

---

# 2023년 대학부설 과학영재교육원 사사과정 연구성과 발표대회

---

- 수학 -



한국과학창의재단  
Korea Foundation for the Advancement of Science & Creativity



# contents

## 수학A

- 01. 경계를 가지는 공간에서의 효율적 공 채우기 문제 ..... 3
- 02. 한국 프로스포츠에서의 승률 예측 공식 ..... 15
- 03. 한글 암호 고안 및 수학적 탐구 ..... 25
- 04. 마르코프 체인을 활용한 학습패턴 분석 ..... 53
- 05. 수학적 규칙을 이용하여 색상 배열 추천하기 ..... 65
- 06. K-Means를 활용한 2024 KBO FA 선수 분석 ..... 79
- 07. 두 평균의 역 문제 ..... 91
- 08. 피보나치 수의 합과 제곱 합의 점화식 ..... 103
- 09. 소수를 활용한 효율적인 스포츠 트레이닝 스케줄 ..... 131
- 10. 일반 파스칼 행렬의 분해 ..... 141
- 11. 정수 계수 부정방정식의 음이 아닌 정수해의 개수를 위한 조합적 계산에 대한 연구 ..... 149

## 수학B

- 01.  $c-v$ ,  $c-s$ ,  $c-f$  연결체를 이용한 공간 테셀레이션 ..... 183
- 02. 페이지랭크(Pagerank) 이해 및 활용 ..... 211
- 03. Conway의 생명 게임 탐구 ..... 225
- 04. 트리플카의 연구 사사수학 논문 ..... 235
- 05. 펠 방정식의 일반화 ..... 265
- 06. 쥐와 고양이 게임의 기하학 ..... 285
- 07. 수의 패턴이 그리는 그림 수 바람개비의 특징 연구 ..... 305
- 08. 최소 힌트의 개수를 갖는 변형스도쿠 만들기 ..... 319
- 09. 진법탐구 ..... 335
- 10. 물체의 회전관성으로 유도한 Fibonacci 수열들의 합 ..... 345
- 11. 함수와 수학적 모델링 ..... 359
- 12. 자연수  $n$ 의  $m$ -regular composition 개수와  $m$ -bonacci 수열 ..... 379



# 수학A

01. 경계를 가지는 공간에서의 효율적 공 채우기 문제 .....	3
02. 한국 프로스포츠에서의 승률 예측 공식 .....	15
03. 한글 암호 고안 및 수학적 탐구 .....	25
04. 마르코프 체인을 활용한 학습패턴 분석 .....	53
05. 수학적 규칙을 이용하여 색상 배열 추천하기 .....	65
06. K-Means를 활용한 2024 KBO FA 선수 분석 .....	79
07. 두 평균의 역 문제 .....	91
08. 피보나치 수의 합과 제곱 합의 점화식 .....	103
09. 소수를 활용한 효율적인 스포츠 트레이닝 스케줄 .....	131
10. 일반 파스칼 행렬의 분해 .....	141
11. 정수 계수 부정방정식의 음이 아닌 정수해의 개수를 위한 조합적 계산에 대한 연구 .....	149



수학A

01\_경계를 가지는 공간에서의 효율적 공 채우기  
문제



# 경계를 가지는 공간에서의 효율적 공 채우기 문제

이재우 (천안동성중학교 2학년)

장우진 (문정중학교 2학년)

조인록 (한솔중학교 2학년)

지도교수 : 고영우 (공주대학교 수학교육과)

## 초록

이 논문에서 우리는 경계를 가지는 공간에서 효율적으로 공을 채우는 방법을 연구하였다. 경계를 가지지 않는 무한한 공간에서 공을 쌓는 문제는 케플러 추측이라 불리며, 이 때 효율적으로 공을 쌓는 방법은 공의 기하학적 성질에만 의존한다. 그러나 유한한 공간에서는 필연적으로 그 공간이 경계를 가질 수 밖에 없고, 이 때 효율적으로 공을 쌓는 방법은 공의 기하학적 성질 뿐만 아니라, 경계의 크기와 모양에도 영향을 받는 훨씬 더 복잡한 문제가 된다.

우리는 먼저 2차원에서 원을 직사각형 영역에 효율적으로 채우는 방법부터 연구하려 한다. 이를 위해 경계의 조건을 세부화하고, 그 각각의 경우에 따라 공을 채우는 새로운 방법을 제시할 것이다.

## I. 연구의 필요성 및 목적

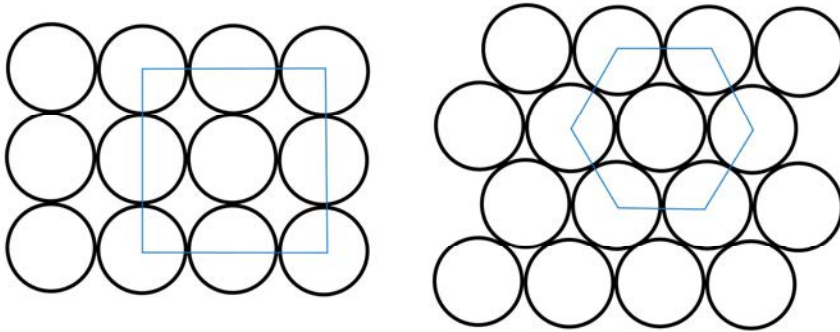
일상에서 바구니에 최대한 많은 물건을 담으려고 애를 써 본 경험은 누구에게나 있을 것이다. 이처럼 제한된 공간에 최대한 효율적으로 물건을 쌓으려는 시도는 실생활에서 너무나 자주 일어나는 중요한 문제이다. 어떠한 방식이 가장 효율적인가 하는 이런 실생활의 중요한 문제는 수학적으로도 당연히 중요한 문제로 취급되었으며, 그동안 많은 이론적 시도가 있었다. 그러나 이 문제는 생각보다도 훨씬 어려운 문제이며, 수학적으로 실제 해결된 것은 무한한 공간에서의 효율성을 계산해보는, 이상적인 상황을 가정한 상태에서만 이루어졌다.

그러나 효율적으로 물건을 쌓는 문제에서, 바탕이 되는 공간에 경계가 존재한다는 가정은 매우 현실적이고 자연스러운 조건이다. 그렇지만 이는 문제를 몹시 복잡하게 만드는데, 이 경계의 모양이나 크기가 우리가 원하는 효율성에 변화무쌍한 효과를 주기 때문이다. 이는 쌓으려고 하는 물건 자체의 기하학적 성질과 맞물려 매우 많은 상황들을 만들어낸다. 따라서 우리는 모든 경우를 완벽하게 다 다룰 수 없으나, 가장 기본적이고 특별한 경우에 대해 효율적 물건 쌓기의 방법이 어떻게 달라지는지를 검토할 예정이다.

## II. 이론적 배경

공 채우기 문제는 오랜 역사를 가진 수학 문제이다. 1611년, 독일의 천문학자 요하네스 케플러는 공을 가장 조밀하게 쌓는 방식에 대한 추측을 내놓았다. 상자가 주어지고, 그 안에 똑같은 크기의 단단한 공을 가능하면 많이 넣으려고 한다. 이때 공을 어떻게 배열해야 가장 밀도 높은 배열을 얻을 수 있을까? 먼저 문제를 단순화시켜, 공의 반지름은 1이라고 가정하자. 이때, 상자의 크기와 모양에 따라 답은 각각 제각각이 될 것이다.

따라서 쉬운 상황을 먼저 고려해보자. 상자의 크기를 무한히 크다고 가정하고, 2차원에서 동전을 가장 밀도 높게 배열하여 평면을 채우는 시도를 해보자. 이 때, 우리는 [그림1]과 같은 두 가지의 방법을 먼저 떠올릴 수 있을 것이다.



[그림1]

[그림1]에서, 동전 사이사이에 동전으로 덮이지 않는 영역을 가능하면 줄여야 밀도가 높은 배열이라 할 수 있을 것이다. 그렇다면 전체 영역에서 동전이 덮는 영역의 넓이를 비율로 나타내보면, 동전이 얼마나 효율적으로 배치되었는지를 비교할 수 있게 된다. 즉, 다음과 같이 밀도를 정의하자.

<p>[정의1]</p> $\text{배열의 밀도} = \frac{\text{단위 공간당 동전이 차지하는 넓이}}{\text{단위 공간의 넓이}}$
---

이제 [그림1]의 배열에서, 사각형 형태의 배열과 육각형 형태의 배열의 밀도는 다음과 같이 간단히 구할 수 있다.

	배열의 밀도
정사각형 형태 배열	$\frac{\pi}{4} = \text{약 } 78.54\%$
정육각형 형태 배열	$\frac{\pi}{\sqrt{12}} = \text{약 } 90.69\%$

위의 계산에서 우리는 자연스럽게 육각형 형태의 배열이 2차원 무한 공간에서 가장 밀도가 높은 배열이라고 생각할 수 있으나, 하지만 이것이 정말로 최선이라는 것을 증명하는 것은 생각보다 어렵다. 동전을 배열하는 무한히 많은 방법을 서로 비교하는 일이 간단하지 않기 때문이다. 이 사실에 대한 증명은 19세기 말 Axel Thue에 의해 얻어졌다.

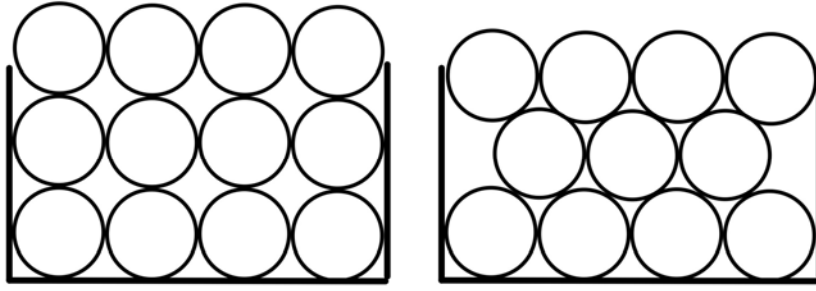
다음으로 우리는 3차원 무한 공간에서 균일한 크기의 공을 효율적으로 쌓는 방법도 생각해 볼 수 있다. 먼저 1층은 2차원에서 채웠던 대로 육각형 형태로 공들을 배열하고, 2층은 세 개의 공이 만나는 움푹 파인 곳에 공을 두는 방식으로 새로운 층을 쌓을 수 있을 것이다. 이는 일상 생활에서도 많이 쓰는 방식이다. 이렇게 공을 쌓았을 때, 공간의 부피 중에서 공의 부피가 차지하는 부피의 비율은  $\frac{\pi}{\sqrt{18}}$ , 즉 대략 74.05%로 구해지게 된다. 이것이 3차원 무한 공간에서 얻을 수 있는 가장 높은 밀도라는 것이 케플러의 추측이다. 그러나 이렇게 직관적으로 당연해보이는 사실은 거의 400년이나 지나서야 Tomas Hales에 의해 증명이 되었다.

또한, 이 문제를 고차원으로 우리의 사고를 확장해 볼 수 있는데, 4차원 무한 공간에서 Viazovska가 이 문제를 해결하였고, (Viazovska는 이 공로로 2022년에 필즈 메달을 수상하였다.) 또한 비슷한 방법으로 24차원의 무한 공간에서 Cohn-Kumar-Miller-Radchenko-Viazovska가 이 문제를 해결하였다.

### III. 연구 방법 및 결과

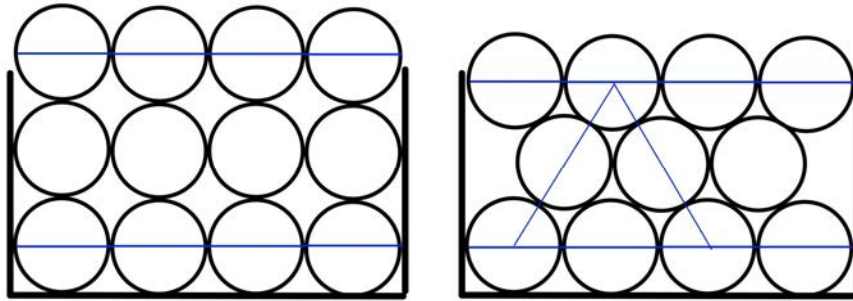
이제 우리는 경계를 가지는 공간에서 공을 채우는 문제를 생각해 보자. 무한한 공간에서의 밀도 최적화 문제도 많은 분야에서 활용되지만, 실제 생활 속의 문제는 공간이 유한한 경우가 대부분이다. 당장, 박스에 공을 담아 이동하는 경우만 생각해도, 경계를 가지는 공간에서 이 문제를 생각하는 것은 매우 자연스럽게.

처음부터 복잡한 경우를 다루기는 어려우니, 우리는 우선 2차원에서 직사각형 모양의 바구니에 공을 담는 방법을 생각해 보겠다. 예를 들어, 아래 [그림2]와 같이 극단적으로 가로 길이가 짧은 상황에서는 육면체 모양의 쌓기보다 정사각 모양 쌓기가 유리할 수 있다.



[그림2]

<경우1> 먼저, 2차원 상자의 가로 길이를  $2n$ 이라 해보자. 한 줄에는 반지름이 1인 원이 최대  $n$ 개가 들어갈 수 있을 것이다. 이 때, 정사각형 형태의 배열의 밀도는 달라지는 것이 없다. 그러나 정육각형 형태의 배열은 양 쪽 경계에서 밀도에 손해를 보게 된다. 가로의 길이를  $2n$ 에 따라 손해를 보는 비율은 달라지게 되는데, 어느 정도의 길이에서 두 쌓기의 밀도 효율이 역전이 일어날까? 이는 실생활에서 매우 많이 등장하는 질문이다.



[그림3]

이 질문에 답하기 위해, 높이를 단위 길이로 설정하자. 두 경우 다 가로의 길이는 같으므로 높이의 비율은 바로 넓이의 비율과 같아진다. 정사각형 형태 배열에서는 단위 높이가 4이고, 이 공간 사이에 배열되는 공의 개수는 모두  $2n$ 개이다. 정육각형 형태 배열에서는 (정사각형의 높이 공식에 의해) 단위 높이가  $2\sqrt{3}$ 이며, 공간 사이에 배열되는 공의 개수는  $2n-1$ 개다. 이를 종합하면 다음과 같다.

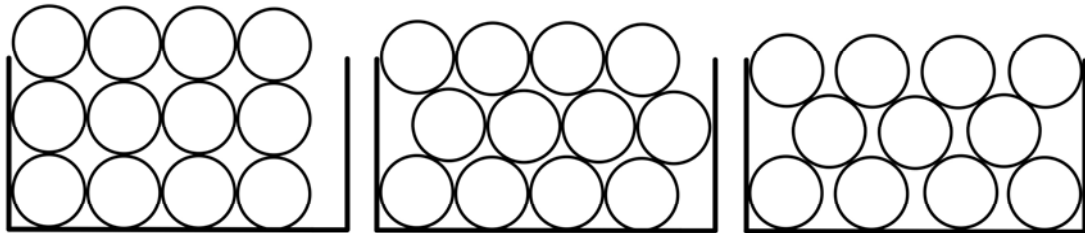
	$n$ 에 따른, 포함되는 공의 갯수 / 단위 공간의 넓이	배열의 밀도	$n \rightarrow \infty$ 일 때, 밀도의 극한값
정사각형 형태 배열	$\frac{2n}{8n}$	$\frac{\pi}{4}$	$\frac{\pi}{4}$
정육각형 형태 배열	$\frac{2n-1}{4\sqrt{3}n}$	$\frac{2n-1}{4\sqrt{3}n}\pi$	$\frac{\pi}{2\sqrt{3}}$

즉, 정육각형 형태의 배열이 정사각형 형태의 배열의 효율을 앞서게 되는 가로  
 길이는  $\frac{2n-1}{4\sqrt{3}n} > \frac{1}{4}$ 의 부등식을 풀면 되고, 이 때의  $n$ 값은  $n > \frac{1}{2-\sqrt{3}} = \text{약 } 3.732$   
 이다. 이 때 경계의 가로의 길이는 정수로 가정하였으므로,  $n \geq 4$ 일 때부터 정육각  
 형 형태가 더 좋은 효율을 가지게 된다. 즉, 무한 공간에서 이론적으로 알고 있는  
 사실과는 다르게,  $n$ 값이 충분히 크지 않을 때는 효율의 역전 현상이 일어난다는 것  
 을 보여준다.

한 편, 정육각형 형태 배열에서  $n \rightarrow \infty$ 를 취하면, 무한 공간에서의 케플러 문제의  
 경우와 같은 밀도인  $\frac{\pi}{\sqrt{12}}$ 가 나온다는 것을 알 수 있다. 즉,  $n$ 값이 커진다는 것은  
 공간이 점점 무한 공간과 비슷해진다는 이야기이고, 이 때 경계가 공의 밀도의 효  
 율에 미치는 영향은 점점 약해진다는 것을 보여준다.

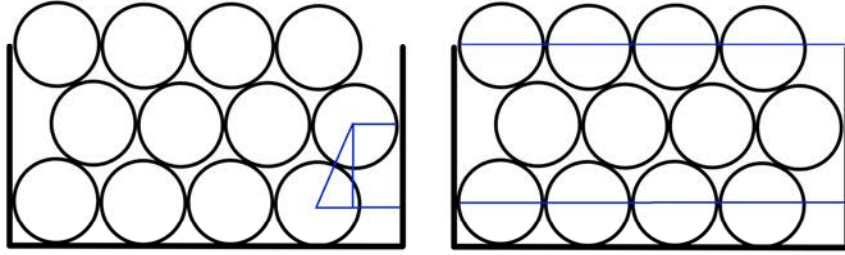
**<경우2>** 다음 질문은, 상자의 가로의 길이가 정확하게  $2n$ 으로 떨어지지 않는  
 경우를 고민해 볼 수 있다. 이 질문은 앞의 질문보다 더욱 현실적인데, 보통의 경  
 우, 상자의 길이는 공의 지름의 정수배로 정확하게 떨어지지 않기 때문이다. 이제,  
 상자의 가로의 길이를  $2n + \alpha$ , ( $0 < \alpha < 2$ )라고 두자.

**<경우2-1>**  $0 < \alpha \leq 1$  일 때.



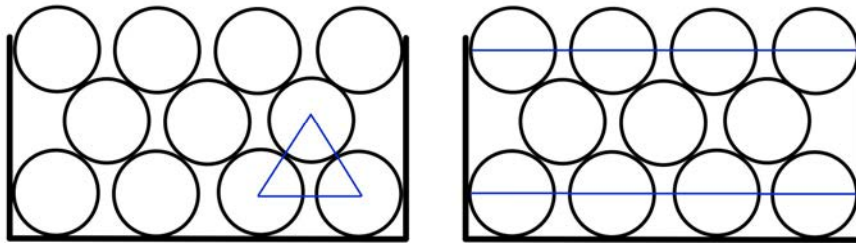
[그림4]

이 때는 정사각형 형태 배열을 사용하면, 상자에 빈 공간이 발생한다. ([그림4]의  
 첫 번째) 그러나 정육각형 배열을 사용하기에는 여분의 공간이 충분하지 않은 경우  
 이다. 이 경우 우리는 두 가지 나은 시도를 할 수 있는데, 충분히 정육각형이 되지  
 않더라도 조금씩 공을 옆으로 밀어보는 경우와, ([그림4]의 두 번째) 또는 각 공의  
 길이를 균등하게 공간을 띄우는 경우이다. ([그림4]의 세 번째) 두 가지 시도는 모  
 두 쌓인 공의 높이를 낮추는 효과를 얻을 수 있다. 우리는 이 두 가지 방법을 비교  
 하여 분석해보겠다.



[그림5]

첫 번째 방법은, [그림5]의 첫 번째 그림의 작은 삼각형을 관찰해보면, 이 삼각형의 빗변은 2, 그리고 밑변은  $(1+\alpha)-1=\alpha$ 이다. 따라서, 이 삼각형의 높이는 피타고라스 정리에 의해  $\sqrt{4-\alpha^2}$ 이 된다. 그러므로, 기준 높이는  $2\sqrt{4-\alpha^2}$ 이고, 이 높이에 공은 모두  $2n$ 개가 들어간다.



[그림6]

[그림6]의 두 번째 방법은, 여분의 공간  $\alpha$ 를 각각의 공 사이에 균등하게 적용시켰으므로, 같은 줄의 공과 공 사이의 거리는  $\frac{\alpha}{n-1}$ 이 된다. (이 때, 항상  $n \geq 2$ 라고 가정한다.) 따라서, 왼 쪽 그림의 작은 삼각형은, 밑변의 길이가  $2 + \frac{\alpha}{n-1}$ 이고, 빗변의 길이가 2인 이등변 삼각형이다. 이 삼각형의 높이는  $\sqrt{3 - \frac{\alpha}{n-1} - \frac{\alpha^2}{4(n-1)^2}}$ 이다. 따라서 기준 높이는  $2\sqrt{3 - \frac{\alpha}{n-1} - \frac{\alpha^2}{4(n-1)^2}}$ 이고, 이 높이에 공은 모두  $2n-1$ 개가 들어간다.

따라서, 두 방법의 효율을 정리하면 다음과 같다.

	$n$ 과 $\alpha$ 따른, 포함되는 공의 개수 / 단위 높이	밀도
--	---------------------------------------	----

첫 번째 형태 배열	$\frac{n}{\sqrt{4-\alpha^2}}$	$\frac{n}{\sqrt{4-\alpha^2}}\pi$
두 번째 형태 배열	$(2n-1)/2\sqrt{3-\frac{\alpha}{n-1}-\frac{\alpha^2}{4(n-1)^2}}$	$(2n-1)\pi/2\sqrt{3-\frac{\alpha}{n-1}-\frac{\alpha^2}{4(n-1)^2}}$

변수가  $n$ 과  $\alpha$ 이기 때문에, 이 두 개의 변수에 고려하여 두 배열의 효율을 비교하는 계산은 간단하지 않다. 따라서, 각각의  $n \geq 2$ 에 대하여 계산을 수행하여 보자. 아래에서, 두 값의 크기를 계산의 편의를 위해, “효율” 값을 (공의 개수 $\times 2$  / 단위 높이)의 제곱으로 정의하여 제곱급을 없애보자. 그러면 본질적으로 두 번째 형태 배열이 첫 번째 형태 배열보다 우수할 때의  $\alpha$ 의 범위는 다음과 같은  $\alpha$ 에 대한 이차 부등식의 형태를 얻는다.

$$\frac{4n^2}{4-\alpha^2} < \frac{(2n-1)^2}{3-\frac{\alpha}{n-1}-\frac{\alpha^2}{4(n-1)^2}}$$

이 이차 부등식은 각각의  $n$ 값과  $\alpha$ 값에 따라 정확히 풀 수 있으나, 우선 고정된  $n$ 값에 대해 각각  $\alpha$ 값의 범위를 계산해서 일반적인 아이디어를 얻어보기로 하자.

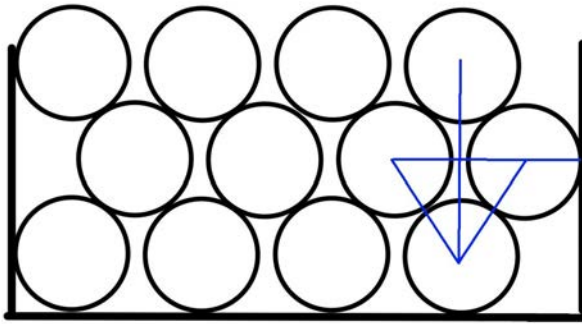
$n$ 의 값	첫 번째 형태 배열의 효율	두 번째 형태 배열의 효율	첫 번째 < 두 번째 일 때의 양수 $\alpha$ 범위
2	$\frac{16}{4-\alpha^2}$	$\frac{9}{3-\alpha-(\alpha^2/4)}$	$\alpha$ 의 범위 없음
3	$\frac{36}{4-\alpha^2}$	$\frac{25}{3-(\alpha/2)-(\alpha^2/16)}$	$\alpha$ 의 범위 없음
4	$\frac{64}{4-\alpha^2}$	$\frac{49}{3-(\alpha/3)-(\alpha^2/36)}$	약 $0 < \alpha < 0.594296$
5	$\frac{100}{4-\alpha^2}$	$\frac{81}{3-(\alpha/4)-(\alpha^2/64)}$	약 $0 < \alpha < 0.729095$
6	$\frac{144}{4-\alpha^2}$	$\frac{121}{3-(\alpha/5)-(\alpha^2/100)}$	약 $0 < \alpha < 0.79084$
⋮	⋮	⋮	⋮
101	$\frac{40804}{4-\alpha^2}$	$\frac{40401}{3-(\alpha/100)-(\alpha^2/40000)}$	약 $0 < \alpha < 0.989999$
⋮	⋮	⋮	⋮
1001	$\frac{4008004}{4-\alpha^2}$	$\frac{40040001}{3-(\alpha/1000)-(\alpha^2/4000000)}$	약 $0 < \alpha < 0.999$

계산의 결과를 보면,  $n$ 값이 증가할수록 각 배열 형태의 효율 값의 분자 부분은 거의 유사한 값으로 수렴하는데, 분모 부분은 1보다 작은  $\alpha$ 에 대해 차이가 점점 벌어진다. 즉,  $\alpha$ 가 충분히 작을 때 첫 번째 방식의 분모는 4에 가깝고, 두 번째 방식의 분모는 3에 가깝다. 이는 효율 값에서 어마어마한 차이를 내며,  $n$ 값이 증가할수록

두 번째 방식의 효율이 압도적으로 증가하게 된다. 식에서 볼 수 있듯이, 상당히 큰  $n$ 값에 대해서는 대부분의  $\alpha$ 에서 두 번째 방식이 효율이 좋음을 알 수 있다. 반대로, 크지 않은  $n$ 값에서는 첫 번째 방법이 항상 우월하거나, 두 방식이 경합한다는 것을 확인할 수 있다.

2)  $1 < \alpha < 2$  일 때.

이 경우, 두 번째 방법은 앞과 같은 방식으로 식이 세워진다. 즉, 두 번째 방법의 밀도는 여전히  $(2n-1)\pi/2\sqrt{3-\frac{\alpha}{n-1}-\frac{\alpha^2}{4(n-1)^2}}$ 이다.



[그림7]

첫 번째 방법의 경우, [그림7]에서 확인할 수 있다시피, 정삼각형 배열에서 조금 더 여유를 가지는 형태로 변화한다. 이 때, 같은 줄과 공과 공 사이의 거리는  $\frac{2(\alpha-1)}{2n-1}$ 이다. 또한, 피타고라스 정리에 의해, 높이는  $\sqrt{3-\frac{2(\alpha-1)}{2n-1}-\frac{(\alpha-1)^2}{(2n-1)^2}}$ 이다.

이 안에 공이  $n$ 개가 들어간다. 즉, 이때의 밀도는  $n\pi/\sqrt{3-\frac{2(\alpha-1)}{2n-1}-\frac{(\alpha-1)^2}{(2n-1)^2}}$ 이다.

이제, 첫 번째 방법이 두 번째 방법보다 효율이 좋을 때의 방정식은, 양 방법의 밀도를 제공하여 정리하면

$$4n^2\left(3-\frac{\alpha}{n-1}-\frac{\alpha^2}{4(n-1)^2}\right) > (2n-1)^2\left(3-\frac{2(\alpha-1)}{2n-1}-\frac{(\alpha-1)^2}{(2n-1)^2}\right)$$

이다. 이를  $\alpha$ 에 대한 2차 부등식으로 정리하면,

$$\left(\frac{(2n-1)^2}{(2n-1)^2}+\frac{-4n^2}{4(n-1)^2}\right)\alpha^2+\left(\frac{2(2n-1)^2}{(2n-1)}-\frac{2(2n-1)^2}{(2n-1)^2}+\frac{-4n^2}{n-1}\right)\alpha+\left(\frac{-2(2n-1)^2}{(2n-1)}+\frac{(2n-1)^2}{(2n-1)^2}+12n^2-3(2n-1)^2\right)>0$$

이고, 이를 다시 간단하게 정리하면

$$(2n-1)\alpha^2 + 4(2n-1)(n-1)\alpha - 8n(n-1)^2 < 0$$

이다. 이 2차 방정식의 두 근은  $\alpha = \frac{2(n-1)}{2n-1}(-2(2n-1) \pm \sqrt{(2n-1)(4n-1)})$ 이고, 이는 둘 다 음수 해이다. 즉, 양수인  $\alpha$ 에서는 위의 부등식이 항상 성립하지 않는다. 즉, 두 번째 방법이 항상 더 효율이 좋다고 말할 수 있다.

이는, 각 층에 항상 같은 개수의 공을 쌓으려고 하는 것보다는, 각 층마다 공의 개수를 하나씩 빼주어 지그재그로 배열하는 것이 더 낫다는 것을 의미한다. 이는 두 경우  $0 < \alpha \leq 1$ 와  $1 < \alpha < 2$ 에서 공통적으로 보여지는 경향이라는걸 확인할 수 있다. (물론,  $0 < \alpha \leq 1$  경우에는 폭이 극단적으로 좁은 경우에 한해 첫 번째 방식이 더 효율적인 경우도 있었다.)

#### IV. 결론 및 제언

이 논문에서 우리는 경계를 가지는 공간에서 효율적으로 공을 채우는 방법을 연구하였다. 특히, 2차원에서 원을 직사각형 영역에 효율적으로 채우는 방법을 집중적으로 탐구하였으며, 이를 위해 경계의 조건을 세부화하였다. 또한, 각각의 경우에 따라 여러 가지 방법들의 우열을 판단할 수 있는 계산 근거를 마련하였다.

이 결과로써, 제한된 폭을 가지는 바구니에 공을 담을 때, 각 층에 배열하는 공의 개수를 지그재그로 하나씩 차이를 두어 배치하는 것이, 모든 층에 같은 개수의 공을 배열하려고 하는 시도보다 대부분의 경우에서 더 나은 경향을 보인다는 사실을 확인할 수 있었다. (물론 폭이 매우 좁은 경우는 성립하지 않을 수 있다.)

이는 실제로 최적화된 방식일 것이라고 여겨지나, 이 배열이 가능한 모든 배열들 중 최선이라는 수학적 증명은 참고문헌을 보더라도 매우 어려운 문제이다. 따라서 이 논문에서는 중등 사사 과정의 한계를 인정하고, 최적의 방법이라 여겨지는 몇 가지 방식들을 비교하고 분석하는 선에서 마무리하였다.

#### V. 참고문헌

- 1) A. Thue, *Om nogle geometrisk-taltheoretiske Theoremer*, **Forhandlingerne ved de Skandinaviske Naturforskeres** 14 (1892), 352 - 353.
- 2) T. C. Hales, *A proof of the Kepler conjecture*, **Ann. of Math.** (2) 162 (2005), no. 3, 1065 - 1185.
- 3) M. S. Viazovska, *The sphere packing problem in dimension 8*, **Ann. of Math.** (2) 185 (2017), no. 3, 991 - 1015.

4) H. Cohn, A. Kumar, S. D. Miller, D. Radchenko, and M. Viazovska, *The sphere packing problem in dimension 24*, **Ann. of Math. (2)** 185 (2017), no. 3, 1017 - 1033.

수학A

## 02\_한국 프로스포츠에서의 승률 예측 공식



# 한국 프로스포츠에서의 승률 예측 공식

김도형 (청주중학교 3학년)

지도교수 : 박영희 (청주교육대학교 수학교육과)

다양한 목적으로 프로스포츠의 승률을 예측하기 위한 공식으로 피타고라스 정리가 사용되어왔다. 이 공식은 팀의 득점수와 실점수의 지수를 조절하여 정확도를 높인다. 이 공식은 리그마다, 경기 종목마다 최적의 지수값이 달라지는데, 본 연구에서는 한국 프로 야구, 프로 농구의 최적 지수값을 찾는다.

롯데 자이언츠 1982~2023년도 기록, 안양 KGC의 10-11시즌 ~ 22-23시즌 기록, 22-23시즌 프로농구 전체 팀의 기록을 사용하여 각 데이터의 최적 지수값을 찾았다. 그 결과 년도별 예측 승률과 실제 승률의 오차 제곱의 총합이 낮게 나타났다. 본 연구에서는 기존 연구를 보완하여 승률 예측 과정에 전년도 승률을 반영하여 오차를 더욱 줄일 수 있도록 하는 공식을 적용하였다. 이 방법에서는 기존보다 더욱 낮은 오차를 얻었다.

## I. 서론

프로스포츠 경기를 보던 중 승률을 예측하여 다양한 용도로 활용할 수는 없을까 생각하였다. 그러던 중 야구의 승률을 예측하는 야구의 피타고라스 정리라는 방법을 알게되었다. 이 공식에서 득점값과 실점값의 지수를 조절하여 승률을 보다 정확하게 예측하려하였다.

또한 지수만을 조절하는 기존 형태와 달리 전년도 승률을 반영하여 승률을 예측하도록 하였다. 전년도 실제 승률과 야구의 피타고라스 정리를 사용한 당해년도 예측 승률을  $c:1$ 로 내분하는 값으로 승률을 구하였다.  $c$ 의 값이 클수록 실제 승률이 전년도 승률보다는 공식을 통한 예측승률에 더 가깝다는 의미이다. 이때 오차가 최소가 되는 최적  $c$ 값을 구해 야구의 승률에 전년도의 승률이 반영되는 정도를 알고자 하였다.

## II. 이론적 배경

야구의 피타고라스 정리에서 예측승률( $W_{pct}$ )은 아래와 같이 득점(RS), 실점(RA)로 표현된다.

$$W_{pct} = \frac{RS^2}{RS^2 + RA^2}$$

이 공식에서는 더욱 정확한 예측을 위해 득점값과 실점값의 지수를 조절해야 하는데, 이를 통해 예측의 오차를 줄이는 것이 본 연구의 목표이다.

또한 스포츠 리그에서 선수의 구성, 감독과 코치진의 구성은 바로 다음 시즌에는 상대적으로 크지 않은 변동이 있다. 따라서 전년도의 승률을 반영하여 더욱 정확하게 승률을 예측하고자 하였다.

### III. 연구 방법

#### 1. 사용한 프로그램

Python을 이용하여 각 경우의 오차 제곱의 총합을 계산하고, 데이터를 시각화하였다. Matplotlib, csv, numpy 라이브러리를 사용했으며, 코드를 쉽게 공유하기 위해 google colabory 환경에서 연구를 진행하였다.

##### 1-1. 알고리즘

기존 연구에서 밝혀진 최적 지수(야구에서 약 2, 농구에서 약 16) 근처의 값에서 지수값을 0.01씩 변화시키며 오차 제곱의 총합을 구하였고 이를 matplotlib를 통해 꺾은선 그래프로 나타내었다. 이를 통해 오차 제곱의 총합이 최소인 최적지수를 구하였다.

또한 전년도 승률을 반영한 방법에서는 그렇게 구한 최적지수값에서 c값을 변화시키며 최적 c 값을 구해 위와 같은 방법으로 꺾은선 그래프로 나타내었다.

##### 1-2. 소스코드

- 롯데 자이언츠 데이터를 불러오는 소스코드 일부

```
import csv
import matplotlib.pyplot as plt
import numpy as np

f = open('anyang.csv', encoding='utf-8')
data = csv.reader(f)
.....
```

- 감마값에 따른 오차 제곱 총합을 구하는 함수 gapSqSum의 코드

```
def gapSqSum(gamma):  
  
    exp = []  
    rea = []  
    gap = []  
  
    for i in range(len(reco)):  
        rea.append(float(reco[i][0]))  
  
        rs = float(reco[i][1])  
        ra = float(reco[i][2])  
        temp = expect(gamma,rs,ra)  
  
        exp.append(temp)  
  
        gap.append(rea[i]-exp[i])  
  
    sum = 0  
    for i in gap:  
        sum += i**2  
  
    return sum
```

## 2. 자료 수집

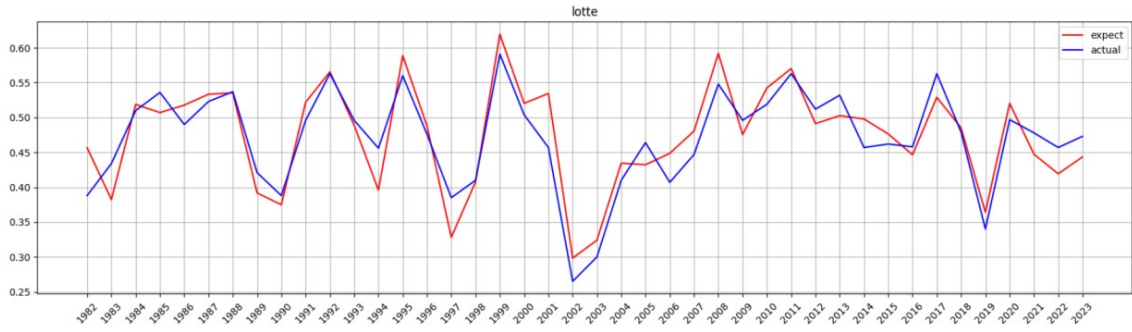
롯데 자이언츠 홈페이지(<https://www.giantsclub.com>) 기록실 -> 역대 기록의 자료에서 1982 ~ 2023년의 승률, 득점, 실점 데이터를 csv파일 형태로 추출하였다.

KBL 홈페이지(<https://www.kbl.or.kr>) LIVE 기록 -> 팀 기록실의 자료에서 안양 KGC의 10-11시즌 ~ 22-23시즌의 승률, 득점, 실점 데이터를 csv파일 형태로 추출하였다.

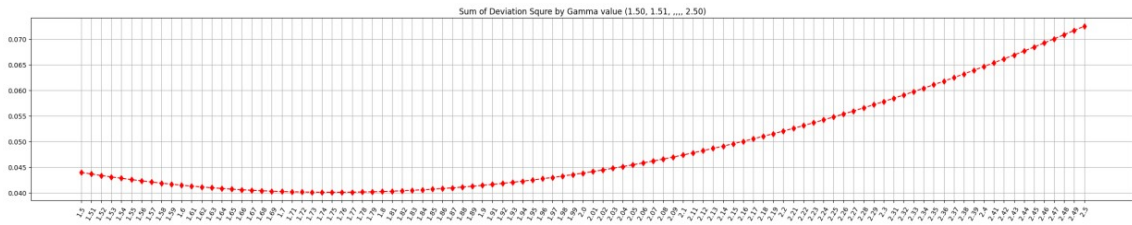
KBL 홈페이지(<https://www.kbl.or.kr>) LIVE 기록 -> 팀 기록실의 자료에서 2022-2023시즌의 10개 팀의 승률, 득점, 실점 데이터를 csv파일 형태로 추출하였다.

#### IV. 연구 결과

##### 1. 야구의 피타고라스 최적지수



위 그래프는 피타고라스 승률 예측공식에서 지수가 2일 때 롯데자이언츠의 연도별 예측승률(빨강)과 실제 승률(파랑)을 나타낸 것이다.



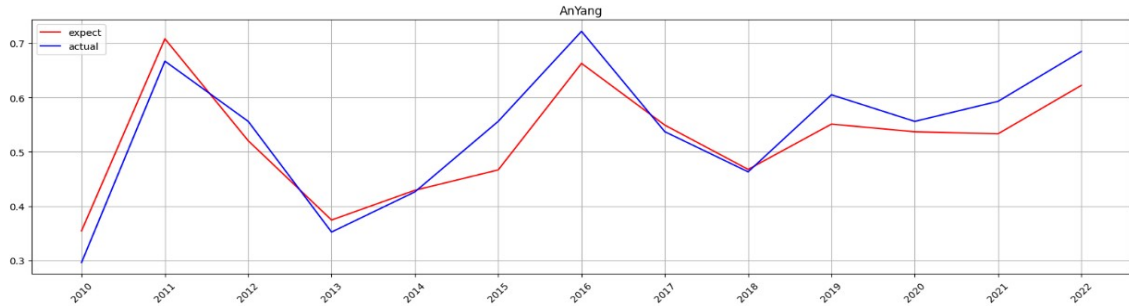
위 그래프는 지수값( $\gamma$ )을 1.50에서 2.50까지 0.01씩 증가시켰을 때 연도별 오차 제곱의 총합을 나타낸 그래프이다. 아래 그래프는 몇 개의 지수값( $\gamma$ )에 따른 오차 제곱의 총합을 소수점 아래 셋째 자리까지 나타낸 것이다.

지수값( $\gamma$ )	1.6	1.75	1.9	2
오차 제곱 총합	0.041	0.040	0.041	0.044

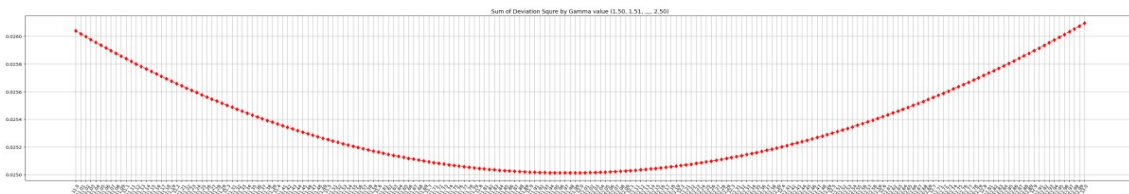
그래프와 표에서 오차 제곱의 총합이 최소가 되는 지수값( $\gamma$ )은 1.75로 구해진다. 이때의 오차 제곱의 총합은 지수값이 2일때보다 0.003 낮게 나타난다.

## 2. 농구의 피타고라스 최적지수

### 2.1. 안양 KGC 데이터 사용



위 그래프는 피타고라스 승률 예측 공식에서 지수가 10일 때 안양 KGC의 연도별 예측승률(빨강)과 실제 승률(파랑)을 나타낸 것이다.



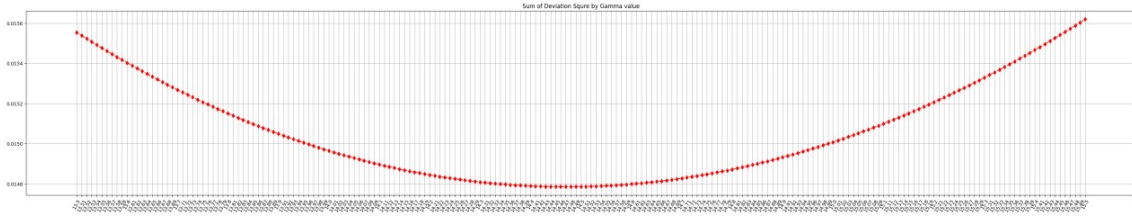
위 그래프는 지수값( $\gamma$ )을 11.00에서 13.00까지 0.01씩 증가시켰을 때 연도별 오차 제곱의 총합을 나타낸 그래프이다. 아래 그래프는 몇 개의 지수값( $\gamma$ )에 따른 오차 제곱의 총합을 소수점 아래 셋째 자리까지 나타낸 것이다.

지수값( $\gamma$ )	10	11	11.97	13
오차 제곱 총합	0.029	0.026	0.025	0.026

그래프와 표에서 오차 제곱의 총합이 최소가 되는 지수값( $\gamma$ )은 11.97로 구해진다. 이때의 오차 제곱의 총합은 지수값이 10일때보다 0.004 낮게 나타난다.

### 2.2. 22-23시즌 프로농구 10개 팀 데이터 사용

아래 그래프는 지수값( $\gamma$ )을 13.50에서 15.50까지 0.01씩 증가시켰을 때 연도별 오차 제곱의 총합을 나타낸 그래프이다. 이때 오차 제곱의 총합이 최소가 되는 지수값( $\gamma$ )은 14.47로 구해진다. 이는 안양 KGC의 데이터만을 사용했을 때보다 높은 수치이다.



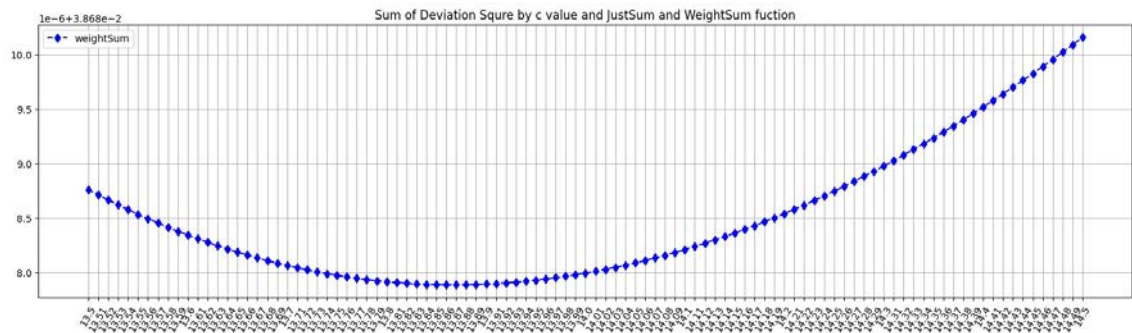
### 3. 야구의 전년도 승률을 반영한 승률예측

$$W = \frac{W_{pct} \times c + W_{작년}}{c + 1}$$

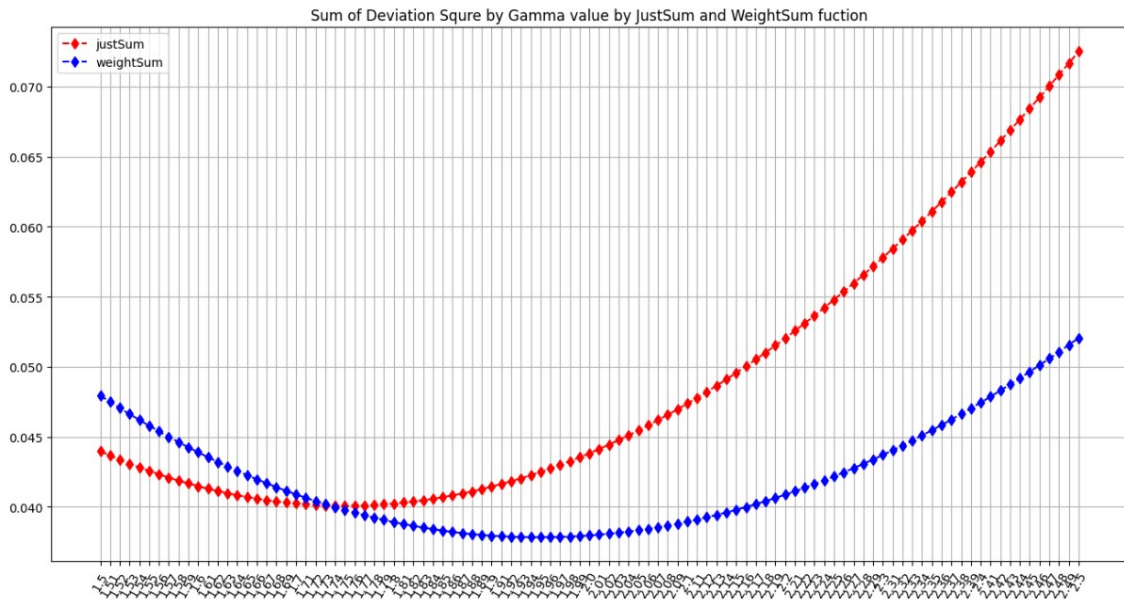
전년도 승률을 반영하여 승률을 예측하는 공식이다.  $W_{pct}$ 는 기존 피타고라스 공식을 통한 예측 승률,  $W_{작년}$ 은 팀의 작년 승률  $W$ 는 최종 예측 승률이며  $c$ 값을 조절하여 예측의 정확도를 높인다.

또한 전년도 실제 승률과 야구의 피타고라스 정리를 사용한 당해년도 예측 승률을  $c:1$ 로 내분하는 값을 의미하게된다.

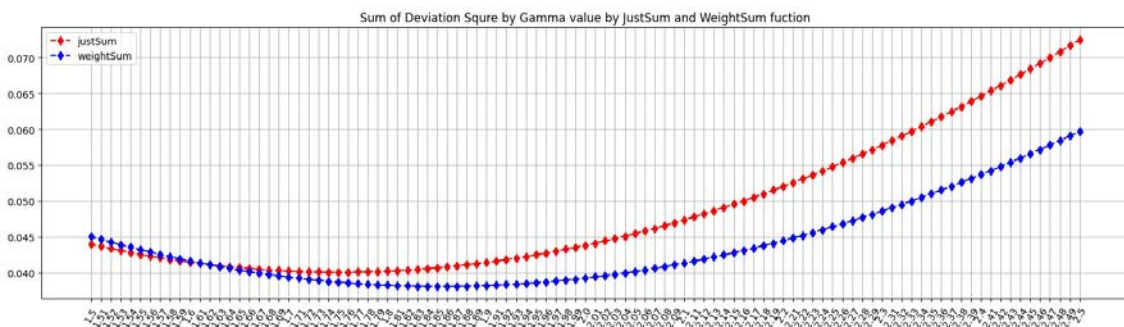
가)



먼저 위 그래프는 지수( $\gamma$ )가 2 일 때 최적의  $c$  값에 따른 오차 제곱의 총합을 나타낸 것이다. 이때의 오차 제곱의 총합이 최소가 되게 하는 최적  $c$  값은 약 6.8임을 알 수 있었다. 또한  $c$ 가 6.8일 때, 최적지수( $\gamma$ )값은 2와 오차가 있는 1.95로 나타났다. 아래는 값마다별 기존 예측의 오차 제곱의 총합(빨강),  $c=6.8$ 로 일정할 때 전년도 승률을 반영한 예측의 오차 제곱의 총합(파랑)의 그래프이다.



나) 또한 아래는 앞서 구한 야구의 피타고라스 최적지수( $\gamma$ ) = 1.75일 때,  $c$  값에 따른 오차 제곱의 총합을 나타내는 그래프이다. 이때 오차 제곱의 총합이 최소가 되도록 하는 최적  $c$  값은 약 13.87임을 알 수 있었다. 또한  $c$ 가 13.87일 때, 최적지수( $\gamma$ )값은 1.85로 나타났다. 아래는 값마다별 기존 예측의 오차 제곱의 총합(빨강),  $c=13.87$ 로 일정할 때 전년도 승률을 반영한 예측의 오차 제곱의 총합(파랑)의 그래프이다.



다) 위의 가) 와 나)에서  $c$ 값과 그에따른 최적지수( $\gamma$ )에 따라 각각 하나의 ( $c$ ,  $\gamma$ )의 순서쌍을 얻었다. 이 두가지 경우와 기존 예측에서의 오차 제곱의 총합을 아래 표로 나타내었다.

	기존 예측 $\gamma = 2$	기존 예측 $\gamma = 1.75$	$c = 6.8$ $\gamma = 1.95$	$c = 13.87$ $\gamma = 1.85$
오차 제곱 총합	0.044	0.040	0.0378	0.0381

따라서 오차 제곱의 총합이 최소인 경우는 전년도 승률을 반영한 공식에서  $c = 6.8$ ,  $\gamma = 1.95$ 인 경우이고, 이때의 오차 제곱 총합은 기존 예측에서  $\gamma = 2$ 인 기존에 사용되던 조건에 비하여 0.0062 낮은 수치를 보였다.

## V. 결론 및 제언

본 연구에서는 기존 야구의 피타고라스 정리를 한국의 프로야구와 프로농구에 적용하였으며, 전년도의 성적을 반영할 수 있도록 공식을 수정하였다. 이를 통해 기존의 공식보다 적은 오차로 승률 예측이 가능하였다. 그리하여 야구에서의 오차가 가장 적은 최종 예측 승률  $W$ 는 득점(RS), 실점(RA)에 대해 아래와 같이 구해진다.

$$W_{pct} = \frac{RS^{1.95}}{RS^{1.95} + RA^{1.95}}, W = \frac{6.8 W_{pct} + W_{작년}}{6.8 + 1}$$

이때 최적  $c$ 값이 6.8로 1보다 크다는 것을 통해 공식을 통한 예측승률이 전년도 승률보다 많이 반영된다는 것을 의미한다는 결론을 내릴 수 있었다. 또한 전년도 승률을 반영하지 않았을 때, 야구와 농구에서의 오차 제곱의 평균, 즉 평균제곱오차(MSE)는 야구에서 0.000952, 농구에서 0.001923로 농구에서 더욱 크게 나타났고 피타고라스 최적지수 또한 농구에서 더욱 크게 나타났다. 대체적으로 야구보다 농구에서 득점과 실점값이 높는데, 이것과 오차 및 최적지수간의 상관관계를 파악하고 축구, 하키등 다른 종목에도 같은 경향이 나타나는지 파악하는 연구가 필요하다고 생각한다.

## VI. 참고문헌

- [1] 이장택, 김용태. (2006) 한국 프로스포츠에서의 승률 추정, Journal of the Korean Data Analysis Society, Vol 8, No 5, 2105-2116.
- [2] 이장택. (2015). 한국프로야구에서의 피타고라스 정리의 정확도 측정, 한국데이터정보과학회지, 26(3), 653-659.
- [3] 이장택. (2019). 한국프로농구에서의 피타고라스 최적지수, 한국데이터정보과학회지, 30(5), 987-995.

수학A

### 03\_한글 암호 고안 및 수학적 탐구



# 한글 암호 고안 및 수학적 탐구

권채은 (전주우아중학교 2학년)  
김채린 (전주동중학교 2학년)  
윤소희 (전주서신중학교 2학년)  
이태환 (신흥중학교 3학년)  
한민주 (전주온고을중학교 2학년)

지도교수 : 정은희 (전북대학교 수학교육과)  
지도조교 : 임수빈 (전북대학교 수학교육과)

## 초록

암호란 비밀을 유지하기 위하여 당사자끼리만 알 수 있도록 꾸민 약속 기호이다. 암호학은 긴 역사를 가지고 있다. 4000년 전 이집트인들이 사용한 단순한 암호부터 20세기 세계대전에서 큰 역할을 한 암호까지 그 방법과 활용은 매우 다양하다. 본 연구의 목적은 암호의 수학적 기반을 탐구하고 이것을 바탕으로 적절한 한글 암호를 고안하는 것이다. 기본적인 암호는 알파벳을 숫자에 대응시킨 후 모듈러 연산을 바탕으로 만들어진다. 본 연구에서는 한글을 숫자에 대응시키는 몇 가지 방법을 제시하고 그 중 가장 모순 없는 방법을 찾았다. 또한 선택한 한글 암호 체계에서 덧셈, 곱셈, 아핀, 다중 문자 치환 암호들을 생각하고, 고안한 한글 암호의 장단점에 대하여 분석하였다.

## I. 서론

비밀을 유지하며 정보를 교환하기 위해 만들어진 암호는 오랜 역사를 가지고 있다. 과거에는 주로 국가 안보와 전략을 보호하는 도구로써 군사, 외교적인 목적으로 사용되었으나 정보통신기술이 발달함에 따라 현재는 민간 분야에서도 다양하게 활용되고 있다. 이 과정에서 암호학에 대한 연구가 활발해졌고, 수학, 역사학, 정치학, 언어학 등의 학문이 다양하게 활용되었다.

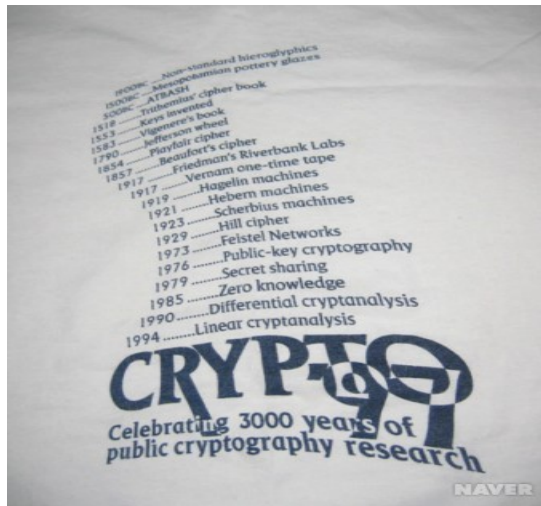
암호학에서는 누구나 읽을 수 있는 문장을 평문이라 부르고 읽을 수 없게 변형된 문장을 암호문이라 부른다. 그리고 평문을 암호문으로 바꾸는 과정을 암호화, 암호문을 평문으로 바꾸는 과정을 복호화라 부른다. 가장 기본적인 암호의 암호화 및 복호화 과정은 수학의 모듈러 연산으로 이해할 수 있다. 때문에 암호는 실생활에 수학이 활용되는 대표적인 예라고 할 수 있다. 하지만 Holden(2007)을 통해 관찰한 알려진 암호들은 알파벳(특히 영어)를 바탕으로 한 평문을 암호화하는 방법들이었다. 우리는 우리의 일상어인 한글을 바탕으로 한 암호 체계를 탐구해보고 싶다는 호기심을 가지게 되었다. 특히, 한글과 알파벳 사이의 뚜렷한 차이가 한글 기반 암

호와 알파벳 기반 암호 사이에 차이를 줄 것으로 예상되어 그것을 탐구하는 것이 의미가 있을 것으로 기대되었다.

본 연구에서는 기존에 알려진 암호화 방법들을 활용하여 한글 암호를 만들고 이것의 수학적 성질을 탐구하는 것을 목표로 한다. 이를 위하여 첫 번째로 효율적인 한글 암호 체계를 찾고 이것을 바탕으로 덧셈 암호, 곱셈암호, 아핀암호, 다중 문자 치환 암호를 만들었다. 두 번째로 각각의 암호의 좋은 키의 성질 및 좋은 키의 개수를 확인하였다. 마지막으로 만들어진 암호의 장단점에 대하여 분석하였다.

## II. 암호의 역사

암호는 긴 역사를 가지고 있다. 1997년, ‘크립토(Crypto)’라는 국제암호학회에서 참가자들에게 나누어준 T셔츠에는 암호의 역사에 대해 기록되어 있다. T셔츠 속 글을 살펴보면, 암호 연구에 대한 역사가 벌써 3,000년이 넘었다는 것을 알 수 있다.



【그림 1】 암호의 역사가 담긴 T-셔츠

암호 기술의 발전 역사는 흔히 두 번의 큰 전환점을 기준으로 고대암호, 근대암호, 현대암호 세 단계로 구분된다. 첫 번째 전환점은 1920년대 제 1·2차 세계대전을 배경으로 무선통신 기술이 발전됨에 따라 여러 기계적, 전자적 암호장치를 개발하고 사용한 것이다. 두 번째 전환점은 1970년대 들어 컴퓨터 사용이 활발해지면서 컴퓨터를 이용한 암호 기술이 발전한 것이다. 이러한 두 전환점을 기준으로 고대부터 제 1·2차 세계대전 이전까지 사용된 초보적인 암호 기술들을 고대암호라고 하며, 두 차례의 세계대전부터 1970년대까지의 복잡한 기계·전자장치들을 이용한 암호 기술을 근대암호, 컴퓨터가 개발된 이후 컴퓨터를 이용하는 발전된 암호 기술을 현대암호라고 부른다. 고대암호 및 근대암호 시기에는 암호 기술이 주로 전문가들에 의해 군사용·첩보용으로 쓰였고, 일반인들은 암호기술에 대해 인식하거나 접할 기회가 없

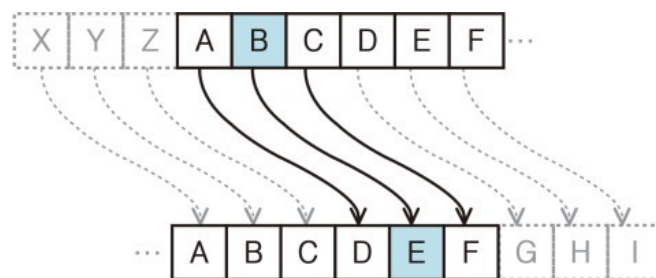
었다. 반면 현대암호 시기에 들어서는 일반인들도 암호기술에 대해 공부하고 널리 사용하게 되었다. 이러한 암호기술의 역사를 살펴보면 인류는 고대로부터 정보를 보호해야 할 필요성을 느꼈고, 이것을 성취하기 위해 그 당시의 최선의 지식과 기술을 이용하여 암호를 고안하고 이용하는 데 많은 노력을 기울여 왔음을 알 수 있다.

고대암호에는 대표적으로 스키테일 암호, 시저 암호, 악보 암호가 있다. 스키테일 암호는 특정 지름을 갖는 막대에 종이를 감아 생성하는 암호로서, 막대의 지름을 아는 이만 암호를 해독할 수 있는 특징이 있다. 암호문 수신자는 특정 지름을 갖는 막대에 종이를 감고 평문을 횡으로 쓴 다음, 종이를 풀어 암호문을 전달한다. 이 종이는 평문의 각 문자가 재배치되어 정보를 인식할 수 없는데, 수신자가 사용한 막대와 지름이 같은 막대를 가진 송신자는 막대에 종이를 감아 횡으로 읽어 암호문을 읽을 수 있다. 여기서 막대의 지름은 송신자와 수신자 사이에 공유된 비밀키가 된다.



【그림 2】 스키테일 암호 예시

시저 암호는 로마의 황제였던 줄리어스 시저(Julius Caesar)가 사용한 암호이다. 시저는 가족과 비밀통신을 할 때 각 알파벳 순으로 세자씩 뒤로 물려 읽는 방법으로 글을 작성했다. 즉 A는 D로, B는 E로 바꿔읽는 방식이다. 수신자가 암호문을 복호화하려면 암호문 문자를 좌측으로 3문자씩 당겨서 읽으면 원래의 평문을 얻을 수 있다. 송신자와 수신자는 몇 문자씩 이동할지를 비밀키로 하여 바꿔가며 사용할 수 있다.



【그림 2】 시저 암호 예시

고대의 단순한 암호 방식은 17세기 근대 수학이 발전하며 암호 기술도 발전하게 되었다. 프랑스의 외교관인 비제네르(Vigenere)가 고안한, 키워드를 이용한 복수 시저 암호형 방식, 플레이페어(Playfair)가 만든 2문자 조합 암호 등 다양한 암호 방식이 발전했다. 20세기 들어서는 통신기술의 발전과 기계식 계산기에 대한 연구를 바탕으로 두 차례의 세계대전을 통해 암호 설계와 해독에 대한 필요성이 높아지면서 암호에 대한 연구가 활발히 진행됐다. 대표적인 근대 암호로는 이그니마(Enigma) 암호가 있다. 이그니마 암호는 2차 세계대전 당시 독일군이 사용하던 암호로서, 각기 다른 몇 개의 전기적으로 연결하여 원문을 입력하면 전기적 연결에 의해 새로운 암호문을 출력하는 방식이다. 즉, 이 기계가 존재하지 않으면 암호를 풀 수 없다.



【그림 3】

독일군이 사용한 이그니마 암호

이렇게 근대 암호가 발전하면서 근대 암호전쟁이 발발했다. 대표적으로 미드웨이 해전의 암호전쟁은 미국이 일본의 암호를 해독하여 전쟁을 승리로 이끈 암호전쟁이다. 태평양 전쟁 당시 일본의 진주만 공습으로 큰 피해를 입은 미국은 일본의 그 다음 공격목표가 어디인지를 알아내야 했다. 미국 해군 정보부의 암호해독반 블랙 챔버는 일본군의 무전이 증가하고 있음을 발견했다. 미군은 암호해독을 통해 일본의 공격목표가 미드웨이라는 사실을 알아낸 후 전투에 대비하고 반격을 준비하여 일본의 태평양함대를 격파하고 전쟁을 승리로 이끌 수 있었다.

마지막으로 현대 암호는 소인수 분해, 일방향 함수, 타원곡선, 양자물리 등을 이용하여 풀기 어렵도록 하고 있다. 대표적으로 RSA 암호란 MIT학생들인 리베스트(Rivest, R.), 샤미르(Shamir, A.), 에이들먼(Adleman, L.)이 1977년에 개발한 암호 체계로, 지금도 널리 쓰이는 대표적인 공개키 암호 체계이다. 공개키 암호 체계의 특성에는 암호화하는 변환으로부터 그 역변환을 알아내는 것이 어려워야 한다는 점이 있다. 이에 따라 이 암호는 두 소수를 곱하는 것은 쉽지만, 두 소수의 곱을 소인수분해하는 것은 어렵다는 사실을 이용하여 완성되었다. 이와 같은 공개키 암호의 도입은 현대암호의 발전에 중요한 계기가 되었다.

### III. 암호에 활용되는 수학<sup>1)</sup>

#### 1. 유클리드 호제법 (또는 유클리드 알고리즘)

유클리드 호제법은 양의 정수  $a, b, q, r$  ( $a > b$ )이  $a = bq + r$  으로 표현될 때,  $a$ 와  $b$ 의 최대공약수  $\gcd(a, b)$ 가  $b$ 와  $r$ 의 최대공약수  $\gcd(b, r)$ 와 같다는 사실을 이용하여 두 정수  $a, b$ 의 최대공약수를 구하는 방법이다. 이것은 암호에서 복호화 키와 좋은 키를 구할 때 이용된다.

구체적으로 정수  $a, b$ 의 최대공약수를 유클리드 호제법을 이용하여 구해보면 다음과 같다.  $a > b$ 일 때,  $a, b$ 는 정수  $q, r$ 을 이용하여 다음과 같이 나타낼 수 있다.

$$a = q_1 b + r_1, \quad 0 \leq r_1 < b$$

$r_1 = 0$ 이라면  $b$ 는  $a$ 의 약수가 되어  $\gcd(a, b) = b$  이다.  $r_1 \neq 0$ 이면,  $b$ 를  $r_1$ 으로 나눠 다음과 같이 나타낼 수 있다.

$$b = q_2 r_1 + r_2, \quad 0 \leq r_2 < r_1$$

$r_2 = 0$ 이면 멈추고, 아니라면 위의 과정을 반복하여 다음 식을 얻는다.

$$r_1 = q_3 r_2 + r_3, \quad 0 \leq r_3 < r_2$$

위의 나눗셈 과정을 0인 나머지가 나타날 때까지 반복한다. 그 결과 다음과 같은 방정식 체계가 나온다.

$$\begin{aligned} a &= q_1 b + r_1, & 0 \leq r_1 < b \\ b &= q_2 r_1 + r_2, & 0 \leq r_2 < r_1 \\ r_1 &= q_3 r_2 + r_3, & 0 \leq r_3 < r_2 \\ &\vdots \\ r_{n-2} &= q_n r_{n-1} + r_n, & 0 \leq r_n < r_{n-1} \\ r_{n-1} &= q_{n+1} r_n + 0 \end{aligned}$$

그러면  $\gcd(a, b) = \gcd(b, r)$ 를 이용하여 아래의 결론을 얻을 수 있다.

$$\gcd(a, b) = \gcd(b, r_1) = \cdots = \gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n$$

위와 같은 방식으로  $a$ 와  $b$ 의 최대공약수가  $r_n$ 임을 구하는 것이 유클리드 호제법이다.

#### 2. 모듈러 연산

모듈러 연산이란 어떤 수를  $n$ 으로 나누어 그 나머지를 구하는 연산이다.

1) Burton, D. M., 기초정수론 6판, 경문사, 2011 (이준복, 이중섭 번역)

정의 1.
<p><math>n</math>을 주어진 양의 정수라 하자. <math>n</math>이 차 <math>a-b</math>를 나누면 즉, 어떤 정수 <math>k</math>에 대해 <math>a-b=kn</math>이면 두 정수 <math>a</math>와 <math>b</math>를 법 (또는 모듈러 <math>n</math>에 대해 합동(congruent modulo <math>n</math>))이라 말하며 기호로 표현하면</p> $a \equiv b \pmod{n}$ <p>이다.</p>

예를 들어, 17을 5로 나눈 나머지가 2이므로  $17 \equiv 2 \pmod{5}$ 라고 표현하고 17은 mod5에서 2와 합동이라고 할 수 있다. 이에 따라 다음 식들을 확인하는 것은 어렵지 않다.

$$3 \equiv 24 \pmod{7}, -31 \equiv 11 \pmod{7}$$

모듈러 연산의 합과 곱에 대한 성질은 다음과 같다.

정리 1.
<p><math>n &gt; 1</math>이 고정되고, <math>a, b, c, d</math>를 임의의 정수라 하자. 그러면 다음 성질이 성립한다.</p> <p>(a) <math>a \equiv a \pmod{n}</math></p> <p>(b) <math>a \equiv b \pmod{n}</math>이면 <math>b \equiv a \pmod{n}</math>이다.</p> <p>(c) <math>a \equiv b \pmod{n}</math>이고 <math>b \equiv c \pmod{n}</math>이면 <math>a \equiv c \pmod{n}</math>이다.</p> <p>(d) <math>a \equiv b \pmod{n}</math>이고 <math>c \equiv d \pmod{n}</math>이면 <math>a+c \equiv b+d \pmod{n}</math>이고 <math>ac \equiv bd \pmod{n}</math>이다.</p> <p>(e) <math>a \equiv b \pmod{n}</math>이면 <math>a+c \equiv b+c \pmod{n}</math>이고 <math>ac \equiv bc \pmod{n}</math>이다.</p> <p>(f) <math>a \equiv b \pmod{n}</math>이면 모든 양의 정수 <math>k</math>에 대해 <math>a^k \equiv b^k \pmod{n}</math>이다.</p>

정리 1의 내용과 같이 합과 곱에 관한 합동( $\equiv$ )의 성질은 보통의 항등(=)과 유사하다.

일반적인 정수 연산에서는 두 정수가 모두 1인 경우를 제외하고 두 정수를 곱하여 1이 되도록 만들 수 없지만 모듈러 연산에서는 가능한 경우가 있다.

정리 2.
<p>양의 정수 <math>n, m &gt; 0</math>이 <math>n &gt; m</math>인 경우,</p> <p>i) <math>n</math>과 <math>m</math>이 서로소이면 <math>m \times k \equiv 1 \pmod{n}</math>를 만족하는 양의 정수 <math>k</math>가 존재한다.</p> <p>ii) <math>n</math>과 <math>m</math>이 서로소가 아니면 i)을 만족하는 <math>k</math>가 없다.</p>

따라서 정리 2에 의하면 모듈러  $n$ 의 연산에서  $n$ 보다 작은 양의 정수  $m$ 이  $n$ 과 서로소인 경우에는  $m$ 의 역수의 역할을 하는 정수  $\bar{m}$ 가 있고 서로소가 아닌 경우에

는 그러한 정수  $\bar{m}$ 는 존재하지 않는다는 것을 알 수 있다. 이것은 추후에 곱셈 암호, 아핀 암호, 다중 문자 치환 암호의 좋은 키를 찾을 때 중요하게 사용된다.

정리 2는 대부분의 기초 정수론 책에서 소개되는 유명한 명제이다. 하지만 우리는 몇 가지 작은 수  $n$ 에 대한 실험을 통하여 정리 2를 유추하였다. 그리고 명제를 부정한 후 모순을 이끌어내어 명제가 참임을 증명하는 귀류법을 이용하여 정리 2의 두 번째 명제가 참임을 직접 증명하였다. 참고 도서의 도움을 받아 유클리드 호제법의 아이디어를 활용하여 첫 번째 명제 또한 참임을 증명하였다.

<i>의 증명>

$n > m, m > r_0 > r_1 \geq 0$ 일 때  $n = m \times q_0 + r_0, m = r_0 \times q_1 + r_1$ 이라 하자.

$r_1 = 0$ 인 경우  $n$ 과  $m$ 이 서로소이면  $r_0 = 1$ 이고 따라서

$$\begin{aligned} 1 &= n - m \times q_0 \\ &= n + m \times (-q_0) \end{aligned}$$

이 성립한다. 즉,  $k \equiv -q_0 \pmod{n}$ 을 만족하는 양의 정수  $k$ 가 원하는 수이다.

이제  $r_1 > 0$ 인 경우를 생각하자.

$$\begin{aligned} n &= m \times q_0 + r_0 \\ m &= r_0 \times q_1 + r_1 \\ r_0 &= r_1 \times q_2 + r_2 \\ &\vdots \\ r_{\alpha-2} &= r_{\alpha-1} \times q_{\alpha} + r_{\alpha} \\ r_{\alpha-1} &= r_{\alpha} \times q_{\alpha+1} + 0 \end{aligned}$$

위의 과정을 반복하면 유한 번 만에  $r_{\alpha+1} = 0$ 이 되는 경우가 생긴다. 그런데  $n$ 과  $m$ 이 서로소이면  $r_{\alpha} = 1$ 이다. 따라서 위의 방정식 체계는

$$\begin{aligned} r_0 &= n - m \times q_0 \\ &\vdots \\ r_{\alpha} = 1 &= r_{\alpha-2} - r_{\alpha-1} \times q_{\alpha} \cdots \textcircled{1} \end{aligned}$$

으로 표현되고 식 ①에 얻어진  $r_j, 0 \leq j \leq \alpha-1$ , 들을 대입하면  $np + mk = 1$ 인 정수  $p, k$ 를 찾을 수 있다. 즉,  $n$ 과  $m$ 이 서로소일 때  $mk = 1 - np$ 을 만족하는 정수  $p, k$ 가 있고,

$$mk \equiv 1 \pmod{n}$$

이다. 그러므로  $(n, m) = 1$  이면  $mk \equiv 1 \pmod{n}$ 를 만족하는 양의 정수  $k$ 가 반드시 존재한다.

<ii>의 증명>

$n$ 과  $m$ 이 서로소가 아닌 경우에 두 수의 최소공배수  $l$ 은  $l > 1$ 이고  $m = al, n = bl$ 으로 표현할 수 있다. 귀류법을 사용하기 위해  $mk \equiv 1 \pmod{n}$ 을 만족하는 양의 정수  $k$ 가 존재한다고 가정하자. 그러면  $al \times k = 1 + bl \times p$ 이 되는 정수  $p$ 가 있다. 즉,

$$l(ak - bp) = 1$$

이다. 그런데  $l$ 과  $ak - bp$ 이 모두 정수이므로  $l = 1$ 일 수 밖에 없다. 그런데 이는  $l > 1$ 이란 사실에 모순이다. 그러므로  $(n, m) \neq 1$ 이면  $mk \equiv 1 \pmod{n}$ 을 만족하는 정수  $k$ 는 없다.

### 3. 행렬과 행렬곱

행렬은 1개 이상의 수를 직사각형의 배열로 나열한 것을 말한다. 예를 들어  $\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}$ ,  $\begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{pmatrix}$ 와 같이 표기한다. 이때, 행렬의 크기는 그 행렬이 내포하는 행(수평선)의

개수와 열(수직선)의 개수로 표현된다. 즉, 행렬  $\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}$ 의 크기는  $2 \times 2$ 이고,

$\begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{pmatrix}$ 의 크기는  $2 \times 3$ 이다. 크기가 같은 두 행렬이 있을 때 각각 성분들 간의 연산으로 행렬들 사이의 덧셈과 뺄셈을 정의한다. 예를 들어 크기가  $2 \times 2$ 인 두 행렬의 덧셈은

$$\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} + \begin{pmatrix} y_1 & y_2 \\ y_3 & y_4 \end{pmatrix} = \begin{pmatrix} x_1 + y_1 & x_2 + y_2 \\ x_3 + y_3 & x_4 + y_4 \end{pmatrix}$$

이다.

두 행렬의 곱도 덧셈처럼 행렬들의 크기가 맞는 경우에 정의된다. 구체적으로 앞 행렬의 열의 수와 뒤 행렬의 열의 수가 같은 경우에만 곱셈이 잘 정의되며, 곱셈 결과 나오는 행렬의 크기는 (앞 행렬의 행의 수)  $\times$  (뒤 행렬의 열의 수)이다. 예를 들어  $2 \times 2$  행렬과  $2 \times 3$  행렬의 곱은 아래와 같은 이  $2 \times 3$  행렬으로 정의된다.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} e & f & g \\ h & i & j \end{pmatrix} = \begin{pmatrix} ae + bh & af + bi & ag + bj \\ ce + dh & cf + di & cg + dj \end{pmatrix}$$

행렬과 행렬곱은 다중 문자 치환 암호에서 계산을 더 편리하게 할 때 이용된다.

## IV. 알려진 암호화 방법2)

수학을 기반으로 한 암호는 각 문자를 수에 대응시킨 후 연산을 이용하여 암호화한다. 이를테면 a는 1, b는 2, ..., z는 26에 대응시킨다. 구체적으로 아래 [표 1]과 같이 대응된다.

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

【표 1】 알파벳의 숫자화

그런데 알파벳의 개수는 26개로 제한적이기 때문에 연산을 통해 얻어진 암호화된 수가 26보다 크면 이에 대응되는 알파벳이 없다. 이 경우에 암호문의 글자수가 줄어들고 제대로 된 복호화가 불가능하다. 즉, 상대방에게 원하는 뜻을 제대로 전달할 수 없는 문제가 발생한다. 이 문제를 해결하기 위해 모듈러 연산을 이용한다.

앞으로 평문을 숫자로 바꾼 것을 P, 암호문의 숫자를 C라고 하자. P를 암호화하여 나온 숫자 C가 26보다 크다면 C에 대응되는 알파벳을 찾기 위해 모듈러 26의 연산을 사용한다. 예를 들어 암호화 결과,  $C=27$ 가 나왔다면  $C \equiv 27 \equiv 1 \pmod{26}$ 가 되어 암호문의 알파벳은 A가 된다.

### 1. 덧셈 암호

덧셈 암호는 평문에 키  $k$ 를 더해 암호화하는 방법이다. 암호화하는 방법을 식으로 나타내면

$$P+k \equiv C \pmod{26}$$

이다. 예를 들어  $k$ 가 3일 때 ‘apple’을 덧셈 암호로 암호화 하면 다음과 같다. 먼저 알파벳을 각각 숫자로 치환하면  $a=1, p=16, l=12, e=6$  이므로,  $apple=(1,16,16,12,5)$ 이고 각 숫자에  $k=3$ 을 더하면  $1+3 \equiv 4 \pmod{26}, 16+3 \equiv 19 \pmod{26}, 12+3 \equiv 15 \pmod{26}, 5+3 \equiv 8 \pmod{26}$ 가 되어  $(4,19,19,15,8)$ 으로 암호화 된다. 이것을 다시 알파벳으로 치환하면 ‘dssoh’가 된다. 따라서 키  $k$ 가 3인 덧셈 암호에서 ‘apple’은 ‘dssoh’으로 암호화된다.

덧셈 암호에서 암호문을 복호화하는 방법은 정리 1의 모듈러 연산의 성질을 활용하면  $C-k \equiv P \pmod{26}$ 임을 알 수 있다. 실제로 키  $k$ 가 3인 덧셈 암호에서 ‘dssoh’를 복호화해보자. 암호문을 숫자에 대응시키면,  $dssoi=(4,19,19,15,8)$ 이고  $4-3 \equiv 1 \pmod{26}, 19-3 \equiv 16 \pmod{26}, 15-3 \equiv 12 \pmod{26}, 8-3 \equiv 5 \pmod{26}$ 이다. 따라서 복호화 결과 얻어진 평문의 수는  $(1,16,16,12,5)$ 이다.  $(1,16,16,12,5)$ 을 알파벳

2) Holden, J., 수학으로 이해하는 암호의 원리, 프리렉, 2007, (허성심 번역)

으로 다시 치환하면 apple이 되어 실제로 복호화가 잘 되었다는 것을 확인할 수 있다. 덧셈 암호에서는 키  $k$ 를 1과 26사이의 임의의 수로 택하더라도 항상 암호화와 복호화가 잘 됨을 알 수 있다.

## 2. 곱셈 암호 (데시메이션 기법)

곱셈 암호는 평문  $P$ 에 키  $k$ 를 곱해 암호화하는 방법이다. 식으로 나타내면

$$P \times k \equiv C \pmod{26}$$

이다. 예를 들어  $k=3$ 일 때 apple을 암호화해보자. apple=(1,16,16,12,5)이므로 각 숫자에  $k$ 를 곱하고 모듈러 연산을 적용하면  $1 \times 3 \equiv 3 \pmod{26}$ ,  $16 \times 3 \equiv 48 \equiv 22 \pmod{26}$ ,  $12 \times 3 \equiv 36 \equiv 10 \pmod{26}$ ,  $5 \times 3 \equiv 15 \pmod{26}$ 을 얻는다. 따라서 apple을 암호화하면 (3,22,22,10,15)=cvvjo가 된다.

곱셈 암호에서 암호문을 복호화하는 방법은  $C \times ('k$ 의 역수')  $\equiv P \pmod{26}$ 으로 생각하는 것이 자연스럽다. 여기서 'k의 역수'는  $k$ 와 곱해졌을 때 모듈러 26에서 1과 합동이 되는 정수이다. 즉, 정수 ' $k$ 의 역수'를 정수  $\bar{k}$ 라고 하면  $k \times \bar{k} \equiv 1 \pmod{26}$ 이다.  $k$ 가 3일 때, 모듈러 26에서  $\bar{k}$ 는 9이다. 이를 활용하여 위에서 얻어진 암호문 cvvjo를 복호화해보자. 대응되는 수는 암호문에 대응되는 수는 (3, 22, 22, 10, 15)이고  $3 \times 9 \equiv 1 \pmod{26}$ ,  $22 \times 9 \equiv 16 \pmod{26}$ ,  $10 \times 9 \equiv 12 \pmod{26}$ ,  $15 \times 9 \equiv 6 \pmod{26}$ 이다. 따라서 복호화 수는 (1, 16, 16, 12, 5)이다. (1, 16, 16, 12, 5)=apple 이므로 cvvjo를 복호화하면 apple이 된다.

만약 키  $k$ 가 '역수'  $\bar{k}$ 를 갖지 못하면 위의 방식으로 복호화를 할 수 없다. 예를 들어  $k=4$ 인 경우를 생각하자. 정리 2의 ii)에 의해 모듈러 26에서 '4의 역수'인  $\bar{4}$ 가 존재하지 않는다. 이 때, ant=(1, 14, 20)를 암호화 하면  $1 \times 4 \equiv 4 \pmod{26}$ ,  $14 \times 4 \equiv 56 \equiv 4 \pmod{26}$ ,  $20 \times 4 \equiv 80 \equiv 2 \pmod{26}$ 이므로 암호문은 (4,4,2)=ddb가 된다. 즉, 서로 다른 알파벳 a와 n이 동시에 b으로 암호화 되었으므로 이 암호문은 복호화를 시킬 수 없다. 실제로 '역수'  $\bar{k}$ 를 갖지 못하는  $k$ 를 곱셈 암호의 키로 갖는 경우에는 서로 다른 알파벳 중에 동시에 하나의 알파벳으로 암호화되는 경우가 항상 발생하고 암호문은 제대로 복호화될 수 없다.

우리는 암호화 후 복호화가 잘 되는 키를 좋은 키라 부르고, 그렇지 않은 키를 나쁜 키라 부른다. 따라서 곱셈 암호는 모든 키가 좋은 키인 덧셈 암호와 달리 나쁜 키를 갖는다. 정리 2에 의하면  $k$ 가 좋은 키가 되려면 26과 서로소인 관계여야 한다는 것을 알 수 있다. 따라서 좋은 키의 개수는 12이다.

## 3. 아핀 암호

아핀 암호는 곱셈 암호와 덧셈 암호를 결합한 암호로, 평문 P에 곱셈 키 k를 곱한 후 덧셈 키 m을 더하여 암호화하는 방법이다. 즉,

$$C \equiv kP + m \pmod{26}$$

으로 나타낼 수 있다. 예를 들어 k=3이고 m=5 일 때 아핀암호를 생각해보자, apple 을 숫자로 치환하면 (1, 16, 16, 12, 5)이고 이것을 암호화하면  $1 \times 3 + 5 \equiv 8 \pmod{26}$ ,  $16 \times 3 + 5 \equiv 1 \pmod{26}$ ,  $12 \times 3 + 5 \equiv 15 \pmod{26}$ ,  $5 \times 3 + 5 \equiv 20 \pmod{26}$ 이다. 따라서 apple은 (8, 1, 1, 15, 20)=haaot으로 암호화 된다.

복호화하는 방법은 모듈러 연산의 성질에 의해  $P \equiv (C - m) \times \bar{k} \pmod{26}$  이다. 여기서  $\bar{k}$ 는  $k \times \bar{k} \equiv 1 \pmod{26}$  가 되는 정수이다. haaot를 복호화하기 위해 k=3의 '역수'  $\bar{k}$ 가 필요하고 이것은 9이다. (8, 1, 1, 15, 20)에서  $(8-5) \times 9 \equiv 1 \pmod{26}$ ,  $(1-5) \times 9 \equiv 16 \pmod{26}$ ,  $(15-5) \times 9 \equiv 12 \pmod{26}$ ,  $(20-5) \times 9 \equiv 5 \pmod{26}$ 이므로 평문은 (1, 16, 16, 12, 5)=apple 이다.

아핀 암호에서 키는 덧셈 키 m과 곱셈 키 k의 순서쌍 (m,k)으로 주어진다. 때문에 곱셈 암호처럼 좋은 키와 나쁜 키를 모두 갖는다. 왜냐하면 m은 0과 25사이의 임의의 수로 선택이 가능하지만 k는 26과 서로소인 경우만 가능하기 때문이다. 따라서 좋은 키의 개수는 312 이다.

#### 4. 다중 문자 치환 암호

다중 문자 치환 암호는 복수의 알파벳을 묶어서 동시에 암호화 하는 방법이다. 예를 들어 두 문자씩 묶어서 암호화할 때는, 단어 abcd를 (a, b), (c, d)와 같이 두 알파벳씩 분리시킨다. 이때, 괄호로 묶은 알파벳 중 첫 번째 알파벳에 대응되는 수를  $P_1$ , 두 번째 알파벳에 대응되는 수를  $P_2$ 라고 두자. 그러면 다중 문자 치환 암호는 네 개의 수로 구성된 키  $k_1, k_2, k_3, k_4$  를 이용하여 다음과 같이 암호화는 방식을 말한다.

$$C_1 \equiv k_1 P_1 + k_2 P_2 \pmod{26}, C_2 \equiv k_3 P_1 + k_4 P_2 \pmod{26}$$

여기서  $C_1$ 은 암호문의 첫 문자에 대응되는 수이고  $C_2$ 는 두 번째 문자에 대응되는 수이다. 예를 들어 키  $k_1=3, k_2=2, k_3=7, k_4=1$ 을 활용하여 math라는 단어를 암호화 하자. 숫자로 바꾸면 (13, 1, 20, 8)이고, 이것을 두 개씩 묶은 후 연산하면  $3 \times 13 + 2 \times 1 \equiv 15 \pmod{26}$ ,  $7 \times 13 + 1 \times 1 \equiv 14 \pmod{26}$ ,  $3 \times 20 + 2 \times 8 \equiv 24 \pmod{26}$ ,  $7 \times 20 + 1 \times 8 \equiv 18 \pmod{26}$ 이 된다, 따라서 (15, 14, 24, 18)=onxr으로 암호화 된다.

다중 문자 치환에서 암호문을 복호화하는 방법은 연립방정식 
$$\begin{cases} C_1 \equiv k_1 P_1 + k_2 P_2 \\ C_2 \equiv k_3 P_1 + k_4 P_2 \end{cases} \pmod{26}$$
을 풀어서 알아낼 수 있다. 구체적으로 위의 식에  $k_4$ 을 곱하고, 아래의 식에  $k_2$ 을 곱하면

$$\begin{cases} k_4 C_1 \equiv k_1 k_4 P_1 + k_2 k_4 P_2 \\ k_2 C_2 \equiv k_2 k_3 P_1 + k_2 k_4 P_2 \end{cases} \pmod{26}$$

을 얻는데 위의 식에서 아래의 식을 빼면

$$k_2 C_2 - k_4 C_1 \equiv P_1 (k_2 k_3 - k_1 k_4) \pmod{26}$$

이 된다. 이 때,  $(k_2 k_3 - k_1 k_4)$ 의 '역수'  $\bar{k}$ 가 있으면  $P_1 \equiv (k_2 C_2 - k_4 C_1) \times \bar{k} \pmod{26}$  이 된다. 평문  $P_2$ 를 찾기 위해 연립방정식의 위의 식에  $k_3$ 를 곱하고 아래의 식에  $k_1$ 을 곱한 후, 두 식 사이의 차를 생각하면  $k_3 C_1 - k_1 C_2 \equiv P_2 (k_2 k_3 - k_1 k_4) \pmod{26}$ 이 된다. 따라서 양변에  $\bar{k}$ 를 곱하여  $P_2 \equiv (k_3 C_1 - k_1 C_2) \times \bar{k} \pmod{26}$ 를 얻는다. 정리하면

$$P_1 \equiv (k_2 C_2 - k_4 C_1) \times \bar{k} \pmod{26}, \quad P_2 \equiv (k_3 C_1 - k_1 C_2) \times \bar{k} \pmod{26}$$

으로 복호화한다.

위에서 선택한  $k_1, \dots, k_4$ 에 대하여  $k_2 k_3 - k_1 k_4 \equiv 11 \pmod{26}$ 이고, 11은 mod 26에서 '역수' 19를 갖는다. 따라서  $(k_1, k_2, k_3, k_4)$ 를 키로 선택한 다중 문자 치환 암호는 복호화 가능하다. 예를 들어 위에서 얻은 암호문 onxr를 복호화 시켜보자. onxr를 숫자로 치환하면 (15, 14, 24, 18)이므로, 두 개씩 묶어서 (15, 14) (24, 18)을 생각하자.

$$\begin{cases} (2 \times 14 - 1 \times 15) \times 19 \equiv 13 \pmod{26}, & (2 \times 18 - 1 \times 24) \times 19 \equiv 2 \pmod{26} \\ (7 \times 15 - 3 \times 14) \times 19 \equiv 1 \pmod{26}, & (7 \times 24 - 3 \times 18) \times 19 \equiv 8 \pmod{26} \end{cases}$$

이므로 (13, 1, 20, 8)=math을 얻는다. 즉, onxr을 math로 복호화 할 수 있다.

다중 문자 치환 암호에서도 역시 좋은 키와 나쁜 키가 존재한다. 좋은 키는 mod 26에서  $k_2 k_3 - k_1 k_4$ 가 역수를 가지는  $(k_1, k_2, k_3, k_4)$ 이다. 따라서  $(k_1, k_2, k_3, k_4)$ 가 좋은 키가 되려면  $k_2 k_3 - k_1 k_4$ 가 26과 서로소가 되어야 한다.

## V. 한글 암호 체계

한글과 영어 알파벳, 더 나아가서 한국어와 영어는 여러 가지 차이점을 갖는다. 대표적으로 영어 단어는 알파벳을 나란히 배열하여 얻어지지만 한국어 단어는 음절 단위로 모아쓰기를 하여 얻어지며 각 음절 단위는 자음과 모음의 특정한 구조로 구성된다. 이러한 한국어와 영어의 차이점 때문에 한글 암호와 영어 암호에 차이가 생긴다. 앞서 4장에서 살펴보았듯이 영어 암호는 총 26개의 알파벳을 쭉 나열하여 암호 체계를 만들 수 있었다. 하지만 한국어 암호의 경우 한국어 문법 체계를 고려하면 기본 자음, 쌍자음, 기본 모음, 결합모음을 합친 40개의 문자를 순서대로 나열하는 것 만 으로 충분하지 않았다. 본 연구에서는 몇 가지 한글 암호 체계를 고려한 후 한국어 문법 체계에 모순이 없고 효율적인 한글 암호 체계를 찾았다.

첫 번째 방법은 영어 알파벳과 유사하게 기본 자음, 쌍자음, 기본 모음, 결합모음을 차례로 나열한 후 숫자에 대응시키는 방법이다. ([표 2] 참고) 하나의 모듈러 연산

을 고려하므로 암호화 과정이 세 가지 방법들 중에서 가장 간단하지만 모아쓰기를 하는 한글에서는 암호화 후 한글 체계가 가장 많이 붕괴되었다. 구체적인 예는 아래의 6장 3절에서 확인한다.

글자	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ
숫자	1	2	3	4	5	6	7	8	9	10		
글자	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅋ	ㆁ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ
숫자	11	12	13	14	15	16	17	18	19	20		
글자	ㅈ	ㅊ	ㅋ	ㆁ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ
숫자	21	22	23	24	25	26	27	28	29	30		
글자	ㅈ	ㅊ	ㅋ	ㆁ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ
숫자	31	32	33	34	35	36	37	38	39	40		

【표 2】 고안한 한글 체계 1

두 번째 방법은 기본 자음과 기본 모음을 나누어서 한글 암호 체계를 만드는 것이다. ([표 3] 참고) 자음과 모음의 개수가 서로 다르므로 이 체계를 이용하여 암호화를 할 때는 두 종류의 모듈러 연산이 필요하다. 평문에서 나타나는 모음과 자음을 구분하여 풀어쓰기를 한 후 암호화하고 다시 모아쓰기를 하여 암호문을 만든다. 자음과 모음을 구분했기 때문에 암호화 후 한글의 구조는 유지할 수 있지만 실제 한글에서 나타나지 않는 이중 자음과 이중 모음이 나타날 수 있다. 그리고 받침의 유무와 이중 자음 또는 모음의 유무에 따라서 한 단어를 구성하는 모음과 자음의 개수가 달라지기 때문에 모아쓰기를 할 때 평문을 주의 깊게 살펴야 되는 번거로움이 있다. 이 경우의 구체적인 예도 아래의 6장 3절에서 확인한다.

자음	ㄱ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	
숫자	1	2	3	4	5	6	7	8	9	10	11	12	13	14								
모음	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ												
숫자	1	2	3	4	5	6	7	8	9	10												

【표 3】 고안한 한글 체계 2

세 번째 방법은 초성, 중성, 종성으로 나누어 암호화를 하는 것이었다. 즉, 각각에 가능한 문자의 총 개수를 고려하여 초성은 모듈러 19, 중성은 모듈러 21, 종성은 모듈러 28의 연산을 수행하여 암호화한다. 글자 중에 종성이 없는 경우가 있으므로 실존하지 않는 28번째 문자를 종성에 포함시켰다. ([표 4] 참고).

초성	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ
숫자	1	2	3	4	5	6	7	8	9	10	11	12	13	14
초성	ㄱ	ㄴ	ㅁ	ㅂ	ㅅ	-	-	-	-	-	-	-	-	-
숫자	15	16	17	18	19	-	-	-	-	-	-	-	-	-

중성	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅞ	ㅟ	ㅠ	ㅡ
숫자	1	2	3	4	5	6	7	8	9	10	11	12	13	14
중성	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	-	-	-	-	-	-	-	-
숫자	15	16	17	18	19	20	21	-	-	-	-	-	-	-

중성	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ
숫자	1	2	3	4	5	6	7	8	9	10	11	12	13	14
중성	ㄱ	ㅂ	ㄱ	ㄴ	ㄴ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㅁ	X
숫자	15	16	17	18	19	20	21	22	23	24	25	26	27	28

【표 4】 고안한 한글 체계 3

이 체계로 한글 암호를 만들면 세 가지 모듈러 연산을 수행해야 하므로 번거롭다. 그렇지만 초성, 중성, 종성이 구분되어 있어서 암호문의 한글 구조가 유지될 뿐만 아니라 모아쓰기를 하는 과정도 단순하다. 암호문에 실제 한글에서 나타나는 이중 자음과 이중 모음들 만 나타나므로 한글 문법 체계를 가장 잘 유지하는 방법이라고 할 수 있다.

## VI. 연구결과

우리는 5장에서 언급한 세 가지 한글 암호 체계 중에서 암호문이 한글 문법 체계를 여전히 잘 따르는 세 번째 방법(초성, 중성, 종성으로 나누는 방법)을 선택했다. 그리고 4장에서 살펴본 암호화 방법들을 적용하여 새로운 암호를 만들었다. 각각의 암호의 좋은 키의 성질 및 좋은 키의 수를 확인하며 수학적 특성을 살폈다. 그리고 5장에서 제시한 첫 번째와 두 번째 한글 체계가 적절하지 않은 이유를 예를 들어 살폈다.

### 1. 암호화 방법에 따른 분류

#### (1) 덧셈 암호

##### ① 평문과 암호문 사이의 규칙

평문에 대응되는 숫자를 P, 암호문에 대응되는 숫자를 C, 암호화 키를 k라 할 때,

$$C \equiv P+k \pmod{p}$$

이다.

② 복호화 방법

복호화 키를  $\bar{k}$ 라 할 때,  $P \equiv C - k \pmod{p}$ 이므로  $\bar{k}$ 는  $\bar{k} \equiv -k \pmod{p}$ 을 만족하는 정수이다. 따라서 복호화할 때는 암호문에  $\bar{k}$ 를 더한다.

③ 좋은 키와 그 개수

덧셈 암호의 좋은 키는 자명한 키를 포함할 시  $0 < k \leq p$ 를 만족하는 모든 자연수이다. 엄밀히 말하자면 모든 자연수지만, 모듈러 연산을 거쳐 중복 되는 것들이 생기므로 위와 같이 말할 수 있다. 따라서 모듈러  $p$  연산을 하는 암호 체계에서 좋은 키의 수는  $p$ 개이다.

우리가 선택한 한글 암호 체계는 초성, 중성, 종성을 따로 고려하며 각각에 대해 모듈러 19, 모듈러 21, 모듈러 28 연산을 한다. 이 경우 초성, 중성, 종성에 서로 다른 암호화 키  $k_1, k_2, k_3$  을 고려하여 독립적으로 암호화할 수 있다. 따라서 좋은 키의 총 개수는  $19 \times 21 \times 28 = 11172$  이다. 하지만 초성, 중성, 종성의 암호화 키들이 하나의 정수로 표현되는 경우로만 제한하면 좋은 키의 개수는 19, 21, 28의 최소공배수인 1596이다.

<p>성질 1.</p> <p>서로 다른 세 양의 정수 <math>n_1, n_2, n_3 &gt; 0</math>의 최소공배수를 <math>m = \text{lcm}(n_1, n_2, n_3)</math>라 하자. 만약 서로 다른 양의 정수 <math>a, b &gt; 0</math> 이 <math>a &lt; b \leq m</math> 을 만족하면</p> $a \equiv b \pmod{n_i}, i = 1, 2, 3 \dots \textcircled{2}$ <p>이 성립할 수 없다.</p>
---

실제로 성질 1에 의하면 1596 보다 작거나 같은 서로 다른 두 양의 정수  $a, b$ 에 대하여  $a$ 만을 키로 갖는 한글 암호와  $b$ 만을 키로 갖는 한글 암호는 절대 같을 수 없다.

<성질 1의 증명>

만약  $a$ 와  $b$ 가 식 ②를 만족한다면 모든  $i=1,2,3$ 에 대해  $b - a = n_i k_i$  을 만족시키는 정수  $k_1, k_2, k_3$ 이 있다. 따라서  $b - a$  는  $n_1, n_2, n_3$  의 공배수가 된다. 하지만 조건에 의해  $b - a < m$  이므로  $m$  이 최소 공배수라는 것에 모순이 된다. 따라서 식 ②는 불가능하다.

④ 한글 암호에 적용하는 방식

한글 암호의 특수성을 고려하여 덧셈 암호의 원리를 한글 암호 체계에 다음과 같이 적용했다. 평문의 한 글자의 초성, 중성, 종성마다 각각 키를 더한 후 각각  $p=19, 21, 28$ 일 때의 모듈러 연산을 적용해 암호문을 만든다. 복호화 역시 암호문에 각각 다른 모듈러 연산에 의한 복호화 키  $\bar{k} \equiv -k \pmod{p}$ 를 더해 평문을 구한다.

⑤ 적용 예시

덧셈 암호에서  $k=3$ 일 때, 평문 ‘암호’를 암호화하자. 각 문자는 아래와 같이 숫자화 시킬 수 있다.

$$\text{암} = (P_1, P_2, P_3) = (8, 1, 5), \text{호} = (P_4, P_5, P_6) = (14, 5, 28).$$

여기서 순서쌍의 첫 번째 성분은 초성, 두 번째 성분은 중성, 세 번째 성분은 종성을 나타내는 숫자이다. 두 번째 문자 ‘호’에는 중성이 없으므로 28번에 대응시킨다. 따라서,

$C_1 = 8 + 3 = 11 \pmod{19}, \dots, C_6 = 28 + 3 = 31 \equiv 3 \pmod{28}$ 과 같은 연산을 통해  $(C_1, C_2, C_3) = (11, 4, 8), (C_4, C_5, C_6) = (17, 8, 3)$ 을 얻는다. 따라서 암호문은 ‘**경뿐**’이다.

이를 복호화 하면  $\bar{k} = -3$ 이고  $(C_1, C_2, C_3) = (11, 4, 8), (C_4, C_5, C_6) = (17, 8, 3)$ 이므로  $P_1 = 11 - 3 = 8 \pmod{19}, \dots, P_6 = 3 - 3 = 0 \equiv 28 \pmod{28}$ 로부터,  $(P_1, P_2, P_3) = (8, 1, 5)$  과  $(P_4, P_5, P_6) = (14, 5, 28)$ 을 얻는다. 따라서 평문은 ‘**암호**’가 된다.

## (2) 곱셈 암호

### ① 평문과 암호문 사이의 규칙

$$C \equiv P \times k \pmod{p}$$

### ② 복호화 방법

$k\bar{k} \equiv 1 \pmod{p}$ 를 만족하는  $\bar{k}$ 가 존재해야만 복호화가 가능하다. 복호화 키  $\bar{k}$ 에 대한 곱셈 암호로 복호화를 이해할 수 있다.  $\bar{k}$ 가 존재하는 경우에 유클리드 호제법을 활용하여 구한다. (아래의 예시 참고)

### ③ 좋은 키와 그 개수

곱셈 암호에서의 좋은 키의 조건은 정리 2에 의해 키  $k$ 가  $p$ 와 서로소라는 것을 알 수 있다. 따라서 좋은 키의 개수는  $(p, k) = 1, 0 < k \leq p$ 를 만족하는 자연수  $k$ 의 개수이다.

한글 암호에서 초성, 중성, 종성에 서로 다른 키를 고려하는 경우 좋은 키의 수는  $18 \times 12 \times 12 = 2592$ 이다. 하지만 암호화 키들이 하나의 정수  $k$ 로 표현되는 경우로 제한 하면 좋은 키의 개수는 432이다. (아래 [그림 5], [그림 6] 참고).

H1599								H1599								
=COUNTIF(H3:H1598, TRUE)								=COUNTIF(H3:H1598, TRUE)								
A	B	C	D	E	F	G	H	A	B	C	D	E	F	G	H	
1	곱셈 암호 좋은 키의 수						결과	1564	1562	4	8	22	TRUE	TRUE	FALSE	FALSE
2	숫자	mod 19	mod 21	mod 28	19와 서로소	21과 서로소	28과 서로소	1565	1563	5	9	23	TRUE	FALSE	TRUE	FALSE
3	1	1	1	1	TRUE	TRUE	TRUE	1566	1564	6	10	24	TRUE	TRUE	FALSE	FALSE
4	2	2	2	2	TRUE	TRUE	FALSE	1567	1565	7	11	25	TRUE	TRUE	TRUE	TRUE
5	3	3	3	3	TRUE	FALSE	TRUE	1568	1566	8	12	26	TRUE	FALSE	FALSE	FALSE
6	4	4	4	4	TRUE	TRUE	FALSE	1569	1567	9	13	27	TRUE	TRUE	TRUE	TRUE
7	5	5	5	5	TRUE	TRUE	TRUE	1570	1568	10	14	0	TRUE	FALSE	FALSE	FALSE
8	6	6	6	6	TRUE	FALSE	FALSE	1571	1569	11	15	1	TRUE	FALSE	TRUE	FALSE
9	7	7	7	7	TRUE	FALSE	FALSE	1572	1570	12	16	2	TRUE	TRUE	FALSE	FALSE
10	8	8	8	8	TRUE	TRUE	FALSE	1573	1571	13	17	3	TRUE	TRUE	TRUE	TRUE
11	9	9	9	9	TRUE	FALSE	TRUE	1574	1572	14	18	4	TRUE	FALSE	FALSE	FALSE
12	10	10	10	10	TRUE	TRUE	FALSE	1575	1573	15	19	5	TRUE	TRUE	TRUE	TRUE
13	11	11	11	11	TRUE	TRUE	TRUE	1576	1574	16	20	6	TRUE	TRUE	FALSE	FALSE
14	12	12	12	12	TRUE	FALSE	FALSE	1577	1575	17	0	7	TRUE	FALSE	FALSE	FALSE
15	13	13	13	13	TRUE	TRUE	TRUE	1578	1576	18	1	8	TRUE	TRUE	FALSE	FALSE
16	14	14	14	14	TRUE	FALSE	FALSE	1579	1577	0	2	9	FALSE	TRUE	TRUE	FALSE
17	15	15	15	15	TRUE	FALSE	TRUE	1580	1578	1	3	10	TRUE	FALSE	FALSE	FALSE
18	16	16	16	16	TRUE	TRUE	FALSE	1581	1579	2	4	11	TRUE	TRUE	TRUE	TRUE
19	17	17	17	17	TRUE	TRUE	TRUE	1582	1580	3	5	12	TRUE	TRUE	FALSE	FALSE
20	18	18	18	18	TRUE	FALSE	FALSE	1583	1581	4	6	13	TRUE	FALSE	TRUE	FALSE
21	19	0	19	19	FALSE	TRUE	TRUE	1584	1582	5	7	14	TRUE	FALSE	FALSE	FALSE
22	20	1	20	20	TRUE	TRUE	FALSE	1585	1583	6	8	15	TRUE	TRUE	TRUE	TRUE
23	21	2	0	21	TRUE	FALSE	FALSE	1586	1584	7	9	16	TRUE	FALSE	FALSE	FALSE
24	22	3	1	22	TRUE	TRUE	FALSE	1587	1585	8	10	17	TRUE	TRUE	TRUE	TRUE
25	23	4	2	23	TRUE	TRUE	TRUE	1588	1586	9	11	18	TRUE	TRUE	FALSE	FALSE
26	24	5	3	24	TRUE	FALSE	FALSE	1589	1587	10	12	19	TRUE	FALSE	TRUE	FALSE
27	25	6	4	25	TRUE	TRUE	TRUE	1590	1588	11	13	20	TRUE	TRUE	FALSE	FALSE
28	26	7	5	26	TRUE	TRUE	FALSE	1591	1589	12	14	21	TRUE	FALSE	FALSE	FALSE
29	27	8	6	27	TRUE	FALSE	TRUE	1592	1590	13	15	22	TRUE	FALSE	FALSE	FALSE
30	28	9	7	0	TRUE	FALSE	FALSE	1593	1591	14	16	23	TRUE	TRUE	TRUE	TRUE
31	29	10	8	1	TRUE	TRUE	TRUE	1594	1592	15	17	24	TRUE	TRUE	FALSE	FALSE
32	30	11	9	2	TRUE	FALSE	FALSE	1595	1593	16	18	25	TRUE	FALSE	TRUE	FALSE
33	31	12	10	3	TRUE	TRUE	TRUE	1596	1594	17	19	26	TRUE	TRUE	FALSE	FALSE
34	32	13	11	4	TRUE	TRUE	FALSE	1597	1595	18	20	27	TRUE	TRUE	TRUE	TRUE
35	33	14	12	5	TRUE	FALSE	FALSE	1598	1596	0	0	0	FALSE	FALSE	FALSE	FALSE
36	34	15	13	6	TRUE	TRUE	FALSE	1599								
37	35	16	14	7	TRUE	FALSE	FALSE									좋은 키의 수: 432

【그림 5】 좋은 곱셈 키의 개수 구하기 (1)      【그림 6】 좋은 곱셈 키의 개수 구하기 (2)

④ 한글 암호에 적용하는 방식

곱셈 암호에서는 덧셈 암호와 비슷하게 평문의 한 글자마다, 그 안에서조차 성, 중성, 종성마다 각각 키를 곱한 후 모듈러 연산을 적용해 암호문을 만든다. 복호화 역시 각각 모듈러 19, 21, 28의 연산에 대한 복호화 키를 구한 다음 모듈러 연산을 거친다.

⑤ 적용 예시

곱셈 암호에서  $k=5$ 일 때, 평문 ‘암호’를 암호화하자.  $(P_1, P_2, P_3) = (8, 1, 5)$  이고  $(P_4, P_5, P_6) = (14, 5, 28)$ 이다. 때문에  $C_1 = 8 \times 5 \equiv 40 \equiv 2 \pmod{19}, \dots$ ,  $C_6 = 28 \times 5 \equiv 140 \equiv 28 \pmod{28}$  과 같은 계산을 통하여  $(C_1, C_2, C_3) = (2, 5, 25), (C_4, C_5, C_6) = (13, 4, 28)$ 을 얻고 암호문은 ‘**늪뻬**’가 된다.

이를 복호화 하려면 먼저 유클리드 호제법의 아이디어를 이용하여 초성, 중성, 종성 각각의 복호화 키  $\overline{k_1}, \overline{k_2}, \overline{k_3}$ 를 구해야 한다. 먼저 종성에 대해 살펴보자.  $\overline{k_3}$ 은 모듈러 28 연산에서 ‘5의 역수’이다. 따라서 28과 5의 관계를 살펴보면

$$\begin{aligned} 28 &= 5 \times 5 + 3 \\ 5 &= 3 \times 1 + 2 \\ 3 &= 2 \times 1 + 1 \end{aligned}$$

이고 정리하면

$1 = 3 - 2 \times 1 = 3 - (5 - 3 \times 1) = 3 \times 2 - 5 = (28 - 5 \times 5) \times 2 - 5 = 28 \times 2 - 5 \times 11$  이 된다.  
 따라서  $1 \equiv 5 \times (-11) \equiv 5 \times 17 \pmod{28}$ 이고 중성의 복호화 키는  $\overline{k_3} = 17$  이다. 같은  
 방법으로  $\overline{k_1} = 4, \overline{k_2} = 17$ 을 구할 수 있다.

암호문의 숫자는  $(C_1, C_2, C_3) = (2, 5, 25), (C_4, C_5, C_6) = (13, 4, 28)$ 이므로  $P_1 = 2 \times 4 \equiv 8 \pmod{19}, P_2 = 5 \times 17 \equiv 1 \pmod{21}, P_3 = 25 \times 17 \equiv 5 \pmod{28}$ 을 얻는다. 비슷하게  
 $(P_4, P_5, P_6) = (14, 5, 28)$ 을 얻을 수 있고, 평문은 '암호'가 된다.

### (3) 아핀 암호

#### ① 평문과 암호문 사이의 규칙

곱셈 키  $k$ , 덧셈 키  $m$ 에 대하여

$$C \equiv Pk + m \pmod{p}$$

이다. 즉, 평문을 키가  $k$ 인 곱셈 암호로 암호화 한 후 키가  $m$ 인 덧셈 암호로 다시  
 암호화 하여 얻는다. 이때, {덧셈 암호 적용 후 곱셈 암호 적용하는 경우}  $\subset$  {곱셈  
 암호 적용 후 덧셈 암호 적용하는 경우} 와 같은 포함관계를 갖는다.

#### ② 복호화 방법

$Pk \equiv C - m \pmod{p}$ 이므로  $P \equiv \overline{k}(C - m) \pmod{p}$ 이 된다. 여기서  $\overline{k}$ 는  $k\overline{k} \equiv 1 \pmod{p}$ 를 만족하는 정수이다.

#### ③ 좋은 키와 그 개수

아핀 암호에서 키는 덧셈 암호 키  $m$ 과 곱셈 암호 키  $k$ 의 순서쌍  $(m, k)$ 으로 주  
 어진다. 따라서 좋은 키의 개수는

$$(\text{덧셈 암호의 좋은 키의 개수}) \times (\text{곱셈 암호의 좋은 키의 개수})$$

으로 주어진다. 따라서 초성의 아핀 암호화에서 좋은 키의 수는  $19 \times 18$ , 중성의 아핀  
 암호화에서 좋은 키의 수는  $21 \times 12$ , 종성의 아핀 암호화에서 좋은 키의 수는  $28 \times 12$   
 이다. 그러므로 한글 아핀 암호의 좋은 키의 개수는  $19 \times 18 \times 21 \times 12 \times 28 \times 12 = 28957824$   
 이다. 하지만 초성, 중성, 종성의 암호화 키들이 하나의 순서쌍  $(m, k)$ 으로 표현되는  
 경우로 제한하면 좋은 키의 수는  $1596 \times 432 = 689472$  이다.

#### ④ 한글 암호에 적용하는 방식

앞의 경우와 마찬가지로 초성, 중성, 종성을 분류한 후 각각에 아핀 연산과 모듈  
 러 연산을 적용하여 암호화한다. 마찬가지로 방법으로 복호화를 할 수 있다. 곱셈 암  
 호와 마찬가지로 곱셈 키  $k$ 의 복호화 키는 초성, 중성, 종성에 대해 독립적으로 찾  
 아야 된다.

#### ⑤ 적용 예시

$k=5, m=3$ 일 때, 앞의 경우와 마찬가지로 평문 '암호'를 암호화하자. 초성, 중성,  
 종성에 대해 다른 모듈러 연산을 적용하므로  $(C_1, C_4) \equiv (5P_1 + 3, 5P_4 + 3) \pmod{19},$   
 $(C_2, C_5) \equiv (5P_2 + 3, 5P_5 + 3) \pmod{21}, (C_3, C_6) \equiv (5P_3 + 3, 5P_6 + 3) \pmod{28}$ 을 계산하  
 여 암호화 할 수 있다. 그런데 암 =  $(P_1, P_2, P_3) = (8, 1, 5),$  호 =  $(P_4, P_5, P_6) = (14, 5, 28)$

이므로 대입하여 연산하면  $(C_1, C_2, C_3) = (5, 8, 28), (C_4, C_5, C_6) = (16, 7, 3)$ 이 된다. 따라서 암호문은 ‘**류똥**’이다.

이를 복호화하려면 각 모듈러 연산에서  $k=5$ 의 ‘역수’를 먼저 찾아야 된다. 이것이 모듈러 19, 모듈러 21, 모듈러 28 연산에서 각각  $\bar{k}_1=4, \bar{k}_2=17, \bar{k}_3=17$ 이 됨을 이미 알고 있다. 그러면  $(P_1, P_4) \equiv ((C_1 - 3) \times 4, (C_4 - 3) \times 4) \pmod{19}$ ,  $(P_2, P_5) \equiv ((C_2 - 3) \times 17, (C_5 - 3) \times 17) \pmod{21}$ ,  $(P_3, P_6) \equiv ((C_3 - 3) \times 17, (C_6 - 3) \times 17) \pmod{28}$  이다.  $(C_1, C_2, C_3) = (5, 8, 28), (C_4, C_5, C_6) = (16, 7, 3)$ 을 대입하고 연산하면  $(P_1, P_2, P_3) = (8, 1, 5), (P_4, P_5, P_6) = (14, 5, 28)$ 이다. 따라서 평문은 ‘**암호**’이다.

#### (4) 다중 문자 치환 암호

##### ① 평문과 암호문 사이의 규칙

다중 문자 치환 암호는 평문의 글자를  $m$ 개씩 무리지은 후 동시에 암호화하는 것이다. 여기서  $m$ 은 1보다 큰 양의 정수 이다. 특별히  $m=2$ 인 경우를 고려하자. 평문의 첫 번째 문자를 나타내는 숫자를  $P_1$ , 평문의 두 번째 문자를 나타내는 숫자를  $P_2$ 라 두고 대응되는 암호문의 숫자를 각각  $C_1$ 과  $C_2$ 이라 두자. 그러면 1과  $p$ 사이의 고정된 정수  $k_1, k_2, k_3, k_4$ 에 대하여

$$C_1 \equiv k_1 P_1 + k_2 P_2 \pmod{p}, \quad C_2 \equiv k_3 P_1 + k_4 P_2 \pmod{p}$$

이다. 3장 3절에서 소개한 행렬을 이용하여 표현하면

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} \pmod{p}$$

이다.

##### ② 복호화 방법

4장에서 살펴보았듯이  $\bar{k}(k_1 k_4 - k_2 k_3) \equiv 1 \pmod{p}$ 이 되는 정수  $\bar{k}$ 가 존재 하는 경우에만 복호화가 가능하고,

$$P_1 \equiv (k_2 C_2 - k_4 C_1) \times \bar{k} \pmod{p}, \quad P_2 \equiv (k_3 C_1 - k_1 C_2) \times \bar{k} \pmod{p}$$

이다. 편의상 1과  $p$  사이의 정수  $m_1, \dots, m_4$ 이 모듈러  $p$  연산에서

$$m_1 \equiv -k_4 \times \bar{k}, \quad m_2 \equiv k_2 \times \bar{k}, \quad m_3 \equiv k_3 \times \bar{k}, \quad m_4 \equiv -k_1 \times \bar{k}$$

을 만족한다고 하자. 그러면

$$\begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \pmod{p}$$

으로 나타낼 수 있다.

##### ③ 좋은 키와 그 개수

$k = k_1 k_4 - k_2 k_3$  의 모듈러  $p$  연산에서 ‘역수’  $\bar{k}$ 가 존재하는 경우에만

$(k_1, k_2, k_3, k_4)$ 가 좋은 키가 된다. 따라서 정리 2에 의하여  $k = k_1k_4 - k_2k_3$  이 p 와 서로소가 되는 순서쌍  $(k_1, k_2, k_3, k_4)$  들이 다중 문자 치환 암호의 좋은 키이다.

엑셀을 이용하여 초성 암호에서 좋은 키의 개수는 123120, 중성 암호에서 좋은 키의 개수는 96768, 종성 암호에서 좋은 키의 개수는 193536이다. ([그림 7] 참고) 따라서 세 값을 모두 곱하면 다중 문자 치환 한글 암호의 좋은 키의 개수가 된다. 하나의 순서쌍  $(k_1, k_2, k_3, k_4)$ 으로 초성, 중성, 종성의 키가 표현되는 경우로 제한한 경우의 좋은 키의 개수는 고려할 데이터의 크기가 커서 엑셀로 계산하기 어려웠다. 하지만  $1 \leq k_1, k_2, k_3, k_4 \leq 19$  이라는 제약조건을 추가적으로 주고 그 원리를 살폈다. 실제로 이 경우에 좋은 키의 개수가 25814이라는 것을 엑셀을 통해 찾을 수 있었다. ([그림 8] 참고)

#### ④ 한글 암호에 적용하는 방식

초성, 중성, 종성의 문자들끼리 따로 묶어서 암호화를 한 후 모아쓰기를 하여 암호문을 만든다. 계산의 편의상 선형연산을 모아서 먼저 한 후 각각에 맞는 모듈러 연산을 적용한다. (적용 예시 참고). 복호화를 할 때는 다른 암호화 방법에서와 마찬가지로 각각의 모듈러 연산에서  $k = k_1k_4 - k_2k_3$  의 '역수'를 찾은 후 복호화식에 대입하여 평문을 찾는다. 이 경우 복호화 키는 초성, 중성, 종성에 대해 달라질 수 있다.

F5		=MOD(\$F\$1*A5, 19)																									
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V					
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19							
2	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	0							
3	2	2	4	6	8	10	12	14	16	18	1	3	5	7	9	11	13	15	17	0							
4	3	3	6	9	12	15	18	2	5	8	11	14	17	1	4	7	10	13	16	0							
5	4	4	8	12	16	1	5	9	13	17	2	6	10	14	18	3	7	11	15	0							
6	5	5	10	15	1	6	11	16	2	7	12	17	3	8	13	18	4	9	14	0							
7	6	6	12	18	5	11	17	4	10	16	3	9	15	2	8	14	1	7	13	0							
8	7	7	14	2	9	16	4	11	18	6	13	1	8	15	3	10	17	5	12	0							
9	8	8	16	5	13	2	10	18	7	15	4	12	1	9	17	6	14	3	11	0							
10	9	9	18	8	17	7	16	6	15	5	14	4	13	3	12	2	11	1	10	0							
11	10	10	1	11	2	12	3	13	4	14	5	15	6	16	7	17	8	18	9	0							
12	11	11	3	14	6	17	9	1	12	4	15	7	18	10	2	13	5	16	8	0							
13	12	12	5	17	10	3	15	8	1	13	6	18	11	4	16	9	2	14	7	0							
14	13	13	7	1	14	8	2	15	9	3	16	10	4	17	11	5	18	12	6	0							
15	14	14	9	4	18	13	8	3	17	12	7	2	16	11	6	1	15	10	5	0							
16	15	15	11	7	3	18	14	10	6	2	17	13	9	5	1	16	12	8	4	0							
17	16	16	13	10	7	4	1	17	14	11	8	5	2	18	15	12	9	6	3	0							
18	17	17	15	13	11	9	7	5	3	1	18	16	14	12	10	8	6	4	2	0							
19	18	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
20	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
21																											
22			k	경우	조성에서의 좋은 키의 개수																						
23	숫자	빈도 수	1	6840	123120																						
24			2	6840																							
25			3	6840																							
26			4	6840																							
27			5	6840																							
28			6	6840																							
29			7	6840																							
30			8	6840																							
31			9	6840																							
32			10	6840																							
33			11	6840																							
34			12	6840																							
35			13	6840																							

【그림 7】 다중 문자 치환 암호의 좋은 키의 개수 구하기  
(초성 암호만 고려)

L24		=AND(MOD(H24,3)>0, MOD(H24,7)>0)																									
이름 상자	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T								
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19								
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38								
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57								
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76								
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95								
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114								
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133								
8	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152								
9	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171								
10	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190								
11	11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209								
12	12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228								
13	13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247								
14	14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266								
15	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285								
16	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304								
17	17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323								
18	18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342								
19	19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361								
20																											
21	숫자	빈도	k	빈도	k값	mod 19	mod 21	mod 28	mod 19에서	mod 21에서	mod 28에서	역수															
22	1	1	1	0	1151	0	0	1	1	1	FALSE	FALSE	FALSE	0													
23	2	2	2	1	440	1	1	1	1	1	TRUE	TRUE	TRUE	440													
24	3	2	2	2	622	2	2	2	2	2	TRUE	TRUE	FALSE	0													
25	4	3	3	3	539	3	3	3	3	3	TRUE	FALSE	TRUE	0													
26	5	2	2	4	670	4	4	4	4	4	TRUE	TRUE	FALSE	0													
27	6	4	4	5	467	5	5	5	5	5	TRUE	TRUE	TRUE	467													
28	7	2	2	6	762	6	6	6	6	6	TRUE	FALSE	FALSE	0													
29	8	4	4	7	433	7	7	7	7	7	TRUE	FALSE	FALSE	0													
30	9	3	3	8	669	8	8	8	8	8	TRUE	TRUE	FALSE	0													
31	10	4	4	9	539	9	9	9	9	9	TRUE	FALSE	TRUE	0													
32	11	2	2	10	614	10	10	10	10	10	TRUE	TRUE	FALSE	0													
33	12	6	6	11	397	11	11	11	11	11	TRUE	TRUE	TRUE	397													
34	13	2	2	12	775	12	12	12	12	12	TRUE	FALSE	FALSE	0													
35	14	4	4	13	391	13	13	13	13	13	TRUE	TRUE	TRUE	391													
36	15	4	4	14	572	14	14	14	14	14	TRUE	FALSE	FALSE	0													
37	16	5	5	15	520	15	15	15	15	15	TRUE	FALSE	TRUE	0													
38	17	2	2	16	621	16	16	16	16	16	TRUE	TRUE	FALSE	0													
39	18	6	6	17	377	17	17	17	17	17	TRUE	TRUE	TRUE	377													
40	19	2	2	18	714	18	18	18	18	18	TRUE	FALSE	FALSE	0													
41	20	4	4	19	369	0	19	19	19	19	FALSE	TRUE	TRUE	0													
42	21	2	2	20	605	1	20	20	20	20	TRUE	TRUE	FALSE	0													
43	22	2	2	21	462	2	0	21	21	21	TRUE	FALSE	FALSE	0													
44	23	0	0	22	506	3	1	22	22	22	TRUE	TRUE	FALSE	0													
45	24	6	6	23	347	4	2	23	23	23	TRUE	TRUE	TRUE	347													

【그림 8】 다중 문자 치환 암호의 좋은 키의 개수 구하기  
(초성, 중성, 종성 모두 고려,  $1 \leq k_1, k_2, k_3, k_4 \leq 19$ )

⑤ 적용 예시

$k_1 = 2, k_2 = 1, k_3 = 3, k_4 = 4$ 일 때, 다중 문자 치환 방법을 활용하여 평문 ‘암호’를 암호화하자. 초성, 중성, 종성끼리의 모듈러 연산이 같으므로  $(C_\alpha, C_{\alpha+3})$ 로 묶어서

$$\begin{pmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \end{pmatrix} = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix} \begin{pmatrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \end{pmatrix}$$

을 고려한다.  $(P_1, P_2, P_3) = (8, 1, 5), (P_4, P_5, P_6) = (14, 5, 28)$ 이므로

$$\begin{pmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 8 & 1 & 5 \\ 14 & 5 & 28 \end{pmatrix} = \begin{pmatrix} 16+14 & 2+5 & 10+28 \\ 24+56 & 3+20 & 15+112 \end{pmatrix} = \begin{pmatrix} 30 & 7 & 38 \\ 80 & 23 & 127 \end{pmatrix}$$

이다. 따라서  $(C_1, C_4) \equiv (30, 80) \equiv (11, 4) \pmod{19}$ ,  $(C_2, C_5) \equiv (7, 23) \equiv (7, 2) \pmod{21}$ ,  $(C_3, C_6) \equiv (38, 127) \equiv (10, 15) \pmod{28}$ 이다. 결과적으로 암호문은 ‘**꺠랏**’이다.

복호화를 하려면  $k = k_1k_4 - k_2k_3 = 5$ 의 각 모듈러 연산에서의 ‘역수’  $\bar{k}$ 를 먼저 찾아야한다. 곱셈 암호 적용 예시에서  $\bar{k} = 4 \pmod{19}$ ,  $\bar{k} = 17 \pmod{21}$ ,  $\bar{k} = 17 \pmod{28}$ 이 됨을 알았다. 이것을 복호화식에 대입하면

$$\begin{aligned} P_1 &\equiv \bar{k}(k_4C_1 - k_2C_4) \equiv 4 \times (44 - 4) \equiv 4 \times 40 \equiv 8 \pmod{19} \\ &\vdots \\ P_6 &\equiv \bar{k}(k_1C_6 - k_3C_3) \equiv 17 \times (30 - 30) \equiv 28 \pmod{28} \end{aligned}$$

의 연산을 통해  $(P_1, P_2, P_3) = (8, 1, 5), (P_4, P_5, P_6) = (14, 5, 28)$ 을 얻는다. 따라서 평문은 ‘**암호**’이다.

## 2. 정리

연구 결과를 표로 정리하면 다음과 같다.

	덧셈 암호	곱셈 암호	아핀 암호	다중 문자 치환 암호
평문과 암호 사이의 규칙	$C \equiv P+k \pmod{p}$	$C \equiv Pk \pmod{p}$	$C \equiv Pk+m \pmod{p}$	$C_1 \equiv k_1P_1 + k_2P_2 \pmod{p}$ , $C_2 \equiv k_3P_1 + k_4P_2 \pmod{p}$
복호화 키	$n \equiv -k \pmod{p}$	$k \times n \equiv 1 \pmod{p}$ 를 만족하는 $n$	$k \times n \equiv 1 \pmod{p}$ 를 만족하는 $n$ 에 대하여 $(-m \times n, n)$	$k = k_1k_4 - k_2k_3$ 라 할 때 $k \times n \equiv 1 \pmod{p}$ 를 만족하는 $n$ 을 이용 $(-k_4 \times n, k_2 \times n, k_3 \times n, -k_1 \times n)$
좋은 키의 개수 (한글 암호)	$0 < k \leq p$ 를 만족하는 모든 자연수 (1596)	$(p,k)=1, 0 < k \leq p$ 를 만족하는 자연수 $k$ 의 개수 (432)	덧셈 암호 좋은 키의 개수 $\times$ 곱셈 암호 좋은 키의 개수 (689472)	$k = k_1k_4 - k_2k_3$ 일 때 $(p,k)=1, 0 < k \leq p$ 를 만족하는 $k_1, k_2, k_3, k_4$ 의 경우의 수 초성에서의 좋은 키 (123120) 전체의 좋은 키 (...)

여기서 한글 암호의 좋은 키의 개수는 초성, 중성, 중성의 키들이 하나의 정수(또는 정수 순서쌍)으로 표현되는 경우로 제한하여 생각했다.

### 3. 첫 번째와 두 번째 한글 암호 체계의 모순

(1) 첫 번째 방법 (기본 자음, 쌍자음, 기본 모음, 결합모음을 하나로 나열)

덧셈 암호를 활용하여 평문 ‘값’을 암호화하자. [표 2]를 이용하여 평문을 숫자에 대응시키면  $값 = (P_1, P_2, P_3, P_4) = (1, 20, 8, 10)$ 이 된다. 덧셈 키  $k=10$ 을 이용하여 암호화하면  $(C_1, C_2, C_3, C_4) \equiv (11, 30, 18, 20) \pmod{40}$ 을 얻는다. 암호문의 숫자를 한글에 대응시키면  $(C_1, C_2, C_3, C_4) = (\text{쓰}, \text{하}, \text{프}, \text{卜})$ 이다. 따라서 암호문은 ‘**ષ패**’가 되어 한 음절의 단어가 두 음절이 되었다. 이번에는 덧셈 키  $k=25$ 를 이용하여 암호화를 해보자. 모듈러 40 연산을 통해  $(C_1, C_2, C_3, C_4) \equiv (26, 5, 33, 35) \pmod{40}$ 을 얻는다. 숫자를 한글에 대응시키면  $(C_1, C_2, C_3, C_4) = (\text{ㄷ}, \text{ㄷ}, \text{ㅋ}, \text{ㄱ})$ 으로 모아쓰기를 할 수 없다. 결과적으로 한글 문법 체계를 따르지 않는 암호문이 존재한다는 것을 알 수 있다.

(2) 두 번째 방법 (자음과 모음 분리)

덧셈 암호를 활용하여 평문 ‘값’을 암호화하자. 자음과 모음을 분리하여 풀어쓰기를 하면

자음 : ㄱ, ㅂ, ㅅ                      모음 : ㅏ

이고, [표 3]을 이용하여 숫자화 하면  $(P_1, P_2, P_3) = (1, 6, 7)$ 과  $P_1' = 1$  을 얻는다. 자음과 모음의 덧셈 키를  $k=2$ 으로 선택하여 암호화를 하면,  $(C_1, C_2, C_3) \equiv (3, 8, 9) \pmod{14}$ 와  $C_1' = 3 \pmod{10}$ 을 얻는다. 암호문의 자음은 (ㄷ, ㅇ, ㅈ)이고 모음은 (ㅏ)이다. 한글 문법 체계에서 받침에 ㅇ과 ㅈ으로 구성되는 이중모음이 존재하지 않으므로 모아쓰기를 하면 암호문 ‘덩ㅈ’을 얻는다.

## VII. 결론 및 제언

우리는 한 음절을 초성, 중성, 종성으로 구분하여 각각 모듈러 연산을 하는 한글 암호 체계를 만들었다. 그리고 여러 암호화 방법들을 적용하며 해당 암호 체계가 한글 문법 체계에 모순이 없는 적합한 체계임을 확인하였다.

우리가 만든 한글 암호 체계는 기본적으로 우리의 일상어인 한글로 표현된 정보를 비밀을 유지하며 교환할 수 있다는 장점이 있다. 뿐만 아니라, 보안성이 꽤 높은 암호라고 할 수 있다. 일반적으로 침입자가 암호문을 취득하더라도 복호화를 하기 어려울 때, 높은 보안성을 갖는다고 한다. 좋은 키의 개수가 많으면 침입자가 복호화를 시도할 때 키의 후보가 많아지므로 쉽게 정보를 획득하기 어렵다. 따라서 공개 키 암호 이전의 암호들의 경우 좋은 키의 개수가 많을수록 보안성이 높은 암호라고 할 수 있다. 그런데 우리의 한글 암호 체계는 알파벳 암호 체계와 비교했을 때 좋은 키의 개수가 월등히 많다.

하지만 우리가 고안한 한글 암호는 세 가지의 모듈러 연산을 고려해야 하므로 암호화와 복호화 시 계산이 더 복잡할 뿐만 아니라, 초성, 중성, 종성의 글자들을 나열하는 방법도 복잡하다는 단점이 있다. 향후 이 단점들을 보완하고 장점은 더욱 늘릴 수 있는 연구가 필요할 것이다.

## VII. 참고문헌

1. 정교일, 이병천, 진승헌. (2008). *확실히 보이는 정보보호*. 서울 : 전자신문사
2. 황규범, 이시창, 정명인. (2009). *암호학의 이해*. 서울 : 교우사
3. Burton, David M. (2011). *기초정수론 6판*. 이준복, 이중섭(번역). 서울 : 경문사
4. Holden, Joshua. (2007). *수학으로 이해하는 암호의 원리*. 허성심(번역). 부천 : 프리렉
5. 모듈러 연산이란? [Khan Academy 게시판] Retrieved from <https://ko.khanacademy.org/computing/computer-science/cryptography/modular-arithmetic/a/what-is-modular-arithmetic>
6. aqua(2021.06.25.). 4. 아핀암호[네이버 블로그] Retrieved from <https://blog.naver.com/PostView.naver?blogId=wolco&logNo=222410227322>
7. RSA 암호화. [나무위키] Retrieved from <https://namu.wiki/w/RSA%20%EC%95%94%ED%98%B8%ED%99%94>
8. jhh0712(2019.01.08.). RSA 암호[네이버 블로그] Retrieved from <https://m.blog.naver.com/vjhh0712v/221437338754>



수학A

## 04\_마르코프 체인을 활용한 학습패턴 분석



# 마르코프 체인을 활용한 학습패턴 분석

남궁채은 (동산중학교 3학년)

소다현 (중앙중학교 3학년)

채강국 (중앙중학교 3학년)

지도교수 : 정수미 (군산대학교 수학과)

각 시행의 결과가 바로 앞의 시행의 결과에만 영향을 받는다는 마르코프 성질을 가진 확률과정을 이용해서 앞으로의 과목별 학습 패턴을 예측해 보자. 이러한 예측 모델을 만드는 과정에서 행렬표현과 계산의 편리성을 확인해 보고, 단순하지만 앞으로를 예측해 보는 새로운 경험도 같이 할 기회를 가지는데 의의를 둔다.

## I. 서론 (또는 연구의 필요성 및 목적)

행렬이란 행과 열로 이루어진 사각형 형태에 원소를 적고, 순서를 주어 정의한 형태이다. 이러한 행렬을 이해하고 관심을 가지기 위해서는 행렬을 어떻게 활용할 수 있는지에 대해 생각해 볼 필요가 있다.

행렬의 역사를 살펴보면, 다양한 연립방정식 문제를 해결하기 위해 행렬식이 정의되었고, 후에 행렬이 정의된다. 방정식 문제 뿐 아니라, 행렬과 관련된 연산을 정의하고 성질을 파악하다 보면, 어떠한 반복적인 상황에 대해서 계산하는 경우 행렬의 거듭제곱 표현이 편리하다는 점을 파악할 수 있다.

예를 들어 오늘 날씨에 기준하여 특정 날씨를 예상하는 경우 또는 렌트카 영업소별 차 회수율이 시간이 지남에 따라 어떻게 변하는지를 예상하는 경우 행렬의 거듭제곱을 통해 그 예측값을 편리하게 찾아낼 수 있다.

또한, 기후의 급격한 변화로 인한 생태계의 변화, 저출산에 의한 학교 통폐합의 가속화, 코로나 이후 일회용품 사용량 급증에 의한 환경 문제 등 다양한 사회적 논점들은 항상 앞으로 이러한 논점들이 어떻게 변화될 것인지에 대한 예측값을 제시하려고 한다. 불확실성이 주요한 특징을 가지는 이러한 논점들을 설명하는데 확률은 중요한 역할을 한다.

따라서 우리는 많은 변수들을 고려하는 예측 시스템은 아니어도, 간단한 조건과 행렬을 활용해서 앞으로 일어날 일에 대한 예측을 할 수 있는 모델링을 만들어 보려고 한다. 여기에 적합한 예측 시스템이 러시아의 수학자 안드레이 안드레예비치 마르코프의 이름을 따라서 붙여진 마르코프 체인이라는 기법이다.

## II. 이론적 배경

마르코프 체인의 목적은 확률을 활용하여 어떤 객체의 상태가 시간에 따라 어떻게 변화할지 모델링 하는 확률과정으로써 예를 들어 날씨 예측, 인구 이동 예측 등과 같이 날씨 또는 인구 등 객체가 시간이 지남에 따라 어떻게 변화할지를 확률을 이용하여 예측하는 것이다. 단, 각 시행의 결과가 바로 앞의 시행의 결과에만 영향을 받는 특징을 가지는 경우에 적용하고 이러한 특징을 마르코프 성질이라 한다. 또한 마르코프 체인은 행렬을 활용하여 표현할 수 있는데 이때 사용하는 행렬을 마르코프 행렬이라 부른다.

일반적으로 시간이 지남에 따라 시스템이 어떤 상태에서 다른 상태로 바뀌는 상태의 변화를 전이라 하는데, 따라서 전이할 확률을 설명할 때 사용하는 행렬을 전이행렬이라 한다. 또한 시간이 지남에 따라 상태간의 전이를 겪는 시스템의 모델을 랜덤 프로세스라 한다. 그런데 과거 상태에 의존하는 프로세스를 학습하는 것은 과거 상태의 영향이 무수히 많기 때문에 엄청난 과제이다.

상대적으로 마르코프 체인은 시스템의 다음 상태로의 전이 확률이 과거 상태와 상관없이 현재 상태에만 의존하는 랜덤 프로세스이다. 그래서 마르코프 체인은 과거에 일어난 모든 일을 잊어버리므로 무기억 랜덤 프로세스라 부르고, 과거 상태에 의존하는 프로세스보다 가정이 단순하므로 유용하게 적용할 수 있다.

그렇지만, 1차 마르코프 체인은 직전 과거값만 영향을 받는데 비해,  $n$ 차 마르코프 체인은 과거  $n$ 개의 값이 현재값에 영향을 주게 되는 점은 생각해야 한다.

마르코프 체인을 활용해서 예측값을 찾아내는 몇 가지 사례를 살펴보자. Markov Chain을 이용한 국내 폐차발생량 예측<sup>1)</sup>에서는 폐차발생 연구에서 나타나는 문제점을 극복하기 위해서 전국대비 지역별 폐차발생분포 상태의 비교적 안정된 경향을 이용해서 마르코프 체인 전이 행렬을 활용한다. 마코브체인 분석을 이용한 범죄전이 연구<sup>2)</sup>에서는 범죄경력의 전이 또는 패턴을 분석을 목적으로 또는 모바일 악성코드를 예측하는 프로세스<sup>3)</sup>로 마르코프 체인을 활용한다.

마르코프 체인을 활용하기 위해서는 전이행렬으로써 마르코프 체인에서 사용하는 마르코프 행렬의 특징 및 몇 가지 성질을 확인할 필요가 있다.

먼저 마르코프 체인에서 사용하는 전이행렬인 마르코프 행렬의 특징을 정리해 보자. 전이행렬은 확률과정에서 시간에 따라 변하는 확률을 표현한 확률행렬이므로 각 행 또는 각 열의 성분의 합은 1이어야 하고 각각의 성분은 0보다 크거나 같아야 한다.

또한, 마르코프 체인의 핵심은 초기 상태 (초기상태 행렬)를 전이행렬의 거듭제곱을 통해서  $n$ 회 이후의 상태를 예측하는 것을 목적으로 한다. 따라서 마르코프 체인에서 사용하는 전이행렬인 마르코프 행렬의 거듭제곱 계산은 특히 중요하다. 그런

1) Markov Chain을 이용한 국내 폐차발생량 예측

2) 마코브체인 분석을 이용한 범죄전이 연구

3) 마코프 체인을 이용한 모바일 악성코드 예측 모델링 기법 연구

데 전이행렬의 크기가 커지거나 또는 행렬의 거듭제곱의 횟수가 커지면 행렬의 거듭제곱 계산의 시간이 많이 필요하다.

이러한 문제점을 해결하기 위해서 케일리-해밀턴 정리를 활용할 수 있다. 먼저 행렬  $A$ 에 대한 특성방정식을 정의해보고, 케일리-해밀턴 정리를 알아 보자. 그리고 이 정리를 활용해서 행렬의 거듭제곱을 계산하는 방법을 정리해 보자.

특성방정식이란, 정사각행렬  $A$ 에 대하여 연립일차방정식  $(A-\lambda I)X=O$ 의 해  $X$ 가 영행렬이 아닌 형태를 가지기 위한 필요충분조건인  $|A-\lambda I|=0$ 를 행렬  $A$ 에 대한 특성방정식이라 한다. 즉, 행렬식의 형태를 가진 방정식  $|A-\lambda I|=0$ 를 행렬  $A$ 에 관한 특성방정식이라 하고, 이 방정식의 해  $\lambda$ 를 행렬  $A$ 에 대한 고유값이라 한다. 그리고 이 고유값에 대한 특성방정식에 행렬  $A$ 를 대입했을 때에도 주어진 방정식이 성립한다는 정리가 케일리-해밀턴 정리이다.

그렇다면 행렬의 거듭제곱에 어떻게 케일리-해밀턴 정리를 적용할 수 있을까? 하나의 예를 들어 거듭제곱 계산과정을 정리해 보자.

행렬  $A = \begin{pmatrix} 0.3 & 0.3 & 0.25 \\ 0.2 & 0.6 & 0.35 \\ 0.5 & 0.1 & 0.4 \end{pmatrix}$ 에 대하여 케일리 해밀턴 정리를 활용하기 위해 먼저 특성

방정식을 찾아보자.

$$|A-\lambda I| = \begin{vmatrix} 0.3-\lambda & 0.3 & 0.25 \\ 0.2 & 0.6-\lambda & 0.35 \\ 0.5 & 0.1 & 0.4-\lambda \end{vmatrix} = -\lambda^3 + \frac{13}{10}\lambda^2 - \frac{8}{25}\lambda + \frac{1}{50} = -\frac{1}{50}(50\lambda^3 - 65\lambda^2 + 16\lambda - 1) = 0$$

이고,  $\lambda^3 = \frac{13}{10}\lambda^2 - \frac{8}{25}\lambda + \frac{1}{50}$ 이 성립한다.

첫 번째 방법은, 특성방정식에 케일리-해밀턴 정리를 적용하면 다음과 같다.

$$A^3 = \frac{13}{10}A^2 - \frac{8}{25}A + \frac{1}{50}I$$

행렬의 차수를 내리기 위해, 다음과 같이 행렬의 차수 내리기를 시행하면,

$$\begin{aligned} A^4 &= A^3A = \left(\frac{13}{10}A^2 - \frac{8}{25}A + \frac{1}{50}I\right)A = \frac{13}{10}A^3 - \frac{8}{25}A^2 + \frac{1}{50}A \\ &= \frac{13}{10}\left(\frac{13}{10}A^2 - \frac{8}{25}A + \frac{1}{50}I\right) - \frac{8}{25}A^2 + \frac{1}{50}A = \frac{137}{100}A^2 + \frac{109}{250}A + \frac{13}{500}I \end{aligned}$$

이 성립하고, 구체적인 값은 위에 계산된 식에  $A^2, A, I$ 를 대입한다. 동일한 방법을 통해 각각  $n \geq 3$ 에 대하여

$$A^n = aA^2 + bA + cI$$

의 형태로 정리할 수 있다. 즉 각각  $n \geq 3$ 에 대하여 계수  $a, b, c$ 를 찾을 수 있다.

두 번째 방법은 특성방정식의 차수를 먼저 내린 후 케일리-해밀턴 정리를 적용하는 방법인데  $|A - \lambda I| = -\frac{1}{50}(50\lambda^3 - 65\lambda^2 + 16\lambda - 1) = 0$ 에서 세 개의 서로 다른 고유값

$\lambda = 1, \frac{1}{10}, \frac{1}{5}$ 이 존재하기 때문에 적용할 수 있다. 즉,  $\lambda^3 = \frac{13}{10}\lambda^2 - \frac{8}{25}\lambda + \frac{1}{50}$ 이므로

$$\lambda^4 = \lambda^3\lambda = \left(\frac{13}{10}\lambda^2 - \frac{8}{25}\lambda + \frac{1}{50}\right)\lambda = \frac{137}{100}\lambda^2 + \frac{109}{250}\lambda + \frac{13}{500}$$

이고 동일한 방법으로 전개하면  $n \geq 3$ 에 대하여  $\lambda^n = a\lambda^2 + b\lambda + c$ 이라 놓을 수 있다.

이 식에  $\lambda = 1, \frac{1}{10}, \frac{1}{5}$ 을 대입하면

$$\begin{cases} 1^n = a + b + c \\ \left(\frac{1}{10}\right)^n = \left(\frac{1}{10}\right)^2 a + \frac{1}{10}b + c \\ \left(\frac{1}{5}\right)^n = \left(\frac{1}{5}\right)^2 a + \frac{1}{5}b + c \end{cases}$$

이고, 연립방정식의 해  $a, b, c$ 를 찾을 수 있다. 다시 케일리-해밀턴 정리를 적용하면, 다음과 같은 행렬  $A$ 에 대한 거듭제곱식을 찾을 수 있다.

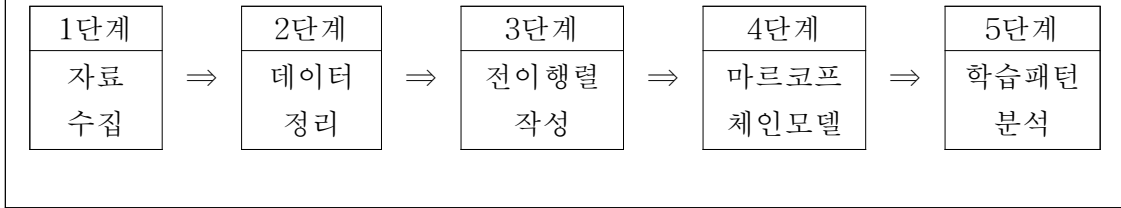
$$A^n = \frac{-225(0.2)^n + 200(0.1)^n + 200}{18}A^2 + \frac{165(0.2)^n - 160(0.1)^n - 5}{12}A + \frac{-45(0.2)^n + 80(0.1)^n + 1}{36}I$$

즉, 정사각행렬  $A$ 가  $n \times n$ 의 크기를 가지는  $n$ 차 행렬이면 특성방정식과 케일리-해밀턴 정리를 활용해서 거듭제곱의 차수를  $(n-1)$ 차 이하의 식으로 표현한 후 계산이 가능하다.

### III. 연구 방법

무기억 랜덤 프로세스로 적합하면서 쉽게 접근할 수 있는 주제로 학습패턴을 분석해 보기로 하자. 몇 가지 과목을 중심으로 현재 상태를 시작으로 학습 패턴이 지속되었을 때 적당한 시간이 지난 후 나의 과목별 학습 패턴을 예측해 보기로 한다. 연구 방법을 단계별로 정리해 보면 다음과 같다.

<마르코프 체인을 활용한 학습패턴 분석 방법>



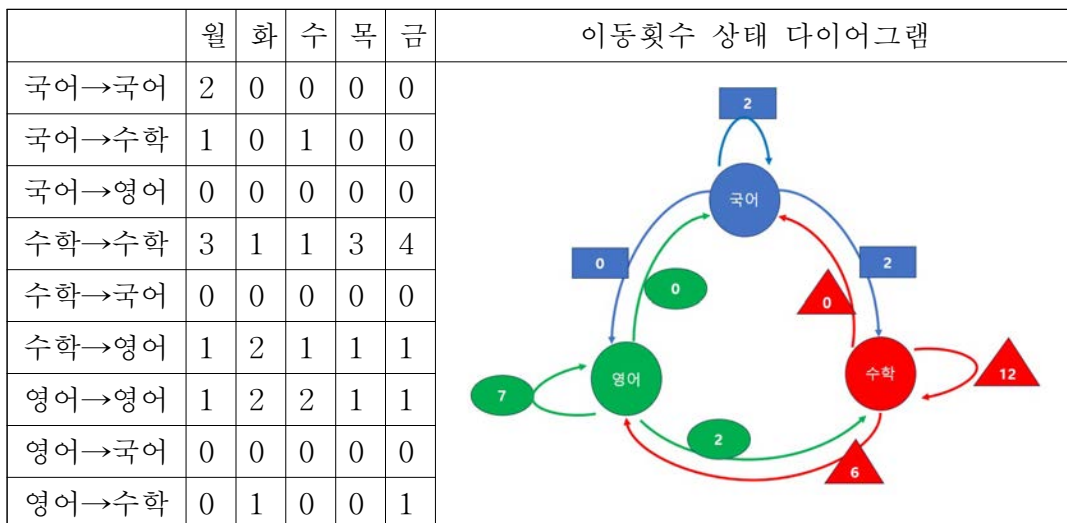
1단계는 자료 수집단계이다. 1주일 또는 2주일간 주로 학습하는 세 개의 교과목을 선정해서 두 가지 자료를 수집하는데 매일 과목별 학습시간과 과목 간 이동 횟수를 표로 작성한다. 그리고 1주 또는 2주의 시간이 지나면, 기간 동안 과목별 학습한 총 시간과 과목별 이동 횟수를 정리한다.

실제로 5일간 과목별 공부시간을 정리한 표와 과목별 이동횟수를 정리한 표이다. (주말의 학습시간은 자료에서 제외한다)

<5일간 과목별 학습시간>

과목/요일	월요일	화요일	수요일	목요일	금요일	총 학습시간
국어	1:01	0:00	0:51	0:00	0:00	$1\frac{5}{6}$ 시간
수학	2:23	3:02	1:49	2:09	3:47	$13\frac{1}{6}$ 시간
영어	2:12	2:47	2:28	1:55	1:20	$10\frac{2}{3}$ 시간

<5일간 과목별 이동횟수>



2단계는 데이터 정리단계이다. 과목별 학습시간을 총 학습시간으로 나눈 비율로 정리하고, 이동 횟수 또한 비율로 정리한다. 국어 학습시간을 A, 수학 학습시간을 B, 영어 학습시간을 C라 놓으면 5일을 주기(토요일 일요일을 제외한 평일만 공부한다고

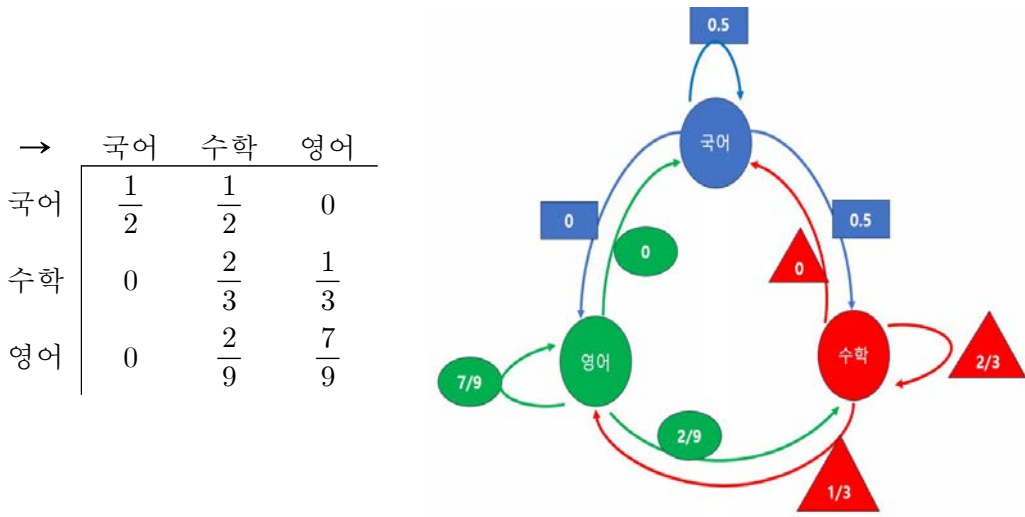
가정할 때)로 하는 과목별 학습시간 비율은 다음과 같다.

$$\text{국어학습시간비율 } a = \frac{A}{A+B+C}$$

$$\text{수학학습시간비율 } b = \frac{B}{A+B+C}$$

$$\text{영어학습시간비율 } c = \frac{C}{A+B+C}$$

그리고 과목별 이동횟수도 비율로 정리한다. 이동횟수를 비율로 정리해야 전이행렬에 대입이 가능하다. 아래 그림은 확률 상태 다이어그램이라고 부른다.



3단계는 데이터를 활용하여 전이행렬을 작성한다. 위에 작성한 상태 확률 다이어그램을 활용하면 다음과 같은 전이행렬  $P$ 를 찾을 수 있다. 행과 열은 첫 번째 행과 열부터 국어, 수학 영어의 순서로 적는다.

→	국어	수학	영어
국어	$\frac{1}{2}$	$\frac{1}{2}$	0
수학	0	$\frac{2}{3}$	$\frac{1}{3}$
영어	0	$\frac{2}{9}$	$\frac{7}{9}$

즉, 전이행렬  $P$ 는 다음과 같다.

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{9} & \frac{7}{9} \end{pmatrix}$$

4단계는 전이행렬을 활용하여 학습패턴을 분석하기 위한 마르코프 체인 모델을 작성한다. 3단계에서 찾은 전이행렬  $P$ 와 2단계에서 정리한 총 학습시간에 대한 과목별 학습시간 비율인 상태행렬  $A = \left( \frac{11}{154} \quad \frac{79}{154} \quad \frac{64}{154} \right)$ 에 대하여 1차 마르코프 체인은

$$AP = \begin{pmatrix} \frac{11}{154} & \frac{79}{154} & \frac{64}{154} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{9} & \frac{7}{9} \end{pmatrix}$$

이다. n회 마르코프 체인을 실행한  $AP^n$ 를 정리한 식을 찾기 위해 먼저 행렬  $P$ 의 거듭제곱  $P^n$ 의 계산을 정리해 보자. 주어진 행렬  $P$ 의 특성방정식을 찾아보면

$$\begin{aligned} 0 = |P - \lambda I| &= \begin{vmatrix} \frac{1}{2} - \lambda & \frac{1}{2} & 0 \\ 0 & \frac{2}{3} - \lambda & \frac{1}{3} \\ 0 & \frac{2}{9} & \frac{7}{9} - \lambda \end{vmatrix} = -\lambda^3 + \frac{35}{18}\lambda^2 - \frac{7}{6}\lambda + \frac{2}{9} \\ &= -\frac{1}{18}(18\lambda^3 - 35\lambda^2 + 21\lambda - 4) = -(\lambda - 1)\left(\lambda - \frac{4}{9}\right)\left(\lambda - \frac{1}{2}\right) \end{aligned}$$

이다. 서로 다른 세 실근  $1, \frac{4}{9}, \frac{1}{2}$ 를 가지므로 특성방정식을 활용한 차수내리는 방법을 적용할 수 있는데 즉,  $\lambda^n = a\lambda^2 + b\lambda + c$ 이라 놓고,  $\lambda = 1, \frac{4}{9}, \frac{1}{2}$ 을 대입해서  $a, b, c$ 를 계산한 후 케일리-해밀턴 정리를 적용하자. 그리고 그 결과를 n회 마르코프 체인에 실행한 결과인  $AP^n$ 을 계산하면 다음과 같다.

$$\begin{aligned} AP^n &= A(aP^2 + bP + cI) \\ &= \left[ \frac{162\left(\frac{4}{9}\right)^n - 180\left(\frac{1}{2}\right)^n + 18}{5} \right] AP^2 \\ &\quad - \left[ \frac{243\left(\frac{4}{9}\right)^n - 260\left(\frac{1}{2}\right)^n + 17}{5} \right] AP + \left[ \frac{81\left(\frac{4}{9}\right)^n - 80\left(\frac{1}{2}\right)^n + 4}{5} \right] A \end{aligned}$$

#### IV. 연구 결과

4단계를 통해서 만들어진 마르코프 체인을 활용해서 5단계는 학습패턴을 분석하는 단계이다. 결과를 해석하는 단계로 과목간 이동 성향에 따라 과목별 학습시간 비율이 주말을 제외하고  $5 \times n$ 일 후마다 어떻게 바뀔 수 있을지 예측할 수 있고, 여기서 찾은 예측값을 활용해서 자신의 학습 패턴을 파악하고 조정하는데 도움을 받을 수 있다.

1차 마르코프 체인을 시행한 결과는

$$AP = \begin{pmatrix} \frac{11}{154} & \frac{79}{154} & \frac{64}{154} \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{9} & \frac{7}{9} \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{2}{9} & \frac{7}{9} \end{pmatrix} = \begin{pmatrix} \frac{1}{28} & \frac{1303}{2772} & \frac{685}{1386} \end{pmatrix}$$

이고, 이 결과값  $\begin{pmatrix} \frac{1}{28} & \frac{1303}{2772} & \frac{685}{1386} \end{pmatrix}$ 는 초기상태 과목별 학습시간이 차지하는 비율인

$A = \begin{pmatrix} \frac{11}{154} & \frac{79}{154} & \frac{64}{154} \end{pmatrix}$ 에 대하여 과목별 이동 패턴인 전이행렬을 적용했을 때 초기상태

후 그 다음 주에 세 개의 교과목에 대한 학습패턴 예측값이라고 할 수 있다.

그렇다면 초기상태 후  $n$ 주 후 학습패턴은 어떤 변화를 가지게 될까?

$AP = \begin{pmatrix} \frac{1}{28} & \frac{1303}{2772} & \frac{685}{1386} \end{pmatrix}$ ,  $AP^2 = \begin{pmatrix} \frac{1}{56} & \frac{22007}{49896} & \frac{13499}{24948} \end{pmatrix}$ 을 계산하고,  $n \geq 1$ 에 대하여

$$x_n = \frac{162\left(\frac{4}{9}\right)^n - 180\left(\frac{1}{2}\right)^n + 18}{5}$$

$$y_n = \frac{-243\left(\frac{4}{9}\right)^n + 260\left(\frac{1}{2}\right)^n - 17}{5}$$

$$z_n = \frac{81\left(\frac{4}{9}\right)^n - 80\left(\frac{1}{2}\right)^n + 4}{5}$$

이라 놓으면,  $AP^n = x_n AP^2 + y_n AP + z_n A$ 로 정리할 수 있다. 다음 표는  $AP^n$  ( $1 \leq n \leq 10$ )을 계산한 결과값이다.

$n$ 주 후	$AP^n = x_n AP^2 + y_n AP + z_n A$	$(AP^n \text{의 결과값}) \sim (AP^n \text{의 근삿값})$ - 근삿값은 소수점 다섯 번째 자리에서 반올림 - $n$ 주 후 과목별 학습시간 비율은 국어, 수학, 영어의 순서로 기입)
$n=1$	$x_1=0, y_1=1, z_1=0$	$\begin{pmatrix} \frac{1}{28} & \frac{1303}{2772} & \frac{685}{1386} \end{pmatrix} \sim (0.0357 \ 0.4701 \ 0.4942)$
$n=2$	$x_2=1, y_2=0, z_2=0$	$\begin{pmatrix} \frac{1}{56} & \frac{22007}{49896} & \frac{13499}{24948} \end{pmatrix} \sim (0.0179 \ 0.4411 \ 0.5411)$
$n=3$	$x_3 = \frac{35}{18}, y_3 = -\frac{7}{6}, z_3 = \frac{2}{9}$	$\begin{pmatrix} \frac{1}{112} & \frac{380095}{898128} & \frac{255007}{449064} \end{pmatrix} \sim (0.0089 \ 0.4232 \ 0.5679)$
$n=4$	$\begin{cases} x_4 = \frac{847}{324} \\ y_4 = -\frac{221}{108} \\ z_4 = \frac{35}{81} \end{cases}$	$\begin{pmatrix} \frac{1}{224} & \frac{6673367}{16166304} & \frac{4710383}{8083152} \end{pmatrix} \sim (0.0045 \ 0.4128 \ 0.5827)$

$n = 5$	$\begin{cases} x_5 = \frac{17711}{5832} \\ y_5 = -\frac{5089}{1944} \\ z_5 = \frac{847}{1458} \end{cases}$	$\left( \frac{1}{448} \frac{118413007}{290993472} \frac{85965463}{145496736} \right)$ $\sim (0.0022 \ 0.4069 \ 0.5908)$
$n = 6$	$\begin{cases} x_6 = \frac{345079}{104976} \\ y_6 = -\frac{103649}{34992} \\ z_6 = \frac{17711}{26244} \end{cases}$	$\left( \frac{1}{896} \frac{2114525639}{5237882496} \frac{155875503}{2618941248} \right)$ $\sim (0.0011 \ 0.4037 \ 0.05952)$
$n = 7$	$\begin{cases} x_7 = \frac{6480719}{1889568} \\ y_7 = -\frac{1990489}{629856} \\ z_7 = \frac{345079}{472392} \end{cases}$	$\left( \frac{1}{1792} \frac{37896964351}{94281884928} \frac{28166153959}{47140942464} \right)$ $\sim (0.0006 \ 0.4020 \ 0.5975)$
$n = 8$	$\begin{cases} x_8 = \frac{119338759}{34012224} \\ y_8 = -\frac{37083137}{11337408} \\ z_8 = \frac{6480719}{8503056} \end{cases}$	$\left( \frac{1}{3584} \frac{680566317815}{1697073928704} \frac{508017048479}{848536964352} \right)$ $\sim (0.0003 \ 0.4010 \ 0.5987)$
$n = 9$	$\begin{cases} x_9 = \frac{2174367167}{612220032} \\ y_9 = -\frac{679834057}{204073344} \\ z_9 = \frac{119338759}{153055008} \end{cases}$	$\left( \frac{1}{7168} \frac{12235193826991}{30547330716672} \frac{9153937632151}{15273665358336} \right)$ $\sim (0.0001 \ 0.4005 \ 0.5993)$
$n = 10$	$\begin{cases} x_{10} = \frac{39391811767}{11019960576} \\ y_{10} = -\frac{12356439953}{3673320192} \\ z_{10} = \frac{2174367167}{2754990144} \end{cases}$	$\left( \frac{1}{14336} \frac{220092181609511}{549851952900096} \frac{164860708331087}{274925976450048} \right)$ $\sim (0.0000 \ 0.4003 \ 0.5997)$

$n$ 주 후 학습패턴을 분석해 보면, 초기상태인

$$A = \left( \frac{11}{154} \frac{79}{154} \frac{64}{154} \right) \sim (0.0714 \ 0.5130 \ 0.4156)$$

에서 10주 후 학습패턴은

$$AP^{10} = \left( \frac{1}{14336} \frac{220092181609511}{549851952900096} \frac{164860708331087}{274925976450048} \right)$$

$$\sim (0.0000 \ 0.4003 \ 0.5997)$$

로 변한다. 과목별 이동횟수가 적었던 국어과목의 경우는 예측이 가능한 수치이지만,

수학과 영어 과목의 경우는 시간에 따라 초기상태의 학습비율과 상당한 차이가 있는 결과로 예측이 된다. 상당히 흥미로운 결과라고 할 수 있다.

이 결과가 학습자의 의도인 부족한 과목에 대한 보충이라면 이 학습패턴을 유지하는 것이 좋지만, 그렇지 않다면 과목별 이동횟수를 변화시키는 노력을 통해 또 다른 전이행렬과 그에 따른 과목별 학습비율을 달성해야 한다는 패턴분석과 조정방법을 이끌어 낼 수 있다.

## V. 결론 및 제언

첫째, 마르코프 체인의 특성상 과거에 영향을 받지 않는다는 무기억 랜덤 프로세스의 특성상 현재의 선택이 추후에 미칠 영향에 대해서만 고려할 수 있다. 따라서 적용할 주제가 비교적 많다. 다만, 결과 예상값이 잘 맞지 않을 수 있으므로 적당한 보완은 필요하다. 따라서 예측값이 실질적으로 의미가 있는지 파악할 수 있는 방법을 공부해 본다.

둘째, 마르코프 체인에 적용한 학습패턴 분석의 조건이나 결론이 다소 미흡할지라도 행렬과 확률의 필요성에 흥미롭게 다가갈 수 있는 방법이었다. 공공데이터<sup>4)</sup>의 자료를 마르코프 체인에 적용해서 예측해 보면서 이 예측 시스템의 필요성을 확인해 본다.

## VI. 참고문헌

1. 김종민, 김민수, 김귀남 (2014), 마코프 체인을 이용한 모바일 악성코드 예측 모델링 기법 연구, 융합보안논문지 14(4), 19-26,
2. 이은아, 최희련, 이홍철 (2012), Markov Chain을 이용한 국내 폐차발생량 예측, Journal of the Korean Institute of Industrial Engineers, 38(3), 208-219,
3. 한영선 (2012), 마코프체인 분석을 이용한 범죄전이 연구, 형사정책연구 Korean Criminological Review 23(4), 257-275

---

4) <https://www.data.go.kr/>

수학A

## 05\_수학적 규칙을 이용하여 색상 배열 추천하기



# 수학적 규칙을 이용하여 색상 배열 추천하기

노하윤 (제주대학교 사범대학부설중학교 1학년)

신훈민 (중앙중학교 1학년)

장혜원 (신성여자중학교 1학년)

홍주원 (서귀포중학교 1학년)

지도교수 : 이호수 (제주대학교 초등교육과)

## I. 서론

### 1. 탐구 동기

디자인 사이트에 들어가 보면 추천하는 색상 배열들이 있다. 우리는 왜 사람들이 이런 색들을 좋아하는지 궁금했다. 또한 그 색들의 관계에도 수학적 관계가 있지 않을까하는 궁금증이 들었다.

### 2. 탐구 목적

모든 색상들은 빛의 삼원색의 선형결합으로 나타낼 수 있다. 따라서 각 색상은 삼원색이 어느 정도 포함되어 있는지 나타내는 성분 3개인 순서쌍으로 나타낼 수 있다. 이 연구에서는 모든 색상을 3차원 좌표(색공간)으로 표현하는 다양한 방법들에 대하여 조사해보고 그들 사이의 변환하는 방법에 대해 조사하였다. 좋은 디자인에는 색상 배열에서 규칙적인 특징을 가지고 있다. 우리는 좋은 색상 배열이 색공간에서 수학적으로 표현할 수 있는 규칙이 무엇인지 살펴보고 이를 이용하여 새로운 색상 배열 추천을 만들어 보았다.

## II. 이론적 배경

### 1. 삼원색

원색이란 다른 색을 섞어서 만들 수 없는 색을 의미한다. 색에는 3가지 원색이 있다. 3개의 원색을 조합하면 여러 가지 색이 나온다. 3원색을 기반으로 각 색을 혼합하여 만들어진 색을 2차색이라고 부르고, 2차색을 기반으로 만들어진 색을 3차색이라고 한다.

### 가. 빛의 삼원색

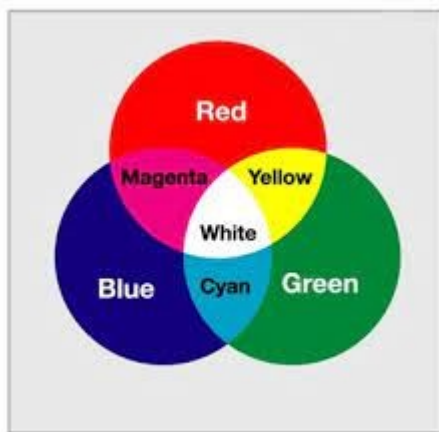
빛의 삼원색은 빨간색, 녹색, 파란색이다. 이 세 가지 색의 빛을 적절히 조절하면 모든 색을 표현할 수 있다. 그리고 빛의 삼원색을 혼합하는 과정을 가산혼합이라고 한다. 빛의 삼원색을 모두 섞으면 흰색이 되고, 빛이 없으면 검은색이 된다. 빛의 삼원색으로 색을 표현하는 방법을 RGB라고 한다.

가산혼합이란 빛의 삼원색으로 색을 만드는 방식을 의미한다. 빛이 전혀 없을 때가 검정이고 기본 3원색에서 더하는 방식으로 색상을 만들기 때문에 가산혼합이라고 말하는 것이다.

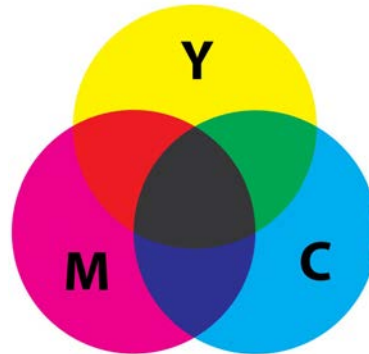
### 나. 색의 삼원색

색의 삼원색은 시안(Cyan), 마젠타(Magenta), 노란색(Yellow)이다. 색의 3원색을 여러 가지 비율로 섞으면 여러 색상을 만들 수 있다. 모든 색상을 섞으면 검은색 계열의 색이 나온다. 색의 삼원색으로 색을 만드는 방식을 감산혼합이라고 한다.

감산혼합은 CYM 혼합 또는 색료 혼합이라고 한다. 시안, 마젠타, 노랑은 색상을 빼내거나 흡수하는 방식으로 색을 만들어 내고, 합치면 합칠수록 어두워지는 성질을 띠고 있다.



<그림1. 빛의 삼원색>



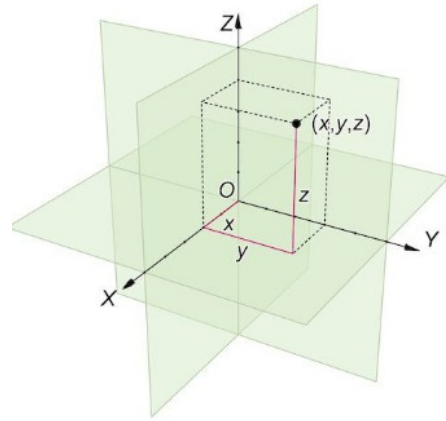
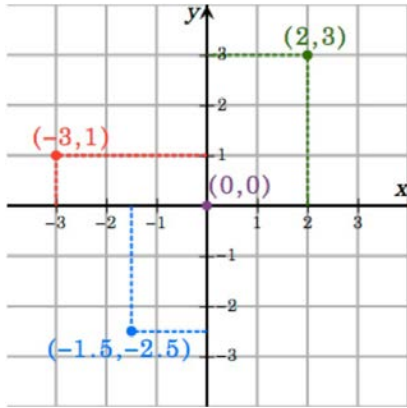
<그림2. 색의 삼원색>

## 2. 좌표계

### 가. 직교좌표계(Cartesian coordinate system)

직교좌표계는 각각의 축 또는 축 평면이 직교성을 유지하는 좌표계이다. 좌표축이 미리 정해진, 일반적으로 사용되는 좌표계이다. 각각의 축이 상호 독립적인 직교성을 가진다. 평면은 2개의 성분을, 공간은 3개의 성분으로 나타내어진다. 평행 이동과 대칭 이동에는 매우

유용한 좌표계이다. 데카르트가 개발하여 데카르트 좌표계로도 불린다.



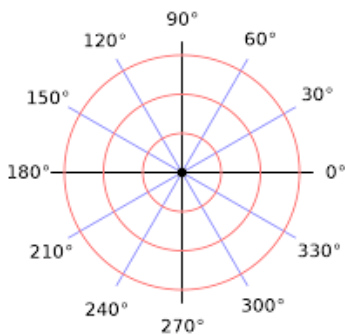
<그림3. 직교좌표계>

나. 극좌표계(Polar coordinate system)

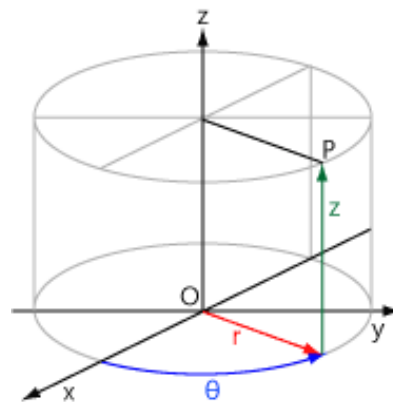
극좌표계는 평면 위의 위치를 각도( $\theta$ )와 거리( $r$ )를 써서 나타내는 2차원 좌표계이다. 극좌표계는 두 점 사이의 관계가 각이나 거리로 쉽게 표현되는 경우에 가장 유용하다. 특히 회전 이동에 대해 유용한 좌표계이다.

다. 원통좌표계(cylindrical coordinate system)

원통좌표계는 2차원 극좌표계에 평면( $r, \theta$ )의 수직인 방향의 높이( $z$ )를 결합하여 만든 3차원 좌표계이다.



<그림4. 극좌표계>

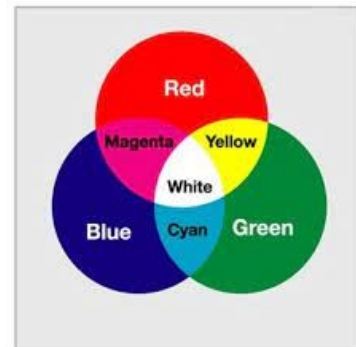


<그림5. 원통좌표계>

3. 색공간

### 가. RGB

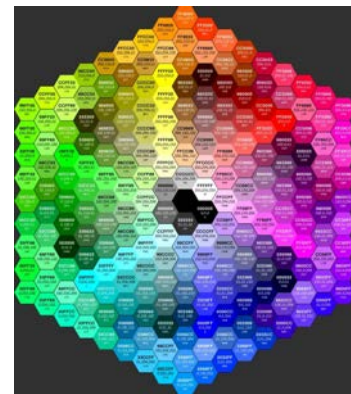
빛의 삼원색을 이용해 색을 표현하는 방식이다. 빛의 삼원색 빨간색, 파란색, 초록색을 0부터 255까지의 수로 표현할 수 있다. 0은 흰색을, 255는 원색을 의미한다. 어느 한 색상에 빛의 삼원색이 포함된 정도로 표현하면 색상 하나는 3차원 직교좌표계로 표현가능하다.



<그림6. RGB>

### 나. HEX

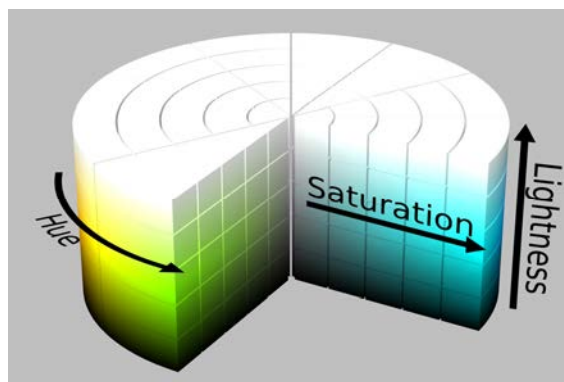
RGB 방식 색상 코드 표기법이다. #과 뒤에 붙는 여섯 자리 또는 세 자리의 숫자로 색상을 표기하는 방식이다. 여섯 자리인 경우는 두 자리씩 끊어서 각각 RGB의 강도를 256단계에 걸쳐 나타낸 것이며, 각 두 자리수는 16진수이므로 00일 때 가장 어둡고 FF일 때 가장 밝다. 또한 헥스 코드로 나타낼 수 있는 색상의 수는 총  $16^6$ , 즉 1677만 7216가지가 된다.



<그림7. HEX>

### 다. HSL

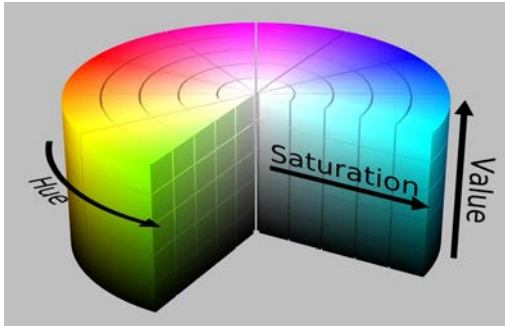
색상(Hue), 채도(Saturation), 밝기(Lightness)로 이루어진 색 공간을 나타낸다. 색상은 원기둥의 방향을 나타내고 채도는 반지름을, 마지막으로 밝기는 높이를 나타낸다. HSL은 원통좌표계에서 나타낼 수 있다. 색상의 채도가 높을수록 점은 원기둥의 겉면에 위치하게 된다. 밝기는 위로 갈수록 밝아지고 윗면은 하얀색을, 밑면은 검은색으로 구성되어 있다.



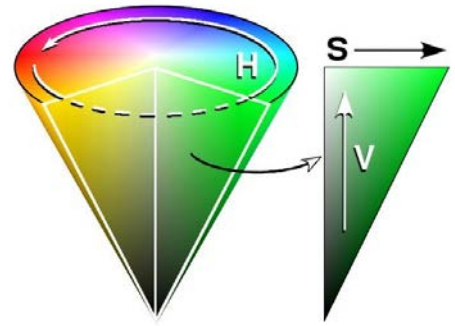
<그림8. HSL>

### 라. HSV

색상, 채도, 명도의 좌표를 이용해 색을 지정하는 방법이다. 색상 값은 각도( $\theta$ )로 표현되며 원기둥의 수평 단면의 방향에 따라 알 수 있다. 채도는 원기둥의 반지름에 해당한다. 정 중앙에 무채색이 위치하며 원기둥의 길면으로 갈수록 진한 채도를 갖는다. 명도는 높이에 해당하며 위로 갈수록 색이 밝아진다. HSV는 원뿔에서도 나타낼 수 있다. 이때 원뿔의 꼭짓점은 명도가 0%이기 때문에 검은색이며 명도가 높을수록 채도 값의 폭은 줄어든다.



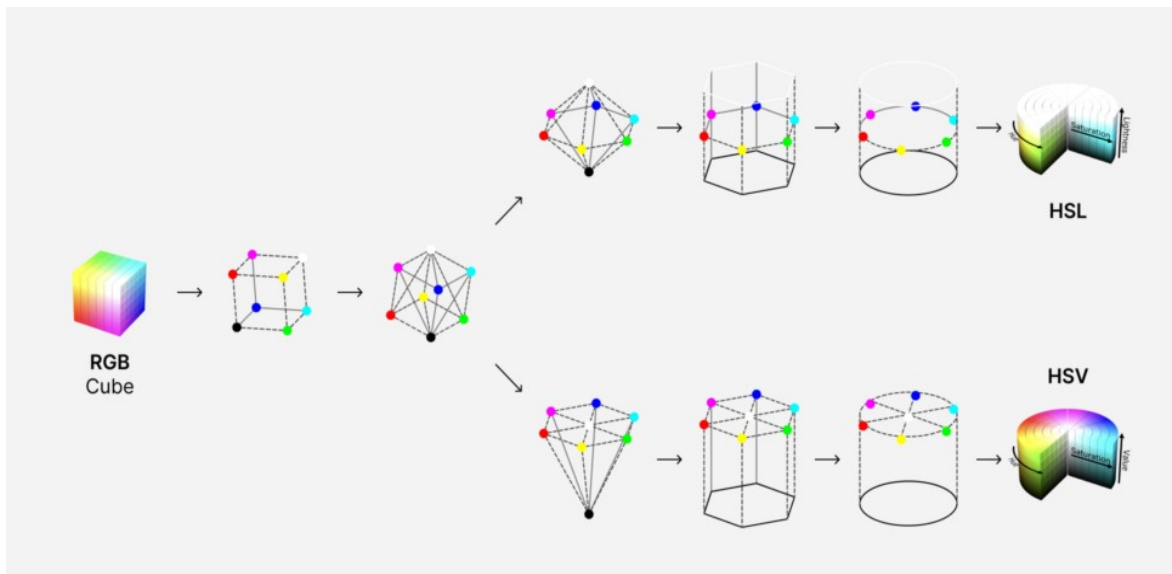
<그림9. HSV 원기둥 모형>



<그림10. HSV 원뿔 모형>

#### 4. RGB를 HSV 혹은 HSL으로 변환하는 과정

RGB 빛의 삼원색으로 표현가능한 색상의 집합은 3차원 직교좌표계에서 정육면체로 표현된다. 이 정육면체를 기울인다. 거기서 육각뿔 모양으로 변환되고 육각기둥, 원기둥 순으로 모양이 바뀌어 HSV를 만들게 된다. HSL은 육각뿔 모양을 2개로 합친 뒤 이를 육각기둥, 원기둥으로 변환해 만들 수 있다.



<그림11. RGB를 HSV 혹은 HSL으로 변환하는 과정>

가. RGB를 HSV 혹은 HSL으로 변환하는 과정을 수식으로 표현하기

(1단계) 0~255 사이의 RGB 색상을 0과 1사이의 수로 바꾸기

R'을 R/255라고 할 때, G'과 B'도 같은 방법으로 표현해본다.

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

(2단계) R', G', B'의 최댓값 Cmax, 최솟값 Cmin, 그리고 이들의 차를 Δ로 두자.

$$C_{max} = \max(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

$$\Delta = C_{max} - C_{min}$$

(3단계) 색상의 방향 H를 구하기

$$H = \begin{cases} 0, & \Delta = 0 \\ 60^\circ \times \left( \frac{G' - B'}{\Delta} \bmod 6 \right), & C_{max} = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right), & C_{max} = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right), & C_{max} = B' \end{cases}$$

(4단계) 채도 S와 색상값 V 구하기

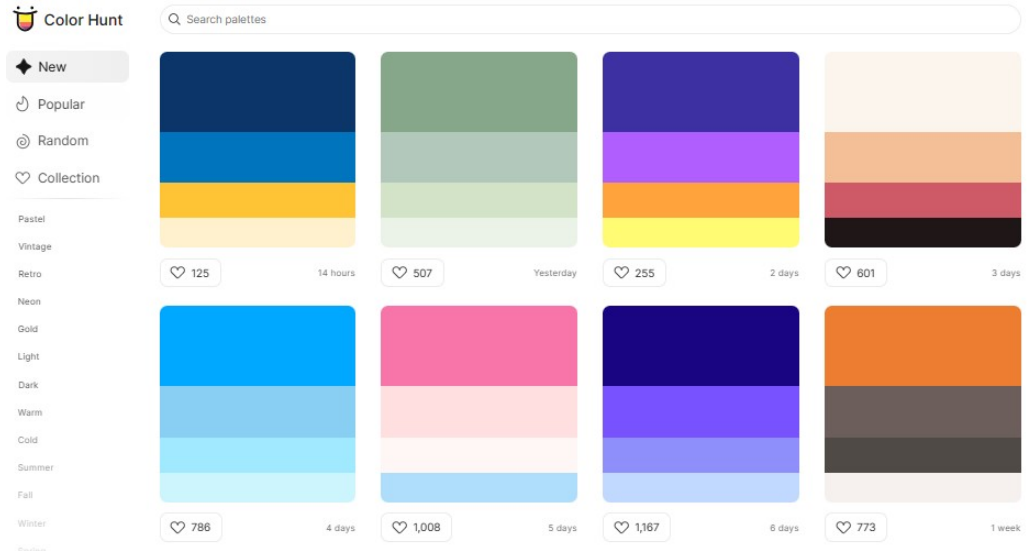
$$S = \begin{cases} 0, & C_{max} = 0 \\ \frac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases}$$

$$V = C_{max}$$

### III. 연구 방법

- 추천 색상 조합의 수학적 규칙찾기 실험 6단계

(1단계) 색상 조합 추천 사이트인 Color Hunt (<https://colorhunt.co>)에서 색상 조합들을 수집한다.



(2단계) 그림판을 이용하여 수집된 색상 조합에 대한 RGB 정보를 찾는다.



(3단계) 단위 변환에 대한 정보를 소개하는 사이트인 Rapid Tables 에서 RGB를 HSV로 색상 변환한다. (<https://www.rapidtables.com/convert/cplor/rgb-to-hsv.html>)

**RapidTables** Q Search

Home > Conversion > Color conversion > RGB to HSV

### RGB to HSV color conversion

Enter 6 digits hex code or enter red, green and blue color levels (0..255) and press the Convert button:

Enter RGB hex code (#):

or

Enter red color (R):

Enter green color (G):

Enter blue color (B):

Hue (H):  °

Saturation (S):  %

Value (V):  %

Color preview:

[HSV to RGB conversion ▶](#)

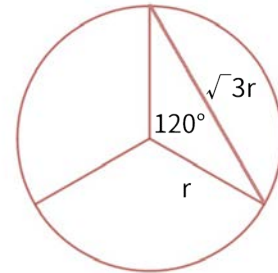
(4단계) HSV 값을 아래와 같이 표로 정리하였다.

H	S	V
251	80.8	40.8
313	91.8	62.4
331	92.7	91.8
323	37.2	96.9
249	49.5	75.3
257	43.1	91
178	40.1	91
141	16.5	100
256	54.2	32.5
256	54.2	83.9
245	45.2	97.3
353	17.6	100

H	S	V
60	56.3	80.8
46	51.8	100
31	92.1	99.2
5	97.5	77.3
345	35.9	98.4
348	18	100
340	5.9	100
35	33.3	100
126	53.7	68.6
82	59	89.8
55	81.1	99.6
44	76.5	100

(5단계) 표의 값에서의 규칙을 조사해보았다.

H	S	V
251	80.8	40.8
313	91.8	62.4
331	92.7	91.8
323	37.2	96.9

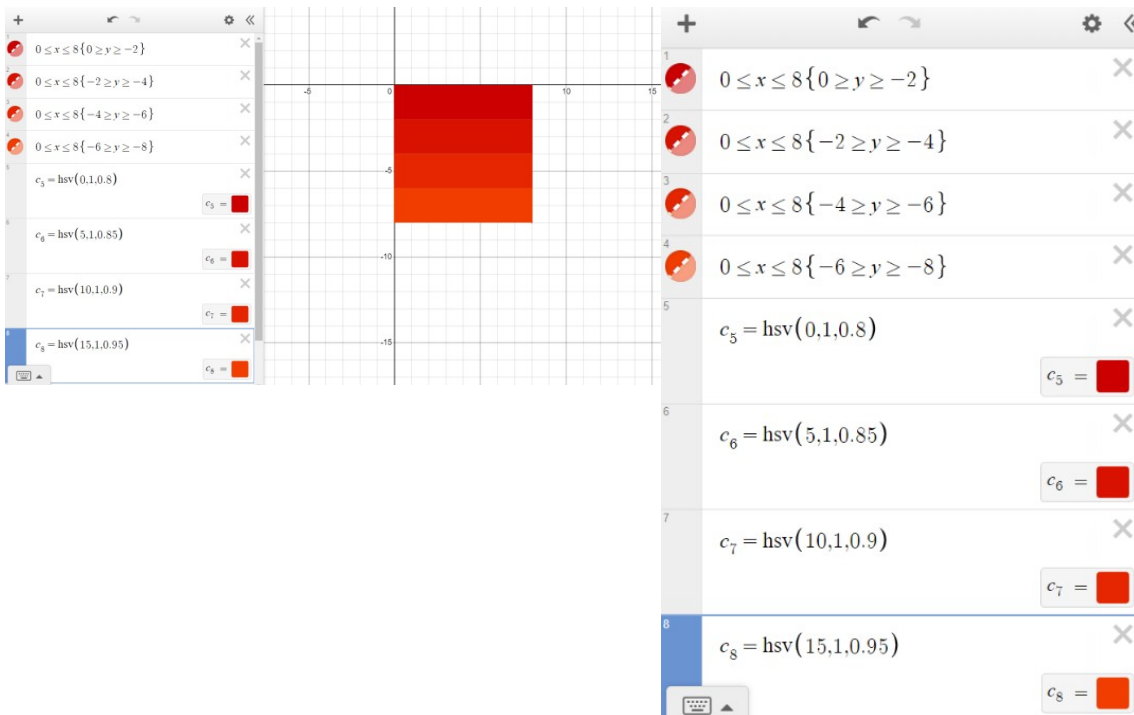


(규칙 1) H의 최댓값과 최솟값이 이루는 각의 크기가 120도를 넘지 않으며 HSV 공간에서  $\sqrt{3}$ 보다 작은 거리를 가진다. 다시 말하여, 추천된 색상 배열은 유사한 색상으로 구성되었다.

(규칙 2) V의 값은 순차적으로 증가하거나 혹은 감소한다.

(6단계) (5단계)에서 찾은 규칙을 이용하여 Desmos에서 색상을 직접 조합해보았다.

<https://www.desmos.com>



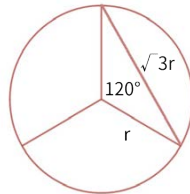
#### IV. 연구 결과

1. “Color Hunt”에서 색상 배열들을 찾고 “Rapid Tables” 사이트를 이용하여 RGB 값을 HSV 값으로 나타내었다.(<https://colorhunt.co/>)



2. HSV 값을 적은 표를 이용하여 다음과 같이 3가지 규칙을 찾았다.

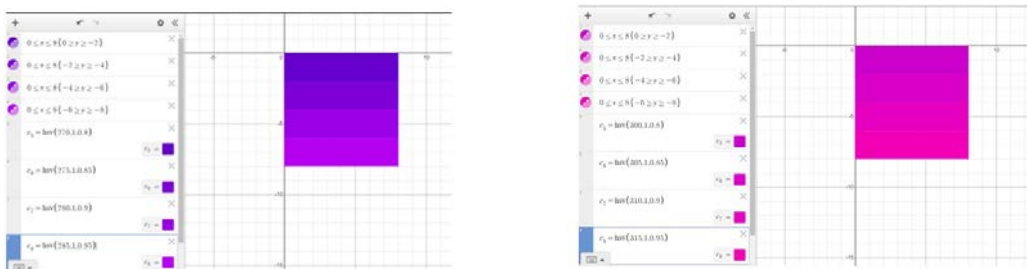
- (1) H의 최댓값과 최솟값이 이루는 각이  $120^\circ$ 를 넘지 않는다.
- (2) H의 최댓값과 최솟값을 나타낸 색상 사이의 거리는  $\sqrt{3}r$ 보다 작다.



- (3) 규칙 1,2가 성립할 때 V의 값이 순차적으로 변화한다.

H(°)	S(%)	V(%)
251	80.8	40.8
313	91.8	62.4
331	92.7	91.8
323	37.2	96.9

3. “Desmos”라는 그래프를 그리는 사이트에서 규칙들을 이용해 색상을 직접 조합했다. (<https://www.desmos.com/calculator?lang=ko>)



## V. 결론 및 제언

### 결론

1. 공간좌표에서 색을 수학적으로 표현하고, 3가지 규칙을 이용해 새로운 색상배열을 만들 수 있었다.
2. RGB 색공간보다 HSV 색공간을 이용했을 때 규칙을 찾기 더 용이하였다.
3. 색상배열에서 다음과 같은 3가지 규칙을 찾을 수 있었다.

(규칙 1) H의 최댓값과 최솟값이 이루는 각의 크기가 120도를 넘지 않는다.

(규칙 2) HSV 공간에서  $\sqrt{3}$ 보다 작은 거리를 가진다.

(규칙 3) (규칙 1), (규칙 2)가 만족할 때 V의 값은 순차적으로 변화한다.

즉, 다시 말하여 색상배열들은 유사한 색상들로 이루어져 있다는 점을 알 수 있다.

### 제언

본 연구는 Color Hunt에서만 제시된 색상배열들을 다루었기 때문에 다른 사이트에서 제시된 색상배열에서는 앞서 찾은 규칙이 성립을 하지 않을 수도 있다는 점이 가장 큰 문제점이다. 이에 본 연구의 연구 방법을 바탕으로 더 많은 정보를 수집하여 더 많은 색상배열들에 적용할 수 있는 규칙을 찾는 후속 연구를 제안한다.

## VI. 참고문헌

-Color Hunt (<http://colorsunt.co>)

-Rapid Tables (<https://www.rapidtables.com/convert/color/rgb-to-hsv.html>)

-Desmos (<http://www.desmos.com>)



수학A

06\_K-Means를 활용한 2024 KBO FA 선수  
분석



# K-Means를 활용한 2024 KBO FA 선수 분석

김규민 (제일여자중학교 2학년)

김바다 (반송중학교 2학년)

김연아 (울하중학교 2학년)

김태준 (삼정자중학교 2학년)

지도교수 : 서셋별 (경남대학교 수학교육과)

순간적인 힘 또는 그날의 운으로만 운영했던 과거의 야구와 달리 오늘날의 야구는 데이터를 활용하여 체계적으로 선수의 기량 컨디션을 분석하여 경기의 승률을 예측 할 수 있다. 데이터 기반으로 알고리즘 스스로가 결론을 도출해 내는 머신러닝에 대해 관심을 가졌으며, 그 중, K-Means는 두 점사이의 거리를 근거로 데이터의 유사성을 계산하고 클러스터링(군집화, Clustering)하는 비지도학습의 하나의 방법이라는 것을 알게 되었다. 본 연구는 K-Means를 활용하여 2024년 KBO에서 FA 자격을 얻게 된 선수 32명에 대한 데이터를 수집하고, 두 점사이의 거리를 이용해 데이터간의 유사도 및 클러스터의 중심을 찾고, 3개의 그룹으로 나누는 클러스터링(Clustering)과정을 통하여 FA선수들을 상, 중, 하위 그룹으로 분류해 보았으며, 이를 통해 가중치를 두어 FA 선수를 30점 만점으로 점수를 측정하였다.

## I. 서론 (또는 연구의 필요성 및 목적)

지난 40여년간 KBO에는 선동열, 류현진, 이승엽 등 수많은 선수들이 각자의 팀을 이끌었다. 이 선수들이 있는 동안, 그들의 팀은 거의 선두에 있었다. 이처럼 선수들의 능력은 팀을 승리로 이끌 것인지 패배로 이끌 것인지를 크게 좌우한다. 순간적인 힘 또는 그날의 운으로만 운영했던 과거의 야구와 달리 오늘날의 야구는 데이터를 활용하여 체계적으로 경기를 운영되고 있다. 데이터 분석이 경기의 결과에 큰 영향을 줄 수 있다. 또한 야구를 관람하고 팀을 응원하는 팬들도 데이터를 활용하여 경기를 예측하곤 한다. 우리도 프로야구팀들이 사용하는 데이터들을 바탕으로 선수와 팀의 기량을 분석해보려고 한다. 2023 시즌 종료 후 FA 자격을 얻게 된 32명의 선수들의 성적 및 정보들을 바탕으로 등급을 나누어 FA 선수들을 분석해보려고 한다.

## II. 이론적 배경

### 가. KBO 및 FA제도

KBO란 1982년에 출범한 대한민국의 프로야구 리그이다. 총 10개의 구단, 12개의 구장에서 리그가 진행되고 있다. 매년 3월 말~4월 초에 시작해서 빠르면 10월 초, 늦

어도 10월 말에 끝난다. 팀당 144경기를 치르며 매주 월요일을 빼고는 화요일부터 일요일까지 빠지지 않고 경기가 열린다. 리그 내 시즌 순위는 승률 순으로 산정하며 승률 순으로 상위 5개 팀이 KBO 포스트시즌에 진출한다. (포스트 시즌은 KBO 리그의 정규시즌 종료 후 리그 상위 5위권에 들어간 팀들이 최종 우승팀을 결정하기 위해 벌이는 경기이다.) 구단으로는 SSG 랜더스, 키움 히어로즈, LG 트윈스, KT wiz, KIA 타이거즈, NC 다이노스, 삼성 라이온즈, 롯데 자이언츠, 두산 베어스, 한화 이글스로 총 10개의 구단이 있다. 자유계약(free agent, FA)이란 일정기간 자신이 속한 팀에서 활동한 뒤에 다른 팀과 자유롭게 계약을 맺어 이적할 수 있는 자유계약선수 또는 그 제도를 일컫는다. 국내 프로야구에서는 1999년 시즌부터 FA제도가 시행되었으며, 타자의 경우 등록일수 145일 이상, 매 시즌 경기 수의 3분의 2 이상 출장하고, 투수는 매 시즌 규정 이닝의 3분의 2 이상 등판한 시즌이 각각 9시즌을 넘어야 FA 자격을 가질 수 있도록 제한되어 있다. 대졸 선수는 8시즌이며, 재 자격 취득은 4시즌이다.

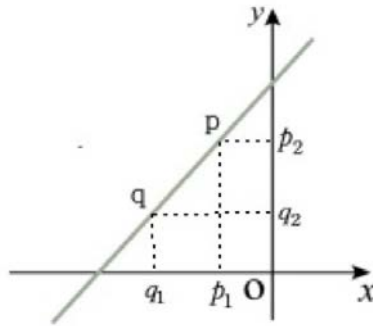
#### 나. 벡터

수학의 관점에서 벡터는 일반적으로 추상적인 벡터공간을 구성하는 원소이다. 물리학과 공학에서는 벡터공간이 유클리드 공간인 경우를 대부분 다루기 때문에, 수학에서의 일반적인 의미가 제한되어 사용되고 있다고 할 수 있다. 이런 맥락에서, 물리학에서의 벡터는 n 차원의 유클리드 공간에서 크기와 방향을 가지고 있는 기하학적 대상으로, 때로는 유클리드 벡터(Euclidean vector), 기하 벡터(geometric vector), 공간 벡터(spatial vector)라고도 부른다. 벡터는 힘, 위치, 속도 등과 같이 크기와 방향을 갖는 물리량을 표현하는 데 편리하기 때문에, 물리학을 기술할 때 매우 기본적인 역할을 한다. 벡터로 표현되는 물리량을 벡터 량이라고 부른다. 또한 벡터는 머신러닝분야에서 입력 데이터를 표시하는 방법으로 사용된다.

#### 다. 유클리드 거리

본 연구의 클러스터링과 선수 분류를 위해 데이터를 처리하여 벡터 형태로 표현하게 된다. 즉, 데이터를 벡터로 표현한 후 이를 비교하여 유사성을 계산한다. 유사성을 계산하기 위해 여러 가지 방법이 있지만 본 연구에서는 유클리드 거리 방법을 적용하였다. 순서를 가지는 n개의 실수의 쌍  $(x_1, x_2, \dots, x_n)$ 을 점이라 하며, 두 점  $(x_1, x_2, \dots, x_n), (y_1, y_2, \dots, y_n)$ 간의 거리를  $\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$ 으로 정의한 공간을 n차원 유클리드 공간이라 한다. 유클리드 거리는 유클리드 공간상에서 두 점을 잇는 선분의 길이이다. 2차원 유클리드 평면인 [그림 1]에서는 다음과 같이 나타낼 수 있다.

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$



[그림 2] 두 점 p, q 사이의 거리

#### 라. 머신러닝(Machine Learning, ML)

머신러닝 또는 기계 학습은 컴퓨터 과학 중 인공지능의 한 분야로, 패턴인식과 컴퓨터 학습 이론의 연구로부터 진화한 분야이다. 머신러닝은 경험적 데이터를 기반으로 학습을 하고 예측을 수행하고 스스로의 성능을 향상시키는 시스템과 이를 위한 알고리즘을 연구하고 구축하는 기술이라 할 수 있다. 머신러닝의 알고리즘들은 엄격하게 정해진 정적인 프로그램 명령들을 수행하는 것이라기보다, 입력 데이터를 기반으로 예측이나 결정을 이끌어내기 위해 특정한 모델을 구축하는 방식을 취한다. 머신러닝의 알고리즘은 크게 지도학습, 비지도 학습으로 구분된다. 지도학습은 훈련 데이터로부터 하나의 함수를 유추해내기 위한 기계 학습이다. 훈련 데이터로부터 하나의 함수가 유추되면 함수에 대한 평가를 통해 파라미터를 최적해나간다. 비지도 학습은 기계 학습의 한 종류이며, 데이터의 구성을 파악하는데 사용된다. 비지도 학습은 입력 값에 대한 목표치가 주어지지 않는다. 본 연구에서는 이 비지도 학습에 속하는 분석 방법인 K-means를 활용했다. k-평균 알고리즘은 처음에는 클러스터 중심을 정하고 클러스터를 만든다. 그 다음 클러스터의 중심을 이동하고 다시 클러스터를 만드는 식으로 반복해서 최적의 클러스터를 구성하는 알고리즘이다. 클러스터 중심이란 k-평균 알고리즘이 많은 클러스터에 속한 샘플의 특성 평균값이다. k-평균 알고리즘의 단점은 클러스터 개수를 사전에 지정해야 한다는 것이다. k-평균 알고리즘의 작동 방식은 다음과 같다. k-평균 알고리즘은 클러스터에 개수 k값에 따라 결과 값이 달라진다는 한계점이 있다. 그렇기에 가장 적절한 k값을 구하여 클러스터링을 진행해야 한다.

### III. 연구 방법

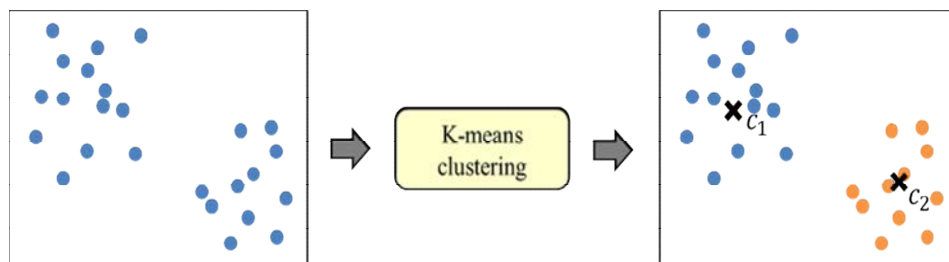
#### 가. 데이터 수집 및 분석 절차

데이터는 KBO 공식 홈페이지 기록실([koreabaseball.com](http://koreabaseball.com))에서 타자, 투수의 기록

데이터를 사용하였다. KBO 공식 기록 데이터에는 타자의 순위, 이름, 팀명, 타율, 경기 수, 타석, 타수, 득점 수, 안타 수, 2루타 수, 3루타 수, 홈런 수, OPS가 기록되어 있으며 투수는 순위, 선수명, 팀명, 평균자책점, 경기 수, 승리 수, 패배 수, 세이브 수, 홀드 수, 승률, 이닝 수, 피안타 수, 홈런 수, 볼넷 수, 사구 수, 삼진 수, 실점 수, 자책점 수, 이닝당 출루허용률(WHIP) 등의 기록을 제공한다. 본 연구는 2023시즌 종료 후 2024년 FA 대상인 32명의 선수들의 기록 중 필요한 데이터들을 추출하여 사용하여 데이터 셋을 구축하고 Python을 통해 K-means를 이용하여 FA선수 데이터간 유사도를 계산하고 기량에 따라 세 그룹으로 나누어질 수 있도록 하였다.

#### 나. K-means 알고리즘(K-means clustering algorithm)

K-means 알고리즘은 클러스터링 방법 중 분할법에 속한다. 즉, K-means 알고리즘의 목적은 각 클러스터와의 거리 차이 분산을 최소화하여 데이터를 분류하는 것이다. 분할법은 주어진 데이터를 여러 그룹(파티션)으로 나누는 방법이다. 클러스터링(clustering)은 지도학습 분류(Classification)와 다른 비지도 학습이다. 분류는 미리 레이블(label)이 붙어 있는 데이터들을 학습해서 그걸 바탕으로 새로운 데이터에 대해 분류를 수행하지만, 클러스터링은 레이블(label)을 모르더라도 그냥 비슷한 속성을 가진 데이터들끼리 묶어주는 역할을 하기 때문이다. 또한, 클러스터링(clustering)은 많은 자료들을 비슷한 것끼리 묶어 다른 그룹과 구분하는 기법을 말한다. 즉, 같은 그룹의 자료들의 동질성을 향상시키고 다른 그룹의 자료들과는 차별성을 향상시키는 방법이다. 이를 통해 자료들 속에 숨어 있는 규칙과 패턴을 찾는 방법을 말한다. K-means 알고리즘은 각 그룹의 중심과 그룹 내의 데이터 오브젝트와의 거리의 제곱합을 비용 함수로 정하고, 이 함수 값을 최소화하는 방향으로 각 데이터 오브젝트의 소속 그룹을 업데이트 해 줌으로써 클러스터링을 수행하게 된다. K-평균 알고리즘은 [그림 2]와 같이 처음에는 클러스터 중심을 정하고 클러스터를 만든다. 그 다음 클러스터의 중심을 이동하고 클러스터를 만드는 식으로 반복해서 최적의 클러스터를 구성하는 알고리즘이다. 클러스터 중심이란 K-평균 알고리즘이 많은 클러스터에 속한 샘플의 특성 평균값이다. K-평균 알고리즘의 단점은 클러스터 개수를 사전에 지정해야 한다는 것이다.



[그림 3] K-means 클러스터링 과정

K-means 알고리즘의 작동 방식은 다음과 같다.

첫째, 무작위로 k개의 클러스터 중심을 정한다. - ①

둘째, 각 샘플에서 가장 가까운 클러스터 중심을 찾아 해당 클러스터의 샘플로 지정한다. - ②

셋째, 클러스터에 속한 샘플의 평균값으로 클러스터 중심으로 변경한다. - ③

마지막으로, 클러스터 중심의 변화가 없을 때까지 ②번으로 돌아가 반복한다. - ④

K-means 알고리즘은 클러스터에 개수 k값에 따라 결과 값이 달라진다는 한계점이 있다. 그렇기에 가장 적절한 k값을 구하여 클러스터링을 진행해야 한다.

K-means 알고리즘은 주어진 데이터를 k개의 클러스터로 묶는 알고리즘이고, 각 클러스터까지의 거리 차이의 분산을 최소화하는 방식으로 동작한다. 이 알고리즘은 자율학습의 일종으로, 레이블이 달려 있지 않은 입력 데이터에 레이블을 달아주는 역할을 한다. 예를 들어 n개의 데이터 오브젝트를 입력받았다고 가정하자. 이 때 분할법은 입력 데이터를 n보다 작거나 같은 k개의 그룹으로 나누는데, 이 때 각 그룹은 클러스터를 형성하게 된다. 다시 말해, 데이터를 한 개 이상의 데이터 오브젝트로 구성된 k개의 그룹으로 나누는 것이다. 이 때 그룹을 나누는 과정은 거리 기반의 그룹 간 비유사도와 같은 비용 함수 (cost function) 을 최소화하는 방식으로 이루어지며, 이 과정에서 같은 그룹 내 데이터 오브젝트끼리의 유사도는 증가하고, 다른 그룹에 있는 데이터 오브젝트와의 유사도는 감소하게 된다. n개의 d-차원 데이터 오브젝트 ( $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ ) 집합이 주어졌을 때, k-평균 알고리즘은 n개의 데이터 오브젝트들을 각 집합 내 오브젝트 간 응집도를 최대화 하는  $k(\leq n)$  개의 집합  $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$  으로 분할한다. 다시 말해,  $\mu_i$ 가 집합  $S_i$ 의 중심점이라 할 때

$$\arg \min_s \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

각 집합별 중심점에서 집합 내 오브젝트 간 거리의 제곱합을 최소로 하는 집합  $\mathbf{S}$ 를 찾는 것이 이 알고리즘의 목표다.

#### 다. 프로그래밍 정보

Python은 웹 애플리케이션, 소프트웨어 개발, 데이터 과학, 기계 학습(ML)에 널리 사용되는 프로그래밍 언어이다. Python의 이점은 다른 많은 언어에 비해 더 적은 코드 줄을 사용하여 Python 프로그램을 작성할 수 있기 때문에 개발자의 생산성을 높일 수 있다는 것이고 기본적인 영어와 유사한 구문을 가지고 있기 때문에 개발자가 쉽게 읽고 이해할 수 있다. Colab이란 웹 브라우저에서 텍스트와 프로그램 코드를 자유롭게 작성할 수 있는 일종의 온라인 텍스트 에디터이다. 그리고 코랩에서 사용할 수 있는 프로그래밍 언어가 바로 'Python'이다

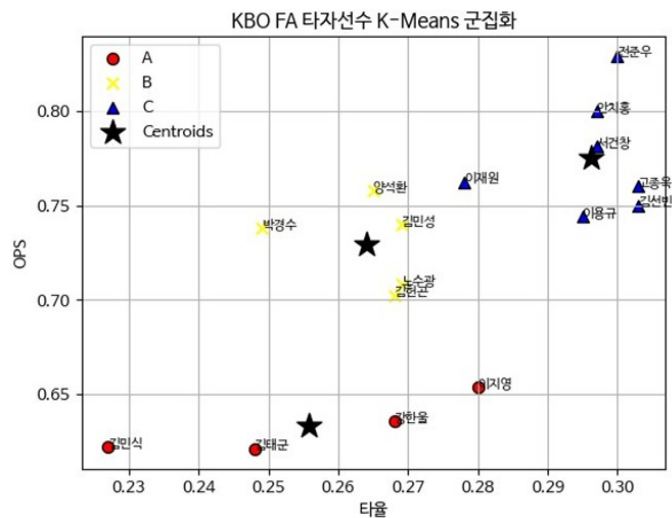
#### IV. 연구 결과

##### 가. 연구 순서

K-means를 활용하여 2024 FA 선수 분석을 하기 위해 두 가지 절차를 거쳐 진행하였다. 먼저 필요한 데이터를 모아 데이터셋을 구축하고, 구축한 데이터셋을 K-means를 활용해서 군집화를 시켜보는 것이다. 먼저, 2023시즌 종료 후 FA 대상인 32명의 선수들 기록 중 필요한 데이터(타자 : 이름, 타율, 나이, OPS / 투수 : 이름, 나이, 평균자책점, 이닝당 출루허용률)들만 추출하 K-means 알고리즘을 사용하였다. 구축한 데이터를 Google colab으로 부른 후 K-means 분석을 진행하였다. 데이터 셋에서 타자와 투수를 나누어서 각각 16명씩 K-means 분석을 진행하였다. 두 변수를 기준으로 하여 분석을 하였다. (타자 : 나이, 타율 / 나이, OPS / 타율, OPS / 투수 : 나이, 평균자책점 / 나이, WHIP / 평균자책점, WHIP) 중간점은 3개로 두고 최적의 중간점을 찾기 위해 반복하였다. 최적의 중간점을 찾고 군집화를 종료하였다.

##### 나. 연구 결과

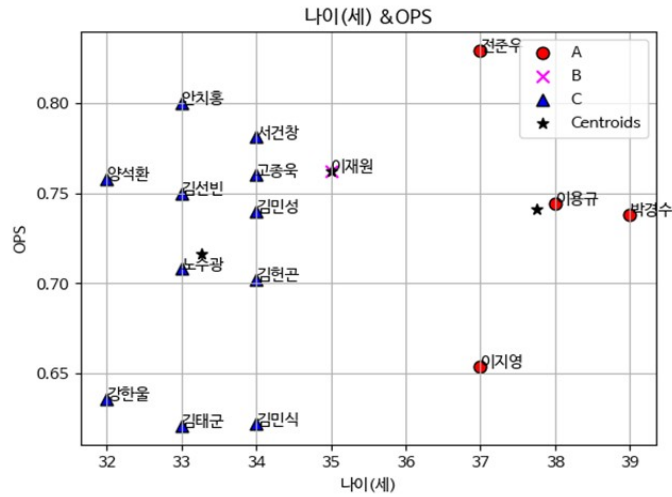
먼저, 2024 FA타자 각각의 나이 타율 OPS를 사용하여 분석해서 군집화한 결과를 바탕으로 2024 FA 타자 등급을 나눠보았다. [그림 3]과 같이 타율과 OPS를 가지고 군집화를 했을 때 A군집은 김민식, 김태군, 김한울, 이지영으로 구성되었고, B군집으로는 박경수, 양석환, 김현곤, 김민성, 노수광선수가, C군집으로는 이재원, 이용규, 안치홍, 서건창, 전준우, 고종욱, 김선빈이 한 그룹으로 나누어졌다. 여기서 A군집을 하위권, B군집을 중위권, C군집을 상위권이라 정의하였다.



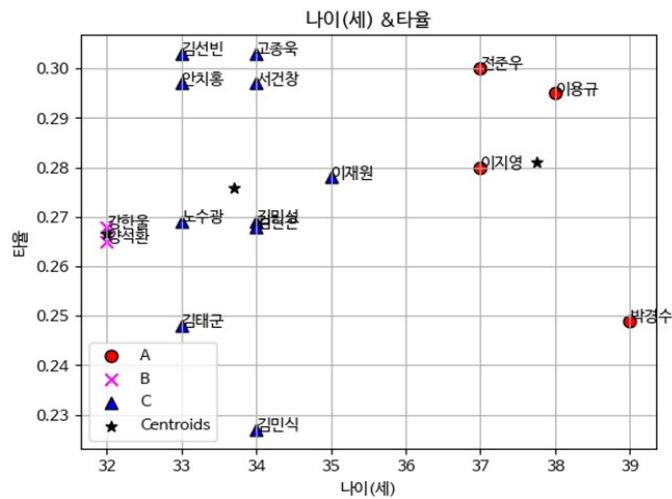
[그림 4] 타율/OPS 기반 KBO FA 타자 Clustering

[그림 4]와 같이 나이와 OPS를 가지고 군집화를 했을 때, A군집은 전준우, 이지영, 이용규, 박경수 B군집은 이재원, C군집은 김한울, 양석환, 안치홍, 김선빈, 노수광, 김

태균, 서건창, 고종욱, 김민성, 김현곤, 김민식과 같이 분석이 되었다. 해당 클러스터링에서는 나이가 적을수록 상위그룹이라 판단하여 A군집을 하위권, B군집을 중위권, C군집을 상위권이라 정의하였다.



[그림 5] 나이/OPS 기반 FA 타자 Clustering



[그림 6] 나이/타율 기반 FA 타자 Clustering

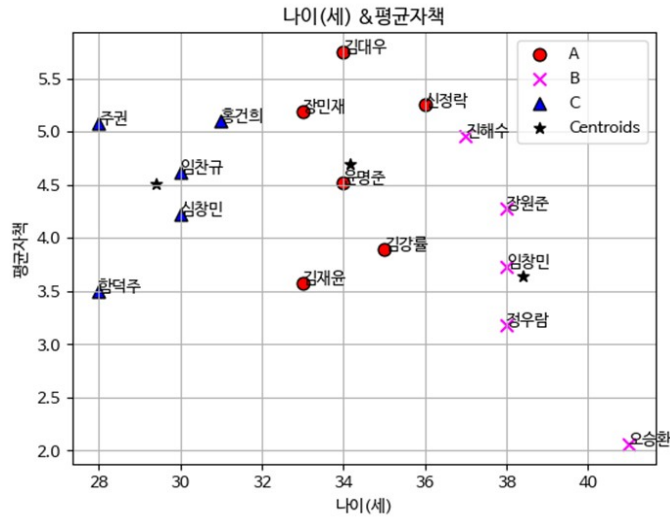
[그림 5]는 타율과 나이를 통한 클러스터링 결과로 A군집은 전준우, 이지영, 이용규, 박경수, B군집은 강한울, 양석환, C군집은 김선빈, 안치홍, 노수광, 김태균, 고종욱, 서건창, 김민성, 김민식, 김현곤, 이재원과 같이 분석이 되었다. [그림 4]와 같이, A군집을 하위권, B군집을 중위권, C군집을 상위권이라 정의하였다.

세 군집화를 바탕으로 상위권에는 30점, 중위권에는 20점, 하위권에는 10점을 준 뒤 타율&OPS에는 가중치 0.6, 나이&OPS에는 가중치 0.3, 나이&타율에는 가중치 0.1을 두고 선수별 기량점수를 측정하였다.

이름	강한올	김민식	김태군	이지영	박경수	양석환	김현곤	김민성	노수광	이재원	이용규	안지홍	서건창	전준우	고종욱	김선빈
타율&OPS	하위권	하위권	하위권	하위권	중위권	중위권	중위권	중위권	중위권	상위권	상위권	상위권	상위권	상위권	상위권	상위권
나이&OPS	상위권	상위권	상위권	하위권	하위권	상위권	상위권	상위권	상위권	중위권	하위권	상위권	상위권	하위권	상위권	상위권
타율&나이	중위권	상위권	상위권	하위권	하위권	중위권	상위권	상위권	상위권	상위권	하위권	상위권	상위권	하위권	상위권	상위권
점수(30점)	17	18	18	10	16	23	24	24	24	27	22	30	30	22	30	30

[그림 7] 클러스터링(상, 중, 하)별 가중치를 부여한 FA 타자 선수의 점수 측정

다음은, 2024 FA투수 각각의 나이 평균자책점 WHIP를 사용하여 분석해서 군집화 한 것을 바탕으로 2024 FA 타자 등급을 나누어보았다.

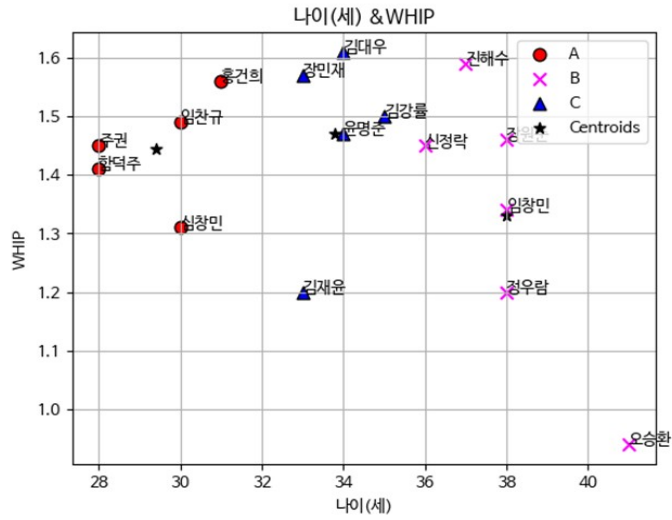


[그림 8] 나이/평균자책 기반 FA 타자 Clustering

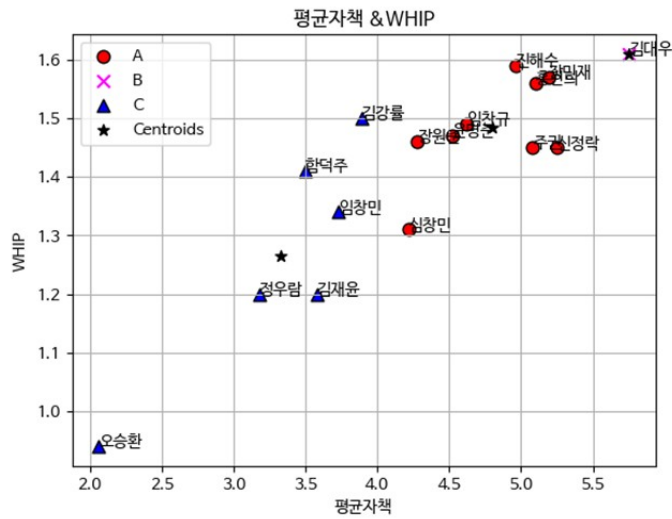
[그림 7]은 나이와 평균자책점을 가지고 군집화한 결과이다. A군집으로는 주권, 함덕주, 임찬규, 홍건희, 심창민, B군집으로 장민재, 김재운, 김대우, 윤명준, 김강률, 신정락, C 군집으로 진해수, 장원준, 임창민, 정우람, 오승환과 같이 클러스터링되었다. A 군집을 상위권, B군집을 중위권, C군집을 하위권이라 정의하였다.

[그림 8]은 나이와 WHIP를 가지고 군집화한 결과이다. A군집으로는 주권, 함덕주, 임찬규, 홍건희, 심창민, B군집으로는 진해수, 임창민, 정우람, 장원준, 신정락, 오승환 C군집으로는 김대우, 장민재, 김강률, 윤명준, 김재운과 같이 세 군집으로 나뉘었다. A군집을 상위권, B군집을 하위권, C군집을 중위권이라 정의하였다.

[그림 9]는 타율과 나이를 가지고 군집화한 결과이다. A 군집은 김대우, B군집은 주권, 임찬규, 홍건희, 심창민, 장민재, 윤명준, 신정락, 진해수, 장원준 C군집은 함덕주, 김재운, 김강률, 임창민, 정우람, 오승환과 같이 분석이 되었다. A군집을 하위권, B군집을 중위권, C군집을 하위권이라 정의하였다.



[그림 9] 나이/WHIP 기반 FA 투수 Clustering



[그림 10] 평균자책/WHIP 기반 FA 투수 Clustering

세 가지 군집화를 바탕으로 상위권에는 30점, 중위권에는 20점, 하위권에는 10점을 준 뒤 나이&평균자책점에는 가중치 0.3, 나이&WHIP에는 가중치 0.1, 평균자책점 &WHIP에는 가중치 0.6을 두고 계산해보았다.

이름	주권	함덕주	임찬규	홍건희	심창민	장민재	김재운	김대우	윤명준	김강률	신정락	김해수	장원준	임창민	정우람	오승환	
나이&평균자책점	상위권	상위권	상위권	상위권	상위권	중위권	중위권	중위권	중위권	중위권	중위권	하위권	하위권	하위권	하위권	하위권	
나이&WHIP	상위권	상위권	상위권	상위권	상위권	중위권	중위권	중위권	중위권	중위권	하위권	하위권	하위권	하위권	하위권	하위권	
평균자책점&WHIP	중위권	상위권	중위권	중위권	중위권	중위권	상위권	하위권	중위권	상위권	중위권	중위권	중위권	상위권	상위권	상위권	
점수(30점)		24	30	24	24	24	20	26	14	20	26	19	16	16	22	22	22

[그림 11] 클러스터링(상, 중, 하)별 가중치를 부여한 FA 투수 선수의 점수 측정

## V. 결론 및 제언

2024 FA 선수들의 기량을 분석하고 FA 시장을 예측해보기 위해 연구를 진행하였다. 본 연구를 통하여 다음과 같은 점을 얻을 수 있었다. 첫째로, 추출한 데이터를 바탕으로 머신러닝을 통해 군집화하는 방법을 학습할 수 있었다. 두 번째로, 클러스터링 방법 중 분할법인 K-means 클러스터링 알고리즘의 방식과 원리를 알게 되었다. 세 번째로, 각 집합별 중심점을 집합 내 오브젝트간 거리의 제곱합을 최소화 하는 집합을 구할 때 유클리디안 거리를 사용하여 구하는 것을 알게 되었다. 하지만 나이를 변수로 두고 군집화 시킨 것들은 나이를 기준으로 군집화가 된 것을 볼 수 있다. 이는 나이의 범위와 다른 변수의 범위 차이가 많이 나서 크기가 큰 나이를 기준으로 군집화가 진행되었다고 보았다. 사용하는 데이터들의 수치 차이가 너무 크지 않도록 조절한다면 중심점을 기준으로 더 확실하게 나누어질 것으로 예상되어 데이터 간 수치 조절이 필요함을 알 수 있었다. 네 번째로, 머신러닝이 임의로 군집화한 결과를 데이터의 상황에 맞게 새롭게 그룹화를 정의하는 과정을 거쳤고, 이를 통해 머신러닝을 기반으로 하는 선수들의 기량을 분석하고 수치화하여 우리만의 점수로 나타내는 과정을 진행하였다. 해당 결과는 현재로는 KBO 선수를 분석하는 데 적용하기는 어렵지만, WAR과 같은 선수들을 분석하는 식을 도출할 수 있을 것으로 기대된다. 마지막으로, 해당 사사연구를 진행하며 프로그램을 작성하는 법에 대해 배우고 친구들과 함께 연구하며 함께 협력하는 방법에 대해 배울 수 있었다.

## VI. 참고문헌

1. k-평균 알고리즘 - 위키백과, 우리 모두의 백과사전 (wikipedia.org).
2. EBS 수학과 함께하는 AI기초, 한국교육방송공사, 2020.
3. 기록실 | 기록/순위 | KBO (koreabaseball.com)
4. 데이터사이언스스쿨(<https://datascienceschool.net/intro.html>)

수학A

## 07\_두 평균의 역 문제



# 두 평균의 역문제

김승은 (서생중학교 2학년)

김이은 (상안중학교 2학년)

심태민 (신일중학교 3학년)

지도교수 : 최진호 (울산대학교 수학과)

## <초 록>

부등식 관계를 갖는 두 평균  $M_1, M_2 (M_1 \leq M_2)$ 에 대해, 모든 양수  $a$ 와  $b (a < b)$ 에 대해 각각 평균  $M_1(x, y)$ 와  $M_2(x, y)$ 가 되는 두 수  $x$ 와  $y$ 가 항상 존재하는 지에 대한 논의를 두 평균의 역문제라고 한다. 본 연구에서는 피타고라스 평균인 산술, 기하 및 조화평균들간의 역문제에 대해 조사해 보았고, 추가적으로 피타고라스 평균들과 유리  $k$ -산술평균과의 역문제들에 대해 조사해 보았다.

주제어 : 평균, 산술평균, 기하평균, 조화평균, 평균의 일반화, 평균의 역문제

## I. 서론 및 이론적 배경

수학 및 프로그래밍에서 중요하게 언급되는 함수는 입력과 출력에 관한 규칙으로 이해할 수 있다. 이는 최근 과학계에서 가장 중요한 주제 중 하나인 인공지능과도 직접적으로 연결되는데, 예를 들면 사진을 입력하면 인공지능은 고양이 사진인지 아닌지에 대한 결과를 출력해 준다.

평균은 여러 숫자를 입력받아 하나의 숫자를 출력해주는 함수라고 할 수 있고, 가장 기본적인 평균으로는 피타고라스 평균이라고 알려진 조화, 기하 및 산술평균이 있다. 피타고라스 평균은 고대 그리스 시대에서부터 음악 및 천문학에서 활용되어 왔으며<sup>[2]</sup> 통계학과 관련해서 주요한 대푯값으로 사용된다. 또한, 임의의  $n$ 개의 양수  $x_1, x_2, \dots, x_n$ 에 대해서 ‘조화-기하-산술 평균 부등식’이라고 알려진 다음 부등식이 잘 알려져있다.

$$H(x_1, x_2, \dots, x_n) \leq G(x_1, x_2, \dots, x_n) \leq A(x_1, x_2, \dots, x_n)$$

여기서,  $H$ ,  $G$  및  $A$ 는 각각 조화, 기하 및 산술평균을 의미한다. 즉,

$$H(x_1, x_2, \dots, x_n) = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}},$$

$$G(x_1, x_2, \dots, x_n) = \sqrt[n]{x_1 x_2 \dots x_n},$$

$$A(x_1, x_2, \dots, x_n) = \frac{x_1 + x_2 + \dots + x_n}{n}$$

이다.

지난 해 수학반 사사과정 논문<sup>[1]</sup>에서는 두 양수에 대한 여러 가지 평균들과 평균들 간의 부등식 관계에 대한 연구 결과를 발표했다. 특히, 산술평균의 일반화로써 유리  $k$ -산술평균에 대해 소개하고, 이 평균들과 피타고라스 평균과의 부등식 관계에 대해 조사했다. 실제로,  $k \geq \frac{1}{2}$ 일 때 두 양수  $x, y$ 의 유리  $k$ -산술평균  $A_k$ 을 다음과 같이 정의하고

$$A_k(x, y) = \frac{x^2 + 2kxy + y^2}{(k+1)(x+y)} \quad (1.1)$$

$k_1 < k_2$ 일 때 평균의 부등식 관계  $A_{k_1}(x, y) \geq A_{k_2}(x, y)$ 을 만족함을 보였다. 특별히,  $k=1$ 일 때는 산술평균이다. 즉,  $A_1(x, y) = A(x, y) = \frac{x+y}{2}$ 이다. 또한, 기하평균 및 조화평균과의 다음 부등식 관계를 보였다.

**(정리 A)**  $\frac{1}{2} \leq k \leq 1$ 이면  $H(x, y) \leq G(x, y) \leq A(x, y) \leq A_k(x, y)$ 이다.

**(정리 B)**  $1 \leq k \leq 3$ 이면  $H(x, y) \leq G(x, y) \leq A_k(x, y) \leq A(x, y)$ 이다.

**(정리 C)**  $k \geq 3$ 이면  $H(x, y) \leq A_k(x, y) \leq A(x, y)$ 이다.

부등식 관계를 갖는 두 양수에 대한 평균  $M_1$ 과  $M_2$  ( $M_1 \leq M_2$ )를 생각하자. 이 두 평균들은 서로 다른 두 개의 양수  $x, y$  ( $x < y$ )를 입력받으면 각각 하나의 숫자씩 두 개의 숫자  $M_1(x, y)$ 와  $M_2(x, y)$ 를 출력하며 다음 부등식 관계를 만족함을 의미한다.

$$x < M_1(x, y) < M_2(x, y) < y \quad (1.2)$$

여기서, 우리는 하나의 질문을 해본다. “임의의 두 양수  $a, b$  ( $a < b$ )가 주어질 때, 이 두 양수가  $M_1$ 과  $M_2$ 의 평균이 되도록 하는 양수  $x, y$ 를 찾을 수 있을까?” 이는 부등식 관계를 갖는 두 평균  $M_1, M_2$  ( $M_1 \leq M_2$ )와 임의의 양수  $a$ 와  $b$  ( $a < b$ )에 대해  $a = M_1(x, y)$ 와  $b = M_2(x, y)$ 가 되는 두 수  $x$ 와  $y$ 가 항상 존재하는 지에 대한 논의로써 **두 평균의 역문제**라고 한다. 또한, 두 평균의 역문제의 해가 단 한 쌍  $(x, y)$ 으로 존재할 때 “**역문제가 성립한다**” 라고 하며, 해  $x, y$ 를 **역문제의 해**라고 한다. 부등식 (1.2)로부터 주어진 양수  $a, b$ 와 역문제의 해  $x, y$ 에 대해 부등식

$$x < a < b < y \quad (1.3)$$

가 성립하는 데, 이 부등식을 역문제의 **필요조건**이라고 한다. 두 평균의 역문제가 성립하지 않을 때 “**역문제가 성립하지 않는다**”라고 한다.

본 연구에서는 피타고라스 평균인 산술, 기하 및 조화평균들간의 역문제에 대해 조사해 보았고, 추가적으로 피타고라스 평균들과 유리  $k$ -산술평균들과 역문제들에 대해 조사해 보았다.

## II. 피타고라스 평균의 역문제

본 절에서는 산술평균과 기하평균, 산술평균과 조화평균, 기하평균과 조화평균의 역문제를 조사해 보았다.

**(정리2.1)** 기하평균  $G(x, y)$ 와 산술평균  $A(x, y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = b - \sqrt{b^2 - a^2}, y = b + \sqrt{b^2 - a^2} \quad (2.1)$$

**증명.**  $a$ 와  $b$ 를  $a < b$ 를 만족하는 임의의 양수라고 하자. 이때 역문제의 해는  $G(x, y) = a$ 이고  $A(x, y) = b$ 를 만족하는 두 양수  $x, y$  ( $x < a < b < y$ ), 즉, 다음 연립방정식의 해를 찾는 문제이다.

$$\sqrt{xy} = a, \quad \frac{x+y}{2} = b \quad (2.2)$$

식 (2.2)의 두 번째 방정식에서 얻은 식  $y = 2b - x$ 을 첫 번째 방정식에 대입한 후 제곱을 하고 정리하면 다음의 이차 방정식을 얻는다.

$$x^2 - 2bx + a^2 = 0$$

근의 공식을 통해 필요조건  $x < b$ 을 만족하는 해

$$x = b - \sqrt{b^2 - a^2}$$

를 얻고 대응되는 연립해

$$y = 2b - x = b + \sqrt{b^2 - a^2}$$

를 얻는다. 여기서  $b < y$ 는 자명하고 다음 식은 해 (2.1)이 역문제의 필요조건  $x < a$ 가 만족함을 보여준다.

$$a - x = \sqrt{b^2 - a^2} - (b - a) = \sqrt{b - a}(\sqrt{b + a} - \sqrt{b - a}) > 0$$

□

**(정리 2.2)** 조화평균  $H(x, y)$ 와 산술평균  $A(x, y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = b - \sqrt{b^2 - ab}, \quad y = b + \sqrt{b^2 - ab} \quad (2.3)$$

**증명.**  $a$ 와  $b$ 를  $a < b$ 를 만족하는 임의의 양수라 하자. 조화평균과 산술평균에 대한 역문제의 해는 다음 연립방정식과 같다.

$$\frac{2xy}{x+y} = a, \quad \frac{x+y}{2} = b \quad (2.4)$$

(2.4)의 두 번째 방정식에서 얻은 식  $y = 2b - x$ 을 첫 번째 방정식에 대입한 후 정리하면 다음 이차 방정식

$$x^2 - 2bx + ab = 0$$

을 얻고, 필요조건  $x < b$ 를 만족하는 연립해 (2.3)을 얻는다. 다음 식은 해 (2.3)이 역문제의 필요조건  $x < a$ 가 만족함을 보여준다.

$$a - x = \sqrt{b^2 - ab} - (b - a) = \sqrt{b - a}(\sqrt{b} - \sqrt{b - a}) > 0$$

□

**(정리 2.3)** 조화평균  $H(x, y)$ 와 기하평균  $G(x, y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = \frac{b^2 - b\sqrt{b^2 - a^2}}{a}, \quad y = \frac{b^2 + b\sqrt{b^2 - a^2}}{a} \quad (2.5)$$

**증명.**  $a$ 와  $b$ 를  $a < b$ 를 만족하는 임의의 양수라 하자. 조화평균과 기하평균에 대한 역문제의 해는 다음 연립방정식과 같다.

$$\frac{2xy}{x+y} = a, \quad \sqrt{xy} = b \quad (2.6)$$

식 (2.6)의 두 번째 방정식에서 양변을 제곱하고 얻은 식  $y = \frac{b^2}{x}$  ( $x > 0$ )을 첫 번째 방정식에 대입한 후 정리하면 다음 이차 방정식을 얻는다.

$$ax^2 - 2b^2x + ab^2 = 0$$

근의 공식을 통해  $x < b$ 를 만족하는 해

$$x = \frac{b^2 - b\sqrt{b^2 - a^2}}{a}$$

를 얻고 대응되는 연립해

$$y = \frac{b^2}{x} = \frac{b^2 + b\sqrt{b^2 - a^2}}{a}$$

를 얻는다.  $b < y$ 는 자명하므로 다음 식

$$a - x = \frac{1}{a} [b\sqrt{b^2 - a^2} - (b^2 - a^2)] = \frac{\sqrt{b^2 - a^2}}{a} (b - \sqrt{b^2 - a^2}) > 0$$

을 통해 해 (2.5)가 역문제의 필요조건을 만족함을 알 수 있다.  $\square$

**[예제 2.4]** 두 양수 1과 2를 생각하자. 이 입력에 대해 피타고라스 평균은 각각  $H(1,2) = \frac{4}{3}$ ,  $G(1,2) = \sqrt{2}$  그리고  $A(1,2) = \frac{3}{2}$ 인 1보다 크고 2보다 작은 수를 출력한다. 이에 반해, 정리 2.1, 정리 2.2 그리고 정리 2.3의 평균의 역문제들은 각각 서로 다른 세가지의 순서쌍의 수  $(2 - \sqrt{3}, 2 + \sqrt{3})$ ,  $(2 - \sqrt{2}, 2 + \sqrt{2})$  그리고  $(4 - 2\sqrt{3}, 4 + 2\sqrt{3})$ 으로 앞 자리는 1보다 작은 수, 뒷 자리는 2보다 큰 수를 출력해준다.

### III. 유리 $k$ -산술평균과 관련된 역평균 문제

이 절에서는 유리  $k$ -산술평균과 관련된 역평균 문제를 살펴본다. 먼저, 정리 A의 산술평균과 유리  $k$ -산술 평균에 대한 역평균 문제를 생각해본다.

**(정리 3.1)**  $\frac{1}{2} \leq k < 1$ 일 때, 산술평균  $A(x,y)$ 와 유리  $k$ -산술평균  $A_k(x,y)$ 는 역문제가 성립하지 않는다.

**증명.**  $\frac{1}{2} \leq k < 1$ 이면  $A \leq A_k$ 이다.  $a$ 와  $b$ 를  $a < b$ 를 만족하는 임의의 양수라 하자. 산술평균과 유리  $k$ -산술평균에 대한 역문제의 해는 다음 연립방정식으로 나타난다.

$$a = \frac{x+y}{2}, \quad b = \frac{x^2 + 2kxy + y^2}{(k+1)(x+y)} \quad (3.1)$$

식 (3.1)의 첫 번째 방정식에서 얻은 식  $y = 2a - x$ 을 두 번째 방정식에 대입해서  $x$ 에 대해 정리하면 다음 이차 방정식을 얻는다.

$$(1-k)x^2 - 2a(1-k)x + a(2a - (1+k)b) = 0 \quad (3.2)$$

이차방정식 (3.2)의 판별식은  $D/4 = a(b-a)(1-k^2) > 0$ 이고  $x < a$ 을 만족하는 이차방정식의 해

$$x = a - \sqrt{\frac{a(b-a)(1+k)}{1-k}} \quad (3.3)$$

를 얻고 이에 대응하는 연립해

$$y = a + \sqrt{\frac{a(b-a)(1+k)}{1-k}} \quad (3.4)$$

를 얻는다. 이때,  $y > b$ 를 만족할 필요충분조건은

$$y - b = a - b + \sqrt{\frac{a(b-a)(1+k)}{1-k}} > 0 \quad (3.5)$$

이고  $b - a > 0$ 이므로 간단한 계산에 의해 부등식 (3.5)는 다음 부등식과 필요충분조건이다.

$$\frac{2}{1-k} > \frac{b}{a} \quad (3.6)$$

을 얻는다. 주어진 양수  $a$ 에 대해  $b = \frac{2a}{1-k}$ 로 두면 부등식 (3.6)을 만족하지 않는다. 따라서, 역평균의 필요조건  $y > b$ 는 만족하지 않는다.  $\square$

**(정리 3.2)**  $k > 1$ 일 때, 유리  $k$ -산술평균  $A_k(x, y)$ 과 산술평균  $A(x, y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = b - \sqrt{\frac{b(b-a)(k+1)}{k-1}}, \quad y = b + \sqrt{\frac{b(b-a)(k+1)}{k-1}} \quad (3.7)$$

**증명.** 정리 3.1의 증명과 같은 방법에 의해  $x < y$ 인 연립해 (3.7)을 얻는다. 식 (3.7)의 두 번째 식으로부터 역평균의 필요조건  $b < y$ 는 자명하고, 첫 번째 식에서 역평균의 필요조건인  $x < a$ 를 만족할 필요충분조건을 계산하면

$$\frac{-2}{k-1} < \frac{a}{b} \quad (3.8)$$

을 얻는다.  $k > 1$ 이므로 부등식 (3.8)는 모든 양수  $a, b$ 에 대해 성립하고, 따라서 연립해 (3.7)의 역평균의 해이다.  $\square$

정리 A, 정리 B와 정리 C에서  $k \geq \frac{1}{2}$ 이면  $H \leq A_k$ 임을 보았다. 다음은 조화평균과 유리  $k$ -산술평균에 대한 역평균 문제를 조사해본다.

**(정리 3.3)**  $k \geq \frac{1}{2}$ 일 때, 조화평균  $H(x, y)$ 와 유리  $k$ -산술평균  $A_k(x, y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = \frac{1}{2} \{ (b-a)k + a + b - \sqrt{(k+1)(b-a) \{ (b-a)k + b + a \}} \} \quad (3.9a)$$

$$y = \frac{1}{2} \{ (b-a)k + a + b + \sqrt{(k+1)(b-a) \{ (b-a)k + b + a \}} \} \quad (3.9b)$$

**증명.**  $k \geq \frac{1}{2}$ 이면  $H \leq A_k$ 이다.  $a$ 와  $b$ 를  $a < b$ 를 만족하는 임의의 양수라 하자. 조화평균과 유리  $k$ -산술평균에 대한 역문제의 해는 다음 연립방정식으로 나타난다.

$$\frac{2xy}{x+y} = a, \frac{x^2 + 2kxy + y^2}{(k+1)(x+y)} = b \quad (3.10)$$

연립방정식 (3.10)을 풀기위해

$$x + y = p, xy = q \quad (3.11)$$

라 두면

$$2q = ap, p^2 + 2(k-1)q = (k+1)bp \quad (3.12)$$

을 얻고, 방정식 (3.12)의 연립해는  $p = q = 0$  또는

$$p = (b-a)k + a + b, q = \frac{a}{2} [(b-a)k + a + b] \quad (3.13)$$

이다. 방정식 (3.11)을  $x, y (x < y)$ 에 대해 풀면

$$x = \frac{1}{2} \{p - \sqrt{p^2 - 4q}\}, y = \frac{1}{2} \{p + \sqrt{p^2 - 4q}\} \quad (3.14)$$

이고, 식 (3.13)의  $p$ 와  $q$ 를 식 (3.14)에 대입하면 연립해 (3.9a)와 (3.9b)를 얻는다. 주어진 두 해가 역평균의 필요조건을 만족함을 보이면 증명이 완성된다.

$$a - x = \frac{1}{2} [\sqrt{(k+1)(b-a)\{(b-a)k + b + a\}} - (b-a)(k+1)]$$

이고  $(b-a)(k+1) > 0$ 이므로,  $x < a$ 일 필요충분조건은

$$(k+1)(b-a)\{(b-a)k + b + a\} - (b-a)^2(k+1)^2 > 0 \quad (3.15)$$

이고 좌변을 정리하면  $2a(b-a)(k+1)$ 로 항상 양수이므로, 부등식 (3.15)는 항상 참이다. 따라서 역평균의 필요조건  $x < a$ 를 만족한다. 유사하게

$$y - b = \frac{1}{2} [(b-a)(k-1) + \sqrt{(k+1)(b-a)\{(b-a)k + b + a\}}] \quad (3.16)$$

이므로,  $k \geq 1$ 일 때  $b < y$ 이다.  $k < 1$ 일 때, 식 (3.16)로부터

$$|(b-a)(k-1)|^2 - (k+1)(b-a)\{(b-a)k + b + a\} < 0 \quad (3.17)$$

이면  $b < y$ 이다. 부등식 (3.17)의 좌변을 정리하면  $-2(b-a)\{(2b-a)k + a\}$ 로 항상 음수이므로, 부등식 (3.17)은 항상 참이다. 따라서, 역평균의 필요조건  $y > b$ 를 만족한다.  $\square$

**참고.** 기하평균과 유리  $k$ -산술평균의 문제는 4차 방정식에 관한 문제로 중학교 과정의 범위를 벗어나는 관계로 도전할 수 없었다.

#### IV. 연구결과 및 고찰

역평균 문제라는 것은, 부등식 관계를 갖는 두 평균  $M_1, M_2 (M_1 \leq M_2)$ 에 대해, 모든 양수  $a$ 와  $b (a < b)$ 에 대해 각각 평균  $M_1(x, y)$ 와  $M_2(x, y)$ 가 되는 두 수  $x$ 와  $y$ 가 항상 존재하는 지에 대한 논의이다. 본 연구에서는 피타고라스 평균들의 역평균 문제 및 유리  $k$ -산술평균과 피타고라스 평균들의 역평균 문제에 대해

조사를 해보았다. 그 결과로써, 다음의 결과를 얻을 수 있었다.

- 기하평균  $G(x,y)$ 와 산술평균  $A(x,y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = b - \sqrt{b^2 - a^2}, y = b + \sqrt{b^2 - a^2}$$

- 조화평균  $H(x,y)$ 와 산술평균  $A(x,y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = b - \sqrt{b^2 - ab}, y = b + \sqrt{b^2 - ab}$$

- 조화평균  $H(x,y)$ 와 기하평균  $G(x,y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = \frac{b^2 - b\sqrt{b^2 - a^2}}{a}, y = \frac{b^2 + b\sqrt{b^2 - a^2}}{a}$$

- $\frac{1}{2} \leq k < 1$ 일 때, 산술평균  $A(x,y)$ 와 유리  $k$ -산술평균  $A_k(x,y)$ 는 역문제가 성립하지 않는다.

- $k > 1$ 일 때, 유리  $k$ -산술평균  $A_k(x,y)$ 과 산술평균  $A(x,y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = b - \sqrt{\frac{b(b-a)(k+1)}{k-1}}, y = b + \sqrt{\frac{b(b-a)(k+1)}{k-1}}$$

- $k \geq \frac{1}{2}$ 일 때, 조화평균  $H(x,y)$ 와 유리  $k$ -산술평균  $A_k(x,y)$ 는 역문제를 만족하며, 그 해는 다음과 같다.

$$x = \frac{1}{2} \{ (b-a)k + a + b - \sqrt{(k+1)(b-a) \{ (b-a)k + b + a \}} \}$$

$$y = \frac{1}{2} \{ (b-a)k + a + b + \sqrt{(k+1)(b-a) \{ (b-a)k + b + a \}} \}$$

이번 연구를 통해 평균의 개념과 역평균 문제의 개념을 폭넓게 이해할 수 있었다. 앞으로 더 다양한 평균들을 공부해 본다면, 역평균 문제에 대한 추가적 연구결과를 낼 수 있을 것으로 기대된다.

## V. 참고논문

- [1] 김태은, 오준엽, 이다빈, 전서훈, 최진호, “산술평균의 일반화”, 탐구와 창의 (Journal of Research and Creativity), 2022. Vol.17, No.1, pp. 1-14.
- [2] P. Pesic, Music and the Making of Modern Science, London, MIT Press, pp. 9-20, 2014.

# Inverse problem of two means

Seung Eun Kim, Yi Eun Kim, Tae Geon Kim, Tae Min Sim  
Jin Ho Choi

(#)

<Abstract>

In this paper, we study an inverse problem for two means of two positive numbers related to Pythagorean means and the rational  $k$ -arithmetic mean.

Key Words : Arithmetic mean, Geometric mean, Harmonic mean, rational  $k$ -arithmetic mean, Inverse problem for means



수학A

## 08\_피보나치 수의 합과 제곱 합의 점화식



# 피보나치 수의 합과 제곱 합의 점화식

김지원 (안남 중학교 2학년)  
변수민 (진해남중학교 2학년)  
조민우 (반 송 중학교 2학년)  
최민혁 (반 송 중학교 2학년)  
하성철 (반 송 중학교 2학년)

지도교수 : 김영호 (창원대학교 수학과)

**초록:** 피보나치수열은 기원전 인도 수학자들에 의하여 연구가 시작되었으며 1200년경 유럽에 알려졌으며 그 후 연구자들에 의해 피보나치 수의 항등식 존재 여부가 계속 연구되었다. 본연구에서는 피보나치수열의 합을 쉽게 계산할 수 있는 일반항의 존재성에 관심을 두고 잘 알려지지 않은 점화식을 찾는 것을 관심의 대상으로 하였다.

고대 수학자들의 업적을 바탕으로 개발된 피보나치 수의  $n$ 항까지 합의 일반항을 중심으로 홀수 번째 항의 합, 짝수 번째 항의 합에 대한 항등식을 찾아보고 또 다른 항등식의 패턴을 탐구하였다. 특히 연속된 피보나치 수 제곱의 합에 대한 일반항을 바탕으로 멀어지는 두 피보나치 수의 곱에 대한 패턴과 피보나치 수 제곱의 합에 대한 패턴을 탐구하여 새롭게 표현되는 피보나치 수 점화식을 찾았다. 또 항등식 패턴을 찾아가는 연구와 잘 알려지지 않는 항등식을 구하는 과정을 통하여 고대 수학자들의 피보나치수열 연구에 대한 열정과 수학의 계통성을 이해하고자 하였다.

## I. 서론 (또는 연구의 필요성 및 목적)

피보나치<sup>1)</sup>의 저서 『산반서(Liber Abaci)』에는 특별한 수열을 다루고 있는데 이 수열을 그의 이름을 따서 피보나치수열이라 한다. 그러나 이 수열은 피보나치가 책을 쓰기 훨씬 이전에 인도 지역에서 이미 알려져 있었다고 한다. 기원전 450년경 인도의 수학자 핑갈라(Pingala)<sup>2)</sup>의 책에 수록되어 있다고 하는데, 명시화된 기록으로는 핑갈라의 책을 기반으로 한 ‘비라한카(Virahanka, 700년 경)’의 저서에 나타난다고 한다. 불행히도 이 책은 소실되었다고 알려지지만 고팔라(Gopala, 1135년 경)<sup>3)</sup>가 자신의 저서에서 다시금 ‘비라한카’를 인용함으로써 기록으로 남아 있다.

인도에서 시작된 연구가 1202년경에 피보나치가 쓴 저서 저서에 인용되어 서구 유

1) 이탈리아의 상인이자 수학자인 피보나치(Fibonacci; 1170~1250)는 중세 유럽의 대수학자이다. 1170년 이탈리아 피사(Pisa)의 상업 중심지에서 태어나 ‘피사의 레오나르도(Leonardo)’라고 불리었다.

2) 아차라 핑갈라(Acharya Pingala, 3~2세기 BC)는 산스크리트 운율에 관한 초기 논문인 상다르슈스트라(Pingala-sutras라고도 함)를 저술한 고대 인도의 수학자였다.

3) 고팔라(Resigned c. 750s - 770s CE)는 인도 대륙 벵골 지역의 팔라 왕조의 시조였다

렵에 잘 알려졌다고 하며, 그 책에서 피보나치는 갓 태어난 토끼 한 쌍의 번식 문제로 이 수열을 설명했다. (네이버 지식백과 참고, [7]).

“들판에 갓 태어난 암수 한 쌍의 토끼가 있다. 토끼들은 한 달이 지나면 성숙하여 두 달의 끝부분에서 암컷이 암수 한 쌍의 토끼를 낳는다. 토끼들은 절대로 죽지 않고, 암컷은 암수 한 쌍의 새끼를 둘째 달부터 계속해서 낳는다고 하면, 1년 후에는 얼마나 많은 토끼가 들판에 있을까”를 질문했다.

토끼의 수 증가와 관련된 피보나치 수는 수식으로 나타내면 다음과 같다.

$$F(n) = F(n-1) + F(n-2),$$

여기서  $F(n)$ 은 피보나치수열의  $n$ 번째 항을 나타낸다. 바로 앞의 두 수의 합으로 만들어지는 것이 피보나치수열이며 처음 몇 개의 항은 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

이 수열은 그 독특한 특성으로 유명하다. 여러 분야에서 활용되고 있으며, 예컨대 자연계에서 나타나는 현상이나 황금비 등과 연관이 있다. 또한, 피보나치수열은 황금비(대략 1.618)에 수렴하는 성질을 지니고 있다. 피보나치 수는 단순한데도 수학적인 규칙이 잘 드러나 있으며, 수학적인 사고력과 호기심 유발을 자극하는 주제로 생각된다.

본 사사 연구의 출발은 간단한 규칙에 따라 숫자가 나열되는 피보나치수열을 통해 수학적 패턴과 규칙을 더 잘 이해할 수 있고 수학뿐만 아니라 과학, 공학 등 여러 분야에서의 논리적 사고를 키우는 데 도움이 될 것이라는 판단에서 시작하였다.

여기서 탐구하고자 하는 주요한 연구 문제는 다음과 같다.

- ✓ 피보나치 수  $n$ 항까지의 합에 의한 기본 항등식 패턴 탐구,
- ✓ 멀어지는 두 피보나치 수 곱 구성법 이해 및 패턴 탐구,
- ✓ 피보나치 수 제곱의 합에 대한 새로운 항등식 점화식 찾아보기
- ✓ 피보나치 수 항등식으로부터 새로운 항등식과 점화식 찾아보기

본 연구 목표는 피보나치 수의 합과 곱에 대한 항등식 이해를 바탕으로 최근 알려진 항등식으로부터 새로운 항등식 패턴을 찾아보고 피보나치 수에는 어떤 것들이 숨어있는지 알아보는 것이다. 아울러 고대 수학자들의 피보나치수열 연구에 대한 열정을 느끼는 것이다. 나아가 알려진 피보나치수열 항등식과 더불어 잘 알려지지 않을 항등식을 탐구하고 새로운 패턴을 찾아보는 과정을 탐구하여 수학의 계통성을 이해하고자 한다.

추가적인 목표는 숨어있는 피보나치수열 항등식을 발견하고 이를 해결하는 과정을 익히는 것은 수학 연구의 본질적인 방향이라 생각하면서 피보나치수열의 항등식 탐구를 통하여 수학을 이해하고 수학적 사고력을 길러 보는 것이다. 나아가 피보나치수열의 항등식 탐구는 학교 수학에서 깊이 다루지 않는 문제이기 때문에 본 사사과정 탐구학습을 통하여 수학적 문제 해결 능력의 깊이를 더해줄 것이라 기대한다.

## II. 이론적 배경

피보나치수열은 수학에서 예기치 않게 자주 나타나기 때문에 피보나치수열만 전문으로 하는 ‘피보나치 쿼터리’(Fibonacci Quarterly)라는 전용 저널이 있을 정도이다. 피보나치 수의 응용 프로그램에는 피보나치 검색 기술 및 피보나치 데이터 구조와 같은 컴퓨터 알고리즘과 병렬 및 분산 시스템을 상호 연결하는 데 사용되는 피보나치 큐브라는 그래프가 포함되는 것으로 알려져 있다([6]). 피보나치 수는 자연에서도 종종 나타나는데 예를 들면, 나무의 가지, 줄기의 잎 배열, 파인애플의 과일 새싹, 아티초크의 개화, 솔방울의 포엽 배열과 같은 생물학적 환경에서 나타난다. 하지만 모든 중에서 발생하는 것은 아닌 것으로 알려져 있다([3]).

또 피보나치 수는 또한 황금 비율과 밀접한 관련이 있다([1, 4]). ‘Binet의 공식’은  $n$  번째 피보나치 수와  $n+1$  번째 수가 황금 비율로 표현됨을 설명하고 있다. 즉 두 개의 연속 피보나치 수의 비율이  $n$ 이 증가함에 따라 황금 비율이 되는 경향이 있음을 의미한다. 피보나치수열을 일반화할 수 있는데 피보나치수열의 정의와 비슷한 정의를 가지면서 일반화된 수열이 ‘루카스 수열’(Lucas sequences)이다.([5]).

위키백과([5, 6])에서 피보나치수열은  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$ 으로 정의되어 있다. 여기서는 위키백과의 피보나치수열 정의를 이용하여 내용을 전개한다.

**피보나치수열의 정의:** 피보나치 수  $F_n$ 은 다음과 같은 초기값 및 점화식으로 정의되는 수열이다.

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$$

첫 번째 피보나치 수부터  $n$  번째 항까지의 첫 번째 피보나치 수의 합이  $(n+2)$  번째 피보나치 수에서 1을 뺀 것과 같다는 다음 항등식이 나타난다. (참고, [1, 5, 6]).

$$\sum_{k=1}^n F_k = F_{n+2} - 1, n \geq 1 \tag{2.1}$$

또 항등식(2.1)과 비슷한 항등식으로, 피보나치 수에서  $F_0$  또는  $F_0, F_1$ 을 제외하고 위치에 따라 합계를 그룹화하면 두 개의 항등식이 성립함이 나타난다.

$$\sum_{k=0}^{n-1} F_{2k+1} = F_{2n} , \quad (2.2)$$

$$\sum_{k=1}^n F_{2k} = F_{2n+1} - 1 . \quad (2.3)$$

위 첫 번째 항등식(2.2)은 홀수 번째 항들의 합은  $2n$ 번째 피보나치 수이고, 두 번째 항등식(2.3)은 첫 항이  $F_2$ 인 짝수 번째 인덱스들의 합이  $2n+1$ 번째 피보나치 수에서 1을 뺀 값이다. 아울러 첫 번째 피보나치 수의 제곱의 합은  $n$ 번째 피보나치 수와  $n+1$ 번째 피보나치 수의 곱이 되는 아래 항등식이 소개되었다.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1} . \quad (2.4)$$

항등식(2.1)~(2.4) 살펴본 바와 같이 피보나치수열에는 다양한 항등식과 점화식이 성립되는 이론적 배경을 가지고 있음을 알 수 있다.

### III. 연구 방법

#### 3. 1. 피보나치 수의 합

피보나치 수의  $n$ 항까지 합에 대한 아래 항등식을 소개하였다.(참고, [1, 5, 6]).

$$\sum_{k=1}^n F_k = F_{n+2} - 1, \quad n \geq 1$$

이 항등식은 피보나치수열의 성질을 이용하여 증명됨을 알 수 있다. 그러나 이 항등식을 다양한 방법으로 증명하거나 수정 보완할 수 있는 여지가 있어 보였다. 여기에서는 수열의 성질 및 한 가지 대표적인 방법인 수학적귀납법을 사용하여 증명한다.

또  $n$ 항까지 합 항등식 외 다양한 항등식은 소개되어 있지 않아 좀 더 깊이 있게 조사하여 새로운 항등식의 존재가 가능한지는 조사하고 증명하였다.  $F_1$ 에서 출발하는 홀수 번째 항들의 합,  $F_2$ 에서 출발하는 짝수 번째 항들의 합 등 아래 피보나치 수의 합의 항등식

$$\sum_{k=1}^n F_{2k-1} = A_1, \quad \sum_{k=1}^n F_{2k} = A_2, \quad \sum_{k=1}^n F_{3k} = A_3, \quad \dots \dots$$

을 조사하고 증명하였다.

#### 3. 2. 멀어지는 두 피보나치 수 곱의 점화식

피보나치 수의  $n$ 항까지 제곱의 합에 대한 아래 항등식이 나타난다. (참고, [5, 6, 7]).

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

이 항등식은 피보나치수열의 성질을 이용하여 증명될 것으로 판단되나 항등식의 증명 과정이 자세히 소개되어 있지 않아  $n$ 항까지 제곱의 합에 대한 항등식을 다양한 방법으로 증명하여 볼 여지가 있어 보였다. 여기에서는 수열의 성질 및 한 가지 대표적인 방법인 수학적귀납법을 사용하여 증명한다. 또 이 항등식의 우변의 두 피보나치 수의 곱에 주목하여 다른 표현의 점화식을 찾아보고자 하였다. 우변의 두 피보나치 수의 곱의 패턴을 찾기 위해  $F_n F_{n+1}$  대신 점차 멀어지는 두 피보나치 수의 곱에 대한 항등식

$$F_n F_{n+2} = A_1, F_n F_{n+3} = A_2, F_n F_{n+4} = A_3, \dots \dots, F_n F_{n+i} = A_{i-1}$$

등을 조사하고 항등식이 성립하는지 수학적귀납법 등으로 증명하였다.

### 3. 3. 피보나치 수 제곱 합의 점화식

3.2절 ‘멀어지는 두 피보나치 수 곱의 점화식’은  $n$ 항까지 제곱의 합에 대한 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

에서 출발하여 우변의 두 피보나치 수의 곱에 주목하였다. 여기서는 좌변의 피보나치 수 제곱의 합에 주목하여 새로운 점화식 발견을 목표로 심화 연구를 하였다. 우변의 두 피보나치 수의 다른 유형을 찾기 위해  $F_n F_{n+1}$  대신 두 피보나치 수의 곱에 대한 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+2} - B_2, \sum_{k=1}^n F_k^2 = F_n F_{n+3} - B_3, \sum_{k=1}^n F_k^2 = F_n F_{n+4} - B_4, \dots \dots,$$

등을 조사하였다. 또 항등식이 일반적인 점화식이 성립하는지 조사하였으며 이러한 항등식이 성립함을 수학적귀납법으로 증명하였다.

### 3. 4. 점화식으로부터의 항등식

3.1절 ‘피보나치 수의 합’에서는 피보나치 수의  $n$ 항까지 합에 대한 항등식

$$\sum_{k=1}^n F_k = F_{n+2} - 1, n \geq 1$$

을 중심으로 피보나치수열의 성질을 이용하여 점화식을 유도하였다. 3.2절 ‘멀어지는 두 피보나치 수 곱의 점화식’은  $n$ 항까지 제곱의 합에 대한 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

에서 출발하여 우변의 두 피보나치 수의 곱에 주목하여 피보나치 수 제곱의 합의 점화식을 유도하였다. 여기에서는 3.1~3.3절에서 주목한 항등식

$$\sum_{k=1}^n F_{2k-1} = F_{2n}, \quad \sum_{k=1}^n F_{2k} = F_{2n+1} - 1, \quad \sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

의 우변의 각 점화식에 주목하여 새로운 점화식 발견을 목표로 심화 연구를 하였다. 우변의 두 피보나치 수의 다른 유형을 찾기 위해 아래 유형의 피보나치 수 항등식

$$\sum_{k=1}^n F_k = D_1, \quad \sum_{k=1}^n F_{2k-1} = D_2, \quad \sum_{k=1}^n F_k^2 = D_3, \quad F_n F_{n+i} = F_{i-1} F_n^2 + F_i \sum_{k=1}^n F_k^2, \dots \dots,$$

등의 일반적인 점화식이 성립하는지 조사하였으며 이러한 항등식이 성립함을 수열의 성질 및 수학적귀납법으로 증명하였다.

#### IV. 연구 결과

##### 4. 1. 피보나치 수의 합

이 절에서는 피보나치 수의  $n$ 항까지 합에 대한 아래 항등식

$$\sum_{k=1}^n F_k = F_{n+2} - 1, \quad n \geq 1 \tag{4.1}$$

의 수학적 특성을 이해하고 이 항등식이 성립됨을 몇 가지 방법으로 증명하여 위키백과([5, 6])에서 찾을 수 없었던 증명을 확인한다. 여기에서는 수열의 성질 및 수학적귀납법을 사용하여 증명한다. 또 새로운 항등식의 존재 가능성을 탐구하고 좀 더 깊이 있게 조사 방법을 위해 기존 사사 논문([2])을 참고하여 연구한다.

**[연구 1-1]** 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 다음 등식이 성립됨을 확인하여 보자.

$$\sum_{k=1}^{n-1} F_k = F_{n+1} - 1, \quad n \geq 2$$

**[풀이 1]** 피보나치 수 각 항이

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$$

을 만족하므로 점화식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_{n+1} &= F_{n-1} + F_n = F_{n-1} + F_{n-2} + F_{n-1} \\ &= F_{n-1} + F_{n-2} + F_{n-3} + F_{n-2} \\ &\dots \\ &= F_{n-1} + F_{n-2} + F_{n-3} + \dots + F_2 + F_1 + F_2 \end{aligned}$$

따라서 위 식으로부터

$$F_1 + F_2 + \dots + F_{n-1} = F_{n+1} - 1$$

임을 알 수 있다.

**[풀이 2]** 수학적귀납법으로 증명하여 보자.

I)  $n=2$ 인 경우

$$(\text{우변}) = \sum_{k=1}^1 F_k = F_1 = 1, (\text{좌변}) = F_3 - 1 = 2 - 1 = 1$$

이므로 (좌변)=(우변)이 성립한다.

ii)  $n$ 인 경우(귀납법 가정)

다음 항등식이 이 성립한다고 하자.

$$F_{n+1} - 1 = \sum_{k=1}^{n-1} F_k, \quad n \geq 2$$

iii)  $n+1$ 인 경우

$$\sum_{k=1}^n F_k = \sum_{k=1}^{n-1} F_k + F_n = F_{n+1} - 1 + F_n = F_{n+1+1} - 1$$

가 성립된다. 따라서 I), ii), iii)에 의하여 항등식 (1) 가 성립됨을 알 수 있다.

**[참고 1]** ‘연구 1-1’에서 피보나치 수에 대한 항등식  $\sum_{k=1}^{n-1} F_k = F_{n+1} - 1$  이 성립됨을

증명하였다. 여기서 우변의 피보나치 수  $F_{n+1}$  대신  $F_{2n+1}, F_{3n+1}$  등을 대입하여 새로운 패턴을 찾아보면 아래와 같은 단순한 패턴

$$\sum_{k=1}^{m-2} F_k = F_m - 1, \quad n \geq 2$$

이 존재함을 알 수 있었다.

다음은 피보나치 수의  $n$ 항까지 합에 대한 아래 항등식에서 좌변의 피보나치 수  $F_k$  대신  $F_{2k-1}$ (홀수)을 대입하여 보면 다음 항등식을 얻을 수 있음을 보였다.

**[연구 1-2]** 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 다음 등식이 성립함을 확인하여 보자.

$$\sum_{k=1}^n F_{2k-1} = F_{2n}, \quad n \geq 1$$

**[풀이]** 피보나치 수 각 항이  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  을 만족하므로 점화식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_{2n+2} &= F_{2n+1} + F_{2n} \\ &= F_{2n+1} + F_{2n-1} + F_{2n-2} \\ &= F_{2n+1} + F_{2n-1} + F_{2n-3} + F_{2n-4} \\ &\dots \\ &= F_{2n+1} + F_{2n-1} + F_{2n-3} + \dots + F_3 + F_1 + F_0 \end{aligned}$$

따라서 위 식으로부터

$$F_1 + F_3 + \dots + F_{2n-1} = F_{2n+2} - F_{2n+1}$$

임을 알 수 있다. 즉 다음과 같이 표현할 수 있다.

$$\sum_{k=1}^n F_{2k-1} = F_{2n}, \quad n \geq 1.$$

다음은 피보나치 수의  $n$ 항까지 합에 대한 아래 항등식에서 좌변의 피보나치 수  $F_k$  대신  $F_{2k}$ (짝수)을 대입하여 보면 다음 항등식을 얻을 수 있음을 보였다.

[연구 1-3] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 다음 등식을 만족하는  $A_2$ 을 구하여 보자.

$$\sum_{k=1}^n F_{2k} = A_2$$

[풀이] 피보나치 수 각 항이

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, \quad n \geq 2$$

을 만족하므로 점화식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} \sum_{k=1}^n F_{2k} &= F_2 + F_4 + \dots + F_{2n} \\ &= (F_0 + F_1) + (F_2 + F_3) + \dots + (F_{2n-2} + F_{2n-1}) \\ &= \sum_{k=1}^{2n-1} F_k \end{aligned}$$

이다. 한편  $\sum_{k=0}^{2n-1} F_k = F_{2n+1} - 1$  이므로

$$\sum_{k=1}^n F_{2k} = F_{2n+1} - 1$$

이 성립한다. 따라서  $A_2 = F_{2n+1} - 1$ 이다.

다음은 피보나치 수의  $n$ 항까지 합에 대한 아래 항등식에서 좌변의 피보나치 수  $F_k$  대신  $F_{3k}$ (3의 배수),  $F_{4k}$ (4의 배수) 및  $F_{5k}$ (5의 배수)을 대입하여 보면 다음 항등식을 얻을 수 있음을 보였다.

[연구 1-4] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 다음 등식을 만족하는  $A_3$ 를 구하여 보자.

$$\sum_{k=1}^n F_{3k} = A_3$$

**[풀이]**

점화식은 다음과 같이 표현할 수 있다.

$$\sum_{k=1}^n F_{3k} = F_3 + F_6 + \dots + F_{3n}, \quad \dots\dots \textcircled{1}$$

$$\sum_{k=1}^n F_{3k} = (F_1 + F_2) + (F_4 + F_5) + \dots + (F_{3n-2} + F_{3n-1}) \quad \dots\dots \textcircled{2}$$

①식과 ②의 양변을 더하면

$$2 \sum_{k=1}^n F_{3k} = (F_1 + F_2) + F_3 + (F_4 + F_5) + F_6 + \dots + (F_{3n-2} + F_{3n-1}) + F_{3n}$$

이다. 한편  $\sum_{k=0}^{3n} F_k = F_{3n+2} - 1$  이므로

$$\sum_{k=1}^n F_{3k} = \frac{1}{2}(F_{3n+2} - 1)$$

이다. 즉  $A_3 = \frac{1}{2}(F_{3n+2} - 1)$ 이다.

[연구 1-5] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 다음 등식을 만족하는  $A_4$ 를 구하여 보자.

$$\sum_{k=1}^n F_{4k} = A_4$$

**[풀이]**

점화식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} \sum_{k=1}^n F_{4k} &= F_4 + F_8 + \dots + F_{4n} \\ &= [F_2 + F_3] + [F_6 + F_7] + \dots + [F_{4k-2} + F_{4k-1}] \end{aligned}$$

우변을 고쳐 쓰면

$$\begin{aligned} \sum_{k=1}^n F_{4k} &= [F_2 + F_3] + [F_6 + F_7] + \dots + [F_{4k-2} + F_{4k-1}] \\ &= \sum_{k=1}^{4n} F_k - \sum_{k=1}^n F_{4k} - \sum_{k=1}^n F_{4k-3} \\ &= F_{4n+2} - 1 - \sum_{k=1}^n F_{4k} - \sum_{k=1}^n F_{4k-3} \end{aligned}$$

양변을 정리하면

$$\sum_{k=1}^n F_{4k} = \frac{1}{2} \left( F_{4n+2} - 1 - \sum_{k=1}^n F_{4k-3} \right)$$

이다. 즉  $A_4 = \frac{1}{2} \left( F_{4n+2} - 1 - \sum_{k=1}^n F_{4k-3} \right)$ 이다.

[연구 1-6] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 다음 등식을 만족하는  $A_5$ 를 구하여 보자.

$$\sum_{k=1}^n F_{5k} = A_5$$

[풀이]

점화식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} \sum_{k=1}^n F_{5k} &= F_5 + F_{10} + \cdots + F_{5n} \\ &= [F_3 + F_4] + [F_8 + F_9] + \cdots + [F_{5k-2} + F_{5k-1}] \\ &= [(F_1 + F_2) + F_4] + [(F_6 + F_7) + F_9] + \cdots + [(F_{5k-4} + F_{5k-2}) + F_{5k-1}] \end{aligned}$$

우변을 고쳐 쓰면

$$\sum_{k=1}^n F_{5k} = \sum_{k=1}^{5n} F_k - \sum_{k=1}^n F_{5k-2} - \sum_{k=1}^n F_{5k}$$

한편  $\sum_{k=0}^{5n} F_k = F_{5n+2} - 1$  이므로 양변을 정리하면

$$\sum_{k=1}^n F_{5k} = \frac{1}{2} \left( F_{5n+2} - 1 - \sum_{k=1}^n F_{5k-2} \right)$$

이다. 즉  $A_5 = \frac{1}{2} \left( F_{5n+2} - 1 - \sum_{k=1}^n F_{5k-2} \right)$ 이다.

[참고 2] [연구 1-3~1-6]의 결과를 참고하여 피보나치 수 중  $j$ 번째 배수 위치의 수들의 합  $\sum_{k=1}^n F_{jk}$ 에 대한 일반적인 우변의 패턴  $A_j$ 가 존재하는지 찾아보고자 하였으나 성공하지 못하였다.

#### 4. 2. 멀어지는 두 피보나치 수 곱의 점화식

이 절에서는 피보나치 수의  $n$ 항까지 제곱의 합에 대한 아래 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1} \tag{4.2}$$

의 수학적 특성을 이해하고 이 항등식이 성립됨을 몇 가지 방법으로 증명하여 위키백과([5, 6])에서 찾을 수 없었던 증명을 확인한다. 여기에서는 수열의 성질 및 수학적귀납법을 사용하여 증명한다. 또 새로운 항등식의 존재가 가능한지는 좀 더 깊이 있게 조사하고 연구한다.

[연구 2-1] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립할 때, 피보나치 수의 다음 항등식이 성립함을 확인하여 보자.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1} \quad .$$

**[풀이 1]**

피보나치 수의 각항이

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$$

을 만족하므로 연속하는 두 피보나치 수의 곱은 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_n F_{n+1} &= F_n (F_n + F_{n-1}) \\ &= F_n^2 + F_n F_{n-1} \\ &= F_n^2 + F_{n-1} (F_{n-1} + F_{n-2}) \\ &= F_n^2 + F_{n-1}^2 + F_{n-1} F_{n-2} \end{aligned}$$

$$\dots$$

$$= F_n^2 + F_{n-1}^2 + F_{n-2}^2 + \dots + F_2^2 + F_2 F_1$$

한편,  $F_1 = 1, F_2 = 1$  이므로 위 식을 정리하면

$$F_n^2 + F_{n-1}^2 + F_{n-2}^2 + \dots + F_2^2 + F_1^2 = F_n F_{n+1}$$

임을 알 수 있다. 즉  $\sum_{k=0}^n F_k^2 = F_n F_{n+1}$  이 성립한다.

**[풀이 2]**

①  $n = 1$ 인 경우

$$(\text{좌변}) = \sum_{k=1}^1 F_k = F_1 = 1, (\text{우변}) = F_3 - 1 = 2 - 1 = 1$$

이므로 (좌변)=(우변)이 성립한다.

②  $n = p$ 인 경우(가정)

$$\sum_{k=1}^p F_k^2 = F_p F_{p+1}$$

③  $n = p+1$ 인 경우

$$\sum_{k=1}^{p+1} F_k^2 = \sum_{k=1}^p F_k^2 + F_{p+1}^2 = F_p F_{p+1} + F_{p+1}^2 = F_{(p+1)} [F_p + F_{p+1}] = F_{(p+1)} F_{p+2}$$

가 성립된다.

따라서 ①, ②, ③에 의하여 항등식이 성립됨을 알 수 있다.

다음은  $n$ 항까지 제곱의 합에 대한 항등식 (4.2) 우변 두 피보나치 수의 곱에 주목하여 ‘떨어지는 두 피보나치 수 곱의 점화식’을 찾아보고자 한다. 먼저 우변의 두 피보나치 수의 곱  $F_n F_{n+1}$  대신  $F_n F_{n+2}$ ,  $F_n F_{n+3}$ ,  $F_n F_{n+4}$ , ... 등에 대한 새로운 항등식을 얻을 수 있음을 보였다.

[연구 2-2] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $B_2$ 을 구하여 보자.

$$F_n F_{n+2} = B_2$$

[풀이]

피보나치 수의 각항이

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$$

을 만족하므로 연속하는 두 피보나치 수의 곱은 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_n F_{n+2} &= F_n (F_n + F_{n+1}) \\ &= F_n^2 + F_n F_{n+1} \\ &= F_n^2 + \sum_{k=1}^n F_k^2 \end{aligned}$$

이 성립한다.

[연구문제 2-3] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $B_3$ 을 구하여 보자.

$$F_n F_{n+3} = B_3$$

[풀이]

피보나치 수의 각항이

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$$

을 만족하므로 연속하는 두 피보나치 수의 곱은 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_n F_{n+3} &= F_n (F_{n+1} + F_{n+2}) \\ &= F_n F_{n+1} + F_n F_{n+2} \\ &= F_n F_{n+1} + F_n [F_n + F_{n+1}] \\ &= F_n^2 + 2 \sum_{k=1}^n F_k^2 \end{aligned}$$

이 성립한다.

[연구 2-4] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $B_4$ 을 구하여 보자.

$$F_n F_{n+4} = B_4$$

**[풀이]**

피보나치 수의 각항이

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$$

을 만족하므로 연속하는 두 피보나치 수의 곱은 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_n F_{n+4} &= F_n (F_{n+2} + F_{n+3}) \\ &= F_n F_{n+2} + F_n F_{n+3} \\ &= F_n^2 + \sum_{k=1}^n F_k^2 + F_n^2 + 2 \sum_{k=1}^n F_k^2 \\ &= 2F_n^2 + 3 \sum_{k=1}^n F_k^2 \end{aligned}$$

이 성립한다.

**[참고 3]** [연구 2-4]의 항등식 좌변 두 번째 피보나치 수의  $F_n F_{n+4}$  대신  $F_n F_{n+5}$ 를 대입하면 새로운 항등식

$$F_n F_{n+5} = 3F_n^2 + 5 \sum_{k=1}^n F_k^2$$

을 얻는다. 따라서 ‘떨어지는 두 피보나치 수 곱의 점화식’을 유추해 보면 다음과 같은 항등식

$$F_n F_{n+i} = F_{i-1} F_n^2 + F_i \sum_{k=1}^n F_k^2, i \geq 2$$

이 성립함을 짐작할 수 있다.

**[연구 2-5]** 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 항등식을 만족함을 증명하라.

$$F_n F_{n+i} = F_{i-1} F_n^2 + F_i \sum_{k=1}^n F_k^2, i \geq 2$$

**[풀이]** 수학적 귀납법

①  $i = 2$ 인 경우

연속하는 두 피보나치 수의 곱은 다음과 같이 표현할 수 있다.

$$F_n F_{n+2} = F_n (F_n + F_{n+1}) = F_n^2 + F_n F_{n+1} = F_n^2 + \sum_{k=1}^n F_k^2$$

이 성립한다.

②  $i = p-1, p$ 인 경우(가정)

$$F_n F_{n+p} = F_{p-1} F_n^2 + F_p \sum_{k=1}^n F_k^2$$

③  $i = p+1$ 인 경우

$$\begin{aligned} F_n F_{n+p+1} &= F_n [F_{n+p} + F_{n+p-1}] = F_n F_{n+p} + F_n F_{n+p-1} \\ &= F_{p-1} F_n^2 + F_p \sum_{k=1}^n F_k^2 + F_n F_{n+p-1} \\ &= F_{p-1} F_n^2 + F_p \sum_{k=1}^n F_k^2 + F_{p-2} F_n^2 + F_{p-1} \sum_{k=1}^n F_k^2 \\ &= [F_{p-1} + F_{p-2}] F_n^2 + [F_p + F_{p-1}] \sum_{k=1}^n F_k^2 \\ &= F_p F_n^2 + F_{p+1} \sum_{k=1}^n F_k^2 \end{aligned}$$

가 성립된다. 따라서 ①, ②, ③에 의하여 항등식이 성립됨을 알 수 있다.

#### 4. 3. 피보나치 수 제곱 합의 점화식

이 절에서는 3.2절의 ‘멀어지는 두 피보나치 수 곱의 점화식’은  $n$ 항까지 제곱의 합에 대한 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

에서 출발하여 좌변의 피보나치 수 제곱의 합에 주목하여 새로운 점화식 발견을 목표로 하였다. 우변의 두 피보나치 수의 다른 유형을 찾기 위해  $F_n F_{n+1}$  대신 두 피보나치 수의 곱에 다른 항을 더한 패턴의 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+2} - B_2, \quad \sum_{k=1}^n F_k^2 = F_n F_{n+3} - B_3, \quad \sum_{k=1}^n F_k^2 = F_n F_{n+4} - B_4, \quad \dots \dots ,$$

등을 조사하였다. 또 항등식이 일반적인 점화식이 성립하는지 조사하였으며 이러한 점화식이 성립함을 수학적귀납법으로 증명할 것이다.

[연구 3-1] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $C_2$ 을 구하여 보자.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+2} - C_2.$$

[풀이]

[연구 2-2]의 결과는 다음과 같이 표현할 수 있다.

$$F_n F_{n+2} = F_n (F_n + F_{n+1}) = F_n^2 + F_n F_{n+1} = F_n^2 + \sum_{k=1}^n F_k^2.$$

따라서

$$\sum_{k=1}^n F_k^2 = F_n F_{n+2} - F_n^2$$

로 표현할 수 있다. 즉  $C_2 = F_n^2 = F_n F_n$  이다.

[연구 3-2] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $C_3$ 을 구하여 보자.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+3} - C_3.$$

[풀이]

[연구 2-3]의 결과는 다음과 같이 표현할 수 있다.

$$\begin{aligned} F_n F_{n+3} &= F_n (F_{n+1} + F_{n+2}) = F_n F_{n+1} + F_n F_{n+2} = F_n F_{n+1} + F_n [F_n + F_{n+1}] \\ &= F_n^2 + 2 \sum_{k=1}^n F_k^2 \end{aligned}$$

따라서

$$\sum_{k=1}^n F_k^2 = F_n F_{n+3} - F_n^2 - \sum_{k=1}^n F_k^2$$

으로 표현할 수 있다. 위의 결과  $\sum_{k=1}^n F_k^2 = F_n F_{n+2} - F_n^2$  를 적용하면

$$\sum_{k=1}^n F_k^2 = F_n F_{n+3} - F_n F_{n+2}$$

을 얻을 수 있다. 즉  $C_3 = F_n F_{n+2} = F_n [F_n + F_{n+1}]$  이다.

[연구 3-3] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $C_4$ 을 구하여 보자.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+4} - C_4.$$

**[풀이]**

[연구 2-4]의 결과는 다음과 같이 표현할 수 있다.

$$F_n F_{n+4} = F_n (F_{n+2} + F_{n+3}) = F_n (F_{n+1} + F_n) + F_n F_{n+3} = F_n F_{n+1} + F_n F_n + F_n F_{n+3}$$

따라서

$$\sum_{k=1}^n F_k^2 = F_n F_{n+4} - F_n F_n - F_n F_{n+3}$$

이다. 따라서  $C_4 = F_n F_n + F_n F_{n+3} = F_n [F_n + F_{n+1} + F_{n+2}]$

이다.

**[연구 3-4]** 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $C_5$ 을 구하여 보자.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+5} - C_5$$

**[풀이]**

[연구 10]의 결과는 다음과 같이 표현할 수 있다.

$$F_n F_{n+5} = F_n (F_{n+4} + F_{n+3}) = F_n (F_{n+3} + F_{n+2} + F_{n+3}) \\ = F_n (F_{n+1} + F_{n+2} + F_{n+2} + F_{n+3})$$

위 식으로부터

$$F_n F_{n+5} = F_n F_{n+1} + F_n (F_n + F_{n+1} + F_{n+2} + F_{n+3})$$

이 얻어진다. 다시 정리하면

$$F_n F_{n+1} = F_n F_{n+5} - F_n (F_n + F_{n+1} + F_{n+2} + F_{n+3})$$

이고

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1} = F_n F_{n+5} - F_n (F_n + F_{n+1} + F_{n+2} + F_{n+3})$$

이다. 따라서

$$C_5 = F_n (F_n + F_{n+1} + F_{n+2} + F_{n+3})$$

이다.

**[참고 4]** [연구 3-1~3-4] 번의 결과를 종합하면  $\sum_{k=1}^n F_k^2 = F_n F_{n+i} - B_i$  와 같은 패턴

이 존재함을 알 수 있다. 우변  $n$ 항까지 피보나치 수의 제곱의 합은 각각  $F_n F_{n+i} - B_i$  와 같이 표현하는 것이 가능할 것으로 짐작되어 다음과 같은 항등식이 성립함을 다음 연구[3-5]에서 증명한다.

$$F_n F_{n+1} = F_n F_{n+i} - F_n (F_n + F_{n+1} + \dots + F_{n+i-2})$$

또는

$$\sum_{k=1}^n F_k^2 = F_n F_{n+i} - F_n \sum_{j=0}^{i-2} F_{n+j}, \quad (i \geq 2).$$

[연구 3-5] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식이 만족 됨을 증명하라.

$$F_n F_{n+1} = F_n F_{n+i} - F_n (F_n + F_{n+1} + \dots + F_{n+i-2}) \quad (4-3)$$

또는

$$\sum_{k=1}^n F_k^2 = F_n F_{n+i} - F_n \sum_{j=0}^{i-2} F_{n+j}, \quad (i \geq 2). \quad (4-3')$$

**[풀이]** 수학적귀납법

①  $i = 2$ 인 경우

$$(우변) = F_n F_{n+2} - F_n (F_n) = F_n (F_{n+2} - F_n) = F_n (F_{n+1} + F_n - F_n) = F_n F_{n+1}$$

이다. 따라서 (좌변)=(우변)이 성립한다.

②  $i = p$ 인 경우(가정)

$$-F_n F_{n+1} = F_n F_{n+p} - F_n (F_n + F_{n+1} + \dots + F_{n+p-2})$$

③  $i = p+1$ 인 경우

먼저 우변을 계산하면

$$\begin{aligned} (우변) &= F_n F_{n+p+1} - F_n (F_n + F_{n+1} + \dots + F_{n+p-2} + F_{n+p-1}) \\ &= F_n (F_{n+p} + F_{n+p-1}) - F_n (F_n + F_{n+1} + \dots + F_{n+p-2} + F_{n+p-1}) \\ &= F_n F_{n+p} - F_n (F_n + F_{n+1} + \dots + F_{n+p-2}) \end{aligned}$$

이다. 위 귀납법 가정에 의하여 (우변) =  $F_n F_{n+1}$  이 성립된다.

따라서 ①, ②, ③에 의하여 항등식이 성립됨을 알 수 있다.

#### 4. 4. 점화식으로부터의 항등식

이 절에서는 3.1절 ‘피보나치 수의 합’에서는 피보나치 수의  $n$ 항까지 합에 대한 항등식

$$\sum_{k=1}^n F_k = F_{n+2} - 1, \quad n \geq 1$$

과 3.2절 ‘멀어지는 두 피보나치 수 곱의 점화식’은  $n$ 항까지 곱의 합에 대한 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

에서 출발하여 우변의 두 피보나치 수의 곱에 주목하여 피보나치 수 곱의 합의

다른 유형의 점화식을 유도하였다. 즉 우변의 두 피보나치 수의 다른 유형을 찾기 위해 아래 유형의 피보나치 수 항등식

$$\sum_{k=1}^n F_k = D_1, \quad \sum_{k=1}^n F_{2k-1} = D_2, \quad \sum_{k=1}^n F_k^2 = D_3, \quad F_n F_{n+i} = F_{i-1} F_n^2 + F_i \sum_{k=1}^n F_k^2, \quad \dots \dots,$$

등의 일반적인 점화식이 성립하는지 조사하였으며 이러한 항등식이 성립함을 수열의 성질을 증명하였다.

[연구 4-1] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $D_1$ 을 구하여 보자.

$$\sum_{k=1}^n F_k = D_1, \quad n \geq 2$$

[풀이]

항등식  $\sum_{k=1}^n F_k = F_{n+2} - 1$  은 아래와 같이 전개된다.

$$\begin{aligned} \sum_{k=1}^n F_k &= (F_n + F_{n+1}) + F_{n+1} - F_{n+1} - 1 \\ &= 2F_{n+1} + F_n - F_{n+1} - 1 \\ &= 2F_{n+1} - F_{n-1} - 1 \\ &= 2(F_n + F_{n-1}) - F_{n-1} - 1 \\ &= 2F_n + F_{n-1} - 1 \end{aligned}$$

여기서 맨 마지막 등식 우변의  $2F_n + F_{n-1} - 1$ 을 다시 쓰면

$$\begin{aligned} 2F_n + F_{n-1} - 1 &= 2(F_{n-1} + F_{n-2}) + F_{n-1} - 1 \\ &= 3(F_{n-2} + F_{n-3}) + 2F_{n-2} - 1 \\ &= 5F_{n-2} + 3F_{n-3} - 1 \end{aligned}$$

로 쓸 수 있다. 같은 방법으로 위 맨 마지막 등식 우변  $5F_{n-2} + 3F_{n-3} - 1$ 을 다시 전개하여 보면

$$5F_{n-2} + 3F_{n-3} - 1 = 8F_{n-3} + 5F_{n-4} - 1$$

임을 알 수 있다. 위 식 우변 역시

$$8F_{n-3} + 5F_{n-4} - 1 = 13F_{n-4} + 8F_{n-5} - 1$$

와 같이 쓸 수 있다. 따라서 피보나치 수의 합 등식

$$\sum_{k=1}^n F_k = D_1 \text{ 에서 우변 } D_1 \text{ 은 새로운 피보나치 수}$$

$$D_1 = F_{j+1} F_{n+2-j} + F_j F_{n+1-j} - 1, \quad (j = 2, 3, 4, \dots; n+1-j > 0)$$

으로 나타남을 알 수 있다.

[참고 5] 위 연구 4-1에서 등식  $\sum_{k=1}^n F_k = F_{n+2} - 1$ 에서 우변은 다음과 같은 각각의

$j$ 에 대하여 새로운 패턴

$$\sum_{k=1}^n F_k = F_{j+1}F_{n+2-j} + F_jF_{n+1-j} - 1, (j=2,3,4,\dots; n+1-j > 0)$$

이 성립됨을 확인하였다.

[연구 4-2] 점화식  $F_0=0, F_1=1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $D_2$ 을 구하여 보자.

$$\sum_{k=1}^n F_{2k-1} = D_2, n \geq 1$$

[풀이]

항등식  $\sum_{k=1}^n F_{2k-1} = F_{2n}$  은 아래와 같이 전개된다.

$$\begin{aligned} \sum_{k=1}^n F_{2k-1} &= F_{2n} \\ &= F_{2n-1} + F_{2n-2} \\ &= F_{2n-1} + F_{2n-2} + F_{2n-2} - F_{2n-2} \\ &= 2F_{2n-2} + F_{2n-1} - F_{2n-2} \\ &= 2F_{2n-2} + F_{2n-3} \end{aligned}$$

여기서 맨 마지막 등식 우변의  $2F_{2n-2} + F_{2n-3}$ 을 다시 쓰면

$$\begin{aligned} 2F_{2n-2} + F_{2n-3} &= 2(F_{2n-3} + F_{2n-4}) + F_{2n-3} \\ &= 3F_{2n-3} + 2F_{2n-4} \end{aligned}$$

로 쓸 수 있다. 같은 방법으로 위 맨 마지막 등식 우변  $3F_{2n-3} + 2F_{2n-4}$ 을 다시 전개하여 보면

$$3F_{2n-3} + 2F_{2n-4} = 5F_{2n-4} + 3F_{2n-5} = 8F_{2n-5} + 5F_{2n-6}$$

임을 알 수 있다. 따라서 피보나치 수의 합 등식

$$\sum_{k=1}^n F_{2k-1} = D_2, n \geq 1 \text{ 에서 우변 } D_2 \text{ 는 새로운 피보나치 수}$$

$D_2 = F_j F_{2n-j+1} + F_{j-1} F_{2n-j}, (j=2,3,4,\dots \mid 2n-j > 0)$  로 나타남을 알 수 있다.

[참고 6] 위 연구 4-2에서는 등식  $\sum_{k=1}^n F_{2k-1} = F_{2n}$  에서 우변은 다음과 같은 각각의

$j$ 에 대하여 새로운 패턴

$$\sum_{k=1}^n F_{2k-1} = F_j F_{2n-j+1} + F_{j-1} F_{2n-j}$$

이 성립됨을 확인하였다.

[연구 4-3] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $D_3$ 을 구하여 보자.

$$\sum_{k=1}^n F_{2k} = D_3$$

[풀이]

항등식  $\sum_{k=1}^n F_{2k} = F_{2n+1} - 1$ 은 아래와 같이 전개된다.

$$\begin{aligned} F_{2n+1} &= F_{2n} + F_{2n-1} \\ &= F_{2n} + F_{2n-1} + F_{2n} - F_{2n} \\ &= 2F_{2n} + F_{2n-1} - F_{2n} \\ &= 2F_{2n} + F_{2n-2} \end{aligned}$$

같은 방법으로 위 맨 마지막 등식 우변  $2F_{2n} + 2F_{2n-2}$ 을 다시 전개하여 보면

$$F_{2n+1} = F_j F_{2n+2-j} + F_{j-1} F_{2n+1-j}$$

임을 알 수 있다. 따라서 피보나치 수의 합 등식

$$\sum_{k=1}^n F_{2k} = D_3, \quad n \geq 1 \text{ 에서 우변 } D_3 \text{ 는 새로운 피보나치 수}$$

$D_3 = F_j F_{2n+2-j} + F_{j-1} F_{2n+1-j} - 1, (j=1, 2, 3, 4, \dots \mid 2n+1-j > 0)$  로 나타남을 알 수 있다.

[참고 7] 위 연구 4-3에서는 등식  $\sum_{k=1}^n F_{2k} = F_{2n+1} - 1$  에서 우변은 다음과 같은 각

각의  $j$ 에 대하여 새로운 패턴

$$\sum_{k=1}^n F_{2k} = F_j F_{2n+2-j} + F_{j-1} F_{2n+1-j} - 1$$

이 성립됨을 확인하였다.

[연구 4-4] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치

수의  $n$ 번째 항을  $F_n$  ( $2 \geq n$ ) 라 할 때, 다음 등식을 만족하는  $D_4$ 을 구하여 보자.

$$\sum_{k=1}^n F_{3k} = D_4$$

[풀이]

항등식  $\sum_{k=1}^n F_{3k} = \frac{1}{2}(F_{3n+2} - 1)$ 은 아래와 같이 전개된다.

$$\begin{aligned} F_{3n+2} &= F_{3n+1} + F_{3n} \\ &= F_{3n} + F_{3n-1} + F_{3n} \\ &= 2F_{3n} + F_{3n-1} \\ &= 2(F_{3n-1} + F_{3n-2}) + F_{3n-1} \\ &= 3F_{3n-1} + 2F_{3n-2} \end{aligned}$$

같은 방법으로 위 맨 마지막 등식 우변  $3F_{3n-1} + 2F_{3n-2}$ 을 다시 전개하여 보면

$$F_{3n+2} = F_j F_{3(n+1)-j} + F_j F_{3n+2-j}$$

임을 알 수 있다. 따라서 피보나치 수의 합 등식

$$\sum_{k=1}^n F_{3k} = D_4, \quad n \geq 1 \text{ 에서 우변 } D_4 \text{ 는 새로운 피보나치 수}$$

$D_4 = \frac{1}{2}(F_j F_{3(n+1)-j} + F_j F_{3n+2-j} - 1)$ , ( $j=1, 2, 3, 4, \dots \mid 3n+2-j > 0$ ) 로 나타낼 수 있다.

[참고 8] 위 연구 4-4에서는 등식  $\sum_{k=1}^n F_{3k} = \frac{1}{2}(F_{3n+2} - 1)$  에서 우변은 다음과 같은

각각의  $j$ 에 대하여 새로운 패턴

$$\sum_{k=1}^n F_{3k} = \frac{1}{2}(F_j F_{3(n+1)-j} + F_j F_{3n+2-j} - 1)$$

이 성립됨을 확인하였다.

[연구 4-5] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n$  ( $2 \geq n$ ) 라 할 때, 다음 등식을 만족하는  $D_5$ 을 구하여 보자.

$$\sum_{k=1}^n F_k^2 = D_5$$

[풀이]

항등식  $\sum_{k=1}^n F_k^2 = F_n F_{n+1}$ 의 피보나치 수  $F_n$ 은 아래와 같이 전개된다.

$$\begin{aligned}
& F_n \\
&= F_{n-1} + F_{n-2} + F_{n-2} - F_{n-2} \\
&= 2F_{n-2} + F_{n-1} - F_{n-2} \\
&= 2F_{n-2} + F_{n-3} \\
&= 2(F_{n-3} + F_{n-4}) + F_{n-3} \\
&= 3F_{n-3} + 2F_{n-4}
\end{aligned}$$

같은 방법으로 위 맨 마지막 등식 우변  $3F_{n-3} + 2F_{n-4}$ 을 다시 전개하여 보면

$$F_n = F_j F_{n+1-j} + F_{j-1} F_{n-j}, \quad (j=1, 2, \dots; n-j > 0) \quad \text{----- ①}$$

이다. 한편 피보나치 수  $F_{n+1}$ 도 비슷하게 전개하면 다음과 같다.

$$F_{n+1} = F_j F_{n+2-j} + F_{j-1} F_{n+1-j}, \quad (j=1, 2, \dots; n+1-j > 0) \quad \text{----- ②}$$

따라서 ①식과 ②식으로부터 피보나치 수 제곱의 합  $\sum_{k=1}^n F_k^2 = D_5$  에서 우변  $D_5$  는

새로운 피보나치 수는 각각의  $j(j=1, 2, 3, 4, \dots \mid n+1-j > 0)$ 에 대하여

$$D_5 = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_j F_{n+2-j} + F_{j-1} F_{n+1-j}) \text{ 이 됨을 알 수 있다.}$$

[참고 9] 위 연구 4-5에서는 등식  $\sum_{k=1}^n F_k^2 = F_n F_{n+1}$  에서 우변은 다음과 같은 각각의

$j$ 에 대하여 새로운 패턴

$$\sum_{k=1}^n F_k^2 = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_j F_{n+2-j} + F_{j-1} F_{n+1-j})$$

이 성립됨을 확인하였다.

[연구 4-6] 점화식  $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 2$  이 성립하고 피보나치 수의  $n$ 번째 항을  $F_n (2 \geq n)$  라 할 때, 다음 등식을 만족하는  $D_6$ 을 구하여 보자.

$$F_i \sum_{k=1}^n F_k^2 = F_n F_{n+i} - F_{i-1} F_n^2 \equiv D_6$$

[풀이]

항등식

$$F_i \sum_{k=1}^n F_k^2 = F_n (F_{n+i} - F_{i-1} F_n) \quad \text{----- ①}$$

의 피보나치 수  $F_n$ 은 아래와 같이 전개된다.

$$\begin{aligned}
& F_n \\
&= F_{n-1} + F_{n-2} + F_{n-2} - F_{n-2} \\
&= 2F_{n-2} + F_{n-1} - F_{n-2} \\
&= 2F_{n-2} + F_{n-3} \\
&= 2(F_{n-3} + F_{n-4}) + F_{n-3} \\
&= 3F_{n-3} + 2F_{n-4}
\end{aligned}$$

같은 방법으로 위 맨 마지막 등식 우변  $3F_{n-3} + 2F_{n-4}$ 을 다시 전개하여 보면

$$F_n = F_j F_{n+1-j} + F_{j-1} F_{n-j} \quad (j=1, 2, \dots; n-j > 0)$$

임을 알 수 있다. 따라서 ①식의 우변은

$$F_n(F_{n+i} - F_{i-1}F_n) = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_{n+i} - F_{i-1}F_n)$$

$F_i \sum_{k=1}^n F_k^2 = D_6$  모양에서 우변  $D_6$ 는 새로운 피보나치 수

$D_6 = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_{n+i} - F_{i-1}F_n)$ ,  $(i, j=1, 2, 3, 4, \dots; n-j > 0)$ 로 나타낼 수 있다.

[참고 10] 위 연구 4-6에서는 등식  $F_i \sum_{k=1}^n F_k^2 = F_n F_{n+i} - F_{i-1} F_n^2$ 에서 우변은 다음과 같은 각각의  $j$ 에 대하여 새로운 패턴

$$F_i \sum_{k=1}^n F_k^2 = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_{n+i} - F_{i-1}F_n)$$

이 성립됨을 확인하였다.

## V. 결론 및 제언

피보나치수열의 덧셈과 제곱 합에 관한 점화식의 연구 결과는 다양한 형태로 나타났다. 아래에는 피보나치수열의 덧셈과 제곱 합에 관련된 몇 가지 점화식을 기반으로 도출한 여러 항등식 중 주요한 결과를 나열하였다.

### ① 피보나치 수의 합:

피보나치 수의 합에서는 이미 알려진 피보나치 수에 대한 항등식

$$\sum_{k=1}^{n-1} F_k = F_{n+1} - 1, \quad n \geq 1$$

이 성립함을 수열의 성질과 수학적귀납법으로 증명하여 항등식이 성립함을 확인하였다. 또  $F_1$ 에서 출발하는 홀수 번째 항들의 합,  $F_2$ 에서 출발하는 짝수 번째 항들의

합 등 아래 피보나치 수의 합의 다음과 같은 항등식이 성립함을 탐구하였다.

$$\sum_{k=1}^n F_{2k-1} = F_{2n}, \quad \sum_{k=1}^n F_{2k} = F_{2n+1} - 1, \quad \sum_{k=1}^n F_{3k} = \frac{1}{2}(F_{3n+2} - 1),$$

$$\sum_{k=1}^n F_{4k} = \frac{1}{2}\left(F_{4n+2} - 1 - \sum_{k=1}^n F_{4k-3}\right), \quad \sum_{k=1}^n F_{5k} = \frac{1}{2}\left(F_{5n+2} - 1 - \sum_{k=1}^n F_{5k-2}\right).$$

### ② 멀어지는 두 피보나치 수 곱의 점화식:

멀어지는 두 피보나치 수 곱의 점화식에서는 피보나치 수의  $n$ 항까지 곱의 합에 대한 아래 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

의 특성을 이해하고 이 항등식이 성립됨을 수열의 성질 및 수학적귀납법을 사용하여 증명하였다. 또 이 항등식의 우변의 두 피보나치 수의 곱에 주목하여 곱의 패턴을 찾기 위해  $F_n F_{n+1}$  대신 점차 멀어지는 두 피보나치 수의 곱에 대한 항등식

$$F_n F_{n+2} = A_1, \quad F_n F_{n+3} = A_2, \quad F_n F_{n+4} = A_3, \quad \dots \dots, \quad F_n F_{n+i} = A_{i-1}$$

등을 조사하였다. 그 결과 멀어지는 두 피보나치 수의 곱에 대하여 다음과 같은 항등식이 성립함을 증명하였다.

$$F_n F_{n+2} = F_n^2 + \sum_{k=1}^n F_k^2, \quad F_n F_{n+3} = F_n^2 + 2 \sum_{k=1}^n F_k^2, \quad F_n F_{n+4} = 2F_n^2 + 3 \sum_{k=1}^n F_k^2,$$

$$F_n F_{n+5} = 3F_n^2 + 5 \sum_{k=1}^n F_k^2.$$

또 위 항등식에서 아래의 멀어지는 두 피보나치 수 곱의 점화식을 찾을 수 있었다.

$$F_n F_{n+i} = F_{i-1} F_n^2 + F_i \sum_{k=1}^n F_k^2, \quad i \geq 2.$$

### ③ 피보나치 수 곱의 합 점화식:

피보나치 수 곱의 합 점화식에서는  $n$ 항까지 곱의 합에 대한 항등식

$$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$$

의 우변의 두 피보나치 수의 다른 유형을 찾기 위해  $F_n F_{n+1}$  대신 두 피보나치 수의 곱에 새로운 점화식을 발견을 목표로 다음과 같은 항등식을 얻었다.

$$\sum_{k=1}^n F_k^2 = F_n F_{n+2} - F_n^2, \quad \sum_{k=1}^n F_k^2 = F_n F_{n+3} - F_n F_{n+2}, \quad \sum_{k=1}^n F_k^2 = F_n F_{n+4} - F_n F_n - F_n F_{n+3},$$

$$\sum_{k=1}^n F_k^2 = F_n F_{n+5} - F_n (F_n + F_{n+1} + F_{n+2} + F_{n+3}).$$

위 결과로부터  $\sum_{k=1}^n F_k^2 = F_n F_{n+i} - B_i$  와 같은 패턴이 존재함을 알 수 있었다. 우변  $n$

항까지 피보나치 수의 제곱의 합은 각각  $F_n F_{n+i} - B_i$  와 같이 표현하는 것이 가능하며, 따라서 다음과 같은 항등식이

$$F_n F_{n+1} = F_n F_{n+i} - F_n (F_n + F_{n+1} + \dots + F_{n+i-2}) \text{ 또는}$$

$$\sum_{k=1}^n F_k^2 = F_n F_{n+i} - F_n \sum_{j=0}^{i-2} F_{n+j}, \quad (i \geq 2)$$

이 성립함을 확인하고 수학적귀납법으로 증명하였다.

#### ④ 점화식으로부터의 항등식:

점화식으로부터의 항등식에서는 피보나치 수의 점화식을 이용하여 얻어진  $n$ 항까지

$$\text{합에 대한 항등식 } \sum_{k=1}^n F_k = F_{n+2} - 1$$

과  $n$ 항까지 제곱의 합에 대한 항등식  $\sum_{k=1}^n F_k^2 = F_n F_{n+1}$  에서 출발하여 우변의 피보나치 수의 표현에 주목하여 피보나치 수 합과 곱의 새로운 항등식을 유도하였다. 그 결과 피보나치 수의 합에 대한 우변의 피보나치 수의 합과 곱에 대한 몇몇 항등식을 찾았다. 요약하면 아래와 같다.

알려진 항등식(1~3절)	합과 곱을 포함하는 항등식
$\sum_{k=1}^n F_k = F_{n+2} - 1$	$\sum_{k=1}^n F_k = F_{j+1} F_{n+2-j} + F_j F_{n+1-j} - 1,$ ( $j = 2, 3, 4, \dots; n+1-j > 0$ )
$\sum_{k=1}^n F_{2k-1} = F_{2n}$	$\sum_{k=1}^n F_{2k-1} = F_j F_{2n-j+1} + F_{j-1} F_{2n-j}$
$\sum_{k=1}^n F_{2k} = F_{2n+1} - 1$	$\sum_{k=1}^n F_{2k} = F_j F_{2n+2-j} + F_{j-1} F_{2n+1-j} - 1$
$\sum_{k=1}^n F_{3k} = \frac{1}{2}(F_{3n+2} - 1)$	$\sum_{k=1}^n F_{3k} = \frac{1}{2}(F_j F_{3(n+1)-j} + F_j F_{3n+2-j} - 1)$
$\sum_{k=1}^n F_k^2 = F_n F_{n+1}$	$\sum_{k=1}^n F_k^2 = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_j F_{n+2-j} + F_{j-1} F_{n+1-j})$
$F_i \sum_{k=1}^n F_k^2 = F_n F_{n+i} - F_{i-1} F_n^2$	$F_i \sum_{k=1}^n F_k^2 = (F_j F_{n+1-j} + F_{j-1} F_{n-j})(F_{n+i} - F_{i-1} F_n)$

연구를 진행하는 중, 참고 문헌([6])에서 “피보나치수열은 황금 비율과 밀접한 관련이 있으며 피보나치 수는 또한 비슷한 점화식을 따르는 ‘루카스 수열’의 루카스 수(Lucas numbers)와 밀접한 관련이 있다.”라고 소개되어 있으나 우리는 시간 관계상이 부분의 탐구는 다음 기회로 미루기로 하였다.

기회가 되면 프랑스 수학자 자크 필립 마리 비네(Jacques Philippe Marie Binet)의

이름을 따서 명명된 비네의 공식으로 알려지게 황금비와 피보나치 수 관계를 공부하여 보고 싶다.

피보나치수열을 설명하는 또 다른 방법으로 행렬을 이용한 선형 차분 방정식의 2차원 시스템이 있다고 간략하게 소개되어 있었으나 역시 우리의 이해를 넘어서는 문제라서 아쉬움이 있었다. 수학을 전공하게 되면 다시 행렬의 차분 방정식 해법에 도전해 보고 싶다.

## VI. 참고문헌

- [1] 강홍식 외 3 번역, 이산수학 (Richard Johnsonbaugh, Discrete Mathematics), (주) 교보문고, (1999)
- [2] 김민준 외, 자연수 제곱 합 및 곱의 합에 대한 공식, 2021년도 창원대학교 과학영재교육원 초등수학 사사반 논문집, (2021)
- [3] 노주석, 김유라, 유명한 수명을 갖는 토끼에 의해 생성되는 피보나치수열의 점화식, 과학영재교육, 10(1), pp. 55-61, (2018.4)
- [4] 양영오, 김태호, 피보나치수열의 일반화에 관한 고찰, 한국수학사학회지, 제21권, 제4호, pp. 87-104, (2008. 11)
- [5] <https://ko.wikipedia.org/wiki/>
- [6] <https://en.wikipedia.org/wiki/>
- [7] <https://terms.naver.com/>

수학A

09\_소수를 활용한 효율적인 스포츠 트레이닝  
스케줄



# 소수를 활용한 효율적인 스포츠 트레이닝 스케줄

안다윗 (왕산중학교 3학년)

진 율 (관동중학교 2학년)

지도교수 : 김광훈 (강릉원주대학교 수학과물리학과)

**초록 :** 추상적 수학 개념의 구체화를 통해 해당 개념의 실생활에서의 쓸모를 고민하는 일은 응용수학적 연구의 여러 가지 목적 중 하나이다. 본 논문에서는 소수라는 개념을 활용하여 ‘효율적인 스포츠 트레이닝 스케줄 설정’이라는 목표를 설정하고 연구를 진행한다. 소수를 활용할 때와 그렇지 않을 때의 스케줄링의 효율성을 정량적으로 비교하기 위하여 여러 가지 운동에 대한 분산도를 지표로 설정한다. 두 경우에 유의미한 차이가 있음을 확인할 수 있고, 이를 통하여 소수라는 개념이 가지는 특성에 대하여 실질적 관점에서의 이해를 도모할 수 있다. 이와 같은 방식의 연구를 통해 수학적 개념의 학습 목표를 공고히 하고 능동적인 학습과 질문 설정과 분석 등의 경험을 통하여 응용수학적 연구를 경험할 수 있다.

## I. 서론

본 연구는 수학적 개념의 실생활 적용 가능성을 탐구함으로써, 특히 소수를 활용한 다양한 응용 분야에 초점을 맞추고 있다. 이를 위해 저자들의 관심사를 바탕으로, 소수의 개념을 이용하여 스포츠 트레이닝 스케줄을 효율적으로 구성하는 방법에 대한 심층적 연구를 수행하였다. 본 논문의 핵심은 소수를 활용하여 스포츠 트레이닝 스케줄을 최적화하는 방안을 제시하고, 이를 파이썬 프로그래밍을 통해 구현하고 분석하는 것이다. 이 과정을 통해 소수라는 수학적 개념이 현실 세계의 다양한 문제 해결에 어떻게 응용될 수 있는지를 보여주며, 이를 통해 수학적 이론의 실용성과 중요성을 강조한다.

## II. 이론적 배경

### (1) 분산 표준편차

분산은 관측값이 평균으로부터 얼마나 흩어져 있는지를 나타내는 통계적 척도로, 수학적 분석에서 중요한 역할을 한다. 이는 각 관측값에서 평균을 뺀 후 제곱하여 평균을 낸 값으로 정의되며, 이의 제곱근이 표준편차이다. 표준편차가 작을수록 데이터는 평균에 가깝게 집중되어 있음을 의미한다. 본 연구에서는 특히 스포츠 트레이닝 스케줄의 최적화를 위해 이 개념을 활용하였다.

## (2) 소수와 서로소의 정의 및 간단한 성질

소수란 1보다 큰 자연수 중에서 오직 1과 자기 자신만을 약수로 가지는 수들을 말한다. 예를 들어, 2, 3, 5, 7 등이 있다. 소수의 중요한 성질 중 하나는 모든 소수가 서로 소라는 점이며, 소수의 개수는 무한하다는 것이다. 반면, 서로소는 최대공약수가 1인 두 개 이상의 양의 정수를 의미한다. 예로는 5와 7, 4와 9 및 25 등이 있다. 서로소인 수들의 최소공배수는 그 수들의 곱과 같다.

## (3) 소수를 쉽게 구할 수 있는 방법

본 연구의 핵심 내용 중 하나가 소수인 만큼, 소수를 쉽게 찾는 방법을 소개한다. 이는 에라토스테네스의 체 방법을 활용한다.

- 범위 설정: 소수를 찾고자 하는 범위 내의 모든 숫자를 나열한다.
- 첫 번째 소수 선택: 가장 작은 소수(일반적으로 2)를 선택한다.
- 선택된 소수의 배수 제거: 선택된 소수의 배수를 모두 제거한다. 이 때, 소수 자신은 제외한다.
- 다음 소수 찾기: 목록에서 제거되지 않은 가장 작은 숫자를 새로운 소수로 선택한다.
- 새로운 소수의 배수 제거: 새로운 소수의 모든 배수를 목록에서 제거한다.
- 반복: 위의 과정을 범위 내에 더 이상 새로운 소수를 찾을 수 없을 때까지 반복한다.
- 소수 목록 완성: 반복 과정을 마치고 남은 숫자들이 해당 범위 내의 모든 소수이다.

이 알고리즘은 범위 내의 소수를 체계적으로 찾는 데 유용하며, 수학적 개념을 실제 문제 해결에 적용하는 좋은 예시가 된다.

## (4) 소수를 실생활에서 사용하는 예

소수가 실생활과 무관할 것이라는 통념과 달리, 소수는 실제로 다양한 분야에서 활용된다. 특히 RSA 암호화 알고리즘에서 소수의 중요성이 드러난다. 이 알고리즘은 두 소수  $p$ 와  $q$ 를 기반으로 하며,  $p-1$ 과  $q-1$ 과 서로소인 정수  $e$ 를 이용한다.  $N=pq$ 를 계산한 후,  $N$ 과  $e$ 를 공개한다. RSA 암호의 해독 과정은  $N$ 과  $e$ 만으로  $p$ 와  $q$ 를 추론해야 하며,  $p$ 와  $q$ 가 서로소이기 때문에 이 과정이 어려워진다. 이처럼 소수는 암호학과 같은 실생활 분야에서 중요한 역할을 한다.

### III. 연구방법

본 연구에서 트레이닝 스케줄 설정의 주요 목표는 두 가지로 정의된다. 첫째, 각 운동을 충분히 실행하여 운동 효과를 극대화하는 것이며, 둘째, 운동을 고르게 분배하여 근육의 회복 시간을 적절히 고려하는 것이다. 이 두 정성적 가치를 반영하기 위해, '각 운동의 평균 운동 횟수'와 '운동 횟수의 표준편차'라는 정량적 지표를 설정한다.

분석 과정에서는 파이썬 프로그래밍 언어와 다양한 라이브러리의 활용이 중요한 역할을 한다. 필수 라이브러리로는 matplotlib, pyplot, numpy, math, pandas가 사용된다. 프로그램 작성과정은 여러 번의 디버깅과 수정을 필요로 했으며, 이 과정에서 오픈소스 교육 자료를 통해 자발적 학습 자료를 다수 활용하였다.

본 연구는 운동 효과와 균형 잡힌 스케줄링을 정량적으로 평가하는 데 중점을 둔다. 이를 위해 파이썬을 사용하여 데이터를 분석하고 그래프를 그리는 등의 방법으로 시각화와 정량화 등의 과정이 사용되었다. 이러한 과정을 통해, 운동 스케줄을 최적화하는데 수학과 프로그래밍의 융합이 어떻게 활용될 수 있는지를 알 수 있다.

### IV. 연구결과

#### (1) 주제 선택 이유:

본 연구는 소수, 스포츠, 트레이닝이라는 공통된 관심사에서 출발하였다. 브레인 스토밍을 통해 '소수를 활용한 효율적인 스포츠 트레이닝 스케줄'이라는 주제가 선정되었는데, 이는 수학적 개념을 실생활에 적용할 수 있는 흥미로운 주제라고 생각되었기 때문이다.

#### (2) 문제상황:

바쁜 일상 속에서 건강을 유지하기 위해 운동을 선택하는 현대인들에게 적절한 운동 스케줄링의 중요성은 명백하다. 너무 강도 높게 계획된 스케줄은 일상에 방해가 될 수 있고, 반대로 너무 여유로운 스케줄은 효과를 보지 못한다. 이에 본 연구에서는 효율적인 스케줄링 방법을 연구하여 이를 해결하고자 하였다.

#### (3) 소수와 서로소의 활용:

파이썬 프로그램을 이용해 31일 동안의 스포츠 트레이닝 스케줄을 분석하였다. 모든 운동을 배우기 때문에 모든 운동을 실행하도록 가정한 첫 번째 날을 제외한 모든 날들에 대해 평균 운동 횟수와 표준편차를 산출 하였으며, 이를 통해 소수 기반 스케줄이 합성수 기반 스케줄보다 효율적임을 밝혀내었다. 표 1에 제시한 바와 같이 평균값은 각각 0.74, 0.76로 큰 차이가 없었지만, 표준편차는 소수를 이용한 스케줄에서 더 낮게 나타나, 운동의 분배가 더 고르게 이루어졌음을 알 수 있다.

	표준편차	평균
소수로 트레이닝 스케줄표를 계획한 경우	0.7	0.74
합성수로 트레이닝 스케줄표를 계획한 경우	1.06	0.76

표 2 소수와 합성수를 활용한 스케줄링에 관한 정량적 비교

또한 파이썬 프로그램의 코딩 결과와 실행 결과를 비교한 자료를 통하여 소수 기반 스케줄이 합성수 기반 스케줄에 비해 운동의 분배가 균일하게 이루어졌음을 그림 2, 3을 통해 시각적으로 확인할 수 있다. 특히, 합성수 기반 스케줄에서는 특정 날에 3개의 운동을 해야 하는 경우가 '4일'이었지만, 소수 기반 스케줄에서는 그러한 날이 없었다는 점이 주목할 만하다. 이러한 분석을 통하여 수학적으로 효율적인 스케줄 계획을 도출할 수 있었다.

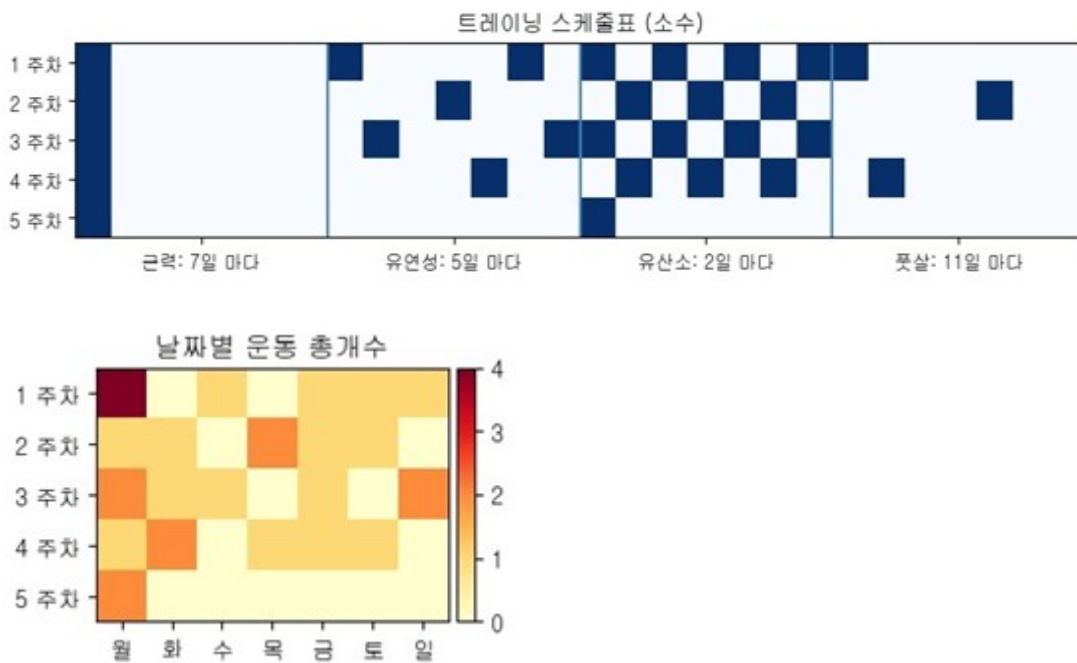


그림 1 소수를 활용한 스케줄링 시각화 결과

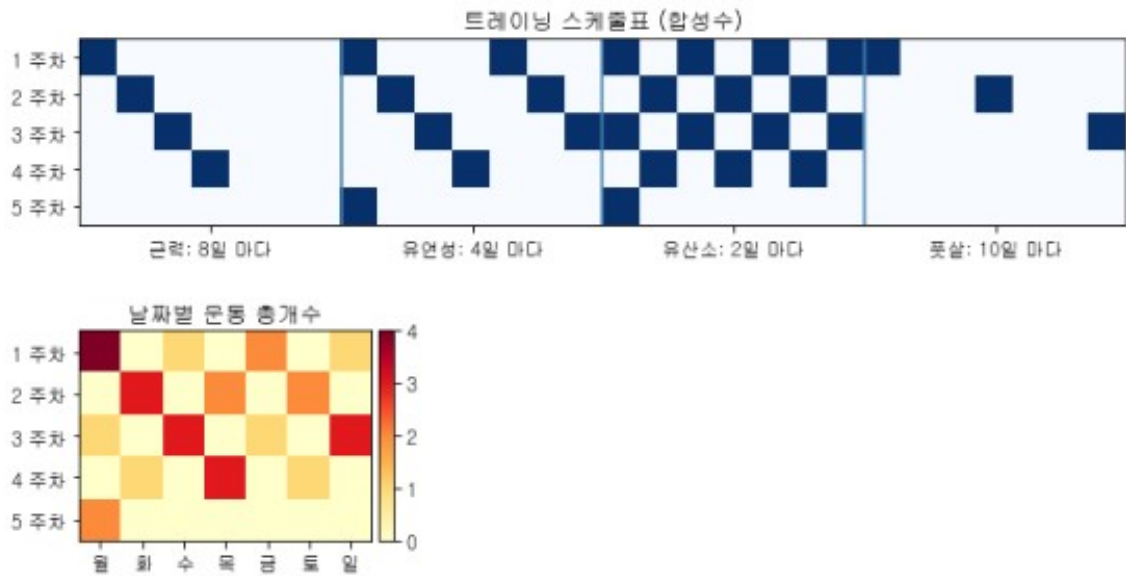


그림 2 합성수를 활용한 스케줄링 시각화 결과

## V. 결론 및 제언

본 논문에서는 '트레이닝 스케줄 설정을 효율적으로 하려면 어떻게 해야 하는가?'에 대하여, 이를 수학적 방법으로 해결할 수 있는 한 가지 방법을 제시하고자 하였다. 중학교 교육과정을 크게 벗어나지 않는 선에서 '소수'라는 개념을 이용하여 접근하였다. 운동 효과를 나타내는 '각 운동의 평균 운동 횟수'를 유사하게 설정한 상태에서 운동의 효율성을 나타내는 '각 운동의 운동횟수의 표준편차'가 적은 트레이닝 스케줄이 보다 이상적이라고 판단할 수 있다. 트레이닝 스케줄을 소수와 합성수 간격으로 설정했을 때의 결과를 정량적, 시각적으로 비교하였다. 구체적으로 소수를 이용하여 스케줄을 짰을 때의 결과와 합성수를 이용하여 스케줄을 짰을 때의 평균 값은 0.74와 0.76으로 차이가 거의 없었지만 표준편차의 경우는 0.7과 1.06으로 상당한 차이를 보임을 확인할 수 있었다. 소수를 활용한 경우에 운동 효과는 비슷하지만 보다 고르게 각 운동이 배치가 되었음을 확인할 수 있었다. 이를 통하여 소수로 스케줄을 계획하였을 때, 운동을 겹치게 하는 날의 수가 많이 줄어들게 되어 신체에 과도한 부하를 주지 않으며 운동을 진행할 수 있음을 확인하였다.

이 논문은 효율적인 트레이닝 스케줄을 수학적으로 설계하는 방법에 대해 탐구하였다. 소수라는 중학교 수학 개념을 활용하여 스케줄을 계획함으로써, 운동 횟수를 균일하게 분배하는 이상적인 방식을 찾았다. 운동 효과는 유사하게 유지하면서도, 운동 분배의 균형을 더 잘 맞출 수 있었다. 소수를 기반으로 한 스케줄은 운동의 중복을 줄여, 무리 없이 지속 가능한 운동 계획을 가능하게 하였다.

이 연구는 소수의 활용을 통해 수학적 이론을 실제 문제에 적용할 수 있는 가능

성을 보여주었다. 그러나 제안된 방식이 최선인지에 대해서는 더 논의하고 연구할 필요가 있다. 소수와 합성수의 다양한 조합을 통해 더 효율적인 스케줄을 탐색할 수 있는 여지가 있다. 또한, 모든 운동을 시작하는 첫째 날이라는 가정에 대한 대안적인 접근법에 대해서도 추가연구가 이루어질 수 있다.

이 연구는 수학의 실용적 적용을 이해하고, 문제 해결 과정을 경험하는 기회가 되었기에 또한 의의를 가진다. 수학적 개념을 적용하고, 가설을 설정하며, 결과를 분석하고 시각화하는 과정을 통해 학습할 수 있었다.

### (파이썬 코드)

본 연구를 진행하여 작성한 코드를 아래에 제시한다.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib import font_manager, rc
4 font_path = "C:/Windows/Fonts/NGULIM.TTF"
5 font = font_manager.FontProperties(fname=font_path).get_name()
6 rc('font', family=font)
7 from mpl_toolkits.axes_grid1 import make_axes_locatable
8
9 def generate_calander(exercise_periods, kind):
10     exercise_periods_for_print = []
11     for key in exercise_periods:
12         exercise_periods_for_print.append(key+' : '+str(exercise_periods[key])+'일 마다')
13     total_days = 30
14
15     # 달력 형식의 2D 배열 생성
16     weeks = (total_days + 6) // 7
17     calendar = np.zeros((weeks, 7 + len(exercise_periods.keys())))
18
19     # 스케줄 설정
20     for idx, (exercise, period) in enumerate(exercise_periods.items()):
21         offset = 0 # 각 운동의 시작 요일을 모두 같게 결정 (첫번째 날은 운동을 배워야하니까)
22
23         for day in range(offset, total_days):
24             # if day % 7 in [5, 6]: # 주말 제외
25             # continue
26             week = day // 7
27             weekday = day % 7
28             if (day - offset) % period == 0:
29                 calendar[week, idx + 7 + weekday] = 1
30
31     # 시각화
32     fig, ax = plt.subplots(figsize=(10,60))
33     cax = ax.imshow(calendar, cmap="Blues")
34

```

```

35 # 라벨 및 타이틀 설정
36 plt.title("트레이닝 스케줄표 (" + kind + ")")
37 plt.yticks(range(weeks), [f"{i+1} 주차" for i in range(weeks)])
38 plt.xticks(np.arange(0, 7 * len(exercise_periods), 7)+3, exercise_periods_for_print)
39 plt.axvline(x=6.5,color='steelblue')
40 plt.axvline(x=13.5,color='steelblue')
41 plt.axvline(x=20.5,color='steelblue')
42
43 plt.show()
44
45 # 날짜별 운동개수 그리기
46 total_calendar = np.sum(calendar.reshape(5,4,7),axis=1)
47
48 # 시각화
49
50 fig, ax = plt.subplots(figsize=(3,10))
51 ax.set_title("날짜별 운동 총개수")
52 cax = ax.imshow(total_calendar, cmap='YlOrRd')
53 divider = make_axes_locatable(ax)
54 setting = divider.append_axes("right", size="5%", pad=0.05)
55
56 plt.colorbar(cax, cax=setting)#. label='날짜별 운동 총개수')
57 # 라벨 및 타이틀 설정
58
59 ax.yaxis.set_ticks(range(weeks))
60 ax.yaxis.set_ticklabels([f"{i+1} 주차" for i in range(weeks)])
61 ax.xaxis.set_ticks(range(7))
62 ax.xaxis.set_ticklabels(['월', '화', '수', '목', '금', '토', '일'])
63 plt.show()
64 return calendar

```

```

1 prime_exercise_periods = {'근력': 7, '유연성': 5, '유산소': 2, '풋살': 11}
2 non_prime_exercise_periods = {'근력': 8, '유연성': 4, '유산소': 2, '풋살': 10}
3 prime_calendar = generate_calander(prime_exercise_periods, '소수')
4 non_prime_calendar = generate_calander(non_prime_exercise_periods, '합성수')

```

```

1 # 소수 관력 스케줄링 결과 각 날짜 (세로: 주차, 가로: 요일-월화수목금토일)
2 np.sum(prime_calendar.reshape(5,4,7),axis=1)

```

```

array([[4., 0., 1., 0., 1., 1., 1.],
       [1., 1., 0., 2., 1., 1., 0.],
       [2., 1., 1., 0., 1., 0., 2.],
       [1., 2., 0., 1., 1., 1., 0.],
       [2., 0., 0., 0., 0., 0., 0.]])

```

```

1 # 합성수 관력 스케줄링 결과 각 날짜 (세로: 주차, 가로: 요일-월화수목금토일)
2 np.sum(non_prime_calendar.reshape(5,4,7),axis=1)

```

```

array([[4., 0., 1., 0., 2., 0., 1.],
       [0., 3., 0., 2., 0., 2., 0.],
       [1., 0., 3., 0., 1., 0., 3.],
       [0., 1., 0., 3., 0., 1., 0.],
       [2., 0., 0., 0., 0., 0., 0.]])

```

```

1 def daily_exercise_load_index(calendar):
2     daily_load = np.sum(calendar.reshape(5,4,7),axis=1)
3     daily_load = daily_load.flatten()[1:]# 첫째 날은 모든 운동을 하기 때문에 제외
4     return np.mean(daily_load), np.std(daily_load)
5
6 # 소수 회복 기간과 합성수 회복 기간에 대한 달력은 이미 생성되어 있다고 가정
7 prime_mean, prime_std = daily_exercise_load_index(prime_calendar)
8 non_prime_mean, non_prime_std = daily_exercise_load_index(non_prime_calendar)
9
10 print("Daily Exercise Load Index with prime numbers:", prime_mean, prime_std)
11 print("Daily Exercise Load Index with composite numbers:", non_prime_mean, non_prime_std)

```

Daily Exercise Load Index with prime numbers: 0.7352941176470589 0.6991096661179242  
Daily Exercise Load Index with composite numbers: 0.7647058823529411 1.0588235294117647

## VI. 참고문헌

1. 파이썬 코딩 수학 with 딥러닝, 박경원, 정보문화사, 2019
2. 소수의 음악 - 수학 최고의 신비를 찾아, 고중숙(역), 승산, 2007
3. Do it! 딥러닝 입문, 박해선, 이지스퍼블리싱, 2019

수학A

## 10\_일반 파스칼 행렬의 분해



# 일반 파스칼 행렬의 분해

최지수 (오마중학교 2학년)

지도교수 : 김도진 (동국대학교 수학과)

**초록:** 파스칼 행렬은 이항계수를 행렬의 원소로 구성된 무한 행렬을 뜻한다. 이러한 행렬은 격자 경로(lattice path)의 수로 표현될 수 있다. 기존의 격자 경로에 대각선 경로를 추가하여 얻어지는 일반 파스칼 행렬의 분해를 수학적 귀납법을 이용하여 증명하였다.

## I. 서론

이항계수  $\binom{i}{j} = \frac{i!}{(i-j)!j!}$ ,  $(i, j = 0, 1, 2, \dots)$ 을  $\binom{i}{j} = \binom{i-1}{j} + \binom{i-1}{j-1}$ ,  $(i, j = 1, 2, \dots) \dots (\star)$

의 점화식을 바탕으로 하여 삼각형 모양으로 나타낸 것을 파스칼 삼각형이라고 한다.

$$\begin{array}{cccccc}
 & & & & & \binom{0}{0} \\
 & & & & \binom{1}{0} & \binom{1}{1} \\
 & & \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & \\
 & \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & \\
 \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} & \\
 \binom{5}{0} & \binom{5}{1} & \binom{5}{2} & \binom{5}{3} & \binom{5}{4} & \binom{5}{5} \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{array}
 \qquad
 \begin{array}{cccccc}
 & & & & & 1 \\
 & & & & 1 & 1 \\
 & & 1 & 2 & 1 & \\
 & 1 & 3 & 3 & 1 & \\
 1 & 4 & 6 & 4 & 1 & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{array}$$

[파스칼 삼각형]

이러한 파스칼 삼각형을 이용하여 행렬( $P$ )의  $(i, j)$ 번째 성분( $P_{i,j}$ )을 이항계수  $\binom{i}{j}$ 로 나타낸 행렬을 파스칼 삼각행렬( $P$ )이라고 한다.

$$P = \begin{bmatrix} \binom{0}{0} & \binom{0}{1} & \binom{0}{2} & \binom{0}{3} & \dots \\ \binom{1}{0} & \binom{1}{1} & \binom{1}{2} & \binom{1}{3} & \dots \\ \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & \binom{2}{3} & \dots \\ \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 1 & 1 & 0 & 0 & 0 & \dots \\ 1 & 2 & 1 & 0 & 0 & \dots \\ 1 & 3 & 3 & 1 & 0 & \dots \\ 1 & 4 & 6 & 4 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

무한 차원인 파스칼 삼각행렬에서 0번째부터  $n$ 번째까지의 행과 열을 선택하여 얻은  $(n+1) \times (n+1)$  크기의 행렬을  $P_n$ 이라고 한다.

$$P_n = \begin{bmatrix} \binom{0}{0} & \binom{0}{1} & \binom{0}{2} & \binom{0}{3} & \cdots & \binom{0}{n} \\ \binom{1}{0} & \binom{1}{1} & \binom{1}{2} & \binom{1}{3} & \cdots & \binom{1}{n} \\ \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & \binom{2}{3} & \cdots & \binom{2}{n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \binom{n}{0} & \binom{n}{1} & \binom{n}{2} & \binom{n}{3} & \cdots & \binom{n}{n} \end{bmatrix}_{(n+1) \times (n+1)}, \quad P_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 \\ 1 & 5 & 10 & 10 & 5 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{bmatrix}_{7 \times 7}$$

특히 행렬  $S$ 의  $(i, j)$ 번째 성분  $(S)_{i,j}$ 을 이항계수  $\binom{i+j}{j}$  또는  $\binom{i+j}{i}$ 로 나타낸 행렬을 대칭 파스칼 행렬이라고 하며  $(n+1) \times (n+1)$  크기의 대칭 파스칼 행렬을  $S_n$ 이라고 표현한다.

$$S = \begin{bmatrix} \binom{0}{0} & \binom{1}{1} & \binom{2}{2} & \binom{3}{3} & \cdots \\ \binom{1}{0} & \binom{2}{1} & \binom{3}{2} & \binom{4}{3} & \cdots \\ \binom{2}{0} & \binom{3}{1} & \binom{4}{2} & \binom{5}{3} & \cdots \\ \binom{3}{0} & \binom{4}{1} & \binom{5}{2} & \binom{6}{3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots \\ 1 & 2 & 3 & 4 & 5 & \cdots \\ 1 & 3 & 6 & 10 & 15 & \cdots \\ 1 & 4 & 10 & 20 & 35 & \cdots \\ 1 & 5 & 15 & 35 & 70 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad S_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 6 & 10 & 15 & 21 \\ 1 & 4 & 10 & 20 & 35 & 56 \\ 1 & 5 & 15 & 35 & 70 & 126 \\ 1 & 6 & 21 & 56 & 126 & 252 \end{bmatrix}_{6 \times 6}$$

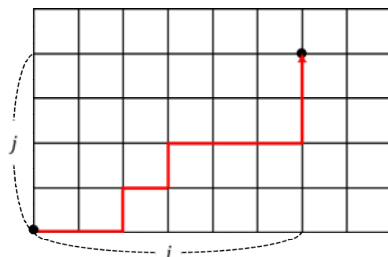
이러한 대칭 파스칼 행렬( $S$ )은 파스칼 삼각행렬( $P$ )로 다음과 같이 분해된다.

$$S = PP^T, \quad S_n = P_n P_n^T, \quad n = 0, 1, 2, 3, \dots$$

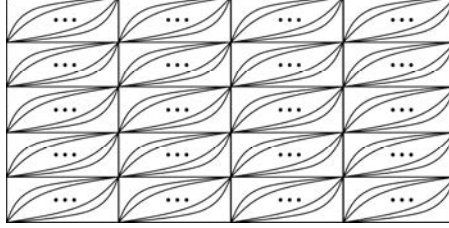
파스칼 삼각형은 다양한 방법으로 설명할 수 있으며 특히 격자 이동(Lattice Path)을 이용하여 나타낼 수 있다. 우리는 일반 격자 이동에서 대각선 격자 이동을 추가하여 일반 파스칼 행렬을 만들고 이를 파스칼 행렬을 이용하여 분해하고자 하며, 수학적 귀납법을 이용하여 이를 증명하고자 한다.

## II. 이론적 배경

아래와 같이 맨 아래 왼쪽의 꼭짓점  $(0,0)$ 에서 꼭짓점  $(i,j)$ 으로 오른쪽( $\rightarrow$ ), 위쪽( $\uparrow$ ) 방향으로 격자점을 따라 이동하여 얻을 수 있는 최소의 경우의 수는  $\binom{i+j}{j}$ 로 나타난다.



특히 우리는 대각선 방향으로  $\alpha$ 개의 경로를 아래와 같이 추가하여 격자점 이동을 생각해 볼 수 있다. 이렇게 추가된  $\alpha$ 개의 대각선을 바탕으로 하여 일반화된 무한 행렬  $H^{(\alpha)}$ 를 만들고 이 행렬의 분해를 생각해 본다.



먼저  $(n+1) \times (n+1)$  크기의 일반 파스칼 행렬  $H_n^{(\alpha)}$ 의  $(i, j)$ 번째 성분  $(H_n^{(\alpha)})_{i,j}$ 은 다음과 같이 나타낼 수 있다.

$$(H_n^{(\alpha)})_{i,j} = \sum_{r=0}^{\min(i,j)} \alpha^r \binom{i+j-r}{i-r, j-r, r}, \quad \binom{i+j-r}{i-r, j-r, r} = \frac{(i+j-r)!}{(i-r)!(j-r)!r!}$$

그리고 행렬  $H_n^{(\alpha)}$ 은 다음과 같은 점화식을 만족한다.

1.  $(H_n^{(\alpha)})_{i,j} = (H_n^{(\alpha)})_{i-1,j} + (H_n^{(\alpha)})_{i,j-1} + \alpha(H_n^{(\alpha)})_{i-1,j-1}$
2.  $1 + (1+\alpha) \sum_{k=0}^{i-1} \sum_{l=0}^{j-1} (H_n^{(\alpha)})_{kl} = (H_n^{(\alpha)})_{ij}$

여기서 우리는 수학적 귀납법이라고 하는 증명 기법을 이용하고자 한다. 증명하고자 하는 식이 수열과 같이 특정 미지수  $n$ 이 자연수(또는 0부터 시작하는 정수)에 따라 그 값이 결정되면, 이론상 우리는 모든 값을 대입해 식이 성립하는지 확인해야만 한다. 하지만 실질적으로 그것은 불가능하므로, 다음과 같은 방법을 생각한다.

1.  $n=0$ (또는 가장 작은  $n$ )에서 식이 성립한다.
2.  $n=k$  ( $k \geq 0$ )에서 성립한다고 가정하고,  $n=k+1$ 일 때 성립한다.

위의 두 명제가 참이 되면 우리는 주어진 식이 모든  $n$ 에 대해 성립한다고 할 수 있다. 이러한 방식의 증명을 수학적 귀납법이라고 한다.

### III. 연구 결과 및 증명

**정리.** 일반 파스칼 행렬  $H_n^{(\alpha)}$ 는  $H_n^{(\alpha)} = P_n D_n^{(\alpha)} P_n^T$ 로 분해된다.

여기에서  $D_n^{(\alpha)}$ 은 다음과 같은 대각행렬을 의미한다.

$$D_n^{(\alpha)} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \alpha+1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & (\alpha+1)^2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & (\alpha+1)^n \end{bmatrix}_{(n+1) \times (n+1)}$$

(증명)

먼저  $n=0$ 일때  $H_0^{(\alpha)} = [1] = [1][1][1] = P_0 D_0^{(\alpha)} P_0^T$  이므로 자명하다.

다음으로  $n=k$ 일때  $H_k^{(\alpha)} = P_k D_k^{(\alpha)} P_k^T$ 임을 가정하자.

이제  $D_{k+1}^{(\alpha)} := \begin{bmatrix} 1 & 0^T \\ 0 & (\alpha+1)D_k^{(\alpha)} \end{bmatrix}_{(k+2) \times (k+2)}$ 로 정의한다. 이는 1행 1열의 값은 1, 그 오른 쪽과 아래쪽은 모두 0, 나머지  $(k+1) \times (k+1)$ 의 부분은  $(\alpha+1)D_k^{(\alpha)}$ 로 이루어진 행렬임을 의미한다. 이때의 0은 모든 항이 0인  $1 \times k$  행렬이다.

I 절과 II 절에서 언급된 점화식들을 활용하기 위해 아래와 같은 행렬  $E_k$ 를 도입한다.

$$E_k = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix}_{(k+1) \times (k+1)}$$

이 행렬  $E_k$ 는 역행렬이 존재하며, 그 역행렬은 다음과 같다.

$$E_k^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{(k+1) \times (k+1)}$$

즉 각 항이 1로 이루어진 하삼각행렬이 된다.

I 절의 점화식 (★)에 의해

$$E_{k+1} P_{k+1} = \begin{bmatrix} 1 & 0^T \\ 0 & P_k \end{bmatrix}$$

가 성립한다. 또한

$$P_{k+1}^T E_{k+1}^T = (E_{k+1} P_{k+1})^T = \begin{bmatrix} 1 & 0^T \\ 0 & P_k^T \end{bmatrix}$$

가 성립한다. 따라서

$$E_{k+1} P_{k+1} D_{k+1} P_{k+1} E_{k+1} = \begin{bmatrix} 1 & 0^T \\ 0 & (\alpha+1)P_k D_k^{(\alpha)} P_k^T \end{bmatrix}$$

수학적 귀납법의 목표에 따라  $H_{k+1}^{(\alpha)} = P_{k+1} D_{k+1}^{(\alpha)} P_{k+1}^T$ 가 성립함을 보여야 하므로 양변의 좌측에  $E_{k+1}^{-1}$ 를, 우측에  $E_{k+1}^{-T}$ 를 곱한다. 즉

$$P_{k+1} D_{k+1} P_{k+1} = E_{k+1}^{-1} \begin{bmatrix} 1 & 0^T \\ 0 & (\alpha+1)P_k D_k^{(\alpha)} P_k^T \end{bmatrix} E_{k+1}^{-T} = E_{k+1}^{-1} \begin{bmatrix} 1 & 0^T \\ 0 & (\alpha+1)H_k^{(\alpha)} \end{bmatrix} E_{k+1}^{-T}$$

우변에 해당하는 행렬의 각 항을 살펴보면

$$\left( E_{k+1}^{-1} \begin{bmatrix} 1 & 0^T \\ 0 & (\alpha+1)H_k^{(\alpha)} \end{bmatrix} E_{k+1}^{-T} \right)_{ij} = 1 + (\alpha+1) \sum_{m=0}^{i-1} \sum_{l=0}^{j-1} (H_k^{(\alpha)})_{ij} = (H_{k+1}^{(\alpha)})_{ij}$$

가 II 절의 점화식 2번에 의해 성립한다.

즉  $H_{k+1}^{(\alpha)} = P_{k+1} D_{k+1}^{(\alpha)} P_{k+1}^T$ 이 성립하므로 수학적 귀납법에 의하여 모든  $n=0,1,2,3,\dots$ 에

대하여  $H_n^{(\alpha)} = P_n D_n^{(\alpha)} P_n^T$  임이 증명되었다.  $\square$

#### IV. 결론 및 제언

일반 파스칼 행렬( $H_n^{(\alpha)}$ )을 수학적 귀납법을 이용하여 파스칼 행렬( $P_n$ )과 대각행렬( $D_n^{(\alpha)}$ )을 이용하여 분해를 증명하였다. 즉 대각선이  $\alpha$ 개 추가된 격자의 경로상에서의 최소 경로의 경우의 수를 모아보면 비교적 단순한 이항계수로 이루어진 행렬과 특정 수의 거듭제곱으로 이루어진 대각행렬로 표현할 수 있음을 시사한다. 특히,  $\alpha=0$ 일 경우 우리가 알고 있는 파스칼 행렬의 분해( $S_n = P_n P_n^T$ )가 됨을 알 수 있다.

#### V. 참고문헌

1. 수학자가 들려주는 수학 이야기-24파스칼이 들려주는 수학적 귀납법 이야기, 자음과 모음, 김정하
2. Introductory Combinatorics (5<sup>th</sup>Edition), Richard A. Braulti.
3. Linear algebra of Pascal matrices by Lindsay Yates, 2014  
<https://www.gcsu.edu/sites/files/page-assets/node-808/attachments/yates.pdf>



수학A

11\_정수 계수 부정방정식의 음이 아닌 정수해의  
개수를 위한 조합적 계산에 대한 연구



# 정수 계수 부정방정식의 음이 아닌 정수해의 개수를 위한 조합적 계산에 대한 연구

박시우 (서울월촌초등학교 6학년)

지도교수 : 박만구 (서울교육대학교 수학교육학과)

본 연구에서는 조합론 중에 주어진 조건을 만족하는 대상의 수를 세는 것을 목표로 알려져 있는 계수적 조합론을 사용하여 정수 계수 부정방정식을 케이스를 줄여서 가짓수를 셀 수 있도록 함으로써 음이 아닌 정수해의 개수를 위한 하나의 알고리즘으로 제시해 보고자 한다. 연구 문제를 중심으로 적합한 문제 모형을 만들어 합동식과 비교 분석을 통해 조합적 계산의 효과를 확인 후 각 문제 모형에서 보여주는 공통된 풀이 과정을 분석하여 새로운 알고리즘을 고안하는 것을 목적으로 했다. 계수적 조합론 방법으로 정수 계수 부정방정식의 음이 아닌 정수해의 개수를 위한 조합적 알고리즘을 적용한 결과, 부정방정식에서 주로 하고 있는 방법인 합동식으로 수행할 때보다 나뉘야 할 케이스를 줄이는 효과를 보였고 효율적인 시간 내에 계산 실수 가능성이 없는 단순하면서도 실용적인 방법이 되었다.

## I. 서론

### 1. 필요성 및 목적

부정방정식은 근이 너무 많아서 일정한 해가 정해지지 않은 대수방정식이다. 합동식을 이용하면 숫자를 줄여서 간단하게 풀 수는 있지만 간단하게 만든 후에 나뉘야 할 케이스가 많고 계산 방법이 복잡해지기 때문에 더러 맞는 계산인지 판별조차 힘들다.

저자가 경험한 예로는 부정방정식의 자연수 혹은 음이 아닌 정수의 순서쌍의 개수를 합동식으로 구할 때, 처음에 범을 몇으로 나누어야 하는지 떠올리기가 어려웠다. 범  $m$ 으로 하고 나눈 나머지를  $n$ 으로 하면 좌변에 문자로 이루어져 있는 식과  $km+n$ 이 같다. 좌변에 한 문자를 두고 우변에 나머지 문자와 숫자를 둔 후 주어진 부정방정식에 대입한다는 것을 생각해 내는 것이 어려웠다. 게다가 좌변에 한 문자의 범위를 기준으로 케이스를 정하는데 이 또한 생각하기 어렵고 케이스를 나눈다고 해도 그 케이스가 너무 많기 때문에 계산 실수 등을 할 확률이 높았다.

그리고 김희재(2005)는 “부정방정식은 아주 적은 몇 가지 형태의 문제만이 고정적인 풀이 형식을 가지고 있으며 대부분의 문제들은 서로 다른 해결방법을 융통성 있게 적용해야하는 경우가 많다.”라고 설명하였다(p.58). 부정방정식은 대수식을 그

목적과 상황에 맞게 변형하는 것이 중요하므로 최대한 조건을 만족하는 해의 개수를 세는 것을 만족하도록 케이스를 줄여서 가짓수를 셀 수 있다면 순서쌍의 개수를 구할 때 긍정적인 영향을 주게 될 것이다.

본 연구에서는 조합론 중에 주어진 조건을 만족하는 대상의 수를 세는 것을 목표로 알려져 있는 계수적 조합론(enumerative combinatorics)을 사용하여 정수 계수 부정방정식을 정하고 케이스를 줄여서 가짓수를 셀 수 있도록 함으로써 음이 아닌 정수해의 개수를 위한 하나의 알고리즘으로 제시해 보고자 한다.

## 2. 가설

“정수 계수 부정방정식에서 자연수 혹은 음이 아닌 정수의 순서쌍의 개수를 구할 때 계수적 조합론을 이용하면 합동식보다 나뉘야 할 케이스를 줄이는 효과가 있다.”라는 가설을 세우고 가설을 확인하기 위한 연구문제를 중심으로 적합한 문제모형을 만들어 검증한다.

## 3. 연구문제

- 1) 일차부정방정식의 자연수 혹은 음이 아닌 정수의 순서쌍 개수를 구하는 방법에 조합적 계산의 연관성은 어떠한가?
- 2) 일차부정방정식 계수의 조건에 따라 조합적 계산방법에 미치는 영향이 있을 것인가?
- 3) 조합적 계산 또는 합동식 이용에 따라 일차부정방정식의 자연수 혹은 음이 아닌 정수의 순서쌍 개수를 구하는 과정에서 어떠한 차이가 있을 것인가?
- 4) 동일 차수 부정방정식의 음이 아닌 정수의 순서쌍 개수를 구하는 방법에 조합적 계산의 연관성은 어떠한가?
- 5) 각 차수가 다른 부정방정식은 조합적 계산 방법에 미치는 영향이 있을 것인가?

# II. 이론적 배경

## 1. 부정방정식[ indeterminate equation, 不定方程式 ]의 기본 개념

부정방정식은 근이 너무 많아서 일정한 해가 정해지지 않은 대수방정식을 말한다. 일반적으로 미지수의 개수보다 식의 수가 적으면 근이 무수히 많은 부정이 된다. 미지수의 개수보다 식의 수가 많으면 근이 생기지 않을 수 있다. 미지수의 개수와 식의 수가 같으면 방정식을 풀 수 있다고 알려져 있다. 디오판토스의 방정식이라고도 한다. 방정식  $x^2+y^2=z^2$ 을 만족하는 정수, 즉 피타고라스의 수를 구하는 문제나 페르마의 문제 또는 예로부터 생각되었던 변과 넓이가 모두 유리수가 되는 삼각형

이나, 브라마굽타가 만든 변과 대각선의 길이가 모두 유리수인 사각형 등은 모두 부정방정식의 문제이다.

### 방법

부정방정식론의 특징은 통일적 이론의 구성이 곤란했다는데 있었는데, 오히려 곤란했기 때문에 깊이 있는 여러 해법이 연구되었다고 할 수 있다. 이들 해법의 대표적인 것을 분류하면 다음과 같다.

- ① 합동식을 이용하는 방법, ② 연분수를 이용하는 방법,
- ③ 대수적 정수론의 응용, ④ 국소체론의 응용, ⑤ 수의 기하학에의 응용,
- ⑥ 디오판토스 근사의 응용, ⑦ 계산기의 응용 등이다.

### 힐베르트의 제10문제

D.힐베르트는 1900년 파리의 국제수학자회의에서 23개의 수학문제를 제출하였는데, 그 중의 제10번째는 임의로 주어진 부정방정식의 정수해가 존재하느냐 존재하지 않느냐 하는 것을 유탄회의 조작으로서 판정할 수 있는 방법을 구하라는 문제였다. M.데이비스, H.푸트넘, J.로빈스 등의 연구결과를 바탕으로 하여 Y.마티야세비치가이 문제에 대하여 부정적 해결을 하였다. 즉, 해가 있느냐 없느냐를 판정하는 방법이 존재하지 않는 그러한 부정방정식이 실제로 있다는 것이었다.

## 2. 합동식[ congruent expression, 合同式 ]의 기본 개념

두 정수  $a, b$ 의 차가 정수  $p$ 의 배수일 때  $a$ 와  $b$ 는  $p$ 를 법으로 하여 합동이라 하고, 이것을  $a \equiv b \pmod{p}$  또는  $a \equiv b \pmod{p}$ 의 식으로 표시하는데, 이 식을 칭한다.

두 정수(整數)  $a, b$ 의 차가 정수  $p$ 의 배수일 때  $a$ 와  $b$ 는  $p$ 를 법으로 하여 합동이라 하고, 이것을  $a \equiv b \pmod{p}$  또는  $a \equiv b \pmod{p}$ 로 표시하고 이 식을 합동식이라 한다.

합동식을 쓰면 정수  $a$ 가 정수  $p$ 로 나누어떨어진다는 것을  $a \equiv 0 \pmod{p}$ 로 쓸 수 있다. 예를 들면  $10 \equiv 1 \pmod{3}$ ,  $50 \equiv 1 \pmod{7}$ ,  $15 \equiv 0 \pmod{5}$ 는 합동식이다.

또 정수를 계수로 하는  $n$ 차인 다항식을  $f(x)$ 라 할 때,  $f(x) \equiv 0 \pmod{p}$ 라는 형의 방정식을  $n$ 차인 합동방정식이라고 하며 이것을 만족시키는 정수를 그 해라고 한다. 예를 들면,  $x^2 - 1 \equiv 0 \pmod{8}$ 의 해는  $1, 3, 5, 7, \dots$ 이다. 이와 같은 방정식에서는  $x^2 - 2 \equiv 0 \pmod{3}$ 과 같이 해를 가지지 않는 것도 있다.

## 3. 조합수학(조합론 combinatorics)의 기본 개념

조합론 또는 조합수학은 유한하거나 가산적인 구조들에 대하여, 어떤 주어진 성질을 만족시키는 것들의 가짓수나 어떤 주어진 성질을 극대화하는 것을 연구하는 수학 분야이다.

## 분류

조합론에서는 다양한 종류의 조합론적 구조들을 다루며, 이들은 다음을 들 수 있다.

- 순열과 조합. 이들을 세는 문제는 12정도라는 이름으로 체계화되어 있다.
- 집합의 분할, 특히 자연수의 분할
- 문자열(영어: word)
- 부분 순서 집합은 순서로 생각할 수 있는 관계를 부여한 집합이며, 특수한 경우로 전순서 집합이나 격자 등이 있다. 이들을 연구하는 분야를 순서론이라고 한다.
- 그래프는 일련의 꼭짓점들과 이들 사이를 잇는 변들로 구성된 조합론적 구조이다. 이들을 다루는 분야를 그래프 이론이라고 한다.
- 매트roid는 그래프를 일반화한 개념이다.
- 유한 기하(영어: finite geometry)는 유한한 수의 점과 선 등으로 구성된 기하학적 공간이다. 이에 대하여 다양한 열거 문제를 고려할 수 있다.
- 계획(영어: design)은 원래 통계학의 실험계획법에서 유래한 개념이다. 라틴 방진이 이의 특수한 경우이다. 계획 이론은 유한 기하학과 코드 이론(영어: coding theory)과 밀접하게 연관되어 있다.

조합론은 다루는 대상 대신 사용하는 기법 또는 목표로서 분류할 수도 있다.

- 계수적 조합론(영어: enumerative combinatorics)은 주어진 조건을 만족하는 대상의 수를 세는 것을 목표로 한다.
- 해석적 조합론(영어: analytic combinatorics)은 해석학적 기법을 조합론에 응용하며, 보통 주어진 대상의 정확한 수보다는 이들의 수의 점근적 공식(영어: asymptotic formula)을 목표로 한다. 계승의 스텔링 공식이 대표적인 예이다.
- 극대 조합론(영어: extremal combinatorics)은 주어진 조건을 만족시키는 대상 가운데 "가장 큰" 또는 "가장 작은" 것 따위의 문제를 다룬다. 극대 그래프 이론은 그래프 이론에서 중요한 연구 분야의 하나이다. 다른 방향으로, 램지 이론도 이에 속한다.
- 위상수학적 조합론(영어: topological combinatorics)은 그래프 따위의 조합론적 구조에 위상을 주어, 보르수크-울람 정리나 호몰로지 대수학을 조합론적 문제에 응용한다. 로바스 라슬로는 보르수크-울람 정리를 사용하여 크네저 그래프(영어: Kneser graph)에 대한 크네저 추측을 증명하였다.

## 4. 선행 연구 분석

### 1) 선행 연구 분석 및 연구의 독창성

① 학술연구정보(RISS) 검색: 한국교육학술정보원에서 운영하는 서비스 리스에서 '부정방정식', '합동식', '조합 수학'으로 검색한 결과, 본 연구와 관련 있는 연구물을 중심으로 분석한 결과는 다음과 같다.

저자명	제목	(1) 선행연구1 내용
오승열 2012	부정방정식 사례 연구	부정방정식이 학교 현장에서 다루어지고 있는 정도를 알아보고, 부정방정식이 들어간 여러 문항 분석을 통해서 부정방정식의 특성을 분석하여, 공식이나 전형적인 해법 이외의 다양한 방법으로 해결할 수 있는 사례를 알아본다.
분석 내용		① 방정식이 제시된 상황에 따라서 식을 변형하거나 인수분해, 법, 홀짝성, 판별식, 근과 계수 관계 등 다양한 방법을 이용하여 문제를 해결. => 부정방정식의 다양한 풀이에 관한 연구. ② 대수식을 그 목적과 상황에 맞게 변형하는 것은 많은 문제를 해결하는 기본적인 방법 => 부정방정식의 대부분의 문제들은 여러 가지 다양한 해결방법을 융통성 있게 적용해야 함.

저자명	제목	(2) 선행연구2 내용
김유리 2005	합동식과 그 응용	합동식의 기본성질과 일차 합동식의 해법, 연립 일차 합동식의 해법을 설명해주는 중국인 나머지 정리를 살펴본다. 합동식 응용에 관한 몇 가지 문제를 연구한다. <u>중고등학교 과정에서 다루어지는 합동개념에 대해 조사해보고, 이러한 개념이 더욱 발전하여 대학과정에서는 어떻게 심화되어 소개되고 있는지 알아본다.</u>
분석 내용		① 나눗셈 판정법과 쉼트 디자인의 예술을 알아보고, 바코드, 주민등록번호, 운전면허번호, 여권번호, ISBN 등 => 합동식의 응용에 관한 연구. ② <u>합동식의 단순한 계산 뿐 아니라 그 개념이 실생활에서 어떻게 응용되는지 연구</u> => 실습을 통해 합동 개념을 습득, <u>수학이 일생활에 직접적으로 응용되는 것을 실감.</u>

저자명	제목	(3) 선행연구3 내용
김효정 2011	중·고등 학교 교과 과정 에서 합동식과 부정방정식에 대한 고찰	우리나라 대학능력평가시험이나 모의수능평가에서는 부정방정식의 내용은 아주 미미하고 혹은 부정방정식을 해결해야만 하는 문제의 경우 아주 기초적인 유형만을 다루고 있다. 정수론과 부정방정식의 내용이 현재 <미지수가 2개인 일차방정식>이라는 한 절로 소개되어 연립방정식의 해를 구하기 전에 자연수범위에서 일차 방정식의 여러 가지 해를 구해보는 정도로 언급되고, 부정방정식이라는 단원이 소개 되지 않은 채 각종 문제지들에 문제를 자연수 혹은 실수 등의 조건에서 부정방정식을 해결하는 유형으로만 소개되고 있다.
분석 내용		① 다양한 수학적 사고 능력을 신장하기 위해 부정방정식이 교과과정에 구성되어야 할 필요성 => 현 교과에서 부정방정식을 학습하는 것에 관한 연구. ② 부정방정식을 합동식을 이용하거나 다른 여러 가지 해법을 통해 문제를 해결 => 부정방정식 내용으로도 여러 가지 창의적인 수학적 활동을 할 수 있음.

저자명	제목	(4) 선행연구4 내용
김기정 2010	합동식과 그 응용에 관한 연구	정수론의 합동식의 기본개념을 정리하고 합동식이 실생활에 활용되는 사례를 수집함으로써, 향후 합동식의 교육에 있어 실생활에 활용되는 사례 소개를 통하여 학습자의 동기를 유발할 수 있는 자료를 제시하고자 한다. 적절한 학습방법을 제시하여 수학을 즐길 수 있도록 할 수 있다면 수학 수업에 변화를 주어 학습에 대한 관심과 흥미를 유도할 수 있을 것이다. 합동식, 기약잉여계, 오일러의 정리, 페르마의 정리, 일차 합동식, 연립일차합동식, 중국인의 나머지 정리 등 합동과 합동식에 관한 기본 개념을 알아본다.
분석 내용		① 통신문의 오류검색, 배수 판정법, 검사숫자, 대진표, RSA암호, RSA전자서명, 별모양 디자인 등. => 합동식이 실생활에서 응용되고 있는 사례. ② 학습단원과 관련된 학생의 관심과 수준에 적합한 학습 자료 제공 => 수학 수업에 변화를 주어 학습에 대한 관심과 흥미를 유도.

저자명	제목	(5) 선행연구5 내용
백창숙 2010	고등학교 '순열과 조합' 단원에서 조합론적 증명 방법을 도입한 수업 지도 방안 연구	교과서에 다루고 있는 순열과 조합 단원에 실려 있는 여러 가지 정리와 문제를 살펴보면 기계적인 풀이와 방법이 주를 이룬다. 순열과 조합 단원은 무엇보다도 직관이 필요한 영역이다. 직관이라는 것은 수학적인 아이디어가 있을 때, 그 기능을 발휘한다. 따라서 학생들에게 단순 한 기계적인 풀이보다는 근본을 이루는 수학적인 아이디어를 심어주는 교수법을 사용하는 것이 좀 더 효율적이고, 학생들의 흥미도 유발시킬 수 있을 것이다. 순열과 조합의 역사와 조합론적 증명의 정의, 조합론적 논증에 의한 문제해결 전략을 유형별로 살펴본다. 대학 과정에서 배운 순열, 조합의 내용이 교과서에 어떻게 소개되어있는지 살펴보고, 다른 점을 비교·분석한 후, 교과서의 '순열과 조합' 단원에서 다루고 있는 증명방법과 조합론적 증명 방법을 비교한다.
분석 내용		① 순열과 조합 단원을 지도하는데 있어 교과서에 제시되고 있는 증명 방법과 다른 조합론적 증명 방법을 도입 => 수업에 활용할 수 있는 본시 학습 지도안과 학생 활동지를 제시함. ② 문제형태를 암기하는 것이 아니라 문제를 논리적으로 판단 => 조합론적 증명은 기존의 대수적 조작에 의존하는 등식의 증명에서 벗어나 창의적이고 수학적으로도 매력적인 등식의 증명.

저자명	제목	(6) 선행연구6 내용
오은주 2007	조합수학 지도에 관한 연구	조합수학은 어떤 조건을 만족하도록 개체들을 배열하는 방법 또는 패턴들의 개수를 세거나 연구하는 분야로서 이산수학의 중요한 부분이다. 즉, 조합은 특별한 이론적 체계를 갖추는 수학분야이기 보다는 수학의 분야에 관계없이 비정형적인 문제를 해결할 때 흔히 나타내는 상이 한 패턴에 대한 개수를 구할 때 많이 쓰이며 전산 및 기타 과학의 여러 분야와 밀접하게 관련되어 있다. 조합수학의 이론적 배경, 조합의 역사, 조합의 정의, 조합의 다양한 성질들을 살펴보고 대수적, 조합론적인 방법으로 각각의 성질들을 증명했다.
분석 내용		① 조합론적인 방법과 대수적인 방법 두 가지로 각각 증명 => 조합과 관련된 여러 성질 이해. ② 주어진 대상에서 일부를 선택하는 과정 및 결과 => 조합수학의 성질을 활용하여 문제 해결이 가능하다.

② Google학술검색(scholar) 구글스칼라에서 ‘indeterminate equation’, ‘congruent expression’, ‘combinatorics’으로 해외 논문과 학술지를 검색한 결과, 본 연구와 관련 있는 연구물을 중심으로 분석한 결과는 다음과 같다.

저자명	제목	(7) 선행연구7 내용
영국리버풀 대학교 과학 및 공학 Fac 2022	Integer Solution for a Class of Indeterminate Equations	<p>부정방정식의 발견 이후 수학자들은 보편적인 해법을 찾기를 희망해왔지만, 불행하게도 아직까지 고정되고 통일된 방법은 존재하지 않는다. 본 논문에서는 부정방정식 <math>x^2+4^k=4y^n</math>의 해법을 선행연구와 법칙의 추론 및 일반화를 바탕으로 설명한다. 이는 이 특정 부정방정식에 대한 일반화된 솔루션을 제공한다. 오늘날 디오판투스방정식을 풀기 위한 통일된 방법이 없기 때문에 정수 해에 대한 조사는 종종 이전 연구와 결합되어야 한다.</p> <p><u>이 논문은 <math>x^2+4^k=4y^n</math>(<math>k=1,2</math> <math>n=3,5</math>)형식의 부정방정식에 대한 정수해 문제 연구에 정수의 가우스 링과 합동 이론을 적는데 중점을 둔다.</u> 첫 번째 섹션에서는 <math>x^2+4^k=4y^n</math>(<math>n=3</math>, <math>k=1</math> 및 <math>2</math>)의 정수해를 분석한다. 두 번째 섹션에서는 <math>x^2+4^k=4y^n</math>의 정수해를 분석한다. 여기서 <math>n=5</math>, <math>k=1,2</math>이다. 세 번째 섹션에서 <math>x^2+4^k=4y^n</math>과 처음 두 섹션의 결과를 연구하여 특별한 경우의 정수 해의 분포 규칙을 제공한다. 이 연구는 특정 사례의 이해를 위해 지역 분포 법칙을 일반화하여 이러한 종류의 문제 분류에 대한 논의를 단순화하고 다른 사람들의 후속 심층 연구를 위한 대안 아이디어를 제공.</p>
분석 내용		<p>① 이 논문에서는 모든 부정방정식이 <math>x=1(\text{mod}2)</math> 및 <math>x=0(\text{mod}2)</math>에서 논의될 수 있다는 결론을 내렸다. =&gt; 첫 번째 시나리오에 대해서는 논의를 위한 템플릿이 제공되며, 두 번째 시나리오에서는 <math>2k-1=n</math>인 경우에만 정수 해 <math>(x,y)=(\pm 2^k,3)</math>만 존재할 수 있다.</p> <p>② 단점은 이러한 특수 부정방정식의 해 패턴을 찾았음에도 불구하고 ‘k’와 ‘n’이 커질수록 이를 푸는 데 필요한 연산 수가 변경된다. =k, n’이 ‘무한대’ 경향이 있고 ‘k n’이 큰 경우 작업 횟수를 예측하기 어렵다.</p> <p>③ 앞으로 저자는 계산을 예측 가능하고 실행 가능하게 만들기 위해 ‘k, n’ 연산을 단순화하거나 분류 논의 횟수를 줄이는 방법을 연구할 것이다. 저자는 보다 일반적인 해법을 찾기 위해 현재 연구에서 상수의 범위를 넓힐 수 있는 방법을 찾고자 한다.</p>

저자명	제목	(8) 선행연구8 내용
Eshim Murotovich Mardonov 2019	About teaching students to methods of solving problems by combinator	<p>수학 학교 과정을 공부할 때 조합 문제를 해결하기 위한 기술 형성의 몇 가지 측면을 보여줍니다. 수학에는 요소가 다른 집합을 사용할 수 있도록 요구하고 <u>특정 규칙에 따라 형성된 요소의 가능한 모든 조합 수를 세는 문제가 많이 있습니다.</u> 이러한 문제를 조합론이라고 하며 이러한 문제를 해결하는 데 관련된 수학 분야를 조합론이라고 합니다. 예를 들어 학생들과 함께 문제를 해결할 때 2,3,5를 사용하여 두 자리 숫자의 수를 셀 수 있으며 선택을 통해 정확히 9개의 숫자(22,23,32,22,23,32,33,25,35,52,53,55) 따라서 두 번의 독립적인 실험을 수행하는 과정에서 가능한 모든 옵션의 수를 찾으려면 다음의 결과를 찾아야 한다는 것을 학생들이 이해하는 것이 중요합니다. 예를 들어 여러 다른 숫자, 숫자가 반복될 수 있는 4자리 홀수를 구성하려면 가능한 모든 옵션이 먼저 첫 번째 숫자에 대해 계산된 다음 두 번째 숫자에 대해 계산됩니다. 결국 이러한 가능한 모든 옵션이 곱해지고 문제에 대한 해결책이 얻어집니다. 예를 들어, 숫자 1, 3, 5, 7, 9로 구성된 두 자리 숫자의 수를 찾을 때 다음을 찾습니다. 숫자 1의 경우: 11, 13, 15, 17, 19, 숫자 3의 경우: 31, 33, 35, 37, 39, 숫자 5의 경우: 51, 53, 55, 57, 59 각 자리마다 2자리 숫자 5개씩 총 25개가 됩니다. 이 패턴은 <math>5^2 = 25</math>로 볼 수 있습니다. 따라서 이 패턴을 계속하려면 이 다섯 자리 숫자로 세 자리 숫자가 어떻게 구성될 수 있는지, 그 숫자가 반복될 수 있는지 확인하세요. 가설 <math>5^3 = 125</math>. 그것이 사실일까요? 이제 학생들은 이 가설을 테스트하는 것이 문제 해결을 위한 다양한 옵션을 찾는 여러 실험에 대한 곱셈 규칙을 테스트하는 방식으로 관련된다는 점을 분명히 알 수 있습니다.</p>
분석 내용		<p>① 숫자의 공식을 논의할 때, 이 숫자 순서의 규칙성을 스스로 찾아내고 특정 값에 대해 올바른지 확인하는 것이 좋다.</p> <p>1. 수식에 따라 숫자의 위치와 유형을 기준으로 특정 숫자를 찾는 데에는 숫자의 위치와 유형이라는 두 가지 매개변수가 있다.</p> <p>2. 이 숫자가 어떤 종류의 숫자에 속하는지 알아보기 위해 숫자로 역 문제를 해결.</p> <p>② 일상생활에서, 우리는 종종 하나가 아닌, 여러 가지 다른 해결책을 가진 문제들에 직면한다.</p> <p>=&gt; 올바른 선택을 하기 위해서는 어떤 것도 놓치지 않는 것이 매우 중요</p> <p>=&gt; 이렇게 하려면 가능한 모든 옵션을 반복한다.</p> <p>=&gt; 이러한 문제를 조합론이라고 함.</p>

저자명	제목	(9) 선행연구9 내용
Constantinescu, Alexandru 2020	Hilbert Functions A connection between algebra, geometry and combinatorics	Matroids는 독립의 일반적인 개념을 추상화하여 도입되었다. 이러한 조합 구조를 정의하는 방법에는 여러 가지가 있다. 이 작업의 목적을 위해 매트로이드를 특정 유형의 단순 복합체로 생각하는 것이 가장 편리하다. 대략적으로 말하면, 매트로이드는 최대 면이 유한 차원 벡터 공간의 기저에 대한 교환 속성과 유사한 교환 속성을 충족하는 추상 단순 복합체이다. Cohen Macaulayness는 매트로이드의 $h$ 벡터가 힐베르트 함수(유한한 경우 $O$ 수열 또는 $M$ 수열이라고도 함)임을 암시한다. Matroids의 Hilbert 함수에 대한 가능한 설명은 Stanley [Sta77]에 의해 추측되었다. Stanley는 매트로이드의 페이스 링이 Cohen Macaulay 그 이상, 즉 수평임을 증명했다. Stanley의 결과는 매트로이드의 $h$ 벡터가 Artinian 수준 대수 브라의 Hilbert 함수임을 의미한다. Stanley의 추측은 동일한 힐베르트 함수를 갖는 단항 아르티니안 대수 (monomial Artinian algebra)도 존재한다고 말한다. 이러한 대수의 힐베르트 함수를 순수 $O$ 수열이라고 한다. 이를 정의하는 조합적 방법, 즉 순수 다중 복합체의 $f$ 벡터도 있다.
분석 내용		<p>① 투영 변량의 경우 힐베르트 다항식의 정도가 변량의 차원 =&gt; 투영 곡선을 다룰 때 오직 두 개의 계수만이 존재하며, 계수는 곡선의 정도와 속을 결정하여 내포된 곡선의 위상 분류를 제공.</p> <p>② 리만-로흐 공식은 적합한 종류의 모듈에 대한 힐베르트 다항식 계산에서 발생 =&gt; 힐베르트 함수는 1970년대에 조합론으로 발전. 교환 고리를 사용한 스탠리의 상한 추론 증명은 이 분야의 영구적인 플레이어로서 교환 대수와 이산 기하학 사이의 연결을 확립한 그 시기의 하이라이트 중 하나였다.</p> <p>③ 매트로이드의 <math>h</math>-벡터에 대한 스탠리의 추측은 힐베르트 함수 사이의 다른 수학적 영역인 조합 설계 이론, 실제 대수 기하학, 열대 기하학, 최적화 및 근사 이론 사이의 하나의 연결이다.</p>

### Ⅲ. 연구 방법

선행연구 조사결과, 학술연구정보서비스 리스(RISS)와 Google 학술검색(scholar)에서 국내 및 해외 검색 결과, 본 연구와 같이 조합적 성질을 이용한 부정방정식에 관련된 연구는 없음. 즉, 본 연구의 필요성을 확인할 수 있다.

#### 1. 연구 대상

‘정수 계수 부정방정식에서 자연수 혹은 음이 아닌 정수의 순서쌍의 개수를 구할 때 계수적 조합론을 이용하면 합동식보다 나뉘야 할 케이스를 줄이는 효과가 있다.’라는 가설을 세우고 가설을 확인하기 위한 연구문제를 중심으로 문제모형을 만들어 검증한다. 이때, 일차부정방정식, 동일 차수 부정방정식, 각 차수가 다른 부정방정식으로 구분하여 확인한다.

#### 2. 연구 방법

일차부정방정식에서 계수적 조합론 방법과 합동식 방법을 비교분석하여 가설을 확인하고 조합적 계산의 풀이 패턴을 분석하여 하나의 알고리즘을 고안한다.

일차부정방정식에서 계수의 조건, 미지수의 조건과 무관하게 앞에서 고안한 알고리즘의 적용 가능성을 확인한다. 이때 각 계수의 차이가 1과 1이 아닌 경우, 미지수를 2개와 3개로 차이를 두고 10개의 문제 모형을 만들어 검증한다.

마지막으로 동일 차수 부정방정식과 각 차수가 다른 부정방정식의 문제 모형 2개를 만들어 검증하고 앞에서 고안한 알고리즘의 적용 가능성을 확인한다.

1) 일차부정방정식의 자연수 혹은 음이 아닌 정수의 순서쌍 개수를 구하는 방법에 조합적 계산의 연관성은 어떠한가?

① 각 계수의 차이가 1인 경우 자연수 순서쌍의 개수 [부록VI-1]

<예제1-1>  $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로도 풀 수 있을까?

② 각 계수의 차이가 1이 아닌 경우 자연수 순서쌍의 개수 [부록VI-2]

<예제1-2>  $40x+42y+47z=2431$ 일 때 자연수 순서쌍  $(x,y,z)$ 의 개수를 구하는 문제로 바뀐다면 조합적 계산으로는 풀 수 없지 않을까?

③ 각 계수의 차이가 1인 경우 음이 아닌 정수 순서쌍의 개수 [부록VI-3]

<예제1-3>  $40x+41y+42z=2431$ 일 때 음이 아닌 정수 순서쌍  $(x,y,z)$ 의 개수를 구하는 문제로 바뀐다면 조합적 계산으로 풀 수 없지 않을까?

2) 일차부정방정식 계수의 조건에 따라 조합적 계산방법에 미치는 영향이 있을 것인가?

<예제2-1>  $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정과  $40x+42y+47z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로 접근할 때 그 방법은 동일할까? [부록VI-4]

3) 조합적 계산 또는 합동식 이용에 따라 일차부정방정식의 자연수 혹은 음이 아닌 정수의 순서쌍 개수를 구하는 과정에서 어떠한 차이가 있을 것인가?

① 미지수가 적으며 각 계수의 차이가 1인 경우 [부록VI-5]

<예제3-1>  $40x+41y=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

<예제3-2>  $40x+41y=2431$ 일 때 음이 아닌 정수 순서쌍  $(x, y)$ 의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

② 미지수가 많으며 각 계수의 차이가 1인 경우 [부록VI-6]

<예제3-3>  $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 문제로 바뀐다면 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

<예제3-4>  $40x+41y+42z=2431$ 일 때 음이 아닌 정수 순서쌍  $(x, y, z)$ 의 개수를 구하는 문제로 바뀐다면 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

③ 미지수가 많으며 각 계수의 차이가 1보다 큰 경우 [부록VI-7]

<예제3-5>  $40x+42y+47z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

<예제3-6>  $40x+42y+47z=2431$ 일 때 음이 아닌 정수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

4) 동일 차수 부정방정식의 음이 아닌 정수의 순서쌍 개수를 구하는 방법에 조합적 계산의 연관성은 어떠한가?

① 동일 차수 2이상, 각 계수의 차이가 1이상인 경우 음이 아닌 정수의 순서쌍의 개수 [부록VI-8]

<예제4-1>  $40k^n + 42l^n + 47m^n = 2431$ 일 때 음이 아닌 정수의 순서쌍  $(k, l, m)$ 의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로도 풀 수 있을까?(단,  $n \geq 2$ )

5) 각 차수가 다른 부정방정식은 조합적 계산 방법에 미치는 영향이 있을 것인가?

① 각 차수의 차이가 있는 경우 음이 아닌 정수의 순서쌍의 개수 [부록VI-9]

<예제5-1>  $40k^3 + 42l + 47m^2 = 2431$ 일 때 음이 아닌 정수의 순서쌍  $(k, l, m)$ 의 개수를 구하는 과정과  $40k^n + 42l^n + 47m^n = 2431$ 일 때 음이 아닌 정수의 순서쌍  $(k, l, m)$ 의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로 접근할 때 그 방법은 동일할까?

## IV. 연구 결과

### 연구문제1. 일차부정방정식의 조합적 계산의 연관성

일차부정방정식에서 자연수 혹은 음이 아닌 정수의 순서쌍 개수를 구할 때 계수적 조합론을 적용할 수 있음을 확인했다. 연구결과, 계수적 조합론을 이용한 알고리즘을 고안할 수 있게 되었다. ‘처음에 주어진 항이  $x, y, z$ 인 부정방정식의 좌변을 가장 작은 계수로 묶은 후 그다음으로 작은 계수로 묶는 과정을 반복한다. 계수 뒤에 괄호 안에 문자들의 합을 각각 다른 문자로 치환한다. ex)  $x+y+z=a, y+z=b, z=c$  단,  $a \geq b \geq c \geq 0$ 이므로 그 문자들 중 가장 큰 문자  $a$ 의 범위를 기준으로 케이스를 나누고 순서쌍( $a, b, c$ )의 개수를 구한다. 이때 음이 아닌 정수 순서쌍( $a, b, c$ )의 개수는 음이 아닌 정수 순서쌍( $x, y, z$ )의 개수와 같으므로 음이 아닌 정수 순서쌍( $x, y, z$ )의 개수를 구할 수 있다.’

### 연구문제2. 일차부정방정식 계수의 조건에 따라 조합적 계산에 미치는 영향

일차부정방정식에서 계수의 조건이 달라져도 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

①  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 로 치환하면  $40a+b+c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 구하는 방법과 같다.  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 로 치환하면  $40a+2b+5c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 구하는 방법은  $a, b, c$ 의 계수만 다를 뿐이지  $a$ 의 범위를 구하여 케이스를 줄여서 문제를 푸는 방법은 동일하다.

### 연구문제3. 일차부정방정식의 조합적 계산과 합동식의 풀이과정 비교

일차부정방정식에서 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다. 계수적 조합론을 이용하면 합동식보다 나뉘야 할 케이스를 줄이는 효과가 있음을 확인했다.

① 부정방정식의 미지수가 적으며 각 계수의 차이가 1이고 자연수 또는 음이 아닌 정수 순서쌍의 개수를 구하는 문제는 합동식을 이용하는 방법과 조합적 계산을 이용하는 방법은 둘 다 단순하다.

② 부정방정식의 미지수가 많으며 각 계수의 차이가 1이고 자연수 순서쌍의 개수를 구하는 문제를 합동식으로 풀 때,  $40x+41y+42z \equiv z-x \equiv 2431 \equiv 12 \pmod{41}$ 이므로

$z-x=41k+12$ (단,  $k$ 는 정수)라 둔 후,  $z$ 의 범위,  $x$ 의 범위,  $y$ 의 범위 모두 알고 있으므로  $z$ 의 범위에 따라  $x, y$ 의 값을 구할 수 있는데 이 부정방정식의  $z$ 의 범위가 1 이상 57이하이므로 나뉘야 할 케이스는 총 57가지이다. 그런데 이 경우는 너무 많으므로 계수적 조합론을 이용하는 방법을 적용했다.  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+b+c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다.  $a$ 의 값은 58이나 59, 60밖에 되지 않아 나뉘야 할 케이스는 총 3가지이다.  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 합동식을 이용하는 것보다 비교적 단순하게 구할 수 있다. 음이 아닌 정수 순서쌍 개수를 구할 때도 동일한 결론이 나왔다.

③ 부정방정식의 미지수가 많으며 각 계수의 차이가 1보다 크고 자연수 순서쌍의 개수를 구하는 문제를 합동식으로 풀 때,  $40x+42y+47z \equiv 5z-2x \equiv 2431 \equiv 37 \pmod{42}$ 이므로  $5z-2x=42k+37$ (단,  $k$ 는 정수)라 둔 후,  $z$ 의 범위,  $x$ 의 범위,  $y$ 의 범위 모두 알고 있으므로  $z$ 의 범위에 따라  $x, y$ 의 값을 구할 수 있는데 이 부정방정식의  $z$ 의 범위가 1이상 51이하인 홀수(왜냐하면 만약  $z$ 가 짝수면 좌변은  $40x, 42y, 47z$ 가 모두 짝수가 되어  $40x+42y+47z$ 는 짝수인데 2431은 홀수이기 때문이다)이므로 나뉘야 할 케이스는 총 26가지이다. 그런데 이 경우는 너무 많으므로 계수적 조합론을 이용하는 방법을 적용했다.  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+2b+5c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다.  $a \geq 3, b \geq 2, c \geq 1$ 이므로  $a_1=a-3, b_1=b-2, c_1=c-1$ 이라 하자.  $a$ 의 값은 51이나 52, ..., 57, 58, 59가 나오기 때문에 나뉘야 할 케이스는 총 9가지이다.  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 합동식을 이용하는 것보다 비교적 단순하게 구할 수 있다. 음이 아닌 정수 순서쌍 개수를 구할 때도 동일한 결론이 나왔다.

#### 연구문제4. 동일 차수 부정방정식의 조합적 계산의 연관성

동일 차수 부정방정식은  $k^n=x, l^n=y, m^n=z$ 로 치환하면, 일차부정방정식으로 바꿔 놓고 풀 수 있으므로 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

#### 연구문제5. 각 차수가 다른 부정방정식의 조합적 계산에 미치는 영향

$n$ 이 2차로 된 정수 계수 부정방정식인 경우 계수를 제외한 각 항을 완전제곱식 형태로 치환하면 일차부정방정식 혹은 우변이 0이 아닌 상수꼴로 인수분해가 가능하기 때문에 연구문제1에서 고안한 알고리즘을 적용할 수 있지만,  $n$ 이 3차 이상인 경우 3차 이상의 합동식은 다루기 힘들고 계수적 조합론 방법은 일반적으로 정수 계수로 인수분해가 가능하지 않기 때문에 연구문제1에서 고안한 알고리즘을 적용할 수 없음을 확인했다.

## V. 결론 및 제언

이 연구는 정수 계수 부정방정식의 음이 아닌 정수해의 개수를 구하는 것에 계수적 조합론을 이용한 하나의 알고리즘을 제시하는 데 목적이 있었다. 그 결과 다음과 같은 결론을 얻었다.

첫째, 부정방정식의 자연수 혹은 음이 아닌 정수의 순서쌍의 개수를 합동식으로 구할 때 처음에 범을 몇으로 나누어야 하는지 떠올리기가 어렵다. 범  $m$ 으로 하고 나눈 나머지를  $n$ 으로 하면 좌변에 문자로 이루어져 있는 식과  $km+n$ 이 같다. 좌변에 한 문자를 두고 우변에 나머지 문자와 숫자를 둔 후 주어진 부정방정식에 대입한다는 것을 생각해 내는 것이 어렵다. 게다가 좌변에 한 문자의 범위를 기준으로 케이스를 정하는데 이 또한 생각하기 어렵고 케이스를 나눈다고 해도 그 케이스가 너무 많기 때문에 계산 실수 등을 할 확률이 높다는 문제 인식을 느껴서 계수적 조합론을 적용해 보기로 했다.

둘째, 연구문제 1을 통해 일차부정방정식에서 계수적 조합론을 적용할 수 있음을 확인했다. 더 나아가 계수적 조합론을 이용한 알고리즘을 고안해 보았다.

셋째, 연구문제 2를 통해 일차부정방정식에서 계수의 조건이 바뀌더라도 연구문제 1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

넷째, 연구문제 3을 통해 일차부정방정식에서 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다. 계수적 조합론을 이용하면 합동식보다 나뉘야 할 케이스를 줄이는 효과가 있음을 확인했다.

다섯째, 연구문제 4를 통해 동일 차수 부정방정식은  $k^n=x$ ,  $l^n=y$ ,  $m^n=z$ 로 치환하면, 일차부정방정식으로 바꿔 놓고 풀 수 있으므로 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

마지막으로 연구문제 5를 통해  $n$ 이 2차로 된 정수 계수 부정방정식인 경우 계수를 제외한 각 항을 완전제곱식 형태로 치환하면 일차부정방정식 혹은 우변이 0이 아닌 상수꼴로 인수분해가 가능하기 때문에 연구문제1에서 고안한 알고리즘을 적용할 수 있지만,  $n$ 이 3차 이상인 경우 3차 이상의 합동식은 다루기 힘들고 계수적 조합론 방법은 일반적으로 정수 계수로 인수분해가 가능하지 않기 때문에 연구문제1에서 고안한 알고리즘을 적용할 수 없다는 유의미한 결과를 얻었다.

본 연구에서는 연구문제를 중심으로 문제 모형을 만들고 풀이과정을 해석하고 합동식과 비교분석을 통해 계수적 조합론으로 접근할 필요성에 대해 알아보았다. 정수 계수 부정방정식에서 음이 아닌 정수의 순서쌍의 개수를 구하기 위해서 합동식 외의 조합적 계산방법을 활용하며, 나아가 계수적 조합론을 이용한 ‘처음에 주어진 부정방정식의 좌변을 가장 작은 계수로 묶은 후 그다음으로 작은 계수로 묶는 과정을 반복한다. 괄호 안에 문자들의 합을 각각 다른 문자  $x+y+z$ 를  $a$ ,  $y+z$ 를  $b$ ,  $z$ 를  $c$ 로 치환한다. 단,  $a \geq b \geq c \geq 0$  이므로 그 문자들 중 가장 큰 문자  $a$ 의 범위를 기준으로

로 케이스를 나누고 순서쌍(a,b,c)의 개수를 구한다.’라는 알고리즘을 고안하고 효율적인 시간 내에 진행해보았다.

이러한 연구 결과를 바탕으로 시사점을 제언하면, 계수적 조합론을 이용한 알고리즘을 제공함으로써 합동식 보다 나뉘야 할 케이스를 줄이는 효과가 있고 효율적인 시간 내에 계산실수 가능성이 없는 단순하면서도 실용적으로 정답에 접근할 수 있을 것으로 그 의미가 있다고 하겠다. 학술연구정보서비스 리스(RISS)와 Google 학술 검색(scholar)에서 ‘부정방정식’, ‘합동식’, ‘조합 수학’으로 국내 및 해외 논문을 검색하고 본 연구와 관련 있는 연구물을 중심으로 분석하는 방법으로 선행 연구 조사 결과, 본 연구와 같이 계수적 조합론을 이용한 부정방정식에 관련된 연구는 없었다. 즉, 본 연구의 필요성과 독창성을 확인할 수 있었다.

## VI.참고문헌

- [1] 김희재(2005). 여러 가지 부정방정식에 관한 연구. 석사 학위 논문. 성균관대학교 교육대학원 수학교육. (p.58).
- [2] 네이버지식백과 두산백과. 부정방정식. <https://terms.naver.com>.
- [3] 네이버지식백과 두산백과. 합동식. <https://terms.naver.com>.
- [4] 위키백과(2023.01.03). 조합론. <https://ko.wikipedia.org>.

## VII. 부록

### [부록VI-1] ① 각 계수의 차이가 1인 경우 자연수 순서쌍의 개수

<예제1-1>  $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로도 풀 수 있을까?

(조합적 계산을 이용하는 방법)  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이다

$\therefore x+y+z=a, y+z=b, z=c$ 라 하자

$\therefore 40a+b+c=2431$ 이다

이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다

$\therefore a$ 의 값은 58이나 59, 60밖에 되지 않는다( $\because a$ 가 58보다 작으면  $b+c$ 는 151이상이므로  $b$ 와  $c$ 중 하나는 57보다 크고,  $a$ 가 60보다 크면  $40a$ 가 2431보다 크게 되기 때문이다)

$a=58$ 일 때  $b+c=111$ 이므로  $b=57, c=54$

$b=56, c=55$

$a=59$ 일 때  $b+c=71$ 이므로  $b=58, c=13$

$b=57, c=14$

○

○

○

$b=36, c=35$

$a=60$ 일 때  $b+c=31$ 이므로  $b=30, c=1$

$b=29, c=2$

○

○

○

$b=16, c=15$ 이다

$\therefore$  총 개수는  $2+23+15=40$ 개

$\therefore$  40개

(해석) 계수적 조합론을 활용하면  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+b+c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍  $(a, b, c)$ 의 개수를 구할 수 있다. 이때, 자연수 순서쌍  $(a, b, c)$ 의 개수는 자연수 순서쌍  $(x, y, z)$ 의 개수가 같으므로 자연수 순서쌍  $(x, y, z)$ 의 개수를 구할 수 있다.

### [부록VI-2] ② 각 계수의 차이가 1이 아닌 경우 자연수 순서쌍의 개수

<예제1-2>  $40x+42y+47z=2431$ 일 때 자연수 순서쌍  $(x,y,z)$ 의 개수를 구하는 문제로 바뀐다면 조합적 계산으로는 풀 수 없지 않을까?

(조합적 계산을 이용하는 방법)  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이다

$\therefore x+y+z=a, y+z=b, z=c$ 라 하자

$\therefore 40a+2b+5c=2431$ 이다

이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다

$\therefore a \geq 3, b \geq 2, c \geq 1$ 이므로  $a_1=a-3, b_1=b-2, c_1=c-1$ 이라 하자

$$40a+2b+5c=40a_1+40+2b_1+2+5c_1+5=40a_1+2b_1+5c_1+47=2431$$

$$\therefore 40a_1+2b_1+5c_1=2384(a_1 \geq b_1 \geq c_1 \geq 0)$$

$\therefore a_1$ 의 값은 51이나 52,  $\circ \circ \circ$ , 57, 58, 59이 나올 수 있다( $\because a_1$ 이 51보다 작으면  $2b_1+5c_1$ 는 384이상인데 그러면  $b_1$ 이나  $c_1$ 중 하나는 52이상이고,  $a_1$ 이 59보다 크면  $40a_1$ 이 2384보다 크다)

$a_1=51$ 일 때  $2b_1+5c_1$ 인 정수해  $b_1, c_1$ 은 존재하지 않는다

$$a_1=52\text{일 때 } 2b_1+5c_1=304\text{이므로 } b_1=47, c_1=42$$

$$a_1=53\text{일 때 } 2b_1+5c_1=264\text{이므로 } b_1=37, c_1=36$$

$$b_1=42, c_1=34$$

$$b_1=47, c_1=32$$

$$a_1=54\text{일 때 } 2b_1+5c_1=224\text{이므로 } b_1=32, c_1=32$$

$$b_1=37, c_1=30$$

$$b_1=42, c_1=28$$

$$b_1=47, c_1=26$$

$$a_1=55\text{일 때 } 2b_1+5c_1=184\text{이므로 } b_1=27, c_1=26$$

$$b_1=32, c_1=24$$

$$b_1=37, c_1=22$$

$$b_1=42, c_1=20$$

$$b_1=47, c_1=18$$

$$a_1=56\text{일 때 } 2b_1+5c_1=144\text{이므로 } b_1=22, c_1=20$$

$$b_1=27, c_1=18$$

$$b_1=32, c_1=16$$

$$b_1=37, c_1=14$$

$$b_1=42, c_1=12$$

$$b_1=47, c_1=10$$

$$a_1=57\text{일 때 } 2b_1+5c_1=104\text{이므로 } b_1=17, c_1=14$$

$$b_1=22, c_1=12$$

$$b_1=27, c_1=10$$

$$b_1=32, c_1=8$$

$$b_1=37, c_1=6$$

$$b_1=42, c_1=4$$

$$b_1=47, c_1=2$$

$$a_1=58\text{일 때 } 2b_1+5c_1=64\text{이므로 } b_1=12, c_1=8$$

$$b_1=17, c_1=6$$

$$b_1=22, c_1=4$$

$$b_1=27, c_1=2$$

$$b_1=32, c_1=0$$

$$a_1=59\text{일 때 } 2b_1+5c_1=24\text{이므로 } b_1=7, c_1=2$$

$$b_1=12, c_1=0$$

$$\therefore \text{총 개수는 } 0+1+3+4+5+6+7+5+2=33$$

$$\therefore 33\text{개}$$

**(해석)** 계수적 조합론을 활용하면  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이므로  $x+y+z=a$ ,  $y+z=b$ ,  $z=c$ 라 하면  $40a+2b+5c=2431$ 이다. 이때,  $x>0$ ,  $y>0$ ,  $z>0$ 이므로  $a>b>c>0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 구할 수 있다.

[부록VI-3] ③ 각 계수의 차이가 1인 경우 음이 아닌 정수 순서쌍의 개수

<예제1-3>  $40x+41y+42z=2431$ 일 때 음이 아닌 정수 순서쌍  $(x,y,z)$ 의 개수를 구하는 문제로 바뀐다면 조합적 계산으로 풀 수 없지 않을까?

(조합적 계산을 이용하는 방법)  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이다

$\therefore x+y+z=a, y+z=b, z=c$ 라 하자

$\therefore 40a+b+c=2431$ 이다

이때,  $x \geq 0, y \geq 0, z \geq 0$ 이므로  $a \geq b \geq c \geq 0$ 이다

$\therefore a$ 의 값은 58이나 59, 60밖에 되지 않는다( $\because a$ 가 58보다 작으면  $b+c$ 는 151이상이므로  $b$ 와  $c$ 중 하나는 58보다 크고,  $a$ 가 60보다 크면  $40a$ 가 2431보다 크게 되기 때문이다)

$a=58$ 일 때  $b+c=111$ 이므로  $b=58, c=53$

$b=57, c=54$

$b=56, c=55$

$a=59$ 일 때  $b+c=71$ 이므로  $b=59, c=12$

$b=58, c=13$

○

○

○

$b=36, c=35$

$a=60$ 일 때  $b+c=31$ 이므로  $b=31, c=0$

$b=30, c=1$

○

○

○

$b=16, c=15$ 이다

$\therefore$  43개

(해석) 계수적 조합론을 활용하면  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+b+c=2431$ 이다. 이때,  $x \geq 0, y \geq 0, z \geq 0$ 이므로  $a \geq b \geq c \geq 0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 음이 아닌 정수 순서쌍  $(a, b, c)$ 의 개수를 구할 수 있다. 이때, 음이 아닌 정수 순서쌍  $(a, b, c)$ 의 개수는 음이 아닌 정수 순서쌍  $(x, y, z)$ 의 개수가 같으므로 음이 아닌 정수 순서쌍  $(x, y, z)$ 의 개수를 구할 수 있다.

연구결과, <예제1-1>, <예제1-2>, <예제1-3>을 통해 '처음에 주어진 항이  $x, y, z$ 인 부정방정식의 좌변을 가장 작은 계수로 묶은 후 그다음으로 작은 계수로 묶는 과정을 반복한다. 계수 뒤에 괄호 안에 문자들의 합을 각각 다른 문자로 치환한다. ex)  $x+y+z=a, y+z=b, z=c$  단,  $a \geq b \geq c \geq 0$ 이므로 그 문자들 중 가장 큰 문자  $a$ 의 범위를 기준으로 케이스를 나누고 순서쌍  $(a, b, c)$ 의 개수를 구한다. 이때 음이 아닌 정수 순서쌍  $(a, b, c)$ 의 개수는 음이 아닌 정수 순서쌍  $(x, y, z)$ 의 개수와 같으므로 음이 아닌 정수 순서쌍  $(x, y, z)$ 의 개수를 구할 수 있다.'라는 계수적 조합론의 알고리즘을 고안할 수 있다.

[부록VI-4] 일차부정방정식 계수의 조건에 따라 조합적 계산방법에 미치는 영향

<예제2-1>  $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정과  $40x+42y+47z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로 접근할 때 그 방법은 동일할까?

( $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 조합적 계산으로 접근할 때, 구하는 방법)  
 $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이다

∴  $x+y+z=a$ ,  $y+z=b$ ,  $z=c$ 라 하자

∴  $40a+b+c=2431$ 이다

이때,  $x>0$ ,  $y>0$ ,  $z>0$ 이므로  $a>b>c>0$ 이다

∴  $a$ 의 값은 58이나 59, 60밖에 되지 않는다(∵  $a$ 가 58보다 작으면  $b+c$ 는 151이상이므로  $b$ 와  $c$ 중 하나는 57보다 크고,  $a$ 가 60보다 크면  $40a$ 가 2431보다 크게 되기 때문이다)

$a=58$ 일 때  $b+c=111$ 이므로  $b=57$ ,  $c=54$

$b=56$ ,  $c=55$

$a=59$ 일 때  $b+c=71$ 이므로  $b=58$ ,  $c=13$

$b=57$ ,  $c=14$

◦

◦

◦

$b=36$ ,  $c=35$

$a=60$ 일 때  $b+c=31$ 이므로  $b=30$ ,  $c=1$

$b=29$ ,  $c=2$

◦

◦

◦

$b=16$ ,  $c=15$ 이다

∴ 총 개수는  $2+23+15=40$ 개

∴ 40개

**( $40x+42y+47z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 조합적 계산으로 접근할 때, 구하는 방법)**

$40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이다

∴  $x+y+z=a$ ,  $y+z=b$ ,  $z=c$ 라 하자

∴  $40a+2b+5c=2431$ 이다

이때,  $x>0$ ,  $y>0$ ,  $z>0$ 이므로  $a>b>c>0$ 이다

∴  $a \geq 3$ ,  $b \geq 2$ ,  $c \geq 1$ 이므로  $a_1=a-3$ ,  $b_1=b-2$ ,  $c_1=c-1$ 이라 하자

$40a+2b+5c=40a_1+40+2b_1+2+5c_1+5=40a_1+2b_1+5c_1+47=2431$

∴  $40a_1+2b_1+5c_1=2384$  ( $a_1 \geq b_1 \geq c_1 \geq 0$ )

∴  $a_1$ 의 값은 51이나 52, ◦ ◦ ◦, 57, 58, 59이 나올 수 있다(∵  $a_1$ 이 51보다 작으면  $2b_1+5c_1$ 는 384이상인데 그러면  $b_1$ 이나  $c_1$ 중 하나는 52이상이고,  $a_1$ 이 59보다 크면  $40a_1$ 이 2384보다 크다)

$a_1=51$ 일 때  $2b_1+5c_1$ 인 정수해  $b_1$ ,  $c_1$ 은 존재하지 않는다

$a_1=52$ 일 때  $2b_1+5c_1=304$ 이므로  $b_1=47$ ,  $c_1=42$

$a_1=53$ 일 때  $2b_1+5c_1=264$ 이므로  $b_1=37$ ,  $c_1=36$

$b_1=42$ ,  $c_1=34$

$b_1=47$ ,  $c_1=32$

$a_1=54$ 일 때  $2b_1+5c_1=224$ 이므로  $b_1=32$ ,  $c_1=32$

$b_1=37$ ,  $c_1=30$

$b_1=42$ ,  $c_1=28$

$b_1=47$ ,  $c_1=26$

$a_1=55$ 일 때  $2b_1+5c_1=184$ 이므로  $b_1=27$ ,  $c_1=26$

$b_1=32$ ,  $c_1=24$

$b_1=37$ ,  $c_1=22$

$b_1=42$ ,  $c_1=20$

$b_1=47$ ,  $c_1=18$

$a_1=56$ 일 때  $2b_1+5c_1=144$ 이므로  $b_1=22$ ,  $c_1=20$

$b_1=27$ ,  $c_1=18$

$$b_1=32, c_1=16$$

$$b_1=37, c_1=14$$

$$b_1=42, c_1=12$$

$$b_1=47, c_1=10$$

$a_1=57$ 일 때  $2b_1+5c_1=104$ 이므로  $b_1=17, c_1=14$

$$b_1=22, c_1=12$$

$$b_1=27, c_1=10$$

$$b_1=32, c_1=8$$

$$b_1=37, c_1=6$$

$$b_1=42, c_1=4$$

$$b_1=47, c_1=2$$

$a_1=58$ 일 때  $2b_1+5c_1=64$ 이므로  $b_1=12, c_1=8$

$$b_1=17, c_1=6$$

$$b_1=22, c_1=4$$

$$b_1=27, c_1=2$$

$$b_1=32, c_1=0$$

$a_1=59$ 일 때  $2b_1+5c_1=24$ 이므로  $b_1=7, c_1=2$

$$b_1=12, c_1=0$$

∴ 총 개수는  $0+1+3+4+5+6+7+5+2=33$

∴ 33개

**(해석)**  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 로 치환하면  $40a+b+c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 구하는 방법과 같다.  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 로 치환하면  $40a+2b+5c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다. 그러므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 구하는 방법은  $a, b, c$ 의 계수만 다를 뿐이지  $a$ 의 범위를 구하여 케이스를 줄여서 문제를 푸는 방법은 동일하다. **연구결과, 일차부정방정식에서 계수의 조건이 달라져도 연구문제1에서 고안한 알고리즘을 적용할 수 있다.**

### [부록 VI-5] ① 미지수가 적으며 각 계수의 차이가 1인 경우

<예제3-1>  $40x+41y=2431$ 일 때 자연수 순서쌍 ( $x, y$ )의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

**(합동식을 이용하는 방법)**  $40x+41y=2431$ 일 때 법 40에서 보면  $40x+41y \equiv y \equiv 2431 \equiv 31 \pmod{40}$

∴  $y \equiv 31 \pmod{40}$ 이므로  $y=40k+31$ 이라 두자(단,  $k$ 는 정수)

∴  $y=31$ 만이 가능하다(∵  $y$ 가 71보다 크면  $71*41=2911>2431$ 이기 때문이다)

$$\therefore 40x+41y=40x+41*31=40x+1271=2431$$

$$\therefore 40x=1160$$

$$\therefore x=29$$

$$\therefore x=29, y=31$$

∴ 1개

**(조합적 계산을 이용하는 방법)**  $40x+41y=40(x+y)+y=2431$ 이다

∴  $x+y=a$ ,  $y=b$ 라 하자

∴  $40a+b=2431$ 이다

이때,  $x>0$ ,  $y>0$ 이므로  $a>b$ 이다

∴  $a$ 의 값은 60밖에 되지 않는다(∵  $a$ 가 60보다 작으면  $b$ 가 60보다 크게 되고  $a$ 가 60보다 크면  $40a$ 가 2431보다 크게 되기 때문이다)

$a=60$ 일 때  $b=31$ 이므로  $x+y=60$ ,  $y=31$ 이다

∴  $x=29$ ,  $y=31$

∴ 1개

**(해석)** 부정방정식의 미지수가 적으며 각 계수의 차이가 1이고 자연수 순서쌍의 개수를 구하는 문제이다. 이 문제를 합동식으로 풀 때,  $40x+41y \equiv y \equiv 2431 \equiv 31 \pmod{40}$ 이므로  $y=40k+31$ (단,  $k$ 는 정수)라 둘 수 있는데,  $y$ 의 범위를 구할 수 있으므로 그  $y$ 의 범위에 맞는 정수를 구하고 각  $y$ 에 대한  $x$ 의 값을 구할 수 있다. 이 문제를 조합적 계산을 이용한 방법으로 풀 때,  $40x+41y=40(x+y)+y=2431$ 이므로  $x+y=a$ ,  $y=b$ 라 하면  $40a+b=2431$ 이다. 이때,  $x \geq 0$ ,  $y \geq 0$ 이므로  $a \geq b \geq 0$ 이므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a$ ,  $b$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a$ ,  $b$ )의 개수는 자연수 순서쌍( $x$ ,  $y$ )의 개수가 같으므로 자연수 순서쌍( $x$ ,  $y$ )의 개수를 구할 수 있다. 부정방정식의 미지수가 적으며 각 계수의 차이가 1이고 자연수 순서쌍의 개수를 구하는 문제는 합동식을 이용하는 방법과 조합적 계산을 이용하는 방법은 둘 다 단순하다.

**<예제3-2>**  $40x+41y=2431$ 일 때 음이 아닌 정수 순서쌍 ( $x$ ,  $y$ )의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

**(합동식을 이용하는 방법)**  $40x+41y=2431$ 일 때 법 40에서 보면  $40x+41y \equiv y \equiv 2431 \equiv 31 \pmod{40}$

∴  $y \equiv 31 \pmod{40}$ 이므로  $y=40k+31$ 이라 두자.(단,  $k$ 는 정수)

∴  $y=31$ 만이 가능하다(∵  $y$ 가 71보다 크면  $71 \cdot 41=2911 > 2431$ 이기 때문이다)

∴  $40x+41y=40x+41 \cdot 31=40x+1271=2431$

∴  $40x=1160$

∴  $x=29$

∴  $x=29$ ,  $y=31$

∴ 1개

**(조합적 계산을 이용하는 방법)**  $40x+41y=40(x+y)+y=2431$ 이다

∴  $x+y=a$ ,  $y=b$ 라 하자

∴  $40a+b=2431$ 이다

이때,  $x \geq 0$ ,  $y \geq 0$ 이므로  $a \geq b$ 이다

∴  $a$ 의 값은 60밖에 되지 않는다(∵  $a$ 가 60보다 작으면  $b$ 가 60보다 크게 되고  $a$ 가 60보다 크면  $40a$ 가 2431보다 크게 되기 때문이다)

$a=60$ 일 때  $b=31$ 이므로  $x+y=60$ ,  $y=31$ 이다

∴  $x=29$ ,  $y=31$

∴ 1개

**(해석)** 부정방정식의 미지수가 적으며 각 계수의 차이가 1이고 음이 아닌 정수 순서쌍의 개수를 구하는 문제이다. 이 문제를 합동식으로 풀 때,  $40x+41y \equiv y \equiv 2431 \equiv 31 \pmod{40}$ 이므로  $y=40k+31$ (단,  $k$ 는 정수)라 둘 수 있는데,  $y$ 의 범위를 구할 수 있으므로 그  $y$ 의 범위에 맞는 정수를 구하고 각  $y$ 에 대한  $x$ 의 값을 구할 수 있다. 이 문제를 조합적 계산을 이용한 방법으로 풀 때,  $40x+41y=40(x+y)+y=2431$ 이므로  $x+y=a$ ,  $y=b$ 라 하면  $40a+b=2431$ 이다. 이때,  $x > 0$ ,  $y > 0$ 이므로  $a > b > 0$ 이므로  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 음이 아닌 정수 순서쌍( $a$ ,  $b$ )의 개수를 구할 수 있다. 이때, 음이 아닌 정수 순서쌍( $a$ ,  $b$ )의 개수는 음이 아닌 정수 순서쌍( $x$ ,  $y$ )의 개수가 같으므로 음이 아닌 정수 순서쌍( $x$ ,  $y$ )의 개수를 구할 수 있다. 부정방정식의 미지수가 적으며 각 계수의 차이가 1이고 음이 아닌 정수 순서쌍의 개수를 구하는 문제는 합동식을 이용하는 방법과 조합적 계산을 이용하는 방법은 둘 다 단순하다.

[부록VI-6] ② 미지수가 많으며 각 계수의 차이가 1인 경우

<예제3-3>  $40x+41y+42z=2431$ 일 때 자연수 순서쌍  $(x, y, z)$ 의 개수를 구하는 문제로 바뀐다면 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

(합동식을 이용하는 방법)  $40x+41y+42z \equiv z-x \equiv 2431 \equiv 12 \pmod{41}$ 이므로

$z-x \equiv 12 \pmod{41}$ 이다

$\therefore z-x=41k+12$ 라 하자(단,  $k$ 는 정수)

$\therefore z=x+41k+12$

이때,  $z$ 의 범위는 1이상 57이하이고  $x$ 의 범위는 1이상 60이하이고  $y$ 의 범위는 1이상 59이하이므로

$z=1$ 일 때  $x=30, y=29$

$z=2$ 일 때  $x=31, y=27$

$z=3$ 일 때  $x=32, y=25$

$z=4$ 일 때  $x=33, y=23$

◦  
◦  
◦

$z=12$ 일 때  $x=41, y=7$

$z=13$ 일 때  $x=42, y=5$

$x=1, y=45$

$z=14$ 일 때  $x=43, y=3$

$x=2, y=43$

$z=15$ 일 때  $x=44, y=1$

$x=3, y=41$

$z=16$ 일 때  $x=4, y=39$

$z=17$ 일 때  $x=5, y=37$

$z=18$ 일 때  $x=6, y=35$

◦  
◦  
◦

$z=35$ 일 때  $x=23, y=1$

$z=36$ 일 때 존재하지 않는다

$z=37$ 일 때 존재하지 않는다

◦  
◦  
◦

$z=53$ 일 때 존재하지 않는다

$z=54$ 일 때  $x=1, y=3$

$z=55$ 일 때  $x=2, y=1$

$\therefore$  총 개수는 40개

$\therefore$  40개

(조합적 계산을 이용하는 방법)  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이다

$\therefore x+y+z=a, y+z=b, z=c$ 라 하자

$\therefore 40a+b+c=2431$ 이다

이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다

$\therefore a$ 의 값은 58이나 59, 60밖에 되지 않는다( $\because a$ 가 58보다 작으면  $b+c$ 는 151이상이므로  $b$ 와  $c$ 중 하나는 57보다

크고, a가 60보다 크면 40a가 2431보다 크게 되기 때문이다)

a=58일 때 b+c=111이므로 b=57, c=54

b=56, c=55

a=59일 때 b+c=71이므로 b=58, c=13

b=57, c=14

◦

◦

◦

b=36, c=35

a=60일 때 b+c=31이므로 b=30, c=1

b=29, c=2

◦

◦

◦

b=16, c=15이다

∴ 총 개수는 2+23+15=40개

∴ 40개

**(해석)** 부정방정식의 미지수가 많으며 각 계수의 차이가 1이고 자연수 순서쌍의 개수를 구하는 문제이다. 이 문제를 합동식으로 풀 때,  $40x+41y+42z \equiv z-x \equiv 2431 \equiv 12 \pmod{41}$ 이므로  $z-x=41k+12$ (단, k는 정수)라 둔 후, z의 범위, x의 범위, y의 범위 모두 알고 있으므로 z의 범위에 따라 x, y의 값을 구할 수 있는데 이 부정방정식의 z의 범위가 1이상 57이하이므로 나뉘야 할 케이스는 총 57가지이다. 그런데 이 경우는 너무 많으므로 계수적 조합론을 이용하는 방법을 적용했다.  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+b+c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이므로 a의 값은 58이나 59, 60밖에 되지 않아 나뉘야 할 케이스는 총 3가지이다. a의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍(a, b, c)의 개수를 구할 수 있다. 이때, 자연수 순서쌍(a, b, c)의 개수는 자연수 순서쌍(x, y, z)의 개수가 같으므로 자연수 순서쌍(x, y, z)의 개수를 합동식을 이용하는 것보다 나뉘야 할 케이스를 줄이는 효과가 있다.

**<예제3-4>**  $40x+41y+42z=2431$ 일 때 음이 아닌 정수 순서쌍 (x, y, z)의 개수를 구하는 문제로 바뀐다면 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

**(합동식을 이용하는 방법)**  $40x+41y+42z \equiv z-x \equiv 2431 \equiv 12 \pmod{41}$ 이므로

$z-x \equiv 12 \pmod{41}$ 이다(단, k는 정수)

∴  $z-x=41k+12$ 라 하자

∴  $z=x+41k+12$

이때, z의 범위는 0이상 57이하이고 x의 범위는 0이상 60이하이고 y의 범위는 0이상 59이하이므로

z=0일 때 x=29, y=31

z=1일 때 x=30, y=29

z=2일 때 x=31, y=27

z=3일 때 x=32, y=25

z=4일 때 x=33, y=23

◦

◦

◦

z=11일 때 x=40, y=9

z=12일 때 x=41, y=7

x=0, y=47

z=13일 때 x=42, y=5

x=1, y=45

z=14일 때 x=43, y=3  
           x=2, y=43

z=15일 때 x=44, y=1  
           x=3, y=41

z=16일 때 x=4, y=39

z=17일 때 x=5, y=37

z=18일 때 x=6, y=35

◦  
 ◦  
 ◦

z=35일 때 x=23, y=1

z=36일 때 존재하지 않는다

z=37일 때 존재하지 않는다

◦  
 ◦  
 ◦

z=52일 때 존재하지 않는다

z=53일 때 x=0, y=5

z=54일 때 x=1, y=3

z=55일 때 x=2, y=1

∴ 총 개수는 43개

∴ 43개

**(조합적 계산을 이용하는 방법)**  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이다

∴  $x+y+z=a$ ,  $y+z=b$ ,  $z=c$ 라 하자

∴  $40a+b+c=2431$ 이다

이때,  $x \geq 0$ ,  $y \geq 0$ ,  $z \geq 0$ 이므로  $a \geq b \geq c \geq 0$ 이다

∴ a의 값은 58이나 59, 60밖에 되지 않는다(∵ a가 58보다 작으면 b+c는 151이상이므로 b와 c중 하나는 58보다 크고, a가 60보다 크면 40a가 2431보다 크게 되기 때문이다)

a=58일 때 b+c=111이므로 b=58, c=53

          b=57, c=54

          b=56, c=55

a=59일 때 b+c=71이므로 b=59, c=12

          b=58, c=13

◦  
 ◦  
 ◦

          b=36, c=35

a=60일 때 b+c=31이므로 b=31, c=0

          b=30, c=1

◦  
 ◦  
 ◦

          b=16, c=15이다

∴ 총 개수는 3+24+16=43개

∴ 43개

**(해석)** 부정방정식의 미지수가 많으며 각 계수의 차이가 1이고 음이 아닌 정수 순서쌍의 개수를 구하는 문제이다.

이 문제를 합동식으로 풀 때,  $40x+41y+42z \equiv z-x \equiv 2431 \equiv 12 \pmod{41}$ 이므로  $z-x=41k+12$ (단, k는 정수)라 둔 후, z의 범위, x의 범위, y의 범위 모두 알고 있으므로 z의 범위에 따라 x, y의 값을 구할 수 있는데 이 부정방정식의 z의 범위가 0이상 57이하이므로 나뉘야 할 케이스는 총 58가지이다. 그런데 이 경우는 너무 많으므로 계수적 조합론을 이용하는 방법을 적용했다.  $40x+41y+42z=40(x+y+z)+(y+z)+z=2431$ 이므로  $x+y+z=a$ ,  $y+z=b$ ,  $z=c$ 라 하면  $40a+b+c=2431$ 이다. 이때,  $x \geq 0$ ,  $y \geq 0$ ,  $z \geq 0$ 이므로  $a \geq b \geq c \geq 0$ 이므로 a의 값은 58이나 59, 60밖에 되지 않아 나뉘야

할 경우는 총 3가지이다. a의 범위를 좁힐 수 있어 범위를 줄여 음이 아닌 정수 순서쌍(a, b, c)의 개수를 구할 수 있다. 이때, 음이 아닌 정수 순서쌍(a, b, c)의 개수는 음이 아닌 정수 순서쌍(x, y, z)의 개수가 같으므로 음이 아닌 정수 순서쌍(x, y, z)의 개수를 합동식을 이용하는 것보다 나뉘야 할 케이스를 줄이는 효과가 있다.

[부록VI-7] ③ 미지수가 많으며 각 계수의 차이가 1보다 큰 경우

<예제3-5>  $40x+42y+47z=2431$ 일 때 자연수 순서쌍 (x, y, z)의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

(합동식을 이용하는 방법)  $40x+42y+47z \equiv 5z-2x \equiv 2431 \equiv 37 \pmod{42}$ 이므로

$5z-2x \equiv 37 \pmod{42}$ 이다

$\therefore 5z-2x=42k+37$ 이라 하자(단, k는 정수)

$\therefore 5z=2x+42k+37$

이때, z의 범위는 1이상 51이하(z는 짝수이면 않된다. 왜냐하면 만약 z가 짝수면 좌변은  $40x, 42y, 47z$ 가 모두 짝수가 되어  $40x+42y+47z$ 는 짝수인데 2431은 홀수이기 때문이다)이고 x의 범위는 1이상 60이하이고 y의 범위는 1이상 57이하이므로

z=1일 때 x=47, y=12

x=26, y=32

x=5, y=52

z=3일 때 x=52, y=5

x=31, y=25

x=10, y=45

z=5일 때 x=36, y=18

x=15, y=38

z=7일 때 x=41, y=11

x=20, y=31

z=9일 때 x=46, y=4

x=25, y=24

x=4, y=44

z=11일 때 x=30, y=17

x=9, y=37

z=13일 때 x=35, y=10

x=14, y=30

z=15일 때 x=40, y=3

x=19, y=23

z=17일 때 x=24, y=16

x=3, y=36

z=19일 때 x=29, y=9

x=8, y=29

z=21일 때 x=34, y=2

x=13, y=22

z=23일 때 x=18, y=15

z=25일 때 x=23, y=8

x=2, y=28

z=27일 때 x=28, y=1

x=7, y=21

z=29일 때 x=12, y=14

$z=31$ 일 때  $x=17, y=7$   
 $z=33$ 일 때  $x=1, y=20$   
 $z=35$ 일 때  $x=6, y=6$   
 $z=37$ 일 때  $x=11, y=5$   
 $z=39$ 일 때 존재하지 않는다  
 $z=41$ 일 때 존재하지 않는다  
 $z=43$ 일 때  $x=5, y=5$   
 $z=45$ 일 때 존재하지 않는다  
 $z=47$ 일 때 존재하지 않는다  
 $z=49$ 일 때 존재하지 않는다  
 $z=51$ 일 때 존재하지 않는다

∴ 총 개수는 36개

∴ 36개

**(조합적 계산을 이용하는 방법)**  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이다

∴  $x+y+z=a, y+z=b, z=c$ 라 하자

∴  $40a+2b+5c=2431$ 이다

이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다

∴  $a \geq 3, b \geq 2, c \geq 1$ 이므로  $a_1=a-3, b_1=b-2, c_1=c-1$ 이라 하자

$40a+2b+5c=40a_1+40+2b_1+2+5c_1+5=40a_1+2b_1+5c_1+47=2431$

∴  $40a_1+2b_1+5c_1=2384(a_1 \geq b_1 \geq c_1 \geq 0)$

∴  $a_1$ 의 값은 51이나 52,  $\dots, 57, 58, 59$ 이 나올 수 있다( $\because a_1$ 이 51보다 작으면  $2b_1+5c_1$ 는 384이상인데 그러면  $b_1$ 이나  $c_1$ 중 하나는 52이상이고,  $a_1$ 이 59보다 크면  $40a_1$ 이 2384보다 크다)

$a_1=51$ 일 때  $2b_1+5c_1=344$ 인 정수해  $b_1, c_1$ 은 존재하지 않는다

$a_1=52$ 일 때  $2b_1+5c_1=304$ 이므로  $b_1=47, c_1=42$

$a_1=53$ 일 때  $2b_1+5c_1=264$ 이므로  $b_1=42, c_1=36$

$$b_1=47, c_1=34$$

$$b_1=52, c_1=32$$

$a_1=54$ 일 때  $2b_1+5c_1=224$ 이므로  $b_1=37, c_1=30$

$$b_1=42, c_1=28$$

$$b_1=47, c_1=26$$

$$b_1=52, c_1=24$$

$a_1=55$ 일 때  $2b_1+5c_1=184$ 이므로  $b_1=27, c_1=26$

$$b_1=32, c_1=24$$

$$b_1=37, c_1=22$$

$$b_1=42, c_1=20$$

$$b_1=47, c_1=18$$

$$b_1=52, c_1=16$$

$a_1=56$ 일 때  $2b_1+5c_1=144$ 이므로  $b_1=22, c_1=20$

$$b_1=27, c_1=18$$

$$b_1=32, c_1=16$$

$$b_1=37, c_1=14$$

$$b_1=42, c_1=12$$

$$b_1=47, c_1=10$$

$$b_1=52, c_1=8$$

$a_1=57$ 일 때  $2b_1+5c_1=104$ 이므로  $b_1=17, c_1=14$

$$b_1=22, c_1=12$$

$$b_1=27, c_1=10$$

$$b_1=32, c_1=8$$

$$b_1=37, c_1=6$$

$$b_1=42, c_1=4$$

$$b_1=47, c_1=2$$

$$b_1=52, c_1=0$$

$$a_1=58\text{일 때 } 2b_1+5c_1=64\text{이므로 } b_1=12, c_1=8$$

$$b_1=17, c_1=6$$

$$b_1=22, c_1=4$$

$$b_1=27, c_1=2$$

$$b_1=32, c_1=0$$

$$a_1=59\text{일 때 } 2b_1+5c_1=24\text{이므로 } b_1=7, c_1=2$$

$$b_1=12, c_1=0$$

∴ 총 개수는  $0+1+3+4+6+7+8+5+2=36$ 개

∴ 36개

**(해석)** 부정방정식의 미지수가 많으며 각 계수의 차이가 1보다 크고 자연수 순서쌍의 개수를 구하는 문제이다. 이 문제를 합동식으로 풀 때,  $40x+42y+47z \equiv 5z-2x \equiv 2431 \equiv 37 \pmod{42}$ 이므로  $5z-2x=42k+37$ (단,  $k$ 는 정수)라 둔 후,  $z$ 의 범위,  $x$ 의 범위,  $y$ 의 범위 모두 알고 있으므로  $z$ 의 범위에 따라  $x, y$ 의 값을 구할 수 있는데 이 부정방정식의  $z$ 의 범위가 1이상 51이하인 홀수(왜냐하면 만약  $z$ 가 짝수면 좌변은  $40x, 42y, 47z$ 가 모두 짝수가 되어  $40x+42y+47z$ 는 짝수인데 2431은 홀수이기 때문이다)이므로 나눠야 할 케이스는 총 26가지이다. 그런데 이 경우는 너무 많으므로 계수적 조합론을 이용하는 방법을 적용했다.  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$  이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+2b+5c=2431$ 이다. 이때,  $x>0, y>0, z>0$ 이므로  $a>b>c>0$ 이다.  $a \geq 3, b \geq 2, c \geq 1$ 이므로  $a_1=a-3, b_1=b-2, c_1=c-1$ 이라 하자.  $a$ 의 값은 51이나 52,  $\dots, 57, 58, 59$ 가 나오기 때문에 나눠야 할 케이스는 총 9가지이다.  $a$ 의 범위를 좁힐 수 있어 범위를 줄여 자연수 순서쌍( $a, b, c$ )의 개수를 구할 수 있다. 이때, 자연수 순서쌍( $a, b, c$ )의 개수는 자연수 순서쌍( $x, y, z$ )의 개수가 같으므로 자연수 순서쌍( $x, y, z$ )의 개수를 합동식을 이용하는 것보다 나눠야 할 케이스를 줄이는 효과가 있다. 연구결과, 일차부정방정식에서 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

**<예제3-6>**  $40x+42y+47z=2431$ 일 때 음이 아닌 정수 순서쌍 ( $x, y, z$ )의 개수를 구하는 과정에서 합동식으로 푸는 방법과 조합적 계산으로 푸는 방법 중 어느 것이 더 단순한가?

**(합동식을 이용하는 방법)**  $40x+42y+47z \equiv 5z-2x \equiv 2431 \equiv 37 \pmod{42}$ 이므로

$$5z-2x \equiv 37 \pmod{42}\text{이다}$$

∴  $5z-2x=42k+37$ 이라 하자(단,  $k$ 는 정수)

$$\therefore 5z=2x+42k+37$$

이때,  $z$ 의 범위는 0이상 51이하( $z$ 는 짝수이면 않된다. 왜냐하면 만약  $z$ 가 짝수면 좌변은  $40x, 42y, 47z$ 가 모두 짝수가 되어  $40x+42y+47z$ 는 짝수인데 2431은 홀수이기 때문이다)이고  $x$ 의 범위는 0이상 60이하이고  $y$ 의 범위는 0이상 57이하이므로

$$z=1\text{일 때 } x=47, y=12$$

$$x=26, y=32$$

$$x=5, y=52$$

$$z=3\text{일 때 } x=52, y=5$$

$$x=31, y=25$$

$$x=10, y=45$$

$$z=5\text{일 때 } x=36, y=18$$

$x=15, y=38$   
 $z=7$ 일 때  $x=41, y=11$   
 $x=20, y=31$   
 $z=9$ 일 때  $x=46, y=4$   
 $x=25, y=24$   
 $x=4, y=44$   
 $z=11$ 일 때  $x=30, y=17$   
 $x=9, y=37$   
 $z=13$ 일 때  $x=35, y=10$   
 $x=14, y=30$   
 $z=15$ 일 때  $x=40, y=3$   
 $x=19, y=23$   
 $z=17$ 일 때  $x=24, y=16$   
 $x=3, y=36$   
 $z=19$ 일 때  $x=29, y=9$   
 $x=8, y=29$   
 $z=21$ 일 때  $x=34, y=2$   
 $x=13, y=22$   
 $z=23$ 일 때  $x=18, y=15$   
 $z=25$ 일 때  $x=23, y=8$   
 $x=2, y=28$   
 $z=27$ 일 때  $x=28, y=1$   
 $x=7, y=21$   
 $z=29$ 일 때  $x=12, y=14$   
 $z=31$ 일 때  $x=17, y=7$   
 $z=33$ 일 때  $x=22, y=0$   
 $x=1, y=20$   
 $z=35$ 일 때  $x=6, y=6$   
 $z=37$ 일 때  $x=11, y=5$   
 $z=39$ 일 때 존재하지 않는다  
 $z=41$ 일 때  $x=0, y=12$   
 $z=43$ 일 때  $x=5, y=5$   
 $z=45$ 일 때 존재하지 않는다  
 $z=47$ 일 때 존재하지 않는다  
 $z=49$ 일 때 존재하지 않는다  
 $z=51$ 일 때 존재하지 않는다

∴ 총 개수는 38개

∴ 38개

**(조합적 계산을 이용하는 방법)**  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이다

∴  $x+y+z=a, y+z=b, z=c$ 라 하자

∴  $40a+2b+5c=2431$ 이다

이때,  $x \geq 0, y \geq 0, z \geq 0$ 이므로  $a \geq b \geq c \geq 0$ 이다

∴  $a$ 의 값은 52이나 53,  $\dots, 57, 58, 59, 60$ 이 나올 수 있다(∵  $a$ 이 52보다 작으면  $2b+5c$ 는 391이상인데 그러면  $b$ 이나  $c$ 중 하나는 52이상이고,  $a$ 이 60보다 크면  $40a$ 이 2431보다 크다)

$a=52$ 일 때  $2b+5c=351$ 인 정수해  $b, c$ 가 존재하지 않는다

$a=53$ 일 때  $2b+5c=311$ 이므로  $b=48, c=43$

$$b=53, c=41$$

$a=54$ 일 때  $2b+5c=271$ 이므로  $b=43, c=37$

$$b=48, c=35$$

$$b=53, c=33$$

$a=55$ 일 때  $2b+5c=231$ 이므로  $b=33, c=33$

b=38, c=31  
 b=43, c=29  
 b=48, c=27  
 b=53, c=25

a=56일 때  $2b+5c=191$ 이므로 b=28, c=27

b=33, c=25  
 b=38, c=23  
 b=43, c=21  
 b=48, c=19  
 b=53, c=17

a=57일 때  $2b+5c=151$ 이므로 b=23, c=21

b=28, c=19  
 b=33, c=17  
 b=38, c=15  
 b=43, c=13  
 b=48, c=11  
 b=53, c=9

a=58일 때  $2b+5c=111$ 이므로 b=18, c=15

b=23, c=13  
 b=28, c=11  
 b=33, c=9  
 b=38, c=7  
 b=43, c=5  
 b=48, c=3  
 b=53, c=1

a=59일 때  $2b+5c=71$ 이므로 b=13, c=9

b=18, c=7  
 b=23, c=5  
 b=28, c=3  
 b=33, c=1

a=60일 때  $2b+5c=31$ 이므로 b=8, c=3

b=13, c=1

∴ a=52일 때 0개

a=53일 때 2개

a=54일 때 3개

a=55일 때 5개

a=56일 때 6개

a=57일 때 7개

a=58일 때 8개

a=59일 때 5개

a=60일 때 2개

∴ 총 개수는  $0+2+3+5+6+7+8+5+2=38$

∴ 38개

**(해석)** 부정방정식의 미지수가 많으며 각 계수의 차이가 1보다 크고 음이 아닌 정수 순서쌍의 개수를 구하는 문제이다. 이 문제를 합동식으로 풀 때,  $40x+42y+47z \equiv 5z-2x \equiv 2431 \equiv 37 \pmod{42}$ 이므로  $5z-2x=42k+37$ (단, k는 정수)라 둔 후, z의 범위, x의 범위, y의 범위 모두 알고 있으므로 z의 범위에 따라 x, y의 값을 구할 수 있는데 이 부정방정식의 z의 범위가 0이상 51이하인 홀수(왜냐하면 만약 z가 짝수면 좌변은  $40x, 42y, 47z$ 가 모두 짝수가 되어  $40x+42y+47z$ 는 짝수인데 2431은 홀수이기 때문이다)이므로 나눠야 할 케이스는 총 26가지이다. 그런데 이 경우는 너무 많으므로 계수적 조합론을 이용하는 방법을 적용했다.  $40x+42y+47z=40(x+y+z)+2(y+z)+5z=2431$ 이므로  $x+y+z=a, y+z=b, z=c$ 라 하면  $40a+2b+5c=2431$ 이다. 이때,  $x \geq 0, y \geq 0, z \geq 0$ 이므로  $a \geq b \geq c \geq 0$ 이다. a의 값은 52이나 53, ..., 57, 58, 59, 60이 나오기 때문에 나눠야 할 케이스는 총 9가지이다. a의 범위를 좁힐 수 있어 범위를

줄여 음이 아닌 정수 순서쌍(a, b, c)의 개수를 구할 수 있다. 이때, 음이 아닌 정수 순서쌍(a, b, c)의 개수는 음이 아닌 정수 순서쌍(x, y, z)의 개수가 같으므로 음이 아닌 정수 순서쌍(x, y, z)의 개수를 **합동식을 이용하는 것보다 나뉘어야 할 케이스를 줄이는 효과가 있다.** 연구결과, 일차부정방정식에서 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

**[부록VI-8] ① 동일 차수 2이상, 각 계수의 차이가1이상인 경우 음이 아닌 정수의 순서쌍의 개수**

<예제4-1>  $40k^n + 42l^n + 47m^n = 2431$ 일 때 음이 아닌 정수의 순서쌍 (k, l, m)의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로도 풀 수 있을까?(단,  $n \geq 2$ )

**(조합적 계산을 이용하는 방법)**

$k^n = x, l^n = y, m^n = z$ 로 치환하면  $40x + 42y + 47z = 2431$ 이 되므로 <예제3-6>의 과정과 동일하다.

**(해석)** 동일 차수 부정방정식은  $k^n = x, l^n = y, m^n = z$ 로 치환하면 일차부정방정식 꼴이 되기 때문에 풀이 과정은 <예제 3-6>과 동일하다. 연구결과, 동일 차수 부정방정식은  $k^n = x, l^n = y, m^n = z$ 로 치환하면, 일차부정방정식으로 바꿔 놓고 풀 수 있으므로 계수의 조건, 미지수의 조건과 무관하게 연구문제1에서 고안한 알고리즘을 적용할 수 있음을 확인했다.

**[부록VI-9] ① 각 차수의 차이가 있는 경우 음이 아닌 정수의 순서쌍의 개수**

<예제5-1>  $40k^3 + 42l + 47m^2 = 2431$ 일 때 음이 아닌 정수의 순서쌍 (k, l, m)의 개수를 구하는 과정과  $40k^n + 42l^n + 47m^n = 2431$ 일 때 음이 아닌 정수의 순서쌍 (k, l, m)의 개수를 구하는 과정에서 합동식으로 접근하지 않고 조합적 계산으로 접근할 때 그 방법은 동일할까?

**(조합적 계산을 이용하는 방법)**

$40k^3 + 42l + 47m^2 = 2431$ 의 순서쌍의 개수를 구하자. 이때,  $k^3 = x, l = y, m^2 = z$ 로 치환하면,  $40x + 42y + 47z = 2431$ 이 되므로 <예제3-6>의 과정과 동일하다.

**(해석)** 이때, (a, b, c)의 순서쌍을 모두 써보면 (55, 49, 43), (56, 44, 37), (56, 49, 35), (56, 54, 33), (57, 39, 31), (57, 44, 29), ... (62, 9, 3), (62, 14, 1)이다. 이때,  $k^3$ 이 55, 56, 57, 58, ..., 62가 될 수 있는데 이 중 세제공수는 존재하지 않으므로 처음에 구하려고 했던 방정식의 해는 존재하지 않는다. (k, l, m)의 순서쌍은 임의의 소수의 제곱꼴이어야 한다. 하지만 이 경우들에서는 존재하지 않는다. 연구결과, n이 2차로 된 정수 계수 부정방정식인 경우 계수를 제외한 각 항을 완전제곱식 형태로 치환하면 일차부정방정식 혹은 우변이 0이 아닌 상수꼴로 인수분해가 가능하기 때문에 연구문제1에서 고안한 알고리즘을 적용할 수 있지만, n이 3차 이상인 경우 3차 이상의 합동식은 다루기 힘들고 계수적 조합론 방법은 일반적으로 정수 계수로 인수분해가 가능하지 않기 때문에 연구문제1에서 고안한 알고리즘을 적용할 수 없음을 확인했다.

# 수학B

01. c-v, c-s, c-f 연결체를 이용한 공간 테셀레이션 .....	183
02. 페이지랭크(Pagerank) 이해 및 활용 .....	211
03. Conway의 생명 게임 탐구 .....	225
04. 트리플카의 연구 사사수학 논문 .....	235
05. 펠 방정식의 일반화 .....	265
06. 쥐와 고양이 게임의 기하학 .....	285
07. 수의 패턴이 그리는 그림 수 바람개비의 특징 연구 .....	305
08. 최소 힌트의 개수를 갖는 변형스도쿠 만들기 .....	319
09. 진법탐구 .....	335
10. 물체의 회전관성으로 유도한 Fibonacci 수열들의 합 .....	345
11. 함수와 수학적 모델링 .....	359
12. 자연수 n의 m-regular composition 개수와 m-bonacci 수열 .....	379



수학B

01\_c-v, c-s, c-f 연결체를 이용한 공간  
테셀레이션



# c-v, c-s, c-f 연결체를 이용한 공간 테셀레이션 탐구

이시훈(세광중학교 3학년)

이찬혁(대제중학교 3학년)

최유담(솔밭중학교 3학년)

지도교수 : 한대회(청주교육대학교 수학교육학과)

## 초록(300자-500자)

도형의 중심과 꼭짓점을 연결하여 만든 도형인 중심연결체에 대해 탐구한 연구를 보았다. 격자도형에서 도형의 중심과 꼭짓점을 연결하면 공간테셀레이션이 만들어진다는 것을 알 수 있었다. 정사면체의 중심연결체는 다이아몬드의 탄소구조와 같다는 점과 정육면체의 중심연결체는 마름모십이면체라는 점에서 흥미를 느껴 이와 관련한 탐구를 진행하게 되었다. 중심을 꼭짓점 외에도 모서리의 수선의 발이나 면의 수선의 발에 이었을 때 또한 공간테셀레이션이 만들어질 수 있음을 알게 되었고 이를 바탕으로 중심연결체의 개념을 확장하였다. 이 공간테셀레이션을 격자도형으로 생각할 때 도형수와 관련있음을 깨달았다. 또한 몇몇의 테셀레이션이 듀얼관계와 연관있음을 발견할 수 있었다.

## I. 서론(또는 연구의 필요성 및 목적)

2021, 2022학년도 청주교육대학교부설 과학영재교육원에서 진행된 선행 연구를 통해 공간 테셀레이션이 가능한 다면체에 관심을 가지게 되었다. 김예주·이연서·전설(2021)의 연구를 통해 정사면체의 중심과 꼭짓점을 이어 만든 도형이 다이아몬드 구조이라는 사실과 이 도형에서 만들어지는 케이지나 윈도우의 기하학적 성질을 알 수 있었다. 또한 고영원(2022)의 연구를 통해 정육면체의 중심과 꼭짓점을 이어 만든 도형을 변형하면 마름모12면체가 나오게 되고 이는 공간 테셀레이션이 가능하다는 것을 알 수 있었다. 그렇다면 중심을 꼭짓점에 잇는 것에서 나아가 모서리와 면의 수선의 발에 내리면 어떤 도형이 나올지에 대해 의문을 가지며 공간 테셀레이션이 가능할 것이라는 추측을 하였다. 그래서 정사면체, 정육면체, 정팔면체의 중심연결체를 모형으로 만들어 도형의 성질을 관찰해보고 길이, 넓이, 부피 등을 계산해보고자 한다.

## II. 이론적 배경

### II-1. 테셀레이션

테셀레이션은 한 종류 또는 여러 종류의 도형을 활용해 평면이나 공간을 빈틈없이 채우는 것을 말한다.

## II-2. 도형수

도형수는 점들을 특정한 기하학적인 모양으로 배열할 때 점의 수를 의미한다.

### II-2-1. 삼각수

삼각수는 점들을 삼각형 모양으로 배열할 때, 점의 수를 의미한다.

n번째 삼각수는 1~n까지 자연수의 합이다.

$$n\text{번째 삼각수} = t_n = 1+2+3+\cdots+n = \sum_{k=1}^n k = \frac{n(n+1)}{2}$$

### II-2-2. 사각수

사각수는 점들을 사각형 모양으로 배열할 때, 점의 수를 의미한다.

n번째 사각수는 1~n번째 홀수들의 합이고, n번째 제곱수이다.

$$n\text{번째 사각수} = r_n = 1+3+5+\cdots+(2n-1) = \sum_{k=1}^n (2k-1) = n^2$$

### II-2-3. 삼각뿔수

삼각뿔수는 점들을 삼각뿔 모양으로 배열할 때, 점의 수를 의미한다.

n번째 삼각뿔수는 1~n번째 삼각수들의 합이다.

$$n\text{번째 삼각뿔수} = T_n = \sum_{k=1}^n t_k = \sum_{k=1}^n \frac{k(k+1)}{2} = \frac{n(n+1)(n+2)}{6}$$

### II-2-4. 사각뿔수

사각뿔수는 점들을 사각뿔 모양으로 배열할 때, 점의 수를 의미한다.

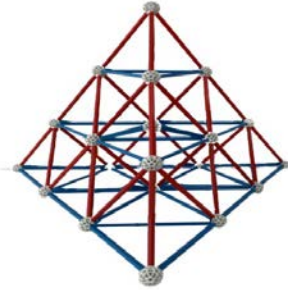
n번째 사각뿔수는 1~n번째 사각수들의 합이다.

$$n\text{번째 사각뿔수} = R_n = \sum_{k=1}^n r_k = \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

위 도형수들은 다음과 같은 특징이 있다.

1.  $t_n + t_{n-1} = r_n$
2.  $T_{n-1} + t_n = T_n$
3.  $T_n + T_{n-1} = R_n$

### II-3. n-격자 정사면체의 성질



3번째 격자 정사면체

n번째 격자 정사면체에 있는 구성요소의 개수를 구하면 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
꼭짓점	1+3	1+3+6	1+3+6+10	1+3+6+10+15	...	$T_{n+1}$
선분	6	6(1+3)	6(1+3+6)	6(1+3+6+10)	...	$6T_n$
면	4	4(1+3) +4	4(1+3+5) +4(1+3)	4(1+3+5+7) +4(1+3+5)	...	$4T_n + 4T_{n-1} = 4R_n$
정사면체	1	1+3	(1+3+6)+ 1	(1+3+6+10) +(1+3)	...	$T_{n-2} + T_n$
정팔면체	0	1	1+3	1+3+6	...	$T_{n-1}$

### III. 연구방법

#### III-1. 중심연결체를 이용한 공간 테셀레이션

김예주·이연서·전설(2021)의 연구와 고영원(2022)의 연구를 보완하고, 중심연결체의 개념을 확장하여 공간 테셀레이션을 이루는 찾아내었다.

#### III-2. 용어 정의

① 중심 : 정다각형과 정다면체의 중심

→ 정다면체는 중심에서 꼭짓점까지의 거리, 변에 내린 수선의 발의 길이, 면에 내

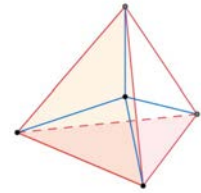
린 수선의 발의 길이가 모두 각각 일정하다.

② 격자 도형 : 도형으로 평면이나 공간을 빈틈없이 채운 도형

③ c-v 연결체(중심-꼭짓점 연결체/center-vertex)

격자 도형의 중심과 격자 도형의 꼭짓점을 모두 연결했을 때 만들어지는 도형

→0차원(중심)과 0차원(꼭짓점)을 연결한다고 생각할 수 있다.



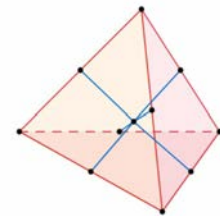
단위 정사면체의  
c-v 연결체

④ c-s 연결체(중심-선 연결체/center-segment)

격자 도형의 중심과 격자 도형의 중심에서 격자 도형의 모서리(변)에 내린 수선의 발을 모두 연결했을 때 만들어지는 도형

→0차원(중심)과 1차원(모서리/변)을 연결한다고 생각할 수 있다.

→중심에서 정다각형의 변과 정다면체의 모서리에 내린 수선의 발은 그 변과 모서리의 중점이다.



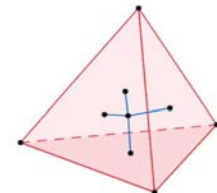
단위 정사면체의  
c-s 연결체

⑤ c-f 연결체(중심-면 연결체/center-face)

격자 도형의 중심과 격자 도형의 중심에서 격자 도형의 면에 내린 수선의 발을 모두 연결했을 때 만들어지는 도형

→0차원(중심)과 2차원(면)을 연결한다고 생각할 수 있다.

→중심에서 정다면체의 면에 내린 수선의 발은 그 면의 중심이다.



단위 정사면체의  
c-f 연결체

⑥ 중심연결체

c-v 연결체, c-s 연결체, c-f 연결체를 모두 포함하는 개념

⑦ 케이지

중심연결체에서 규칙적으로 만들어지는 도형 중 정사면체, 정팔면체, (정사각뿔 두 개를 합친)팔면체 등과 같은 기존의 수학 용어 명칭을 가지고 있지 않는 도형

⑧ 원도우

하나의 정다면체와 함께 케이지를 이루는 도형

→ 케이지이 밖에 노출되는 면이 있어야 한다.

#### IV. 연구 결과

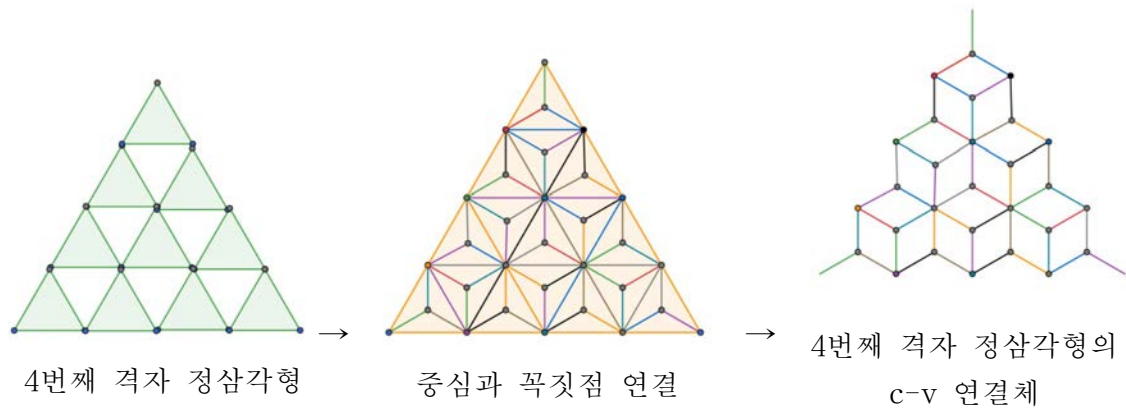
##### IV-1. 평면 중심연결체

2차원 도형은 0차원과 1차원으로 구성되어 있어 c-v 연결체와 c-s 연결체가 존재

한다. 2차원 도형에서는 그 2차원 평면 외의 다른 평면이 존재하지 않기 때문에 중심과 면을 연결하는 c-f 연결체는 존재하지 않는다. 만약 2차원 도형에서 중심과 그 2차원 평면과 연결한다면 점이 될 것이다.

#### IV-1-1. 격자 정삼각형의 중심연결체

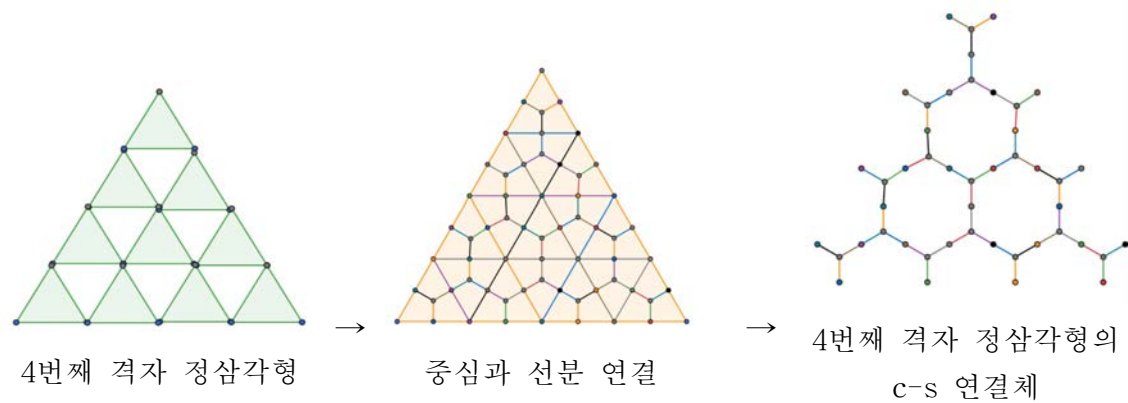
##### ① c-v 연결체



n번째 격자 정삼각형의 c-v 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
꼭짓점	1+3	(1+3+5)+1	(1+3+5+7)+(1+2)	(1+3+5+7+9)+(1+2+3)	...	$r_{n+1} + t_{n-1}$
선분	3	3(1+3)	3(1+3+5)	3(1+3+5+7)	...	$3r_n$
마름모	0	3	3(1+2)	3(1+2+3)	...	$3t_{n-1}$

##### ② c-s 연결체

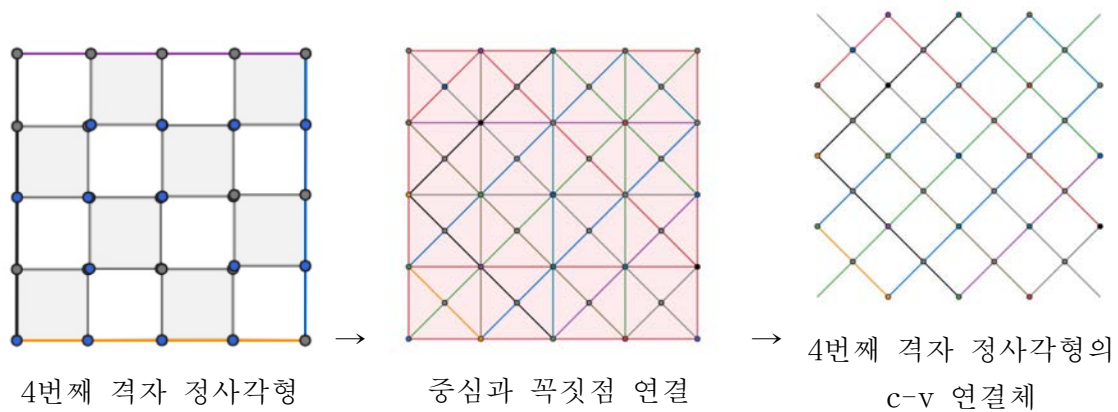


n번째 격자 정삼각형의 c-s 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
꼭짓점	4	$4(1+2)+1$	$4(1+2+3)+(1+2)$	$4(1+2+3+4)+(1+2+3)$	...	$4t_n + t_{n-1}$
선분	3	$3(1+3)$	$3(1+3+5)$	$3(1+3+5+7)$	...	$3r_n$
정육각형	0	0	1	$1+2$	...	$t_{n-2}$

#### IV-1-2. 격자 정사각형의 중심연결체

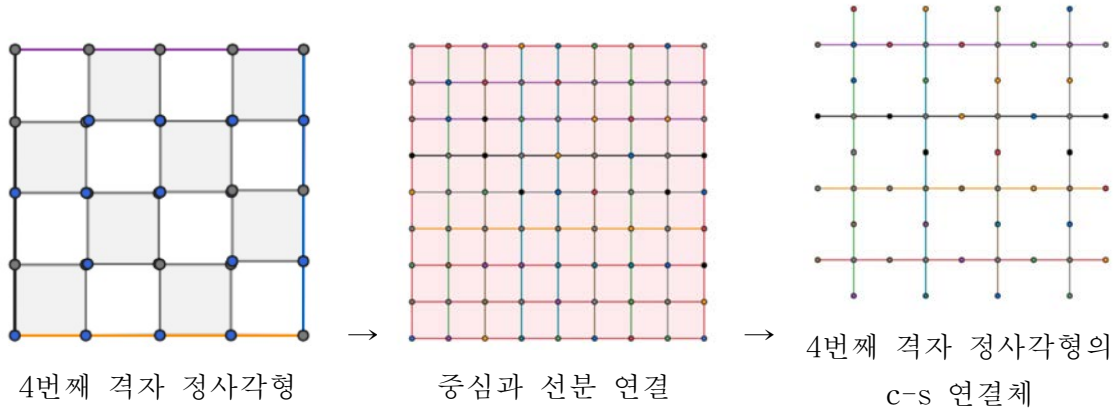
##### ① c-v 연결체



n번째 격자 정사각형의 c-v 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
꼭짓점	$(1+3)+1$	$(1+3+5)+(3+1)$	$(1+3+5+7)+(5+3+1)$	$(1+3+5+7+9)+(7+5+3+1)$	...	$r_{n+1} + r_n$
선분	4	$4(1+3)$	$4(1+3+5)$	$4(1+3+5+7)$	...	$4r_n$
정사각형	0	4	$4(1+2)$	$4(1+2+3)$	...	$4t_{n-1}$

##### ② c-s 연결체



n번째 격자 정사각형의 c-s 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
꼭짓점	2+3	2×2+ 3(1+3)	2×3+ 3(1+3+5)	2×4+ 3(1+3+5+7)	...	$2n + 3r_n$
선분	4	4(1+3)	4(1+3+5)	4(1+3+5+7)	...	$4r_n$
정사각형	0	1	1+3	1+3+5	...	$r_{n-1}$

#### IV-1-3. 격자 정육각형의 중심연결체

##### ① c-v 연결체



n번째 격자 정육각형의 c-v 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
꼭짓점	$3(1+2)$ -2	$3(1+2+3)$ -2	$3(1+2+3+4)$ -2	$3(1+2+3+4+5)$ -2	...	$3t_{n+1} - 2$
선분	6	$6(1+2)$	$6(1+2+3)$	$6(1+2+3+4)$	...	$6t_n$
마름모	0	3	$3(1+2)$	$3(1+2+3)$	...	$3t_{n-1}$

② c-s 연결체



n번째 격자 정육각형의 c-s 연결체에 있는 구성요소의 개수는 다음과 같다.

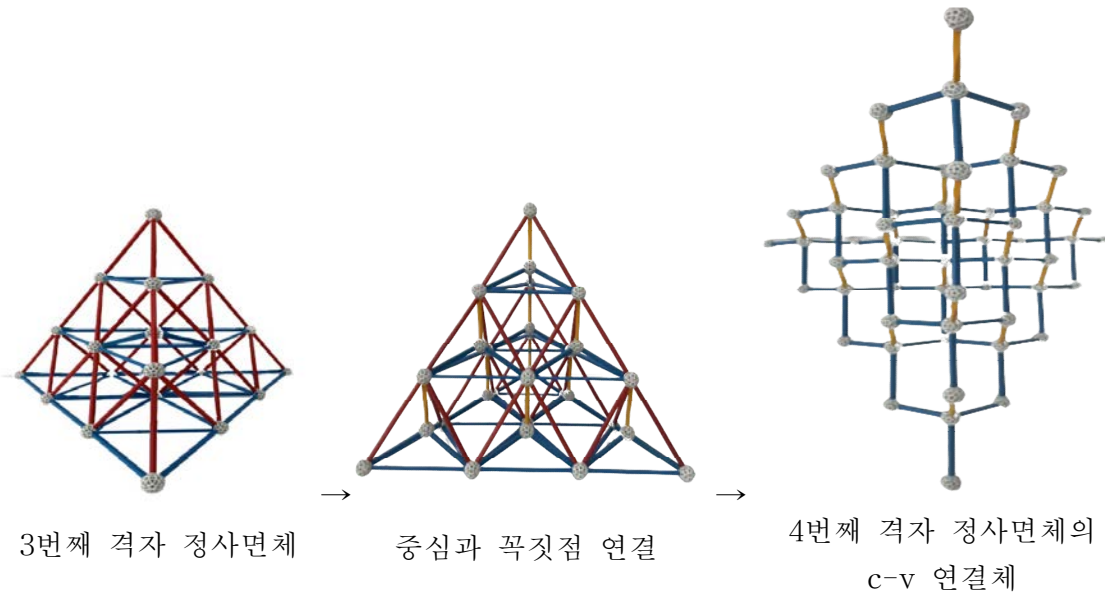
개수 \ n번째	1	2	3	4	...	n
꼭짓점	6+1	$6 \times 2 +$ $(1+2+3)$	$6 \times 3 +$ $(1+2+3+4+5)$	$6 \times 4 +$ $(1+2+3+4+5+6+7)$	...	$6n + t_{2n-1}$
선분	6	$6(1+2)$	$6(1+2+3)$	$6(1+2+3+4)$	...	$6t_n$
정삼각형	0	1	1+3	1+3+5	...	$r_{n-1}$

IV-2. 공간 중심연결체

3차원 도형은 0차원, 1차원, 2차원으로 구성되어 있어 c-v 연결체, c-s 연결체, c-f 연결체가 존재한다.

IV-2-1. 격자 정사면체의 중심연결체

① c-v 연결체



n번째 격자 정사면체의 c-v 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
중심	1	1+3	1+3+6	1+3+6+10	...	$T_n$
꼭짓점	1+3+1	1+3+6+1	1+3+6+10+1	1+3+6+10+15+1	...	$T_{n+1} + T_n$
모서리	4	4(1+3)	4(1+3+6)	4(1+3+6+10)	...	$4T_n$
정삼각형	0	4	4(1+3)	4(1+3+6)	...	$4T_{n-1}$
이등변삼각형	0	12	12(1+3)-6	12(1+3+6)-6(1+3)	...	$12T_{n-1} - 6T_{n-2}$
케이지	0	1	1+3	1+3+6	...	$T_{n-1}$
윈도우	0	4	4(1+3)	4(1+3+6)	...	$4T_{n-1}$
역정사면체	0	0	1	1+3	...	$T_{n-2}$

n번째 격자 정사면체의 c-v 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 선분의 수)	한 면의 넓이 (그 면의 수)	부피	단위 정사면체 와의 부피비
정삼각형	1	$\frac{\sqrt{3}}{4}$		
이등변삼각형	1(1), $\frac{\sqrt{6}}{4}$ (2)	$\frac{\sqrt{2}}{8}$		
케이지	$\frac{\sqrt{6}}{4}$	$\frac{\sqrt{3}}{4}$ (4), $\frac{\sqrt{2}}{8}$ (12)	$\frac{5\sqrt{2}}{12}$	1:5
윈도우	$\frac{\sqrt{6}}{4}$	$\frac{\sqrt{3}}{4}$ (2), $\frac{\sqrt{2}}{8}$ (6)	$\frac{\sqrt{2}}{12}$	1:1

역정사면체	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{2}}{12}$	1:1
-------	---	----------------------	-----------------------	-----

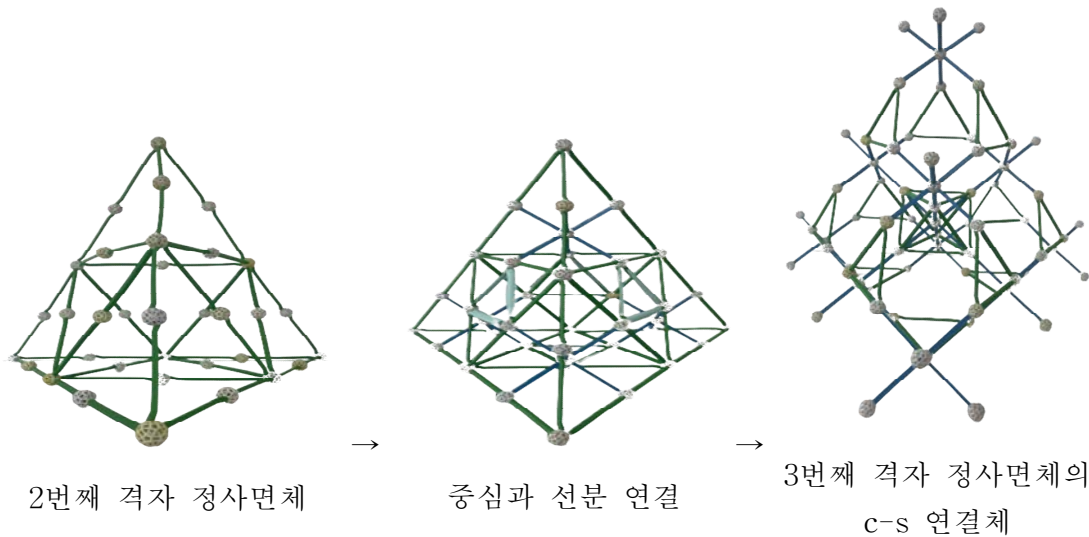
격자 정사면체의 c-v 연결체의 케이지는 정삼각형 4개와 이등변삼각형 12개로 이루어진 16면체이다. 정삼각형의 각 변에는 이등변삼각형이 붙어 있고, 이등변삼각형 세 개는 한 점에서 모이도록 만난다. 격자 정사면체의 c-v 연결체의 원도우는 정삼각형 2개와 이등변삼각형 6개로 이루어진 8면체이다. 케이지와 같이 정삼각형의 각 변에 이등변 삼각형이 붙어있다. 두 개의 정삼각형이 서로 마주보고 있다. 케이지는 단위 정사면체 크기의 정사면체에 네 개의 원도우를 정사면체의 각면체에 하나씩 붙여 만들 수 있다. 케이지는 변끼리 이웃하지 않은 정팔면체의 네 면에 삼각뿔을 붙여 놓은 형태라고도 할 수 있다.

격자 정사면체의 c-v 연결체의 케이지와 정사면체(역정사면체)로 공간 테셀레이션이 가능하다는 것을 확인할 수 있었다.

격자 정사면체에서 정팔면체가 있던 자리에 케이지가 생기게 된다.

정사면체 c-v 연결체는 김예주·이연서·전설(2021)의 연구와 고영원(2022)의 연구에 선행연구가 많이 진행된 도형이다. 이 도형은 다이아몬드 구조와 같다. 중심에 탄소가 있고 그 주위에 탄소 4개가 단일 공유 결합을 하고 있는 상태이다.

② c-s 연결체

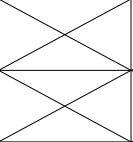
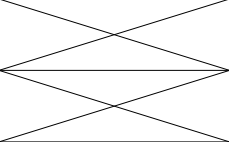
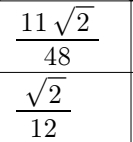
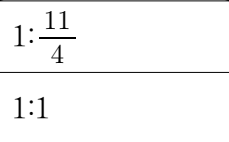


n번째 격자 정사면체의 c-s 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 <del>n번째</del>	1	2	3	4	...	n
중심	1	1+3	1+3+6	1+3+6+10	...	$T_n$
꼭짓점	7	7(1+3)	7(1+3+6)	7(1+3+6+10)	...	$7T_n$

모서리	6	$6(1+3)+12$	$6(1+3+6)+12(1+3)$	$6(1+3+6+10)+12(1+3+6)$	...	$6T_n + 12T_{n-1}$
정삼각형	0	4	$4(1+3)+4$	$4(1+3+6)+4(1+3)$	...	$4T_{n-1} + 4T_{n-2}$
육각형	0	6	$6(1+3)$	$6(1+3+6)$	...	$6T_{n-1}$
케이지	0	1	$1+3$	$1+3+6+1$	...	$T_{n-1} + T_{n-3}$
삼각뿔대	0	4	$4(1+3)$	$4(1+3+6+1)$	...	$4(T_{n-1} + T_{n-3})$
정팔면체	0	0	1	$1+3$	...	$T_{n-2}$

n번째 격자 정사면체의 c-s 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 선분의 수)	한 면의 넓이 (그 면의 수)	부피	단위 정사면체 와의 부피비
정삼각형	$\frac{1}{2}$	$\frac{\sqrt{3}}{16}$		
육각형	$\frac{1}{2}(2), \frac{\sqrt{2}}{4}(4)$	$\frac{3}{8}$		
케이지	$\frac{1}{2}(12), \frac{\sqrt{2}}{4}(12)$	$\frac{\sqrt{3}}{16}(4), \frac{3}{8}(6)$	$\frac{11\sqrt{2}}{48}$	$1:\frac{11}{4}$
정사면체	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{2}}{12}$	1:1
윈도우 (삼각뿔대)	$\frac{1}{2}(3), \frac{\sqrt{2}}{4}(3), 1(3)$	$\frac{\sqrt{3}}{16}(1), \frac{3}{16}(3), \frac{\sqrt{3}}{4}(1)$	$\frac{7\sqrt{2}}{192}$	$1:\frac{7}{16}$
정팔면체	$\frac{1}{2}$	$\frac{\sqrt{3}}{16}$	$\frac{\sqrt{2}}{24}$	$1:\frac{1}{2}$

격자 정사면체의 c-s 연결체의 케이지는 6개의 육각형과 4개의 정삼각형으로 이루어져 있다. 정삼각형의 각 변에는 육각형이 붙어있다. 그리고 3개의 육각형이 한 점에서 만난다.

케이지 안에 정사면체가 있다고 생각하면 하나의 케이지 안에 윗면과 밑면이 정삼각형인 4개의 삼각뿔대가 생긴다. 이는 격자 정사면체의 c-s 연결체의 윈도우이다.

격자 정사면체의 c-s 연결체의 케이지와 정팔면체로 공간 테셀레이션이 가능하다는 것을 알 수 있었다.

1단계에서 정팔면체가 있던 공간에 케이지가 생기고, 역정사면체가 있던 자리에는 정팔면체와 거꾸로된 케이지가 생긴다.

1단계에 있던 역정사면체의 중심까지 생각하여 중심연결체를 만들게 되면 정육면체의 테셀레이션이 되어버린다. 바깥 부분은 새로운 도형처럼 보이지만 바깥과 만나

지 않는 안쪽 부분은 정육면체들만 모이게 된다. 즉 새로운 테셀레이션의 배열을 찾기 위해서는 역정사면체의 중심을 생각하지 않아야 한다. 그래서 역정사면체의 중심을 생각하지 않고 위 결과들을 구하였다.

정리 I-1. 케이지의 부피는  $\frac{11\sqrt{2}}{48}$ 이다.

증명 시작) 여기서 생기는 케이지는 정삼각형 4개와 육각형 6개로 이루어진 10면체이다. 이 육각형은 정사각형의 이웃하지 않는 두 꼭짓점을 정사각형의 2분점을 기준으로 자른 도형이다. 케이지 또한 정육면체의 이웃하지 않은 네 꼭짓점을 정육면체의 2분점을 기준으로 자른 도형이다. 이 둘이 비슷한 성질을 갖는 이유는 케이지가 이 육각형 6개로 이루어져 있기 때문이다. 이를 통해 쉽게 케이지의 부피를 구할 수 있다. 정육면체와 케이지 사이의 부피비를 먼저 구하였다. 부피가 1인 정사면체에 삼각뿔을 붙여 만든 정육면체의 부피는 3이다. 이때 빼야하는 삼각뿔의 부피는  $3 \times \frac{1}{8} \times \frac{1}{6} = \frac{1}{16}$ 이다. 케이지의 부피는 정육면체에서 4개의 삼각뿔을 뺀 것이므로  $3 - 4 \times \frac{1}{16} = \frac{11}{4}$ 이다. 한 모서리의 길이가 1인 단위 정사면체의 부피는  $\frac{\sqrt{2}}{12}$ 이므로 실제 케이지의 부피는  $\frac{\sqrt{2}}{12} \times \frac{11}{4} = \frac{11\sqrt{2}}{48}$ 이다.

<증명 끝>

정리 I-2. 육팔면체의 부피는  $\frac{5\sqrt{2}}{24}$ 이다.

증명 시작) 이 케이지를 다르게 분석해 볼 수 있다. 육팔면체에서 꼭짓점끼리도 이웃하지 않은 정삼각형의 네 면에 삼각뿔을 붙인 도형으로도 볼 수 있다. 이를 통해 육팔면체의 부피 또한 구할 수 있다. 위의 부피비를 다시 적용하면 케이지의 부피는  $\frac{11}{4}$ , 삼각뿔의 부피는  $\frac{1}{16}$ 이다. 육팔면체의 부피는 케이지에서 삼각뿔 네 개를 뺀 것이므로  $\frac{11}{4} - 4 \times \frac{1}{16} = \frac{5}{2}$ 이다. 한 모서리의 길이가 1인 단위 정사면체의 부피는  $\frac{\sqrt{2}}{12}$ 이므로 실제 육팔면체의 부피는  $\frac{\sqrt{2}}{12} \times \frac{5}{2} = \frac{5\sqrt{2}}{24}$ 이다.

<증명 끝>

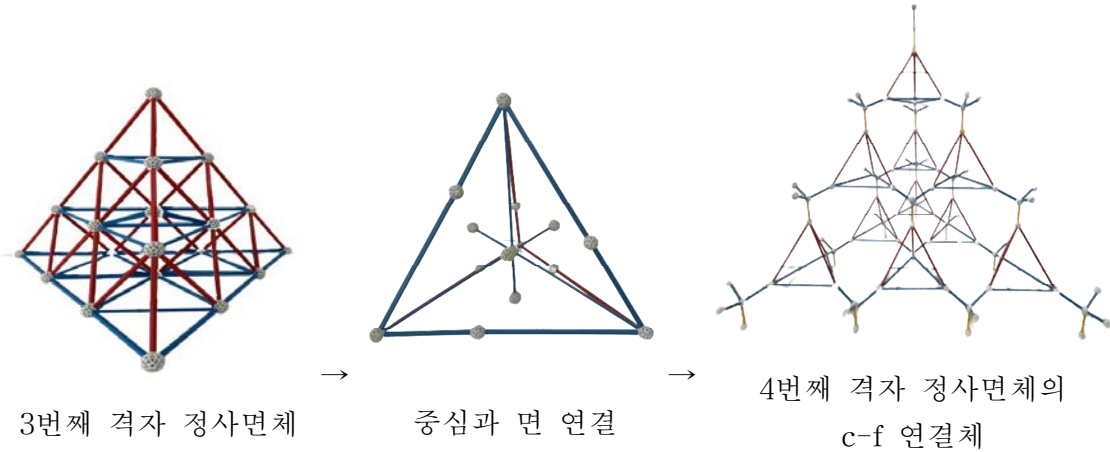
정리 I-3. 삼각뿔대의 부피는  $\frac{7\sqrt{2}}{192}$ 이다.

증명 시작) 마지막으로 이 케이지는 정사면체에 삼각뿔대가 붙은 모양이라고도 볼 수 있다. 이 역시 위의 부피비를 적용하면 케이지의 부피는  $\frac{11}{4}$ , 정사면체의 부피는 1이다. 삼각뿔대의 부피는 케이지에서 정사면체를 뺀 후 4로 나눈 값이므로

$(\frac{11}{4}-1) \div 4 = \frac{7}{16}$ 이다. 한 모서리의 길이가 1인 단위 정사면체의 부피는  $\frac{\sqrt{2}}{12}$ 이므로  
 실제 삼각뿔대의 부피는  $\frac{\sqrt{2}}{12} \times \frac{7}{16} = \frac{7\sqrt{2}}{192}$ 이다.

<증명 끝>

③ c-f 연결체



n번째 격자 정사면체의 c-f 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
중심	1	1+3	1+3+6	1+3+6+10	...	$T_n$
꼭짓점	5	5(1+3)	5(1+3+6)	5(1+3+6+10)	...	$5T_n$
모서리	4	4(1+3)+6	4(1+3+6)+6(1+3)	4(1+3+6+10)+6(1+3+6)	...	$4T_n + 6T_{n-1}$
작은 정삼각형	0	4	4(1+3)	4(1+3+6)	...	$4T_{n-1}$
큰 정삼각형	0	0	4	4(1+3)	...	$4T_{n-2}$
사다리꼴	0	0	12	12(1+3)-6	...	$12T_{n-2} - 6T_{n-3}$
작은 정사면체	0	1	1+3	1+3+6	...	$T_{n-1}$
케이지	0	0	1	1+3	...	$T_{n-2}$
정팔면체	0	0	1	1+3	...	$T_{n-2}$
윈도우 (삼각뿔대)	0	0	4	4(1+3)	...	$4 \times T_{n-2}$
큰 정사면체 (역정사면체)	0	0	0	1	...	$T_{n-3}$

n번째 격자 정사면체의 c-f 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 선분의 수)	한 면의 넓이 (그 면의 수)	부피	단위 정사면체 와의 부피비
--	-----------------	---------------------	----	-------------------

작은 정삼각형	$\frac{2}{3}$	$\frac{\sqrt{3}}{9}$		
큰 정삼각형	1	$\frac{\sqrt{3}}{4}$		
사다리꼴	$\frac{2}{3}(1), \frac{\sqrt{6}}{12}(2),$ 1(1)	$\frac{5\sqrt{2}}{36}$		
작은 정사면체	$\frac{2}{3}$	$\frac{\sqrt{3}}{9}$	$\frac{2\sqrt{2}}{81}$	1: $\frac{8}{27}$
케이지	$\frac{2}{3}(12), \frac{\sqrt{6}}{12}(12),$ 1(12)	$\frac{\sqrt{3}}{9}(4), \frac{5\sqrt{2}}{36}(12),$ $\frac{\sqrt{3}}{4}(4)$	$\frac{127\sqrt{2}}{324}$	1: $\frac{127}{27}$
정팔면체	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{2}}{3}$	1:4
윈도우 (삼각뿔대)	$\frac{2}{3}(3), \frac{\sqrt{6}}{12}(3),$ 1(3)	$\frac{\sqrt{3}}{9}(1), \frac{5\sqrt{2}}{36}(3),$ $\frac{\sqrt{3}}{4}(1)$	$\frac{19\sqrt{2}}{1296}$	1: $\frac{19}{108}$
큰 정사면체 (역정사면체)	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{2}}{12}$	1:1

격자 정사면체의 c-f 연결체의 케이지는 작은 정삼각형 4개, 사다리꼴 12개, 큰 정삼각형 4개로 이루어진 20면체이다.

격자 정사면체의 c-f 연결체의 케이지는 정팔면체 1개와 삼각뿔대 4개가 합쳐진 도형이다. 이 때 삼각뿔대는 격자 정사면체의 c-f 연결체의 윈도우라 할 수 있다.

작은 정사면체, 케이지, 큰 정사면체를 사용하여 공간 테셀레이션이 가능함을 알 수 있었다.

격자 정사면체에서 원래 정팔면체가 있던 자리에 작은 정사면체가 생기고, 원래 역정사면체가 있던 자리에 케이지가 생겼다. 만든 모형에는 큰 정사면체가 드러나 있지 않지만 있다고 생각해야 테셀레이션이 완성된다.

정리 I-4. 케이지의 부피는  $\frac{127\sqrt{2}}{324}$ 이다.

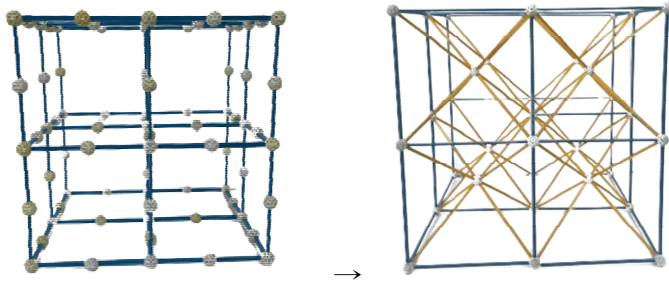
증명 시작) 이 중심연결체에서 생기는 케이지는 정팔면체의 변끼리 이웃하지 않은 네 면에 삼각뿔대를 붙인 도형이다. 케이지의 부피를 구하기 위해 먼저 격자 정사면체의 부피를 1로 보았을 때의 부피비를 구하였다. 이때 정팔면체의 부피는 4이다. 삼각뿔대는 정사면체를 4등분한 정삼각뿔의 3등분점을 잡아 그중 밑면에 가까운 세 점이 윗면의 꼭짓점이 되도록 자른 것이다. 따라서 삼각뿔대의 부피는

$1 \times \frac{1}{4} \times \frac{19}{27} = \frac{19}{108}$ 이다. 케이지의 부피는 정팔면체에 삼각뿔대 네 개를 붙인 것과 같으므로  $4 + 4 \times \frac{19}{108} = \frac{127}{27}$ 이다. 한 모서리의 길이가 1인 단위 정사면체의 부피는  $\frac{\sqrt{2}}{12}$ 이므로 실제 케이지의 부피는  $\frac{\sqrt{2}}{12} \times \frac{127}{27} = \frac{127\sqrt{2}}{324}$ 이다.

<증명 끝>

#### IV-2-2. 격자 정육면체의 중심연결체

##### ① c-v 연결체



2번째 격자 정육면체

중심과 꼭짓점 연결

n번째 격자 정육면체의 c-v 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	4	...	n
중심	1	$2^3$	$3^3$	$4^3$	...	$n^3$
꼭짓점	$2^3 + 1$	$3^3 + 2^3$	$4^3 + 3^3$	$5^3 + 4^3$	...	$(n+1)^3 + n^3$
모서리	8	$8 \times 2^3$	$8 \times 3^3$	$8 \times 4^3$	...	$8 \times n^3$
삼각형	0	$12 \times 2^3 - 12 \times 2$	$12 \times 3^3 - 12 \times 3$	$12 \times 4^3 - 12 \times 4$	...	$12 \times (n^3 - n)$
마름모	0	$2 \times 1^2 \times 3 \times 2$	$3 \times 2^2 \times 3 \times 2$	$4 \times 3^2 \times 3 \times 2$	...	$6n(n-1)^2$
팔면체	0	$3 \times 2^2 \times 1$	$3 \times 3^2 \times 2$	$3 \times 4^2 \times 3$	...	$3n^2(n-1)$
마름모십이면체	0	0	$3 \times 2^2 \times 1$	$3 \times 3^2 \times 2$	...	$3(n-1)^2(n-2)$

n번째 격자 정육면체의 c-v 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 선분의 수)	한 면의 넓이 (그 면의 수)	부피	단위 정육면체와의 부피비
삼각형	$\frac{\sqrt{3}}{2}(2), 1(1)$	$\frac{\sqrt{2}}{4}$		

마름모	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$		
팔면체	$\frac{\sqrt{3}}{2}$ (8), 1(4)	$\frac{\sqrt{2}}{4}$	$\frac{1}{3}$	$1:\frac{1}{3}$
마름모십이면체	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	2	1:2

정육면체의 c-v 연결체를 만들면 2단계에서 단위 정육면체 안에 정사각뿔 6개가 만들어진다. 본래의 정육면체의 모서리를 제거한 3단계 도형에서는 정사각뿔 2개가 합쳐져 팔면체가 된다.

6개의 팔면체가 하나의 정육면체의 중심에서 만나면 마름모십이면체가 만들어진다. 또한 정육면체를 중심으로 6개의 면에 정사각뿔이 붙어도 마름모 십이면체가 만들어진다.

정리 II-1. 팔면체에서 한 면의 넓이는  $\frac{\sqrt{2}}{4}$ , 마름모십이면체에서 한 면의 넓이는  $\frac{\sqrt{2}}{2}$ 이다.

증명 시작) 다음 그림은 정육면체를 직각이등변삼각기둥 2개가 만들어지도록 잘랐을 때 생기는 면을 나타낸 것이다. 이 면의 중심은 원래의 정육면체의 중심이고, 직사각형의 두 대각선은 정육면체의 두 대각선이다.

팔면체의 한 면이 두 개 합쳐져 만들어짐으로 마름모십이면체에서 한 면의 넓이는  $\frac{\sqrt{2}}{4}$ 의 두배인  $\frac{\sqrt{2}}{2}$ 이다.

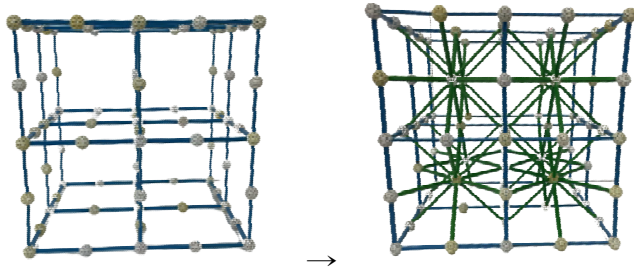
<증명 끝>

정리 II-2. 팔면체의 부피는  $\frac{1}{3}$ , 마름모십이면체의 부피는 2이다.

증명 시작) 단위 정육면체의 부피가 1이고 정육면체 안에 합동인 정사각뿔이 6개가 생기므로 정사각뿔의 부피는  $\frac{1}{6}$ 이다. 팔면체의 부피는 정사각뿔의 두 배이므로  $\frac{1}{3}$ 이고, 마름모십이면체의 부피는 정사각뿔의 12배이므로 2이다.

<증명 끝>

## ② c-s 연결체



2번째 격자 정육면체

중심과 선분 연결

n번째 격자 정육면체의 c-s 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 n번째	1	2	3	...	n
중심	1	$2^3$	$3^3$	...	$n^3$
꼭짓점	$2 \times 2 \times 3 + 1$	$2 \times 3 \times 3 \times 3 + 2^3$	$3 \times 4 \times 4 \times 3 + 3^3$	...	$3n(n+1)^2 + n^3$
모서리	12	$12 \times 2^3$	$12 \times 3^3$	...	$12n^3$
정삼각형	0	$8 \times (2^2 \times 1 \times 3 + 1)$	$8 \times (3^2 \times 2 \times 3 + 2^3)$	...	$8 \times \{3n^2(n-1) + (n-1)^3\}$
정사면체	0	8	$8 \times 2^3$	...	$8 \times (n-1)^3$
정팔면체	0	$2^2 \times 1 \times 3 + 1$	$3^2 \times 2 \times 3 + 2^3$	...	$3n^2(n-1) + (n-1)^3$

n번째 격자 정육면체의 c-s 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 수)	선분의 한 면의 넓이 (그 면의 수)	부피	단위 정육면체 와의 부피비
정삼각형	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{8}$		
정사면체	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{8}$	$\frac{1}{24}$	$1: \frac{1}{24}$
정팔면체	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{8}$	$\frac{1}{6}$	$1: \frac{1}{6}$

정사면체는 1단계의 격자 정육면체에서 외부에 노출되지 않는 꼭짓점에 8개씩 생긴다.

정팔면체는 1단계의 격자 정육면체에서 외부에 노출되지 않는 면에 1개씩, 1단계의 격자 정육면체에서 외부에 노출되지 않는 꼭짓점에 1씩 생긴다.

정리 II-3. 정팔면체의 개수는  $3n^2(n-1) + (n-1)^3$ 이다.

증명 시작) 정육면체의 c-s 연결체에서 만들어지는 정팔면체는 두 가지로 나눌 수

있다. 정팔면체의 중심이 정육면체의 한 면에 있는 경우와 꼭짓점에 있는 경우이다. 정팔면체의 중심이 정육면체의 한 면에 있는 경우에서 정팔면체의 개수는 정육면체의 노출되지 않은 면의 개수와 같다. 따라서 이 때 정팔면체의 개수는  $3n^2(n-1)$ 이다. 정팔면체의 중심이 정육면체의 한 꼭짓점에 있는 경우에서 정팔면체의 개수는 정육면체의 노출되지 않은 꼭짓점의 개수와 같다. 따라서 이 때 정팔면체의 개수는  $(n-1)^3$ 이다. 따라서 정팔면체의 총 개수는  $3n^2(n-1)+(n-1)^3$ 이다. 여기서 ‘노출되지 않은’이 붙는 이유는 노출된 곳에서 만들어지는 정팔면체는 1단계 정육면체의 바깥에 생기기 때문이다.

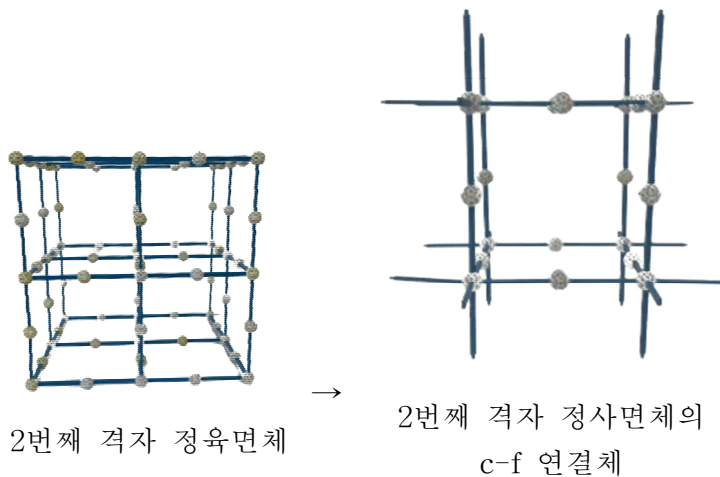
<증명 끝>

정리 II-4. 정삼각형의 개수는  $8 \times 3n^2(n-1)+(n-1)^3$ 이다.

위 정리 II-3로 정삼각형의 개수를 구할 수 있다. 위 중심연결체에서 생기는 정삼각형은 정팔면체를 이룬다. 정사면체를 이루는 정삼각형은 정팔면체를 이루는 정삼각형과 겹치기 때문에 고려하지 않아도 된다. 따라서 정팔면체를 이루는 정삼각형의 개수는 8개이므로  $8 \times 3n^2(n-1)+(n-1)^3$ 이다.

<증명 끝>

### ③ c-f 연결체

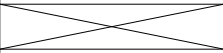
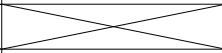


n번째 격자 정육면체의 c-f 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \ n번째	1	2	3	...	n
중심	1	$2^3$	$3^3$	...	$n^3$
꼭짓점	$1^2 \times 2 \times 3 + 1$	$2^2 \times 3 \times 3 + 2^3$	$3^2 \times 4 \times 3 + 3^3$	...	$3n^2(n+1) + n^3$
모서리	6	$6 \times 2^3$	$6 \times 3^3$	...	$6n^3$

정사각형	0	$1 \times 2 \times 3$	$2^2 \times 3 \times 3$	...	$3n(n-1)^2$
정육면체	0	1	$2^3$	...	$(n-1)^3$

n번째 격자 정육면체의 c-f 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 수)	선분의 한 면의 넓이 (그 면의 수)	부피	단위 정육면체 와의 부피비
정사각형	1	1		
정육면체	1	1	1	1:1

정육면체의 c-f 연결체는 위와 같이 간단하므로 설명을 생략한다.

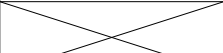
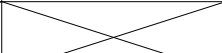
#### IV-2-3. 격자 정팔면체의 중심연결체

##### ① c-v 연결체

n번째 격자 정팔면체의 c-v 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 n번째	1	2	3	4	...	n
중심	1	1+4 +1	1+4+9 +4+1	1+4+9+16 +9+4+1	...	$R_n + R_{n-1}$
꼭짓점	1+5+ 1	1+5+13 +5+1	1+5+13+25 +13+5+1	1+5+13+25+41 +25+13+5+1	...	$4R_{n-1} + 3n^2 + (n+1)^2$
모서리	6	$6 \times (1+4+1)$	$6 \times (1+4+9+4+1)$	$6 \times (1+4+9+16+9+4+1)$	...	$6 \times (R_n + R_{n-1})$
정사각형	0	4+4+4	4+4+16+12 +16+4+4	4+4+16+12+36+2 4 +36+12+16+4+4	...	$8 \times R_{n-1} + 4(T_{n-1} + T_{n-2})$
정육면체	0	0	$2 \times 4$	$2 \times (4+12)$	...	$8 \times T_{n-2}$

n번째 격자 정팔면체의 c-v 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 수)	선분의 한 면의 넓이 (그 면의 수)	부피	단위 정팔면체 와의 부피비
정사각형	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$		

정육면체	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	$\frac{\sqrt{2}}{4}$	$1:\frac{3}{4}$
------	----------------------	---------------	----------------------	-----------------

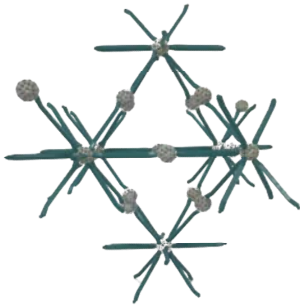
정리 III-1. 정사각형의 개수는  $8 \times R_{n-1} + 4(T_{n-1} + T_{n-2})$ 이다.

증명 시작) 정팔면체의 C-V연결체에서의 정사각형은 바닥과 나란한 경우와 수직인 경우로 나눌 수 있다. 바닥과 나란한 경우는 정사각형의 C-V연결체의 정사각형의 개수( $4t_{n-1}$ )로, 수직인 경우는 정사각형의 C-V연결체의 선분의 개수( $4r_n$ )로 나타낼 수 있다. 정팔면체를 정확히 반으로 나누는 층은 정사각형이 바닥과 나란한 층이므로 나란한 정사각형은  $4t_{n-1}$ 의 1부터 n까지의 합과 n-1까지의 합을 더한  $4(T_{n-1} + T_{n-2})$ 이다. 수직인 정사각형은  $4r_n$ 의 1부터 n-1까지의 합에 2배를 한  $8 \times R_{n-1}$ 이다. 따라서 전체 정사각형 수는  $8 \times R_{n-1} + 4(T_{n-1} + T_{n-2})$ 이다.

정리 III-2. 정육면체의 개수는  $8 \times T_{n-2}$ 이다.

증명 시작) 정육면체의 개수는 나란한 정사각형의 개수에서 가운데 층에 있는 정사각형의 개수를 뺀 것과 같다. 즉  $4t_{n-1}$ 의 1부터 n까지의 합과 n-1까지의 합을 더한 것에서 n인 경우를 빼는 것이므로  $4t_{n-1}$ 의 1부터 n-1까지의 합의 2배와 같다. 따라서 정육면체의 개수는  $8 \times T_{n-2}$ 이다.

② c-s 연결체



2번째 격자 정팔면체의  
c-s 연결체

n번째 격자 정팔면체의 c-s 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 \n번째	1	2	3	...	n
중심	1	1+4+1	1+4+9+4+1	...	$R_n + R_{n-1}$
꼭짓점	12+1	$12(1+4+1) + (1+4+1) - (2 \times 4+4)$	$12(1+4+9+4+1) + (1+4+9+4+1) - \{2 \times (4+4+16) + 12\}$	...	$13R_n + 5R_{n-1} - 8T_{n-2} - 4t_{n-1}$
모서리	12	12(1+4+1)	12(1+4+9+4+1)	...	$12(R_n + R_{n-1})$

정삼각형	0	8	$8(1+4+1)+2\times 4$	...	$8(R_{n-1} + R_{n-2} + t_{n-2})$
정사각형	0	1	$1+4+1$	...	$R_{n-1} + R_{n-2}$
정사각뿔	0	2	$2(1+4+1)$	...	$2(R_{n-1} + R_{n-2})$
정팔면체	0	1	$1+4+1$	...	$R_{n-1} + R_{n-2}$
정사면체	0	0	$2\times 4$	...	$8t_{n-2}$

n번째 격자 정팔면체의 c-s 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 선분의 수)	한 면의 넓이 (그 면의 수)	부피	단위 정팔면체 와의 부피비
정삼각형	1	$\frac{\sqrt{3}}{4}$	<del></del>	<del></del>
정사각형	1	1	<del></del>	<del></del>
정사각뿔	1	$\frac{\sqrt{3}}{4}$ (4), 1(1)	$\frac{\sqrt{2}}{6}$	$1:\frac{1}{2}$
정팔면체	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{2}}{3}$	1:1
정사면체	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{2}}{12}$	$1:\frac{1}{4}$

정팔면체의 c-s 연결체는 본래 격자 정팔면체와 같게 나타난다.

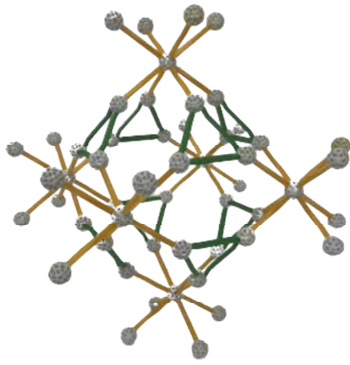
정팔면체의 c-s 연결체에서 n번째의 정삼각형, 정사각형, 정사각뿔, 정팔면체, 정사면체의 개수는 모두 n-1번째 격자 정팔면체에서 다음 도형이 만들어지는 개수와 같다. 이 이유는 정팔면체의 c-s 연결체에서 1단계의 정팔면체의 모서리의 2배인 격자 정팔면체가 생기기 때문이다.

정리 III-3. 정팔면체의 c-s 연결체에서 만들어지는 정팔면체는 단위 정팔면체와 크기가 같다.

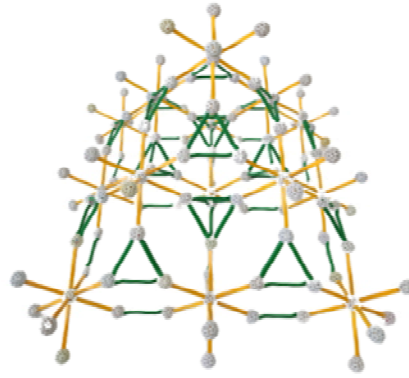
증명 시작) 격자 정팔면체는 정팔면체가 모서리로 연결되어 있다. 따라서 이웃하는 정팔면체의 c-s 연결체는 이웃한 그 모서리에서 중심과 모서리의 수선의 발을 연결하는 선분이 한 선분으로 연결되고 선분의 길이는 중심에서 모서리의 수선의 발까지 이은 것의 2배가 된다. 이는 단위 정팔면체의 한 모서리의 길이와 같으므로 정팔면체의 c-s 연결체에서 만들어지는 정팔면체는 단위 정팔면체와 크기가 같다.

<증명 끝>

③ c-f 연결체



2번째 격자 정팔면체의  
c-f 연결체



3번째 격자 정팔면체의  
c-f 연결체

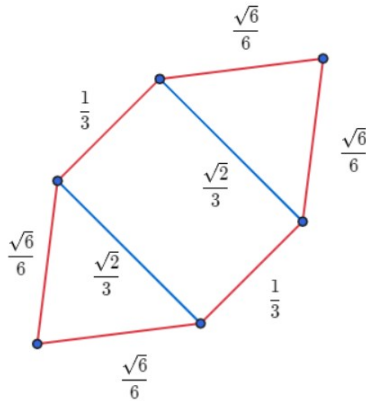
n번째 격자 정팔면체의 c-f 연결체에 있는 구성요소의 개수는 다음과 같다.

개수 n번째	1	2	3	4	...	n
중심	1	1+4+1	1+4+9+4+1	1+4+9+16+9+4+1	...	$R_n + R_{n-1}$
꼭짓점	9	9(1+4+1)	9(1+4+9+4+1)	9(1+4+9+16+9+4+1)	...	$9(R_n + R_{n-1})$
모서리	8	$8(1+4+1)$ $+ 2 \times 3 \times 4 \times$	$8(1+4+9+4+1)$ $+ 2 \times 3 \times 4(1+2)$ $+ 2 \times 6 \times 4$	...	...	$8(R_n + R_{n-1})$ $+ 24t_{n-1}$ $+ 48T_{n-2}$
정삼각형	0	8	$8 \times (1+4+1)$ $+ 2 \times 4 \times 1$	$8 \times (1+4+9+4+1)$ $+ 2 \times 4 \times (1+2)$	...	$8(R_{n-1} + R_{n-2})$ $+ 8t_{n-2}$
육각형	0	12	$12 \times (1+4+1)$ $- 2 \times 4 \times 1$	$12 \times (1+4+9+4+1)$ $- 2 \times 4 \times (1+4)$	...	$12(R_{n-1} + R_{n-2})$ $- 8R_{n-2}$
케이지	0	1	1+4+1	1+4+9+4+1	...	$R_{n-1} + R_{n-2}$
정팔면체	0	1	1+4+1	1+4+9+4+1	...	$R_{n-1} + R_{n-2}$
윈도우 (삼각빨대)	0	8	8(1+4+1)	8(1+4+9+4+1)	...	$8(R_{n-1} + R_{n-2})$
정사면체	0	0	$2 \times 1 \times 2 \times 2$	$2 \times$ $(1 \times 2 \times 2 + 2 \times 3 \times 2)$	...	$\frac{4n(n-1)(n-2)}{3}$

n번째 격자 정팔면체의 c-f 연결체에 있는 구성요소의 크기는 다음과 같다.

	길이 (그 선분의 수)	한 면의 넓이 (그 면의 수)	부피	단위 정팔면체와 의 부피비
정삼각형	$\frac{1}{3}$	$\frac{\sqrt{3}}{12}$	<del>                    </del>	<del>                    </del>
육각형	$\frac{1}{3}(2), \frac{\sqrt{7}}{6}(4)$	$\frac{1}{9} + \frac{\sqrt{6}}{18}$	<del>                    </del>	<del>                    </del>
케이지	$\frac{1}{3}(24), \frac{\sqrt{7}}{6}(24)$	$\frac{\sqrt{3}}{12}(8),$ $\frac{1}{9} + \frac{\sqrt{6}}{18}(12)$	$\frac{\sqrt{3}}{2} + \frac{13\sqrt{2}}{81}$	1: $1 + \frac{13\sqrt{6}}{123}$

정팔면체	1	$\frac{\sqrt{3}}{4}$	$\frac{\sqrt{3}}{2}$	1:1
원도우 (삼각뿔 대)	1(3), $\frac{1}{3}(3)$	$\frac{\sqrt{3}}{4}$ , $\frac{\sqrt{3}}{36}$	$\frac{\sqrt{2}}{9}(3)$ , $\frac{13\sqrt{2}}{648}$	1: $\frac{13\sqrt{6}}{984}$
정사면체	$\frac{1}{3}$	$\frac{\sqrt{3}}{12}$	$\frac{\sqrt{2}}{324}$	1: $\frac{1}{108}$



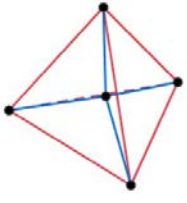
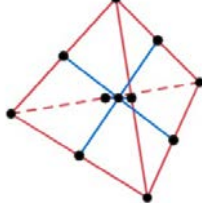
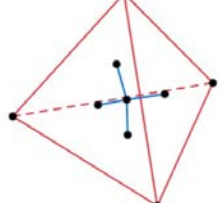
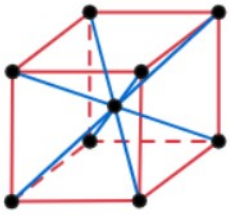
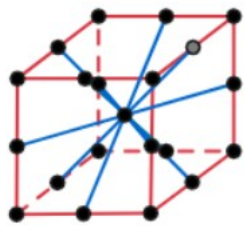
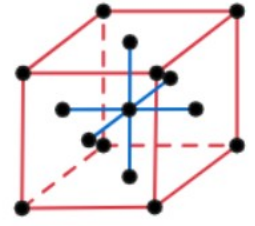
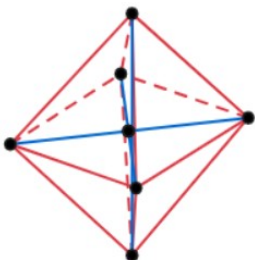
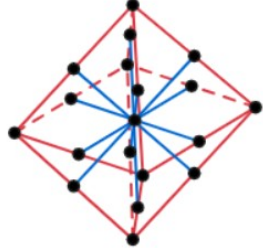
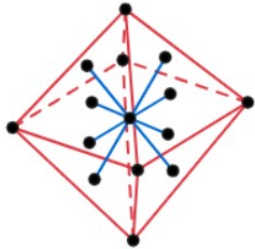
격자 정팔면체의 c-f 연결체의 케이지의 육각형인 한 면

격자 정팔면체의 c-f 연결체의 케이지는 육각형 12개와 정삼각형 8개로 이루어진 도형이다. 정삼각형 하나를 중심으로 육각형 6개가 붙어있다. 또한 네 개의 육각형이 모여 한 꼭짓점을 이룬다.

12개의 육각형을 위, 중간, 아래 부분으로 각각 4개씩 나눌 수 있는데 이 중 중간층은 육각형이 서로 수직하게 만난다. 이 때문에 정팔면체의 공간테셀레이션과 비슷한 모습이 나타난다. 정팔면체의 정삼각형이 생기는 위치와 정팔면체의 c-f 연결체의 정삼각형이 생기는 위치가 같다. 또한 정팔면체의 모서리의 위치와 정팔면체의 c-f 연결체의 육각형이 생기는 위치가 같다. 특히 정사면체의 개수나, 정사면체로 인해 생기는 정삼각형의 개수는 같게 나타난다.(정팔면체의 n번째 개수와 정팔면체의 c-f 연결체의 n-1번째 개수가 같게 나타난다.) 이는 정팔면체와 케이지가 정사면체를 형성하는 방법 때문이다. 각 층의 정사면체 중 가장자리에 있는 정사면체는 세 개의 정팔면체/케이지가 만나 형성되고, 그 외의 안쪽에 있는 정사면체는 네 개의 정팔면체/케이지가 만나 형성된다.

격자 정팔면체의 c-f 연결체의 케이지는 정팔면체의 각 면에 삼각뿔대가 붙은 형태라고 볼 수 있다. 이 때 이 삼각뿔대가 격자 정팔면체의 c-f 연결체의 원도우가 된다

IV-3. 단위 중심연결체끼리의 상관관계

	c-v 연결체	c-s연결체	c-f연결체
정사면체	 <p>선분 수 : 4 연결체 도형 : 정사면체</p>	 <p>선분 수 : 6 연결체 도형 : 정팔면체</p>	 <p>선분 수 : 4 연결체 도형 : 정사면체</p>
정육면체	 <p>선분 수 : 8 연결체 도형 : 정육면체</p>	 <p>선분 수 : 12 연결체 도형 : 입방팔면체</p>	 <p>선분 수 : 6 연결체 도형 : 정팔면체</p>
정팔면체	 <p>선분 수 : 6 연결체 도형 : 정팔면체</p>	 <p>선분 수 : 12 연결체 도형 : 입방팔면체</p>	 <p>선분 수 : 8 연결체 도형 : 정육면체</p>

선분 수 : 단위 중심연결체의 선분 수

연결체 도형 : 단위 중심연결체의 각 이웃한 꼭짓점을 연결했을 때 만들어지는 도형

\* 위와 같은 ‘연결체 도형’이 나타나는 이유

c-v연결체와 관련	각 모서리의 중점을 이웃하는 것끼리 이을 때 생기는 도형 (c-s연결체와 관련)	듀얼 정다면체 (c-f연결체와 관련)
정사면체	정팔면체	정사면체
정육면체	입방팔면체	정팔면체
정팔면체	입방팔면체	정육면체

연결체 도형의 c-v연결체는 모두 본래의 연결체과 같다. 예를 들어 정육면체의 c-s 연결체의 연결체 도형은 입방팔면체이다. 입방팔면체는 정육면체의 c-s연결체의 꼭짓점을 이어 만든 것이므로 입방팔면체의 꼭짓점과 정육면체의 c-s연결체의 꼭짓점이 일치한다. 따라서 정육면체의 c-s연결체와 입방팔면체의 c-v연결체는 일치한다. 이로 인해 연결체 도형이 같은 단위 중심연결체는 서로 도형이 일치한다. 연결체 도형이 입팔면체인 경우를 예로 들어 보자. 정육면체의 c-s연결체와 입방팔면체의 c-v연결체가 일치한다는 것, 정팔면체의 c-s연결체와 입방팔면체의 c-v연결체가 일치한다는 것을 위 문단을 통해 알 수 있다. 이를 합쳐 생각하면 정육면체의 c-s연결체와, 정팔면체의 c-s연결체, 입방팔면체의 c-v연결체 이 셋 모두가 일치한다는 것을 알 수 있다. 이 이유는 두 중심연결체의 연결체 도형이 입방팔면체로 같다는 점에서 기인하였다.

## V. 결론

연구를 통해 중심연결체들 사이에서 나타나는 공통점을 찾아보았다.

- 1) 중심의 개수는 그 도형의 격자도형의 개수와 같다.
- 2) 꼭짓점의 개수는 중심의 개수와 본래 도형의 꼭짓점/모서리/면의 개수와 같다. 예를 들어 정팔면체의 c-s연결체에서 꼭짓점의 개수는 중심의 개수와 본래의 도형 즉, 정팔면체의 모서리 개수의 합과 같다. 위 표의 있는 방법은 이를 사용하지 않고 보이는 꼭짓점의 수를 직관적으로 분석하여 그 규칙을 통해 구한 것이다. 만약 이 방법을 사용한다면 꼭짓점의 개수를 더 쉽게 구할 수 있을 것이다.
- 3) 정사면체의 중심연결체의 구성요소의 개수는  $T_n$ , 정팔면체의 중심연결체의 구성요소의 개수는  $R_n$ 에 대한 식으로 주로 정리되고, 정육면체의 중심연결체의 구성요소의 개수는  $T_n$ 과  $R_n$ 으로 정리되지 않는다.  
그 이유는 다음과 같다.  
 $T_n$ 은 본래 삼각뿔로부터,  $R_n$ 은 본래 사각뿔로부터 나왔다. 정사면체는 삼각뿔의 특수한 경우이므로  $T_n$ 으로 정리되는 경우가 많다. 정팔면체는 사각뿔 두 개로 쪼갤 수 있기 때문에  $R_n$ 으로 정리되는 경우가 많다. 이 때 정리된 식에  $R_n$ 과  $R_{n-1}$ 이 모두 쓰이는 경우가 많은데 이는 사각뿔이 정확히 반으로 나뉘지는 것이 아니라 한 단계 밑의 정사면체로 나뉘지기 때문이다. 정육면체의 경우 이 둘과 연관되어 있지 않으므로  $T_n$ 과  $R_n$ 으로 정리되지 않는 것이다.

## VI. 참고 문헌

김예주·이연서·전설(2021). 도형수와 다이아몬드 구조. 청주교육대학교 과학영재교육원 중등 사사과정 보고서.

고영원(2022). 격자다면체의 중심연결체를 이용한 공간테셀레이션. 청주교육대학교 과학영재교육원 중등 사사과정 보고서.

이정용(2022). 재료결정학. 교문사.

수학B

## 02\_페이지랭크(Pagerank) 이해 및 활용



# 페이지랭크 이해 및 활용

서승연 (영주중학교 2학년)

이은호 (봉화중학교 2학년)

지도교수: 김재명 (국립안동대학교 수학교육과)

**초록** : 일상생활에서 검색사이트를 정보 및 자료 검색을 위해 자주 사용하고 있다. 검색사이트에 찾고자 하는 키워드를 입력하면 해당 키워드가 포함된 웹 페이지들이 순서 있게 나열이 된다. 검색 키워드를 조금 바꾸어서 검색하면 웹 페이지들이 다른 순서로 나열이 된다. 웹페이지의 순서를 어떻게 줄 수 있는지를 간단한 그래프 및 확률 행렬을 통해서 알아보려고 한다. 중학교 2학년 수학교육 과정 중에서 일상생활에서 많이 적용되는 확률을 활용한 페이지랭크(Pagerank:페이지 순위)를 이해하고, 도로망 연결에 적용해 보려고 한다.

**주요용어** : 검색엔진(search engine) ,마코프 행렬(Markov matrix), 확률(Probability), 페이지 랭크(Page rank)

## I. 서론

우리는 일상생활에서 검색사이트를 정보 및 자료 검색을 위해 자주 사용하고 있다. 검색사이트에 찾고자 하는 키워드를 입력하면 해당 키워드가 포함된 웹페이지들이 순서 있게 나열이 된다. 검색 키워드를 조금 바꾸어서 검색하면 웹페이지들이 다른 순서로 나열이 된다. 우리는 검색사이트들을 사용만 할 뿐 그 속에 있는 원리에 대해서 깊게 생각하지 않는다. 그렇기에 그 원리를 알아보고 탐구해 보아야겠다는 생각이 들었다.

여러 가지 원리 중에서, 페이지랭크(PageRank)는 정보화 사회에서 그 진가가 발휘된다. 정보의 바다라고 불리는 인터넷에서 가중치를 부가하고 평가하여 검색자에게 가장 필요한 자료를 찾아준다. 우리 사회는 점점 정보화 사회가 극대화될 것이다. 정보화 사회에서 정보는 지식이고 무기가 된다. 그 정보를 가질 수 있게 도와줄 수 있는 것이 페이지랭크이라 생각하며, 인공지능에서 광범위하게 활용되고 있다. 국립안동대학교 과학영재교육원에서 중학교 2학년 수학교육 과정 중에서 일상생활에서 가장 많이 사용되고 점점 그 가치가 높아질 확률을 활용한 페이지 순위를 이번 학기 주제로 선정하게 되었고, 페이지 순위를 탐구하고 일상생활에 적용해 보려고 한다.

## II. 이론적 배경

### II.1. 마르코프 체인의 정의 및 성질

마르코프 체인은 마르코프 성질을 가진 확률적인 시행으로, 어떠한 상태가 시간에 따라 어떻게 변화할지를 예측하는 것이다. 마르코프 성질이란 각 시행의 결과가 바로 앞의 결과에만 영향을 받는 성질을 말한다. 즉, 현재 상태는 이전 상태에만 의존한다. 마르코프 체인을 활용하면, 복잡한 알고리즘 없이 과거의 통계기반만으로 간단하게 예측을 할 수 있다. 따라서, 마르코프 체인을 사용하면, 충분히 반복된 사건을 기반으로 미래를 확률적으로 예측할 수 있다. 또한, 현재 상태는 이전



그림1. 안드레이 안드레예비치 마르코프(러시아어:Андрей Андреевич Марков,1856년 6월 14일-1922년 7월 20일) 출처:위키

상태에만 의존한다는 마르코프 체인의 특성 때문에, 마르코프 체인은 많은 문제에 적용된다. 예를 들어, 구글의 페이지 랭크, 날씨 예보, 마케팅 등 과거와 현재가 독립되지 않은 상태의 사건들이 마르코프 체인을 사용할 수 있다. 그러나, 과거에 등장하지 않았던 패턴에 대해서는 예측할 수 없다는 단점이 있다. (마르코프 체인과 관련해서는 [1], [2]를 참고하기 바란다.)

#### - 마르코프 체인 활용 예제

[4]에 있는 내용을 바탕으로, 아래 표는 1월 달 11번가 쇼핑몰에서 A라는 물건 구매자 중에서 그 다음 달 재 구매자를 확률로 나타낸 것이다.

1월	2월(단위: 확률)	
구매 (3만명)	구매 0.8	비구매 0.2
비구매 (1만명)	구매 0.1	비구매 0.9

표1. 구매자의 변동 추이

질문) 위 표를 통해서 3월 달의 구매자 수와 비 구매자 수는 어떻게 예측할 수 있을까?

풀이) 1월달에서 2월달 구매자의 이동을 나타내는 전이행렬(transition matrix)를 구

해 보자. 전이행렬을 구하기 위해서 우리는 다음과 같은 기호를 도입하자.  
 $p_{i,j}$  : 1월달 구매/비구매 상태  $i$ 에서 2월달 구매/비구매  $j$ 로 이동할 확률,  
 $P = [p_{i,j}]_{1 \leq i,j \leq N}$  :  $p_{i,j}$ 으로 구성된 전이행렬

위의 표를 상태전이도와 전이행렬로 나타내 보자.

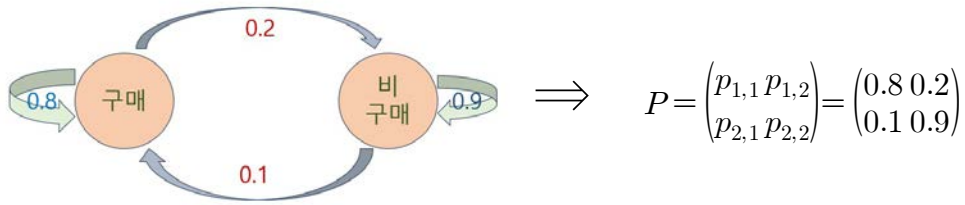


표2. 상태전이도와 전이행렬

여기서,  $p_{1,1}$ : 1월 달 구매자가 2월 달 구매할 확률,  
 $p_{1,2}$ : 1월 달 구매자가 2월 달 구매하지 않을 확률,  
 $p_{2,1}$ : 1월 달 구매하지 않은 사람이 2월 달 구매할 확률,  
 $p_{2,2}$ : 1월 달 구매하지 않은 사람이 2월 달 구매하지 않을 확률.

1월달에서 2월달로의 구매자/ 비 구매자수의 전이행렬  $P = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix}$ 를 통해서  
 3월달 구매자/ 비 구매자수 전이행렬은  $P^2 = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix} \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix} = \begin{pmatrix} 0.66 & 0.34 \\ 0.17 & 0.83 \end{pmatrix}$ 으로  
 구할 수 있다.

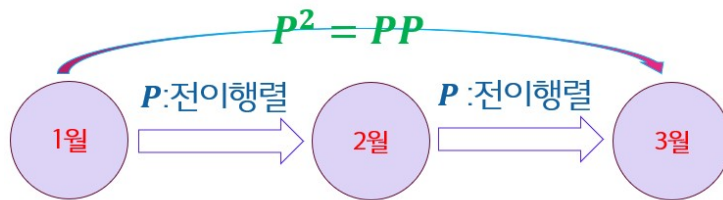


그림2. 월별 전이도(전이 행렬)

따라서 월별 구매/비 구매자의 수에 대한 예측 결과를 나타내면 다음과 같다.

1월 현황(기준)	2월 구매자 현황(%)		3월 구매자 예측(%)	
	구매	비 구매	구매	비 구매
구 매 (3만명)	24,000(80%)	6,000(20%)	19,800(66%)	10,200(34%)
비 구매 (1만명)	1,000(10%)	9,000(90%)	1,700(17%)	8,300(83%)
합계 (4만명)	25,000(63%)	15,000(37%)	21,500(54%)	18,500(46%)

표3. 3월 A 물건 구매 예측

예측 결과에 따른 수치적 해석) 1월 구매자 중 2월 비 구매자의 비율은 20%(6,000명)이었으나, 현재의 구매행태를 보면 3월에는 34%(10,200명)으로 구매자 이탈이 가속화됨을 확인 할 수 있다. 2월 비 구매자의 3월 재구매 예측은 10%에서 17%로 다소 상승했으나, 구매자의 이탈을 막기에는 역부족으로 보인다.

예측 결과를 통한 구매자 이탈을 방지하기 위한 방안) 비 구매 고객의 구매 전환도 중요하겠지만 구매 고객의 재구매 활동에 주력하는 것이 더 유의미해 보인다.

이렇게 과거에 대한 확률을 분석하고 그 결과를 미래를 예측하는 데에 사용함으로써, 대안을 결정하고 판단할 수 있게 된다.

## II.2. 페이지랭크란 무엇인가?

페이지랭크란 인터넷망에 있는 여러 웹페이지를 정리하기 위해 각각의 페이지들에게 점수를 부여해주는 하나의 알고리즘이다. 페이지랭크는 스탠퍼드 대학교에 재학 중이던 래리 페이지와 세르게이 브린이 당시 존재하던 단어의 포함을 검색결과 기준으로 삼은 비효율적인 검색엔진을 보완하기 위해서 최초로 생각되었다. 이 개발은 1995년에 시작되어 결국 현재 대표적인 검색엔진인 구글이라는 서비스로 발전하게 되었고 대부분의 인터넷서비스 바탕이 되었다.

[2.5]에 있는 내용을 바탕으로, 웹 페이지들의 연결(connected)되어 있을 때, 각 웹페이지에서 나가는 간선 또는 들어오는 간선을 통해서, 각각의 웹 페이지들은

$\frac{\text{자신의 페이지랭크 점수}}{\text{나가는 이웃의 수}}$  만큼 다른 웹 페이지에게 점수를 부여한다. 이때 그림과 같은 페이지들의 연결이 있다고 할 때 각 웹사이트들의 페이지랭크 점수는  $r_n$ 이다. 웹사이트  $j$ 의 페이지랭크 점수를 구한다고 할 때  $j$ 로 투표하는 투표값들의 합을 구하면 된다.

즉  $j$ 로 들어오는 간선인  $i, k$ 로부터 들어오는 투표값( $\frac{\text{자신의 페이지랭크 점수}}{\text{나가는 이웃의 수}}$ )을 더하면 되고 결국  $r_j = \frac{r_i}{3} + \frac{r_k}{4}$ 가 될 것이다. 이렇게 각각의 웹페이지에서의 투표 값을 같은 방법으로 구할 수 있으며, 이것을 일반화하면, 다음 수식을 얻어 낼 수 있다.

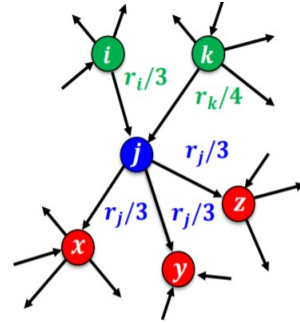
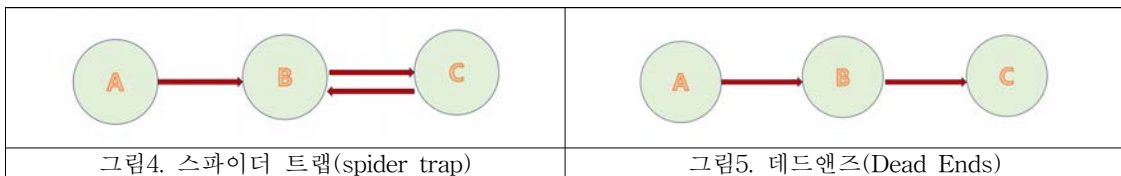


그림3.

$$r_j = \sum_{i \in N_{in}(j)} \frac{r_i}{d_{out}(i)}$$

앞에서 언급한 알고리즘을 바탕으로, 각 웹페이지가 마코프 체인이라 가정하면, 한 명의 웹서퍼가 있다고 가정하고 그 웹서퍼가 균일한 확률로 웹들을 클릭하는 방법으로 웹을 옮겨 다닌다고 생각한다. 이 웹서퍼가  $t$ 번째 방문한 사이트가  $i$ 일 확률을  $p_i(t)$ 라고 한다. 이때  $p_i(t)$ 는 길이가 웹페이지 수(정점의 수)와 같은 확률분포 벡터가 된다. 이때 아래 수식과 같은  $p_j(t+1) = \sum_{i \in N_{in}(j)} \frac{p_i(t)}{d_{out}(i)}$ 라는 식이 나오게 된다. 이와 같은 과정을 업데이트 과정이라 부른다. 이때  $t$ 가 한없이 커질 시  $p(t)$ 는 어떤 하나의 값으로 수렴하게 되고, 이때의 수렴 값  $p$ 를 각 페이지들이 페이지랭크 점수이다. 하지만 아래와 같은 경우에는  $p(t)$ 가 수렴하지 않을 수 있다.



첫 번째 경우는 스파이더 트랩으로 웹 서퍼가 들어오는 간선만 있고 나가는 간선이 없는 웹 페이지들에 갇힐 수 있다. 위 그림에서 보면,  $A$ 에서  $B$ 로 들어오게 될 때,  $B$ 에서 다른 곳으로 갈 수 있는 간선이 없어 웹 서퍼가 갇히게 된다. 즉,  $B$ 에서는  $C$ 로, 다시  $B$ 로 무

한 루프에 빠지게 된다. 이러한 경우에는, 웹 서퍼가 페이지들 사이에 갇히게 되고 결국 페이지랭크 점수가 수렴하지 않는다.

두 번째 경우는 막다른 정점이다. 웹 서퍼가 막다른 정점에 도달해 다른 곳으로 이동할 수 없는 경우 웹 서퍼는 한곳에 머무르면서 갇히게 된다. 이 경우 웹 서퍼는 더는 움직일 수 없게 되므로 페이지랭크의 점수가 0이 된다. 위 그림에서 보면, 웹 서퍼가 막다른 정점에 도달하게 될 때, 웹 서퍼가 정지하게 된다.

이와 같은 문제들을 우리는 페이지랭크를 마르코프체인의 관점으로 봄으로써 해결할 수 있게 된다.

웹 페이지 $j$ 에서의 페이지랭크 값	$r_j = \sum_{i \in N_{in}(j)} \frac{r_i}{d_{out}(i)}$
벡터 표현	$r^{t+1} = Pr^t, P = [p_{ij}]_{1 \leq i, j \leq n}$ : 전이행렬
텔레포트 개념 도입 벡터 표현	$r^{t+1} = Ar^t, A = \beta P + (1-\beta)\left[\frac{1}{N}\right]_{N \times N}$

표4. 각 개념에 대한 수식 표현

여기서,  $r^0$ ( $t=0$ 일 때의 초기값, 데이터를 통해 주어진 값),

$$r^t = \begin{pmatrix} r_1^t \\ r_2^t \\ \vdots \\ r_j^t \\ \vdots \\ r_N^t \end{pmatrix}_{N \times 1}, \quad r^{t+1} = Pr^t \Leftrightarrow \begin{pmatrix} r_1^{t+1} \\ r_2^{t+1} \\ \vdots \\ r_j^{t+1} \\ \vdots \\ r_N^{t+1} \end{pmatrix}_{N \times 1} = r^{t+1} = [P_{ij}]_{1 \leq i, j \leq N} r^t = \begin{pmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & \vdots & \vdots \\ p_{N1} & \cdots & p_{NN} \end{pmatrix}_{N \times N} \begin{pmatrix} r_1^t \\ r_2^t \\ \vdots \\ r_j^t \\ \vdots \\ r_N^t \end{pmatrix}_{N \times 1}.$$

텔레포트(teleport)는 스파이더 트랩 그리고 데드엔즈와 같은 문제점을 해결하기 위해서 도입한 개념으로 웹 서퍼가 어느 웹 페이지에 도달했을 때, 일정한 확률로 간선을 타고 이동하거나 다른 곳으로 순간 이동한다는 설정이다. 이때의 확률은 감쇠 계수(damping coefficient)로 보통 0.8~0.9 값을 사용한다.

이러한 텔레포트를 통해서 기존의 전이행렬  $P$ 를 새로운 행렬인 전이행렬  $A$ ,  $A = \beta P + (1-\beta)\left[\frac{1}{N}\right]_{N \times N}$ 로 변경한다. 이때  $\beta$ 는 감쇠 계수다. 이와 같은 식으로  $\beta$ 의 확률로 간선을 타고 가거나  $1-\beta$ 의 확률로  $N$ 개의 웹페이지들 중 무작위의 확률로 다른 사이트로 이동한다는 텔레포트의 개념을 만족시킬 수 있다. 이때의 행렬  $A$

가 초창기 구글행렬엔진이라 부르며, 페이지랭크를  $r^{t+1} = Ar^t$ 로 수정할 수 있다.

☞ 용어 설명

Li의 강의노트 [3]에 따라 다음과 같이 용어를 본 사사 논문에서 사용한다.

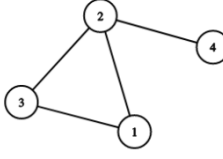
그래프 이론(Graph theory)	네트워크 이론(Network theory)	해당 항목
		
꼭지점(Vertex)	노드(Node)	1,2,3,4 (총 4개)
선(Edge)	간선(Edge) 또는 링크(Link)	$\overline{12}$ , $\overline{13}$ , $\overline{23}$ , $\overline{24}$ (총 4개)

그림6. 용어 비교 설명

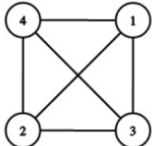
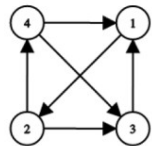
	
무향 그래프(Undirected graph)	방향 그래프(Directed graph)

그림7. 방향에 따른 그래프

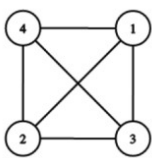
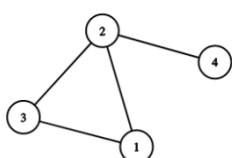
	
완전 연결 그래프 (Complete connected graph)	단순 연결 그래프 (Simple connected graph)

그림8. 연결 그래프의 종류

- Loop: 임의의 점에서 시작해서 그 같은 점에 도착하는 선
- Connected graph: 선에 의해 연결된 그래프
- Complete graph: 임의의 두 점이 하나의 선에 연결된 그래프
- Simple graph: 2개 이상의 선 또는 루프(loop)를 가지고 있지 않은 그래프

### III. 연구 방법

일상생활에서 일어나는 문제를 합리적으로 이해, 추론하기 위해서 확률 행렬의 한 종류인 마르코프 체인을 학습, 이해하고, 서로의 생각의 차이를 공유하였다. 현실에서 많이 사용되고 있는 마르코프 체인을 기반으로 하는 페이지 랭크를 함께 공부하면서, 검색 알고리즘을 이해하는 것에 초점을 맞추어 학습하였다. 이렇게 이해한 내용을 우리 주변에서 볼 수 있는 단순한 도로망에 적용해 봄으로써 실생활 문제를 논리적 또는 수학적인 도구를 사용해서 해결하고자 한다.

### IV. 연구 결과

본 과정은 결과 도출 단순화를 위한 방향 그래프이며, 단순 연결 그래프라 가정하고 결과를 도출하였다.

서울시 언북초 주변 어린이보호구역은 여러 일방통행로로 이루어져 있다(간선으로 나타낼 시 무향 그래프가 아닌 유향 그래프로 나타낼 수 있음)



그림9. 언북초 주위의 도로, 교차로 모습

이를 인접 그래프로 나타내기 위해 도로가 만나는 교차로를 노드, 도로를 간선으로 나타내었다. 인접 그래프를 이용해서 전이 행렬을 구하면 아래와 같이 나타낼 수 있다.

위 사진일 경우 텔레포트를 도입할 필요가 없어진다. 하지만, 실제 도로에서는 스파이더 트랩이나 테스트랩이 있는지, 없는지를 모르기에, 감쇠계수를 포함해야 한다.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$



그림10. 언북초 주위의 도로, 교차로를 방향그래프로 나타낸 모습

이런 경우에선 E부분에서 테스트 트랩이 나타나 감쇠 계수를 적용해야한다. 전이행렬을 구하면 아래가 된다.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



그림11. 방향그래프로 나타낸 모습

이 전이행렬을 구글행렬( $A = \beta P + (1-\beta)[\frac{1}{N}]_{N \times N}$ ) M은 전이행렬, N은 노드의 개수,  $\beta$ 는 감쇠계수 0.8, 초기값 r을  $\frac{1}{N}$  로 계산한다.

$$0.8 \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + 0.2 \begin{pmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{pmatrix} = \begin{pmatrix} \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{23}{25} & \frac{1}{25} & \frac{23}{25} & \frac{23}{25} \\ \frac{25}{25} & \frac{75}{25} & \frac{25}{25} & \frac{75}{25} & \frac{75}{25} \\ \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix}$$

구글행렬 계산 후 페이지랭크 공식,  $r^{t+1} = Ar^t$ 을 반복해서 값을 업데이트한다.

$$\begin{aligned}
 r^0 &= \begin{pmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \end{pmatrix} \implies r^1 = \begin{pmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{25} \end{pmatrix} = \begin{pmatrix} \frac{1}{5} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix} \begin{pmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \end{pmatrix} \implies r^2 = \begin{pmatrix} \frac{121}{625} \\ \frac{121}{625} \\ \frac{283}{1875} \\ \frac{121}{625} \\ \frac{121}{625} \end{pmatrix} = \begin{pmatrix} \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix} \begin{pmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \end{pmatrix} \\
 \implies r^3 &= \begin{pmatrix} \frac{1739}{9375} \\ \frac{473}{3125} \\ \frac{1339}{9375} \\ \frac{1739}{9375} \\ \frac{287}{9375} \end{pmatrix} = \begin{pmatrix} \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix} \begin{pmatrix} \frac{121}{625} \\ \frac{121}{625} \\ \frac{283}{1875} \\ \frac{121}{625} \\ \frac{121}{625} \end{pmatrix} \implies r^4 = \begin{pmatrix} \frac{34903}{234375} \\ \frac{11101}{78125} \\ \frac{88469}{703125} \\ \frac{41303}{234375} \\ \frac{6523}{234375} \end{pmatrix} = \begin{pmatrix} \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix} \begin{pmatrix} \frac{1739}{9375} \\ \frac{473}{3125} \\ \frac{1339}{9375} \\ \frac{1739}{9375} \\ \frac{287}{9375} \end{pmatrix} \\
 \implies r^5 &= \begin{pmatrix} \frac{486949}{3515625} \\ \frac{49021}{390625} \\ \frac{411829}{3515625} \\ \frac{506149}{3515625} \\ \frac{87313}{3515625} \end{pmatrix} = \begin{pmatrix} \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{21}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix} \begin{pmatrix} \frac{34903}{234375} \\ \frac{11101}{78125} \\ \frac{88469}{703125} \\ \frac{41303}{234375} \\ \frac{6523}{234375} \end{pmatrix}
 \end{aligned}$$

실제로 어느 값에 도달하며 수렴하게 되지만 그 부분은 너무 길어 생략한다.

## V. 결론 및 제언

일상생활에서 일어나는 문제를 합리적으로 이해, 추론하기 위해서 마르코프 체인을 학습, 이해한 후, 마르코프 체인을 바탕으로 하는 페이지랭크의 원리에 대해 살펴본 논문이다. 이러한 원리를 중학생 사사과정 학생들이 주도적으로 이해하고 실제 우리 주변에서 볼 수 있는 도로망에 적용해 봄으로써 학생들에게 새로운 수학 관점을 제시한 측면에서 가치가 있다고 생각한다.

## VI. 참고문헌

- [1] 바닥부터 배우는 강화 학습, 노승은, 영진닷컴
- [2] 수학으로 풀어보는 강화학습 원리와 알고리즘, 박성수, 위키 북스
- [3] Jiatong Li (LOGEN), A BRIEF OVERVIEW OF GRAPH THEORY, Lecture note.  
([http://math.uchicago.edu/~may/REU2021/REUPapers/Li,Jiatong\(Logen\).pdf](http://math.uchicago.edu/~may/REU2021/REUPapers/Li,Jiatong(Logen).pdf))

- 참고 웹페이지

- [4] 마코프 체인을 활용한 구매자 이동  
매드타임스(MADTimes): <http://www.madtimes.org>
- [5] 페이지랭크 (velog.io):  
<https://velog.io/@babydeveloper/%ED%8E%98%EC%9D%B4%EC%A7%80%EB%9E%AD%ED%81%AC-%EC%A0%84%ED%8C%8C-%EB%AA%A8%EB%8D%B8>



수학B

## 03\_Conway의 생명 게임 탐구



# Conway의 생명 게임 탐구

김지안 (솔밭중학교 3학년)

최여진 (서경중학교 3학년)

지도교수 : 김동원 (청주교육대학교 수학교육과)

(논문 구성은 아래에 제시한 목차를 활용하여 작성하며, 연구주제의 특성에 따라 변경할 수 있습니다.)

단, 초록은 글꼴(신명조) 및 크기(10pt), 줄간격(130%),

본문 내용의 글꼴(신명조), 크기(11pt), 줄간격(160%)는 변경 불가합니다.)

초록(300-500자) : (글자크기 10) 반드시 포함

## I. 서론 (또는 연구의 필요성 및 목적)

어렸을 때부터 보드게임 하는 것을 좋아했기에 ‘콘웨이의 생존게임’이라는 주제에 흥미를 가지게 되었고, 매우 간단한 규칙임에도 그 복잡성과 가능성이 무한하다는 점에 주목하여 연구하게 되었다.

## II. 이론적 배경

### 정의 1. (콘웨이의 생명게임, Conway's Game of Life)

바둑판처럼 정사각형의 여러 칸으로 나뉘어진 공간에서 한 칸에 한 마리씩 있는 세포들의 삶과 죽음이 펼쳐지는 게임으로, 영국의 수학자 존 호튼 콘웨이 가 만들었다고 하여 ‘콘웨이의 생명게임’ 이라고 불린다.

### 정의 2. (에덴 동산, Gardens of Eden)

에덴 동산은 부모 세대를 갖지 않는 패턴이다. 즉, 어떤 패턴의 진화에서도 절대 나타날 수 없는 패턴이다.

### 정의 3. (oscillator)

일정한 주기를 가지고 끊임없이 반복되는 구조의 세포들을 말한다

### 정의 4. (토큰, token)

여기서 토큰은 수월한 설명을 위해 주고 받을 수 있는 무언가라는 의미로 사용한 단어이다.

### <규칙>

콘웨이의 생명게임은 하나의 세대를 기점으로 생존한 세포를 표시하는데, 이때 임의의 칸에 접한 8개의 칸 중 살아있는 세포가 있는 칸 수에 따라 해당 칸의 세포가 다음 세대에 생존할지 죽을지가 결정된다.

- 세포가 죽어있는 칸과 인접한 칸 중 세포가 살아있는 칸이 정확히 3개일 때  
>> 다음 세대에 죽은 세포가 살아난다
- 세포가 살아있는 칸과 인접한 칸 중 세포가 살아있는 칸이 2개 혹은 3개일 때  
>> 해당 칸의 세포는 살아있는 상태를 유지한다
- 그 외의 경우(세포가 죽어있는 칸과 인접한 칸 중 세포가 살아있는 칸이 3개가 아니거나, 세포가 살아있는 칸과 인접한 칸 중 세포가 살아있는 칸이 2개 혹은 3개가 아닐때)  
>> 세포가 죽은 상태를 유지하거나 살아있는 세포가 죽는다

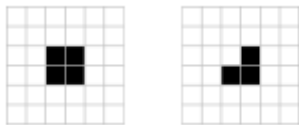
### III. 연구방법

#### [과제 I] 에텐 동산의 존재

[정리 1] 에텐 동산은 항상 존재한다.

$n$ 을 1 이상의 임의의 자연수로 정하고  $6n \times 6n$  정사각형 내의 모든 패턴을 고려한다. 1세대 중앙  $(6n-2) \times (6n-2)$  정사각형의 내용은 0세대의 원래  $6n \times 6n$  정사각형의 내용에만 의존한다. 중앙의  $(6n-2) \times (6n-2)$  정사각형은  $(6n-2)^2$ 개의 세포를 포함하며, 각각은 살아있거나 죽었으므로 중앙 정사각형을 채울 수 있는 패턴은  $2^{(6n-2)^2}$ 개가 있다. 이때 우리는  $n$ 이 충분히 클 때 부모가 없는 패턴이 존재함을 보이면 된다. 이를 위해  $6n \times 6n$  사각형을  $6 \times 6$  크기의  $n^2$ 개의 타일로 분할한다. 각 타일은 36개의 세포를 가지며, 이때 각 세포는 죽었거나 살았거나의 두 가지 상태가 있으므로 총  $2^{36}$ 만큼의 서로 다른 타일이 있다.

그러나 <그림 1>의 두 타일은 주변에 배치된 다른 타일에 관계없이 동일한 방식으로 진화한다. 따라서 우리는 첫 번째 타일을 사용하는 모든 타일에서 전체 패턴의 진화를 변경하지 않아도 그것을 두 번째 타일로 대체 가능하다. 즉, 1세대에 존재 가능한 모든  $(6n-2) \times (6n-2)$  패턴을 분류하고 싶다면, 0세대의  $2^{36-1}$ 의 서로 다른 타일로 만든 패턴의 자식만을 고려하면 된다.



<그림 1>

$6n \times 6n$  정사각형에는 이러한 타일이  $n^2$ 개 만큼 있으므로, 가능한 자식 (1세대에 존재하는 패턴)에는 최대  $(2^{36-1})^{n^2}$ 개가 있다. 또한 중앙의  $(6n-2) \times (6n-2)$

정사각형을 채울 수 있는 패턴에는  $2^{(6n-2)^2}$  만큼이 있다. 이때 이들 중 최대  $(2^{36}-1)^{n^2}$  개는  $6n \times 6n$  정사각형의 자식으로 나타난다. 남은 것은  $n$ 이 충분히 클 때  $(2^{36}-1)^{n^2} < 2^{(6n-2)^2}$  임을 보이는 것이다. 양변에  $n^2$ 근을 취하면 위의 식은  $2^{36}-1 < 2^{(6-2/n)^2}$  이 되고,  $n$ 을 무한대로 보내면  $2/n$ 은 0에 수렴하므로 위의 식은 성립한다. ■

### [과제 II] still life 문제

[정리 1] (still life density)  $n \times n$  상자 내의 still life는  $n^2/2+2n$  개를 초과하는 세포를 가지지 않는다. 또,  $n$ 이 무한대로 갈 때 still life density는  $1/2$ 을 넘지 않는다

[comments]  $n$ 이 매우 커지면  $n^2$ 에 비해  $2n$ 은 상대적으로 작기에 없는 취급을 해도 되기 때문에, 첫 번째 명제가 증명되면 두 번째 명제도 함께 증명된다는 것을 알 수 있다.

<증명>

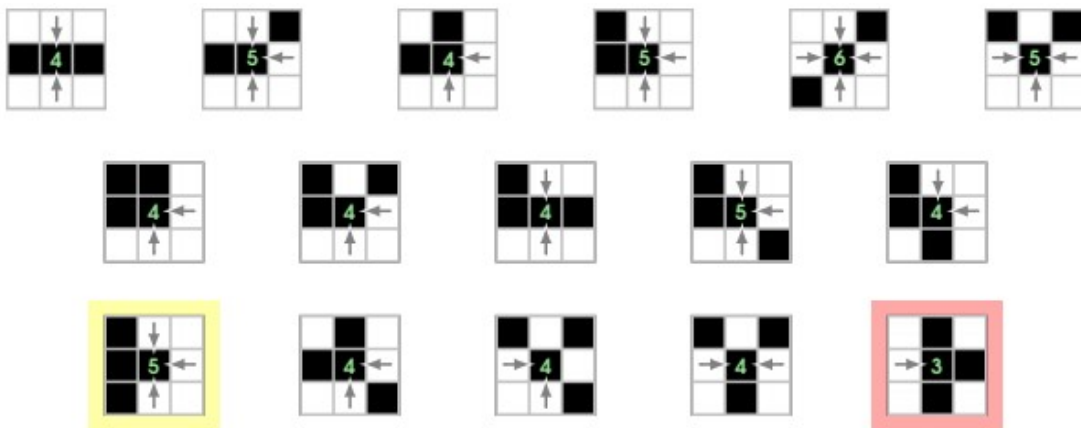
먼저 살아있는 세포와 죽은 세포 모두 2개의 토큰을 가지고 있다고 한다. 그리고 이 토큰들은 수평과 수직 방향으로 인접한 세포 (<그림 7>에서 검은색으로 색칠된 부분)들에게 재분배된다.



<그림 2>

결과적으로 각 세포들은 최소 4개의 토큰을 가지게 되는데, 다음은 이 사실을 증명하는 과정이다.

우선, 가운데 세포가 계속 생존하기 위해선 주변에 2개 혹은 3개의 살아있는 세포만을 두어야 한다. 가운데 세포를 중심으로 2개 혹은 3개의 살아있는 세포가 있는 경우들을 모두 찾으면 <그림 8>과 같다. 단, 돌리거나 뒤집어서 같아지는 경우는 제외한다.



<그림 3> 가운데 세포를 중심으로 2개 또는 3개의 살아있는 세포가 있는 16가지 경우

이때 회색 화살표는 죽은 세포들이 어디로 토큰을 주는지(단, 세포를 어떻게 분배할지는 (<그림 9> 참고), 초록색 숫자는 가운데 세포가 최종적으로 가지는 토큰의 개수를 의미한다.



<그림 4>

죽은 세포가 인접한 세포에 토큰을 줄 때 <그림 9>와 같이 인접한 세포가 1~2 개라면 토큰을 1개 씩, 인접한 세포가 3개라면 마주보는 세포에 각각 1개씩, 인접한 세포가 0,4개라면 토큰을 주지 않는다.

이 숫자들을 보면 16번째 경우(빨간색 그림)를 제외하면 모두 4 이상이다. 즉, 증명을 완료하기 위해 이 경우를 해결해야 하는데, 16번째 경우는 12번째 경우(노란색 그림)와 본질적으로 같다. 왜냐하면 16번째 경우의 오른쪽 줄 가운데 세포(이 세포는 12번째 경우의 가운데 세포와 동일)는 이미 3개의 살아 있는 세포와 인접하므로 더 이상 살아있는 세포와 인접할 수 없다. 16번째 경우의 오른쪽 방향으로 세로줄이 하나 더 있다고 하면, 그 줄에는 살아있는 세포가 존재하지 않음을 의미하고, 따라서 12번째 경우와 16번째 경우는 동일한 구조임을 알 수 있다. 따라서 각 세포들은 최소 4개의 토큰을 가진다.

이제 이 사실을 이용하여 주어진 명제를 증명할 것이다.

$mn$  상자 내의 살아있는 세포 개수를  $K$ 개라고 하면 각 세포마다 최소 4개의 토큰을 가지므로 총 토큰의 수는  $4K$ 개 이상이다. 따라서 다음 부등식이 성립한다.

$$\begin{aligned}
 4K &= \langle n \times n \text{ 상자 내의 살아있는 세포 전체의 토큰 수} \\
 &= \langle n \times n \text{ 상자 내의 모든 세포의 토큰 수} \\
 &= \langle 2(n^2+4n) \text{ (이 값은 } mn \text{ 상자 내의 모든 토큰 수 } 2n^2 \text{과 이웃한 } 4n \text{개의 세포로부터 온 토큰 수인 } 8n \text{의 합이다.)}
 \end{aligned}$$

따라서  $K = \langle n^2/2 + 2n$ 이 성립한다.

[문제 1]  $m \times n$  상자 내의 still life는  $mn/2 + m + n$ 개를 초과하지 않는다.

<해답>

[정리 1]의 내용을 수용하되, 결론 부분만 약간 수정하면 된다.

$$\begin{aligned}
 4K &= \langle m \times n \text{ 상자 내의 살아있는 세포 전체의 토큰 수} \\
 &= \langle m \times n \text{ 상자 내의 모든 세포의 토큰 수} \\
 &= \langle 2mn + 4m + 4n \\
 &\text{(이 값은 } mn \text{ 상자 내의 모든 토큰 수 } 2mn \text{과 이웃한 } 2m + 2n \text{개의 세포로부터 온 토큰 수의 합이다.)}
 \end{aligned}$$

∴ 따라서  $K \leq mn/2+m+n$ 이 성립한다.

[정리 1-2] (still life density)  $n \times n$  상자 내의 still life는  $6n^2/11+24n/11+24/11$ 개 를 초과하는 세포를 가지지 않는다.  
또,  $n$ 이 무한대로 갈 때 still life density는  $6/11$ 을 넘지 않는다.

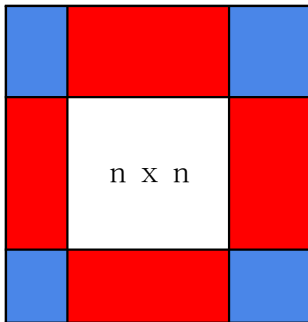
[comments] [정리 1-2] 는 [정리 1] 과 [정리 2]를 연계하여 보다 특정된 수치를 이 끌어냈다.

<증명>

이 증명에선 [정리 1]의 증명에서의 분배 방식과 다른 방식으로 토큰을 분배할 것이다.

여기서의 분배 방식은 ‘죽은 세포는 근처의 8개의 세포 중 살아있는 세포에게 최소 1개의 토큰을 준다’ 이다. 이때 우리는 [정리 2] 에서 죽어있는 세포 근처에 올 수 있는 살아있는 세포 수는 최대 6개라는 사실을 증명했으니 토큰 수가 음수가 되지 않게 하기 위해선 세포가 최소 6개의 토큰을 가져야 함을 알 수 있다. 그러니 [정리 1]의 증명에서 세포 하나당 2개의 토큰을 가졌다고 한 가정과 달리 세포 하나당 6개의 토큰을 가졌다고 하자.

이후엔 [정리 1]의 증명과 동일하다. 다만 [정리 1]에선 살아있는 세포가 최소 4개의 토큰을 가지지만, 지금의 분배 방식으로 토큰을 분배하면 살아있는 세포가 최소 11개의 토큰을 가지게 된다. 또한 인접한 세포들로부터 오는 토큰 수 역시 구하는 방법이 [정리 1] 과는 다르다. [정리 1]에선 단순히 이웃한 세포 수  $4n$ 에 2를 곱하면 되었으나, 분배 방식이 달라진 지금, 변끼리 이웃한 세포만이 아닌 꼭짓점끼리



<그림 5>

맞닿은 세포들도 고려해야 한다. (<그림 10> 참고)

[정리 1]에선 빨간색 부분의 세포들만 고려했다면, 여기선 파란색 세포들까지 고려한다.

결국  $mn$  상자 밖에서 오는 토큰 수는  $6(4n+4)=24n+24$ 이다.

따라서 최종 식(중간 과정은 [정리 1]과 동일하여 생략)은

$$11K \leq 6n^2+24n+24$$

따라서  $K \leq 6n^2/11+24n/11+24/11$ 이 성립한다.

[정리 2] 임의의 죽어있는 세포 주변에는 살아 있는 세포 수가 6개 이하이다

<증명>

편의 상, 3 x 3 상자의 각 칸에 <그림 11> 과 같이 번호를 부여한다.

1	2	3
4	5	6
7	8	9

<그림 6>

이 상자에서 세포 5가 죽은 세포라고 할 때, 나머지 1, 2, 3, 4, 6, 7, 8, 9의 8개 세포 중 살아있는 세포가 6개를 초과하지 않는다는 것을 보이면 된다. still life의 근본적 특징인 세포의 구성이 바뀌지 않는다는 것이다.

즉, 죽어있는 세포는 계속 죽어있는 상태여야 하고, 살아있는 세포는 계속 살아있어야 그 모양을 still life 라고 부를 수 있다. 따라서 세포 5를 둘러싸고 있는 나머지 8개의 세포 중 살아있는 세포는 4개 이상의 살아 있는 세포를 이웃으로 가질 수 없다.

인구 밀집으로 인해 살아있던 세포가 죽어서 세포의 구조가 바뀌기 때문이다. 이때 세포 1, 3, 7, 9는 고려할 필요가 없다. 왜냐하면 4개의 세포의 주위에 올 수 있는 세포들은 각각 2,4,5 / 2,5,6 / 4,5,8 / 5,6,8로 모두 최대 3개이기 때문에 인구 밀집이 발생할 수 없기 때문이다.

반면 세포 2, 4, 6, 8은 각각 이웃세포로 1,3,4,6 / 1,2,7,8 / 2,3,8,9 / 4,6,7,9의 최대 4개를 가질 수 있으므로 이 세포들을 중심으로 살펴봐야 한다.

여기서 발견할 수 있는 흥미로운 특징 중 하나는 세포 2, 4, 6, 8 이 이웃 세포로 가질 수 있는 세포들은 5를 제외한 세포들이 두 번 씩 중복되어 있다는 점이다.

따라서 4개의 세포 2, 4, 6, 8이 최대로 가질 수 있는 이웃 세포의 수는  $3 \times 4 = 12$  인데, 이 중 절반이 중복되므로 그림 상에서 최대로 가질 수 있는 이웃 세포의 수는  $3 \times 4 \div 2 = 6$ 개가 된다.

### [과제 III] oscillator

[정리 1] 주기 4의 oscillator에는 주기가 4인 세포가 반드시 2개 이상이다.

comments. 주기 4의 oscillator 라도 그 세포들은 주기가 4가 아니어도 된다.

여기서 말하는 ‘주기’ 는 가능한 한 작은 숫자로 합을 전제로 하기에 주기 4의 oscillator 이더라도 이를 구성하는 세포들은 주기가 4의 약수들이나 것도 가능하기 때문이다.

<증명>

[정리 1]은 주기 4의 oscillator 에서 주기가 4인 세포가 오직 1개만 존재하는 것이 불가능하다는 말과 같다. 즉, 이 논제를 대신 증명하여도 된다.

4의 주기를 가지는 유일한 세포를 K 라고 하자. K의 상태(죽었는지 살았는지)는 주기 0,1,2 와 주기 4에서 각각 같을 수도 있고, 주기 0,1 과 주기 3,4 에서 각각 같을 수도 있다. 주기 0,2 와 주기 1,3이 각각 같을 경우 주기가 4가 아닌 2가 되기에 불가능하다. 또한 주기 0 과 주기 1,2,3 에서 각각 같은 경우는 앞서 말한 주기 0,1,2 와 주기 4 에서 각각 같은 경우와 동일하므로 제외한다.

K의 상태로 가능한 경우가 2가지(중복제외)임을 확인하였으므로 각 경우에서 모순이 생김을 보이면 된다.

(i) 주기 0,1,2 와 주기 4가 같은 경우 - a a a b 꼴

K를 제외한 나머지 세포들은 주기가 4는 아니지만 주기 4의 oscillator 의 구성원이므로 comments. 에서 말했던 것처럼 주기가 1이거나 2이다. 즉 주기 0과 주기 2에서 상태가 동일하다. K 자신 역시도 a a a b 꼴이므로 주기 0과 주기 2에서 상태가 동일하므로, K는 주기 1에서와 주기 3에서도 상태가 동일해야 한다. 주기 1,3을 결정짓는건 바로 이전 단계인 주기 0,2 이기 때문이다. 이는 K의 주기가 4라는 전제에 위배되므로 모순이다.

(ii) 주기 0,1 과 주기 3,4 가 같은 경우 - a a b b 꼴

보다 효과적인 설명을 위해 a가 생존 상태, b가 죽은 상태라고 가정하자 (반대여도 논리는 동일하다). 주기 4에서 주기 0으로 넘어갈 때 죽어있던 K가 살아나야 하므로 주기 4에서는 주위 세포가 3개여야 한다. 이때, K의 상태가 변하더라도 나머지 세포의 주기에는 변화가 없으므로 (i)과 같이 K 외의 세포들은 주기 0, 주기 2와 주기 1, 주기 3에서 상태가 각각 같다. 이는 주기 4에서 주위 세포가 3개라면 주기 2에서도 주위 세포가 3개여야 함을 의미하고, 이렇게 되면 주기 2(생존 상태) 에서 주기 3(죽은 상태)으로 넘어가는 게 불가능하므로 모순이다. ■

#### IV. 연구결과, V. 결론

- [정리 1] 에덴 동산은 항상 존재한다.
- [정리 1] (still life density)  $n \times n$  상자 내의 still life는  $n^2/2+2n$  개를 초과하는 세포를 가지지 않는다. 또,  $n$ 이 무한대로 갈 때 still life density는  $1/2$ 을 넘지 않는다
- [문제 2]  $m \times n$  상자 내의 still life는  $mn/2+m+n$ 개를 초과하지 않는다.
- [정리 1-2] (still life density)  $n \times n$  상자 내의 still life는  $6n^2/11+24n/11+24/11$ 개를 초과하는 세포를 가지지 않는다. 또,  $n$ 이 무한대로 갈 때 still life density는  $6/11$ 을 넘지 않는다.
- [정리 2] 임의의 죽어있는 세포 주변에는 살아 있는 세포 수가 6개 이하이다.

#### VI. 참고문헌

Conway's game of life Mathematics and construction  
by Nathaniel Johnston and Dave Greene

수학B

## 04\_트리플카의 연구 사사수학 논문



# 트리플카의 연구

## 카프리카, 켄프리카, 리버시카

강은서 (용문초등학교 6학년)

김지민 (해원초등학교 6학년)

이경민 (동래초등학교 6학년)

임지훈 (명륜초등학교 6학년)

주지오 (해원초등학교 6학년)

지도교수 : 김성준 (부산 교육대학교 수학교육학과)

### 초록

수학에는 여러 가지의 규칙과 특별한 수들이 있습니다. 우리 연구에서는 이러한 수 중 카프리카 상수에서 우리 팀만의 특수한 두 상수를 만들어냈습니다. 수에서 그 자릿수들의 수로 배열할 수 있는 두 번째로 큰 자연수에서 두 번째로 작은 수를 지속해서 뺀을 때 반복해서 나오는 수를 만들고 이를 켄프리카 상수라고 이름을 붙였고, 수를 뒤집었을 때 더해서 자릿수가 모두 다 같은 수가 나오는 수를 리버시카 상수라고 이름을 붙였습니다. 그리고 그 두 수에 관한 연구를 하며 그 수들의 규칙들과 공식들, 그 상수에는 어떤 수가 있는지 찾아내기도 했습니다. 거기에 더해 파이썬(python)이랑 스크래치(scratch)인 두 코딩 프로그램을 이용하여 코딩을 해 수를 입력하면 주기가 얼마인지 나오는 켄프리카 수 프로그램을 만들어내고, 또 수를 입력하면 리버시카 상수인지 아닌지 판단할 수 있는 리버시카 상수 프로그램을 만들어냈습니다.

# I. 서론

## 1. 연구 동기

연구주제 선정을 하면서 특수한 숫자나 공식들을 조사하고 있는 중간에 카프리카 상수를 발견하여 더 알아보면서 흥미를 느끼기 시작했다. 그러다 카프리카 상수를 조사하다 카프리카 상수의 특수한 규칙이 신비하게 느껴졌다. 조사를 더욱 하다 보니 우리 팀만의 상수를 만드는 것이 재밌을 것이라고 느꼈기에 우리 팀만의 규칙으로 켄프리카 상수랑 리버시카 상수를 만들어 규칙, 공식, 증명 등의 연구를 시작했다.

## 2. 연구 목적

가. 카프리카 상수의 정의와 수들을 알아본 다음, 카프리카 상수의 규칙을 찾아보고 그 규칙들을 연구해 본다.

나. 카프리카 상수의 규칙을 활용하여 우리만의 상수인 켄프리카 상수와 리버시카 상수를 만들어본다.

다. 켄프리카 상수와 리버시카 상수를 파이썬이랑 스크레치 같은 프로그램을 사용하여 코딩으로 나타내 본다.

## 3. 연구의 필요성

이 상수들의 정의, 공식, 규칙, 증명을 보여줌으로써 수학에 흥미를 느끼지 않는 사람들도 이 상수들을 보고 수학에 대해서 흥미를 느끼고 재미를 느끼고 더 깊게 조사하게 될 수 있다.

## 4. 연구문제

가. 켄프리카 상수를 정의하고 그 규칙을 찾아 증명하고 켄프리카 상수를 찾고 확인하기 위한 코딩을 통해 켄프리카 상수를 탐구한다.

리버시카 상수의 규칙, 조건을 구하고 증명을 한 뒤 리버시카 상수의 자릿수에 따른 모든 리버시카 상수들을 찾는 코딩과 리버시카 상수인지 찾는 코딩을 만들며 리버시카 상수를 연구했다.

## II. 이론적 배경

1. 카프리카 상수의 유래: 카프리카 상수는 인도 수학자 D. R. Kaprekar (1905 - 1986)에 의해 1949년에 발명이 되었다. 카프리카가 살던 마을에 3025라고 적혀 있는 표지판이 있었는데 그 표지판이 넘어지면서 쪼개어져 30과 25로 나누어졌다. 그러면서 55의 제곱은 3025이다 라는 발견에서부터 카프리카 수가 생겨났다.

2. 카프리카 상수의 정의: 수를 abcd를 예로 두면 a,b,c,d로 만들 수 있는 가장 큰 수에서 가장 작은 수를 빼는 것을 반복했을 때 다시 abcd가 나오는 신기한 상수를 말한다.

3. 카프리카 상수의 규칙: 두 자릿수의 경우에는 두 자릿수가 같은 수인 경우를 제외했을 때 9가 나오게 되는 규칙을 볼 수 있고, 세 자릿수의 경우에는 자릿수가 모두 같은 경우를 제외했을 때 규칙을 반복하다 보면 495가 나온다는 것을 알 수 있다. 네 자릿수도 항상 6174에 도달한다.

예) 95

$$95-59=36$$

$$63-36=27$$

$$72-27=45$$

$$54-45=9$$

예) 325

$$532-235=297$$

$$972-279=693$$

$$963-369=594$$

$$954-459=495$$

예) 1872

$$8721-1278= 7443$$

$$7443-3447=3996$$

$$9963-3699=6264$$

$$6642-2466=4176$$

$$7641-1467=6174$$

4. 카프리카 상수의 원리: 네 자릿수의 경우 a, b, c,d,0 순으로 같지 않고 큰 수부터 나열했다고 하자. 그러면  $abcd-dbca=ABCD$

그러면  $10+d-a= D$

$9+c-b=C$

$B=b-1-c$

$A=a-d$

ABCD가 될 수 있는 조합은 abcd, bacd, abdc,...등으로 24가지가 있다. 그중 하나인 abdc를 고르면 연립했을 때 나오는 결과는 a=7, b=6, c=4, d=1

따라서 ABCD는 6174가 되는 것을 볼 수 있다. 다른 경우도 마찬가지로의 결과가 나온다.

그럼 세 자릿수를 증명하면 a, b, c, 0 순서로 큰 수부터 나열했다고 했을 때  $abc-cba=ABC$

$C=10+c-a$

$B=9$

$A=a-c-1$

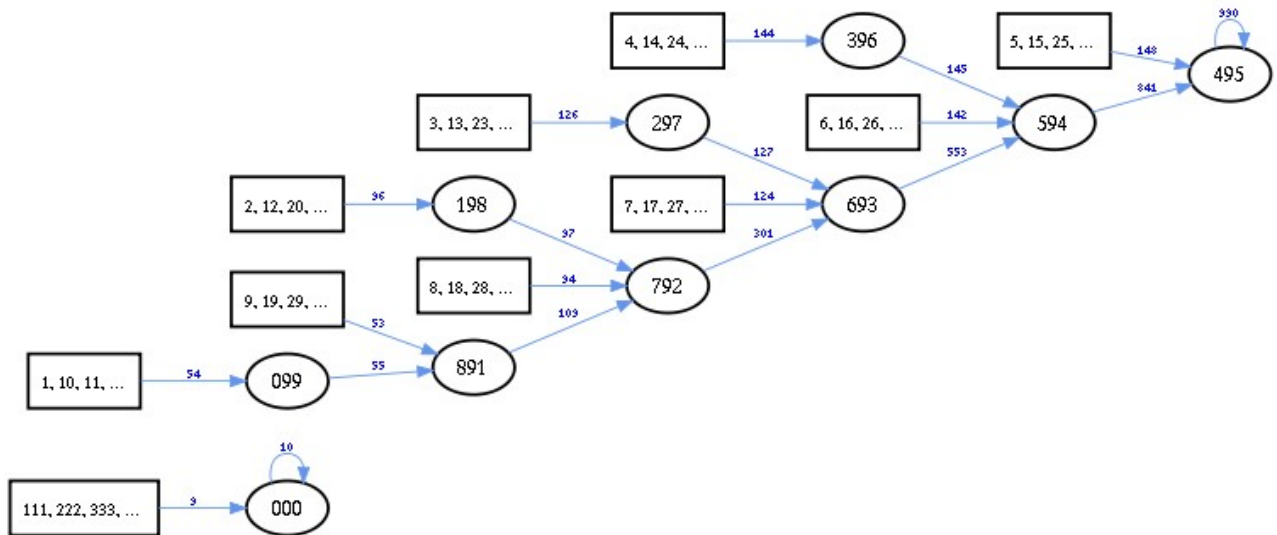
ABC가 될 수 있는 조합은 여기서 6가지이다. 이 중 하나인 bac를 고르자면 a=9 b=5 c=4

따라서 ABC는 495가 되는 것을 볼 수 있다.

5. 대표적인 카프리카 상수: 6174는 7641-1467을 했을 때 6174로 나오기로 유명한 카프리카 수 중 대표적인 수이다.

6. 카프리카 상수의 세 자릿수, 네 자릿수 과정

### Kaprekar Process for Three Digits





## 7. 카프리카 수의 코딩

카프리카 수의 코딩을 직접 하여 밑에 나타난 것처럼 코딩하였다.

```
main.py +
1 def kaprekar_constant(n):
2     while n != 6174:
3         num_str = str(n).zfill(4) # 4자리 숫자로 만들기 위해 0으로 채움
4         asc_num = int(''.join(sorted(num_str)))
5         desc_num = int(''.join(sorted(num_str, reverse=True)))
6         n = desc_num - asc_num
7         print(f"{desc_num} - {asc_num} = {n:04d}")
8
9     return n
10
11 initial_number = int(input("4자리 양의 정수를 입력하세요: "))
12 kaprekar_constant(initial_number)
```

프로그램을 실행시켰을 때 임의로 1234를 넣어보자.

```
1 def kaprekar_constant(n):
2     while n != 6174:
3         num_str = str(n).zfill(4) # 4자리 숫자로 만들기 위해 0으로 채움
4         asc_num = int(''.join(sorted(num_str)))
5         desc_num = int(''.join(sorted(num_str, reverse=True)))
6         n = desc_num - asc_num
7         print(f"{desc_num} - {asc_num} = {n:04d}")
8
9     return n
10
11 initial_number = int(input("4자리 양의 정수를 입력하세요: "))
12 kaprekar_constant(initial_number)
```

Ln: 1, Col: 1

Run Share Command Line Arguments

```
1234
4321 - 1234 = 3087
8730 - 378 = 8352
8532 - 2358 = 6174

** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

직접 계산을 해 보았을 때 1234에서

$$4321-1234=3087$$

$$8730-378=8352$$

$$8532-2358=6174$$

결과적으로 6174가 나옴을 볼 수 있다. 프로그램을 활용했을 경우 위의 식까지 나오며 정확한 것을 볼 수 있다.

### Ⅲ. 연구방법

#### 1. 켄프리카 상수의 연구방법

켄프리카 상수의 정의를 찾은 뒤, 세 자리, 네 자리수 같은 조건들을 넣어 더 세밀한 연구를 진행했다. 그러면서 켄프리카 상수의 조건을 찾아내고, 겹치는 수, 홀수, 짝수 등에 있는지 검산을 했다. 그 상수가 어떤 경우에도 성립하는지 알아보고 만약 그렇지 않다면, 변수들을 찾아낸다. 그다음 단계로, 이 상수가 왜 우리가 정한 조건들에서 만족하는지 증명을 하여 일반화했다.

수정한 뒤 스크래치(Scratch)를 이용하여 기본값을 코딩을 해내었다. 프로그램에서 얻은 값들을 표로 정리를 하여 연구를 끝냈다.

#### 2. 리버시카 상수의 연구 방법

리버시카 상수란 원래 있던 수를 뒤집었을 때 모든 자릿수가 다 똑같은 수가 나오는 수라고 정의를 하여 이름을 붙였다. 여기서 뒤집었다, 반대로라는 뜻이 되는 'reverse'이라는 영어 단어를 활용하여 reverse-kar이라는 이름을 만들어내었다.

리버시카 상수를 자릿수로 나누어 조사하여 자릿수마다 예시들을 구해내면서 규칙들을 구할 수 있을지 구해보았다. 예시를 들어 두 자릿수에서는 12, 32, 45, 63등으로 무엇이 되는지, 세자릿수에는 121, 345, 246, 135등의 어떤 것들이 되는지 조사를 해 보았다.

리버시카 수의 개수에 대한 공식을 세우기 위해 두 자릿수, 세 자릿수의 개수를 구해보았다. 조사한 결과와 규칙들을 바탕으로 리버시카 수의 자릿수(n)에 따른 개수의 공식을 만들어보았다. 그 뒤 공식이나 연구에 문제나 계산 실수가 없는지 다시 검산하여 공식이 정확한지 보았다. 그 뒤로 파이썬(Python)이라는 코딩 프로그램을 이용하여 리버시카 상수인지 아닌지 알려주는 프로그램을 만들어 검사해 보았다.

## IV. 연구결과

### 1. 켈프리카 상수

가. 정의 : 임의의 수에서 자릿값을 바꾸어 배열한 수 가운데 두 번째로 큰 수에서 두 번째로 작은 수를 빼는 과정을 반복하면 나오는 수라고 정의를 하였다. (세 자릿수, 네 자릿수 만 가능하다)

<두 번째로 큰 수-두 번째로 작은 수>

나. 규칙 : 예외적인 경우를 제외하면 세 자릿수에서의 켈프리카 수는 450, 네 자릿수에서의 켈프리카 상수는 9009가 된다.

켈프리카 세 자릿수의 예시를 들어보았다.

597을 예시로 두자면

597

$$957-597=360$$

$$603-063=540$$

$$504-054=450$$

$$504-054=450$$

그러면 이번에는 758을 예시로 두자면

$$857-587=270$$

$$702-072=630$$

$$603-063=540$$

$$504-054=450$$

450을 다시 계산하면  $504-054=450$ 으로 계속 반복되므로, 세자릿수에서 450이 결과적으로 나온다는 것을 볼 수 있다.

이번에는 네 자릿수를 예시로 들어보자.

1348을 예시로 뒤 보자면

$$8413-1384=7029$$

$$9702-0297=9405$$

$$9504-0495=9009$$

2314도 예로 들어보자.

$$4312-1243=3069$$

$$9603-0396=9207$$

$$9702-0297=9405$$

$$9504-0495=9009$$

따라서 네 자릿수는 9009가 결과적으로 나옴을 볼 수 있다.

다. 세 자릿수인 쟈프리카 수에 대한 증명:

$A > B > C$ 라고 가정하자.

두 번째로 큰 수- 두 번째로 작은 수=

$$(100A+10C)-(100C+10A)=100(A-C)-10(A-C)=90(A-C)임을 볼 수 있다.$$

$9 \geq A > B > C \geq 0$ 이므로

$5.90(A-C) > 0$ 이고  $A-C=1$ 이 될 수 없음을 볼 수 있다.

따라서  $A-C=2, 3, 4, 5, 6, 7, 8, 9$ 임을 볼 수 있다.

$A-C=2$ 가정

$$90 \times 2 = 180$$

$$801-081=720$$

$$702-072=630$$

$$603-063=540$$

$$504-054=450$$

$A-C=3$ 가정

$$90 \times 3 = 270$$

$$702-072=630$$

$$603-063=540$$

$$504-054=450$$

$A-C=4$ 가정

$$90 \times 4 = 360$$

$$603-063=540$$

$$504-054=450$$

$A-C=5$ 가정

$$90 \times 5 = 450$$

$A-C=6$ 가정

$$90 \times 6 = 540$$

$$504-054=450$$

$A-C=7$ 가정

$$90 \times 7 = 630$$

$$603-063 = 540$$

$$504-054 = 450$$

A-C=8가정  
90X8=720  
702-072=630  
603-063=540  
504-054=450

A-C=9가정  
90X9=810  
801-081=720  
702-072=630  
603-063=540  
504-054=450

앞의 결과에 따라 알 수 있는 것들은

2를 넣었을 때 4번  
3을 넣었을 때 3번  
4를 넣으면 2번  
7을 넣으면 2번  
8을 넣으면 3번

9를 넣으면 4번 만에 나오는데 이는 거울처럼 과정이 접었을 때 겹치는 수와 같다는 것을 볼 수 있다. 단, 4, 5는 450과 540으로 예외에 속한다. 그러나 이것도 0번, 1번으로 큰 차이는 없다.

라. 잼프리카 상수의 예외:

① 모든 자릿수의 숫자가 짝수이거나 홀수면 마지막에 7227에서 4545가 되는 결과를 볼 수 있다.

예시로 7513을 들자면  
7513-1375=6138  
8613-1386=7227  
7272-2727=4545

예시를 이번에는 8624로 두자.

.8624-2486=6138  
8613-1386=7227  
7272-2727=4545  
ex.8602-0286=8316  
8613-1386=7227  
7272-2727=4545

따라서 위의 규칙이 성립한다는 것을 볼 수 있다.

② 큰 수, 작은 수의 개념이 네 자릿수가 같으면 없으므로 자릿수가 모두 같으면 절대 안 된다는 규칙도 존재한다.

③ 같은 수 안에서 자릿값이 2개까지 똑같은 것은 가능하다. 그러나, abab, abba, aabb같은 네 자릿수의 경우는 결괏값이 909로 세 자릿수가 나온다.

예시: 9191:  $9191-1919=7272$

$7272-2727=4545$

$5454-4545=909$

예시: 6226:  $6262-2626=3636$

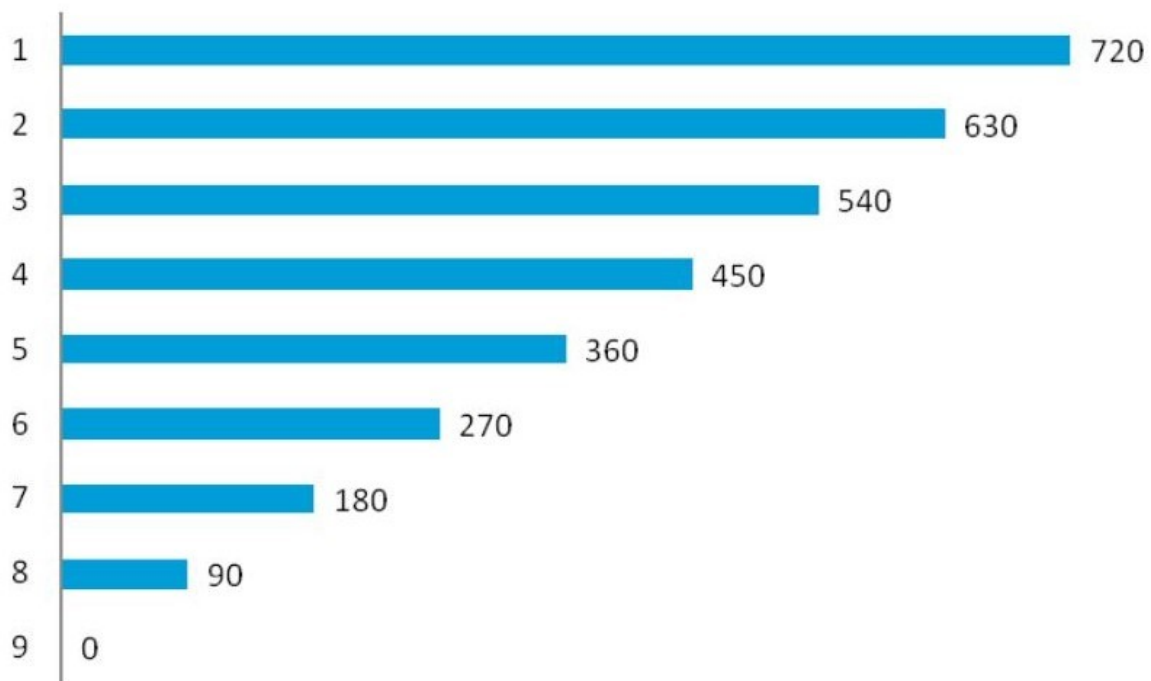
$6363-3636=2727$

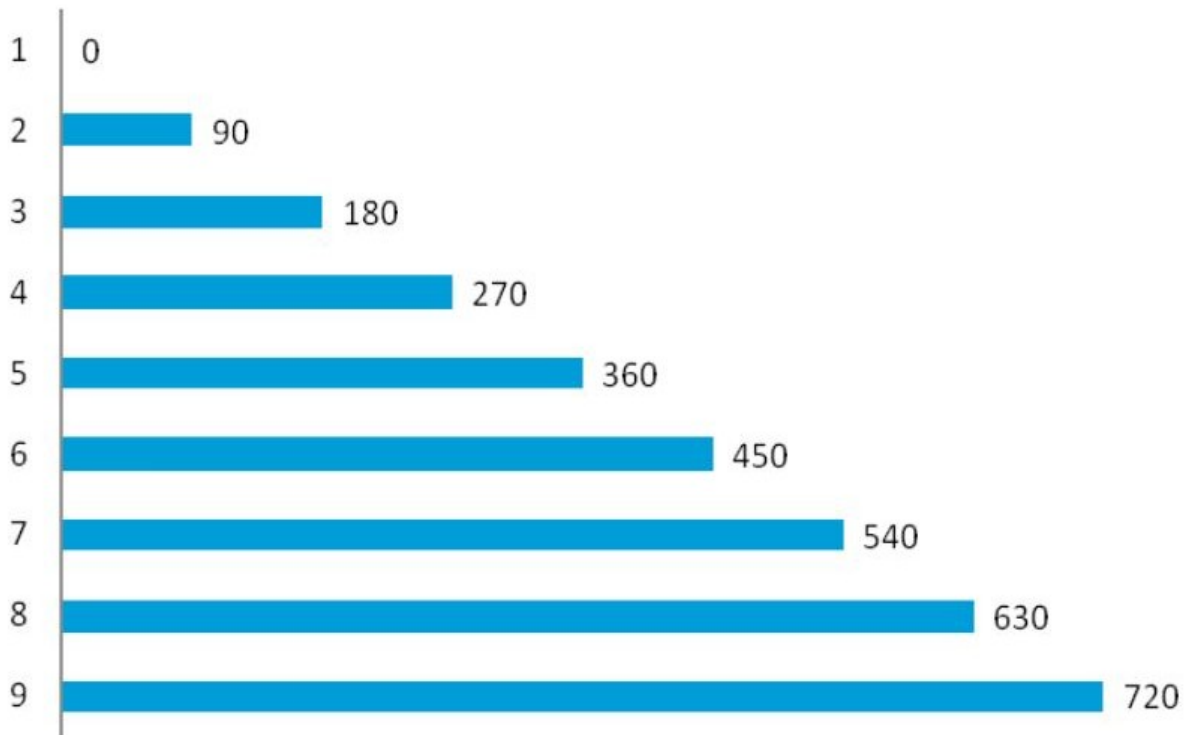
$7272-2727=4545$

$5454-4545=909$

예시: 8877:  $8787-7878=909$

4. aaba, abaa 같은 경우는 특이한 특징이 있다는 것을 볼 수 있다.

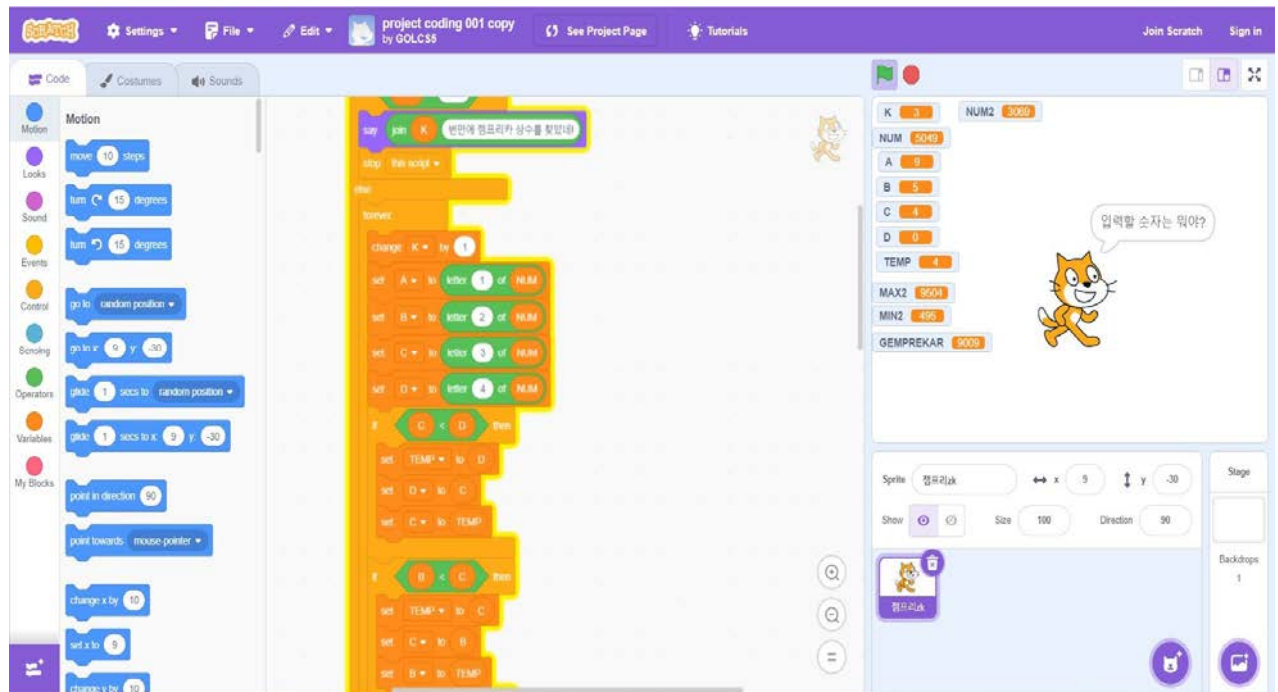




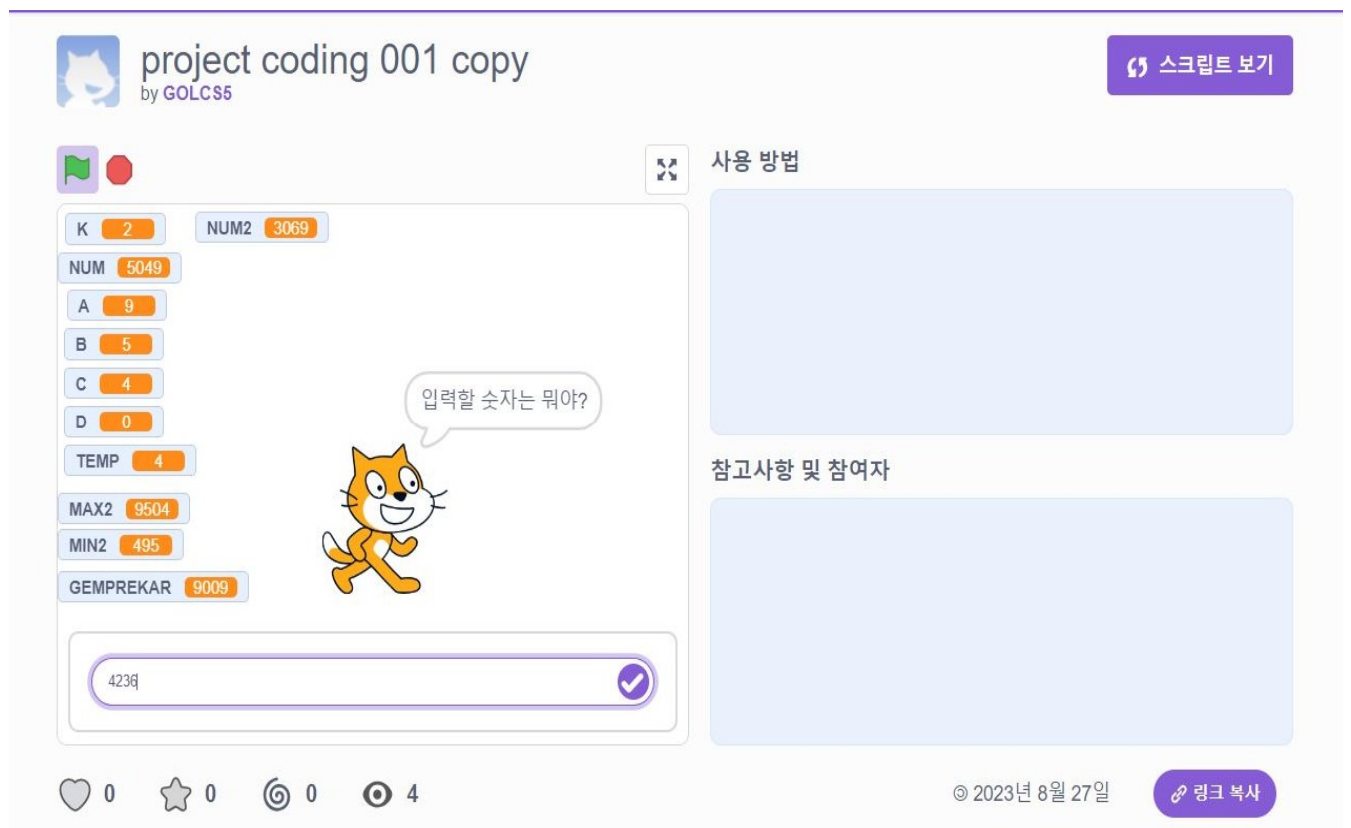
예를 들어,  $b$ 가 1이고,  $a$ 가 2이면 720이고, 9이면 90처럼  $a$ 가 1씩 늘 때마다 곱값은 90씩 빠지는 규칙이 있다. 이 경우는 매우 특이하다고 생각이 돼 그래프로 나타내어 보았다.

또한, 이 그래프는  $b$ 가 9라고 하였을 때  $a$ 가 1부터 8, 720부터 90으로 가능함으로써 모든 수가 720에서 90까지가 될 수 있다.

## 마. 쟈프리카 상수의 코딩



위와 같이 쟈프리카 상수를 스크래치를 이용하여 코딩했다. 코딩에 의해서 결과값이 9009가 항상 나옴을 볼 수 있고, 9009까지 몇 번의 계산을 해야지 나오는지 알 수 있다.



만약 이 프로그램에 4236이라는 숫자를 입력하여 보자.

4236:

6423-2364=4059

9504-0495-9009

실제로 프로그램에서도 2번 만에 찾았다는 것을 볼 수 있다.

그럼 이번에는 7425를 예시로 입력하여 보자.

7425:

7524-2475=5049

9504-0495-9009

project coding 001 copy  
by GOLCS5

스크립트 보기

사용 방법

참고사항 및 참여자

© 2023년 8월 27일 링크 복사

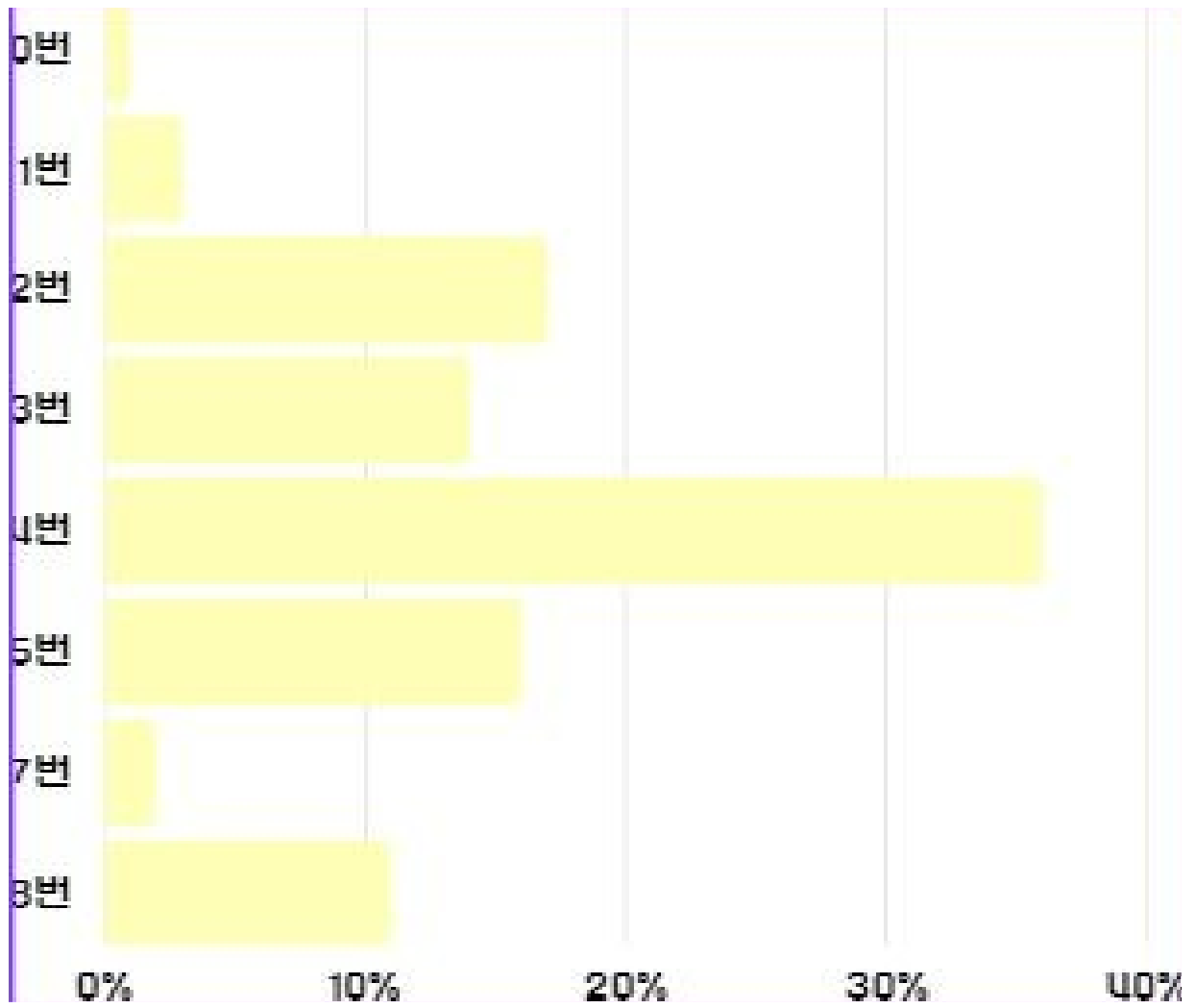
이번 결과에서도 9009가 두 번 만에 나오는 결과를 볼 수 있다.

위의 코딩은 우리가 직접 제작을 한 코딩이며, 정확하다는 것을 볼 수 있다.

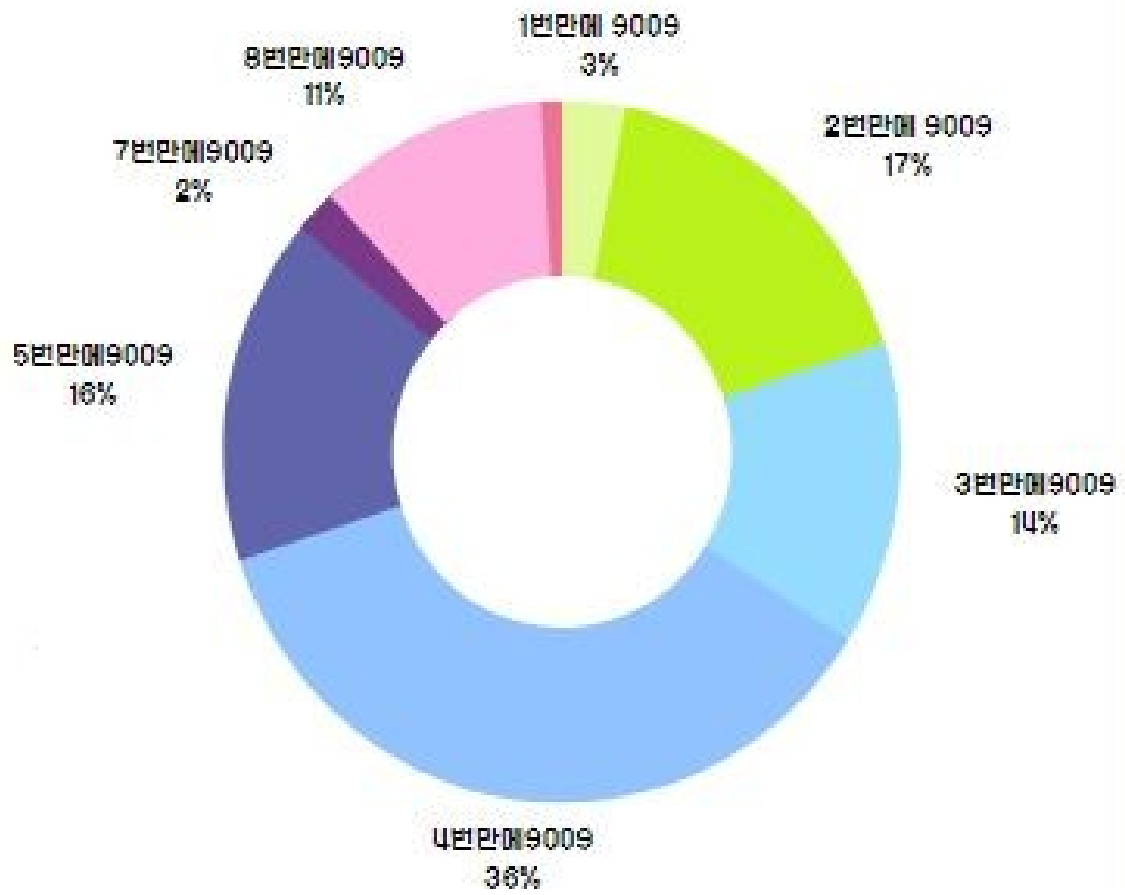
바. 잼프리카 상수의 변수를 제외한 나머지 지수들의 그래프

우리가 직접 만든 스크래치 코딩으로 여러 가지 네 자릿수를 대입시킨 후 엑셀로 정리한 원그래프와 띠 그래프로 나타내었다.

\*예외는 제외(앞에 있는 예외들)



↑ 몇 번 만에 네 자릿수가 나오는지 백분율로 나타낸 띠 그래프 ↑



↑ 몇 번 만에 네 자릿수가 나오는지 백분율을 나타내는 원그래프 ↑

## 2. 리버시카 상수

가. 정의 : 숫자를 거꾸로 하여 더했을 때, 그 값의 자릿수가 모두 똑같은 수

예) 1122, 3030 등

나. 규칙/공식

1. 두 자릿수: 자릿수의 합이 0보다 크고 10보다 작은 모든 수

$$(0 < a < 10) \rightarrow (0123456789)$$

2. 세 자릿수: 다 똑같은 자릿수를 가졌거나 연속되는 세 자릿수를 가진 수 (단, abc라고 했을 때  $a+c < 10$ ,  $b < 5$ 이어야 함)

$$(예: 111, 222, 123, 246 \text{ 등})$$

자릿수가 3 초과인 짝수일 때의 증명: abcdef에서  $a+f$ ,  $b+e$ ,  $c+d < 10$

합=9:

$$a+f, b+e, c+d = (9\text{가지}, 10\text{가지}, 10\text{가지})$$

합=8

$a+f, b+e, c+d = (8, 9, 9)$  합=1일 때까지 이 규칙으로 반복

자릿수가 3 초과인 짝수일 때 위의 규칙은 반복 그때 공식은

( $k=1\sim 9$ 인 수)

$$n-2/2$$

$$k \times (k+1)$$

2. 자릿수가 3 초과인 홀수면 증명: abcdefg에서  $a+g$ ,  $b+f$ ,  $c+e < 10$ 인 짝수,  $d < 5$

합=8: ( $a+g, b+f, c+e$ ) = (8가지, 9가지, 9가지)

합=6, 4, 2일 때 (6, 7, 7), (4, 5, 5), (2, 3, 3)

자릿수가 3 초과인 홀수일 때 규칙이 반복되므로 규칙은

$$n-3/2$$

$k \times (k+1)$ 임을 볼 수 있다.

( $k=1\sim 9$ 인 짝수의 경우를 모두 다 더함)

$$n-3/2$$

$k \times (k+1)$ 임을 볼 수 있다.

다. 리버시카 상수의 공식 증명

자릿수가 3 초과인 짝수일 때의 증명:

abcdef (여섯 자릿수 중 임의의 수)에서  $a+f$ ,  $b+e$ ,  $c+d$ 는 모두 10보다 작아야 한다.

합이 9인 경우 ( $a, f$ )는 (1, 8), (2, 7), (3, 6), (4, 5), (5, 4), (6, 3), (7, 2), (8, 1), (9, 0)으로 9가지임을 볼 수 있다. ( $b+e$ ), ( $c+d$ )는 둘 다 (9, 0), (1, 8), (2, 7), (3, 6), (4, 5), (5, 4), (6, 3), (7, 2), (8, 1), (9, 0)으로 각각 10가지이다.

합이 8인 경우에는 (a, f)는 (1, 7), (2, 6), (3, 5), (4, 4), (5, 3), (6, 2), (7, 1), (8, 0)으로 8가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (8, 0), (1, 7), (2, 6), (3, 5), (4, 4), (5, 3), (6, 2), (7, 1), (8, 0)으로 각각 9가지이다.

합이 7인 경우에는 (a, f)는 (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1), (7, 0)으로 7가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (0, 7), (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1), (7, 0)으로 각각 8가지이다.

합이 6인 경우에는 (a, f)는 (1, 5), (2, 4), (3, 3), (4, 2), (5, 1), (6, 0)으로 6가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (0, 6), (1, 5), (2, 4), (3, 3), (4, 2), (5, 1), (6, 0)으로 각각 7가지이다.

합이 5인 경우에는 (a, f)는 (1, 4), (2, 3), (3, 2), (4, 1), (5, 0)으로 5가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (5, 0), (1, 4), (2, 3), (3, 2), (4, 1), (5, 0)으로 각각 6가지이다.

합이 4인 경우에는 (a, f)는 (1, 3), (2, 2), (3, 1), (4, 0)으로 4가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (0, 4), (1, 3), (2, 2), (3, 1), (4, 0)으로 각각 5가지이다.

합이 3인 경우에는 (a, f)는 (1, 2), (2, 1), (3, 0)으로 3가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (0, 3), (1, 2), (2, 1), (3, 0)으로 각각 4가지이다.

합이 2인 경우에는 (a, f)는 (1, 1), (2, 0)으로 2가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (0, 2), (1, 1), (2, 0)으로 각각 3가지이다.

합이 1인 경우에는 (a, f)는 (1, 0)으로 1가지임을 볼 수 있다. (b+e), (c+d)는 둘 다 (0, 1), (1, 0)으로 각각 2가지이다.

따라서 자릿수가 3 초과인 짝수일 때 위의 규칙이 반복된다는 것을 볼 수 있다.

그때 공식은 (k=1~9인 경우 모두 다 더함)

$$\frac{n-2}{2} \times (k+1)$$

임을 볼 수 있다.

2. 자릿수가 3 초과인 홀수면 증명:

abcdefg에서  $a+g$ ,  $b+f$ ,  $c+e$ 는 10보다 작은 짝수,  $d$ 는 5보다 작은 수( $a+g$ ,  $b+f$ ,  $c+e$ 의 절반)이어야 한다.

합은 8인 경우에는  $(a+g)$ 는 (1, 7), (2, 6), (3, 5), (4, 4), (5, 3), (6, 2), (7, 1), (8, 0)으로 8가지이고  $b+f$ ,  $c+e$ 는 (8, 0), (1, 7), (2, 6), (3, 5), (4, 4), (5, 3), (6, 2), (7, 1), (8, 0)으로 9가지,  $d$ 는 4로 정해져 있다.

합은 6인 경우에는  $(a+g)$ 는 (1, 5), (2, 4), (3, 3), (4, 2), (5, 1), (6, 0)으로 6가지이고  $b+f$ ,  $c+e$ 는 (0, 6), (1, 5), (2, 4), (3, 3), (4, 2), (5, 1), (6, 0)으로 7가지,  $d$ 는 3으로 정해져 있다.

합은 4인 경우에는  $(a+g)$ 는 (1, 3), (2, 2), (3, 1), (4, 0)으로 4가지이고  $b+f$ ,  $c+e$ 는 (0, 4), (1, 3), (2, 2), (3, 1), (4, 0)으로 5가지,  $d$ 는 3으로 정해져 있다.

합은 2인 경우에는  $(a+g)$ 는 (1, 1), (2, 0)으로 2가지이고  $b+f$ ,  $c+e$ 는 (0, 2), (1, 1), (2, 0)으로 3가지,  $d$ 는 1로 정해져 있다.

따라서 자릿수가 3 초과인 홀수일 때 위의 규칙이 반복되므로 공식이 ( $k=1\sim 9$ 인 짝수의 경우를 모두 다 더함)

$$n-3/2$$

$k \times (k+1)$ 임을 볼 수 있다.

라. 리버시카 상수의 코딩

리버시카 상수 또한 카프리카 상수와 켈프리카 상수와 마찬가지로 파이썬 베타 (코딩 프로그램)으로 코딩을 하였다.

### 1. 리버시카 상수인지 아닌지 판단하는 코딩 프로그램

The image shows two screenshots of a Python IDE. The top screenshot shows the code for reversing a number. The code defines a function `reverse_number` that takes a number and returns its reverse. It then prompts the user to enter a 4-digit number. If the number is between 1000 and 9999, it reverses the number and prints the result, then calculates the sum of the original and reversed numbers and prints it. If the number is not in the range, it prints an error message.

```
main.py +
3
4 def reverse_number(number):
5     reversed_number = int(str(number)[::-1])
6     return reversed_number
7
8 input_number = int(input("4자리 수를 입력하세요: "))
9
10 if 1000 <= input_number <= 9999:
11     reversed_input = reverse_number(input_number)
12     print("역수로 변환된 숫자:", reversed_input)
13
14     sum_number = input_number + reversed_input
15     print("역수와 합한 수는:", sum_number, "입니다.")
16
17     compare_number = sum_number
18 else:
Ln: 1, Col: 1
[Stop] [Share] Command Line Arguments
4자리 수를 입력하세요:
7963
```

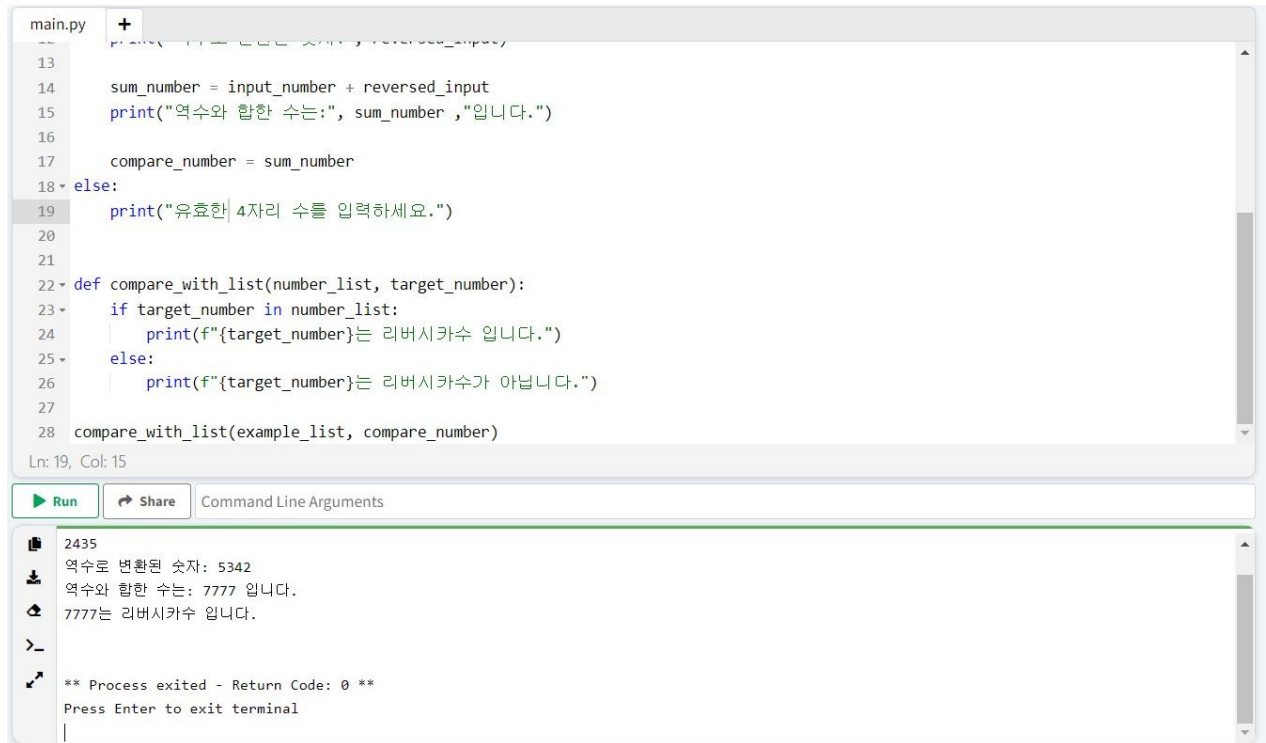
The bottom screenshot shows the same code but with an additional function `compare_with_list` that checks if a number is a palindromic number. It prompts the user to enter a 4-digit number. If the number is between 1000 and 9999, it reverses the number and prints the result, then calculates the sum of the original and reversed numbers and prints it. If the number is not in the range, it prints an error message. The `compare_with_list` function checks if the number is in a list of palindromic numbers and prints the result.

```
main.py +
9
10 if 1000 <= input_number <= 9999:
11     reversed_input = reverse_number(input_number)
12     print("역수로 변환된 숫자:", reversed_input)
13
14     sum_number = input_number + reversed_input
15     print("역수와 합한 수는:", sum_number, "입니다.")
16
17     compare_number = sum_number
18 else:
19     print("유효한 4자리 수를 입력하세요.")
20
21
22 def compare_with_list(number_list, target_number):
23     if target_number in number_list:
24         print(f"{target_number}는 리버시카수 입니다.")
Ln: 1, Col: 1
[Run] [Share] Command Line Arguments
7963
역수로 변환된 숫자: 3697
역수와 합한 수는: 11660 입니다.
11660는 리버시카수가 아닙니다.
** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

앞의 사진처럼 예시를 들어 7963을 예시로 들어보았다.

$7963+3697=11660$ 이므로 리버시카 상수가 아님을 판단할 수 있다.

그러면 이번에는 다른 수를 넣어보자.



The screenshot shows a Python IDE with a file named 'main.py'. The code defines a function 'compare\_with\_list' that checks if a target number is a palindromic number. The terminal output shows the execution of the program with the input '2435'. The output is as follows:

```
Ln: 19, Col: 15  
Run Share Command Line Arguments  
2435  
역수로 변환된 숫자: 5342  
역수와 합한 수는: 7777 입니다.  
7777는 리버시카수 입니다.  
** Process exited - Return Code: 0 **  
Press Enter to exit terminal
```

2435를 넣어보면

$2435+5342=7777$ 로 리버시카 상수임을 볼 수 있다.

따라서 코딩 프로그램이 옳음을 볼 수 있다.

## 2. 리버시카 상수를 찾는 프로그램

```
3
4 for num in range(1000, 10000):
5     sym_sum = num + int(str(num)[::-1]) # 대칭값을 더한 합 계산
6
7     if (sym_sum == 1111) | (sym_sum == 2222)|(sym_sum == 3333)|(sym_sum == 4444)|(sym_sum == 5555) | (sym_sum == 6666) | (sym_sum ==
8         print(f"{num}")
```

Ln: 1, Col: 1

Run Share Command Line Arguments

8080  
8091  
8170  
8181  
8260  
8271  
8350  
8361  
8440

```
3
4 for num in range(1000, 10000):
5     sym_sum = num + int(str(num)[::-1]) # 대칭값을 더한 합 계산
6
7     if (sym_sum == 1111) | (sym_sum == 2222)|(sym_sum == 3333)|(sym_sum == 4444)|(sym_sum == 5555) | (sym_sum == 6666) | (sym_sum ==
8         print(f"{num}")
```

Ln: 1, Col: 1

Run Share Command Line Arguments

4703  
4714  
4725  
4804  
4815  
4905  
5050  
5061  
5070

위처럼 프로그램을 실행했을 때 모든 리버시카 네 자릿수 상수들이 나오는 것을 볼 수 있다.

## V. 결론

젼프리카 상수:

1. 우리만의 규칙을 사용하여 젼프리카 상수를 만들었다.
2. 직접 어떤 수가 젼프리카 상수인지 구했다.
3. 젼프리카 상수를 증명하여 규칙을 찾아내었다.
4. 젼프리카 상수의 예외들을 찾아 직접 확인해 보았다.
5. 젼프리카 상수를 스크래치 코딩으로 만들어 예외가 아닌 수들을 계산해 보았다.
6. 젼프리카 상수의 규칙을 조사하여 원그래프와 띠그래프로 나타냈다.
7. 젼프리카 상수의 최댓값과 최솟값을 구하고 가장 많은 비율의 수도 조사했다.

리버시카 상수:

1. 우리만의 규칙을 사용하여 리버시카 상수를 만들었다.
2. 리버시카 상수의 정의를 직접 구해보았다.
3. 리버시카 상수를 자릿수의 홀수, 짝수에 따라 증명해 보고, 그를 기반으로 규칙을 찾았다.
4. 규칙을 기반으로 리버시카 상수의 코딩을 하였다.
5. 첫 번째 코딩 : 수를 대입하였을 때 그 수가 리버시카 상수인지 판정해 준다.
6. 두 번째 코딩 : 리버시카 상수를 찾아준다.

알게 된 점:

이경민:수학이라는 것은 생각보다 어렵다는 것을 알게 되었다. 카프리카 수를 만든 D.R.Kaprekar 같은 특수적인 수를 만드는 훌륭한 수학자들이 정말 대단하다는 사실을 알게 되었다.

김지민:카프리카 상수의 규칙을 보며 단순한 규칙이어도 더 깊이 파고 들어가면 더 심화한 내용을 찾을 수 있다는 점이 놀라웠다.

강은서:젼프리카 상수를 연구하며 규칙과 패턴을 찾을 수 있었고, 젼프리카의 조건들에 의해 세 자릿수부터 가능하다는 것을 알게 되었다. 또한, 수의 세계에서는 젼프리카, 리버시카 이외에도 정말 많고 다양한 상수들이 존재한다는 것도 알 수 있었다.

임지훈:카프리카 상수를 알게 되었고 이에 대한 우리가 직접 만든 쟀프리카, 리버시카 수로 인해 더 많은 것을 배울 수 있었다. 또한, 직접 만든 스크래치 코딩으로 코딩에 대해서도 배울 수 있었다. 물론, 배우는 과정에서 다툼이 있긴 했어도 언제나 협동심이 필요하다는 걸 알았다. 1명이라도 부족하면 안 되니까 말이다.

주지오: 수학의 세계를 탐구하는 것은 가능성이 무한하다. 그리고 우리는 수학의 세계에서 쟀프리카와 리버시카라는 상수를 찾았고 그것을 코딩하며 코딩에 관해서도 깊이 생각해 보았다. 그리고 무엇보다도 팀원과의 협력이 중요하다는 것을 알았다.

## VI. 참고문헌

<https://post.naver.com/viewer/postView.naver?volumeNo=31377745&memberNo=5565159>

<https://plus.maths.org/content/mysterious-number-6174>

[https://ko.wikipedia.org/wiki/%EC%B9%B4%ED%94%84%EB%A6%AC%EC%B9%B4\\_%EC%83%81%EC%88%98](https://ko.wikipedia.org/wiki/%EC%B9%B4%ED%94%84%EB%A6%AC%EC%B9%B4_%EC%83%81%EC%88%98)

카프리카 상수 네자릿수, 세자릿수 순서도:

By Rdhettinger - 자작, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=15502883>



수학B

## 05\_펠 방정식의 일반화



# 펠 방정식의 일반화

최민준 (형곡중학교 3학년)

지도교수 : 김민훈 (경북대학교 수학교육과)

## 초 록

본 연구에서는 기존에 알려진 펠 방정식의 다양한 성질들과 해를 구하는 여러 방법들 그리고 다각수와의 연관성들을 다양한 예시와 함께 정리하고 소개하였다. 펠 방정식과 쌍곡선과의 관계성에 착안하여 펠 방정식을 고차원으로 일반화하는 방법을 제시하였다. 이 일반화된 고차원 펠 방정식의 해를 파이썬을 이용하여 찾는 방법을 다루었다. 이 과정에서 찾은 해들 사이의 규칙을 발견하였고 이를 이용하여 일반화된 고차원 펠 방정식의 무한한 해의 존재성을 증명하였다.

## I. 서론

그림 1과 같이 쌍곡선  $x^2 - 2y^2 = 1$  위에 있는 모든 정수 격자점을 찾는 문제를 생각해보자.

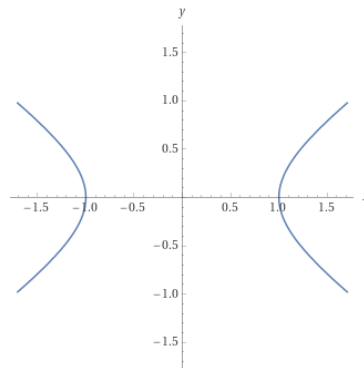


그림 1. 쌍곡선  $x^2 - 2y^2 = 1$  그래프 개형.

쉽게 찾을 수 있는 격자점은  $(1, 0)$ ,  $(3, 2)$  등이다. 실제로 이 점들이 쌍곡선 위에 놓이는 사실은  $1^2 - 2 \times 0^2 = 1$ ,  $3^2 - 2 \times 2^2 = 1$ 의 식으로 확인할 수 있다. 이어지는 질문은 이 쌍곡선이 무한히 많은 자연수 격자점을 포함하는지의 여부이고 이는 자연스럽지만 쉽게 알 수 있는 내용은 아니다. 일반적으로 자연수  $d$ 에 대해 방정식  $x^2 - dy^2 = 1$ 을 펠의 방정식 (Pell's equation)이라고 한다. 역사적으로 이 펠의 방정식의 정수해를 찾는 문제가 Archimedes, Bhaskara, Fermat, Lagrange 등에 의해 활발히 연구되었고  $d$ 가 완전제곱수인 경우에는 정수해가 유한하지만,  $d$ 가 완전제곱수가 아닌 경우에는 무한히 많은 정수해가

존재한다는 사실이 밝혀졌다. 이 과정 속에서 다각수(polygonal numbers), 연분수(continued fraction), 무리수  $\sqrt{d}$ 의 유리수 근사(rational approximation)등과 펠 방정식의 정수해와의 연관성들이 발견되며 정수론의 기초적이면서 중요한 주제가 되었다.

본 연구에서는 펠 방정식이 2차원 유클리드 평면 위의 쌍곡선에 의해서 주어진다는 점에 착안하여, 기하학적으로 고차원 유클리드 공간 안의 쌍곡공간(hyperboloid)에 의해서 주어지는 고차원 펠 방정식을 생각하였다. 이 일반화된 펠 방정식 중 일부가 무한히 많은 정수해를 가진다는 것을 증명하였다.

이 논문은 다음과 같이 구성되어 있다. II절에서는 펠 방정식을 포함하는 일반적인 디오판틴 방정식의 예제들과 흥미로운 사실들을 다룬다. III절에서는 펠 방정식과 다각수와의 연관성을 다룬다. IV절에서는 기존에 알려진 펠 방정식의 해를 찾는 여러 가지 방법들을 다룬다. V절에서는 일반적인 고차원 펠 방정식을 소개하고 주요결과인 특정 일반화된 펠의 방정식이 무한히 많은 정수해를 가진다는 사실의 증명을 다룬다.

## II. 이론적 배경

### 1. 디오판틴 방정식

**디오판틴 방정식(Diophantine equation)**은 다항방정식의 정수해 또는 유리수해를 찾는 것으로 정의된다. 정리하자면 정수 혹은 유리수 계수를 갖는 다항식  $F_1, F_2, \dots, F_k$ 에 대해  $F_1(x_1, \dots, x_m) = \dots = F_k(x_1, \dots, x_k) = 0, x_i \in \mathbb{Z} \text{ or } \mathbb{Q}$ 의 해를 찾는 것이다.

(예시) 디오판틴 방정식의 여러 예시들이다.

(i) 일차 부정방정식  $ax + by = c$ 의 정수해.

(ii) 피타고라스 삼원쌍(Pythagorean triple)  $x^2 + y^2 = z^2$ 의 정수해 또는 유리수해. 중요한 내용이기예 (2.피타고라스 삼원쌍)에서 자세히 다루겠다.

(iii) 펠 방정식  $x^2 - dy^2 = 1$ 의 정수해.

(iv) 페르마의 마지막 정리  $x^n + y^n = z^n$ 의 정수해 또는 유리수해.

(v) 중국인의 나머지 정리(Chinese Remainder Theorem)

## 2. 피타고라스 삼원쌍(Pythagorean triple)

직각삼각형의 세 변의 길이를 각각  $a, b, c$ 라고 할 때 피타고라스 정리  $a^2 + b^2 = c^2$ 을 만족하는 순서쌍  $(a, b, c) \in \mathbb{N}$ 을 피타고라스 삼원쌍이라 한다.

(정리) 피타고라스 삼원쌍  $(a, b, c) = (k(2mn), k(m^2 - n^2), k(m^2 + n^2))$ 이다.

(증명)  $a^2 + b^2 = c^2$  ( $a, b, c \in \mathbb{N}$ ) 식 변형을 하면, 다음과 같다.

$$\begin{aligned} a^2 + b^2 &= c^2 \\ \left(\frac{a^2}{c^2}\right) + \left(\frac{b^2}{c^2}\right) &= 1 \\ \left(\frac{a}{c}\right)^2 + \left(\frac{b}{c}\right)^2 &= 1 \end{aligned}$$

위 식은 원의 방정식 풀이고,  $\left(\frac{a}{c}\right) = x$ ,  $\left(\frac{b}{c}\right) = y$ 라 하면  $x^2 + y^2 = 1$ 이다.

따라서  $a^2 + b^2 = c^2$  ( $a, b, c \in \mathbb{N}$ )은 부정방정식의 해를 찾는 문제 즉 정수론적인 문제에서 단위원 위에 존재하는 유리점의 좌표를 찾는 기하학적 문제로 바뀌었다.

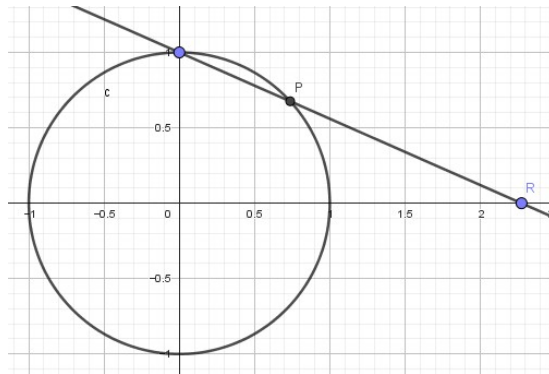


그림 2. 문제의 기하학적 표현

앞서 말한 내용을 기하학적으로 그려보면 그림 2과 같다. 위 그림에서 유리점은 점  $P$ 이고, 문제를 해결하기 위해선 점  $P$ 의 좌표를 구해야 한다. 또한 점  $R(r, 0)$  ( $r > 1$ )과  $(0, 1)$ 을 지나는 직선의 방정식은  $y = -\frac{x}{r} + 1$ 이다. 따라서 점  $P$ 의 좌표는 원의 방정식과 직선의 교점 이므로 다음과 같이 쉽게 구할 수 있다.

$$\begin{cases} x^2 + y^2 = 1 \\ y = -\frac{x}{r} + 1 \end{cases}$$

$$x^2 + \left(1 - \frac{x}{r}\right)^2 = 1$$

$$\frac{(r^2 + 1)x^2}{r^2} - \frac{2x}{r} = 0$$

$$\therefore x = \frac{2r}{r^2+1}, y = \frac{r^2-1}{r^2+1}$$

$$\therefore P\left(\frac{2r}{r^2+1}, \frac{r^2-1}{r^2+1}\right)$$

점  $P$ 의 좌표가  $r$ 에 대한 식으로 표현되었다.

(명제)  $r$ 은 0이 아닌 유리수이다.

(증명)

$$\begin{aligned} &\Leftrightarrow \frac{2r}{r^2+1}, \frac{r^2-1}{r^2+1} ; 0\text{이 아닌 유리수} \\ &\Leftrightarrow \frac{2r}{r^2+1}, \frac{r^2-1}{r^2+1} + 1 ; 0\text{이 아닌 유리수} \\ &\Leftrightarrow \frac{2r^2}{r^2+1} \times \frac{r^2+1}{2r} ; 0\text{이 아닌 유리수} \\ &\Leftrightarrow r ; 0\text{이 아닌 유리수} \quad \blacksquare \end{aligned}$$

따라서  $r$ 은 0이 아닌 유리수 이므로 다음과 같이 표현 된다.

$$\therefore r = \frac{m}{n} (\gcd(m,n) = 1, m > n)$$

명제에 의해 점  $P$ 의 좌표는  $m, n$ 에 대해 다시 써진다.

$$P\left(\frac{2r}{r^2+1}, \frac{r^2-1}{r^2+1}\right) = \left(\frac{2mn}{m^2+n^2}, \frac{m^2-n^2}{m^2+n^2}\right) = \left(\frac{a}{c}, \frac{b}{c}\right)$$

이다. 이때  $a, b, c$ 의 값은 다음과 같다.

$$\begin{aligned} a &= k(2mn), \quad b = k(m^2 - n^2), \quad c = k(m^2 + n^2) \\ &(k ; \text{양의 정수}) \end{aligned}$$

결론적으로 피타고라스 삼원쌍  $(a, b, c)$ 는 다음과 같다.

$$(a, b, c) = (k(2mn), k(m^2 - n^2), k(m^2 + n^2)) \quad \blacksquare$$

모든 피타고라스 삼원쌍  $(a, b, c)$ 는 자연수  $m, n$ 의 값으로 구할 수 있게 되었다.[1]

### 3. 펠 방정식(Pell's equation)

자연수  $d$ 에 대해,  $x^2 - dy^2 = 1$  꼴의 방정식을 펠 방정식(Pell's equation)이라 한다.

[1] 특별한 경우에서 자연수  $m, n$ 이 모두 홀수 이면 세변의 길이가 각각 서로소인 직각삼각형이 아니기 때문에 2로 나누어 주어야 한다.

#### 4. 펠 방정식에 대한 기본적인 이해

(예)  $x^2 - 2y^2 = 1$ 의 그래프를 그려보자.

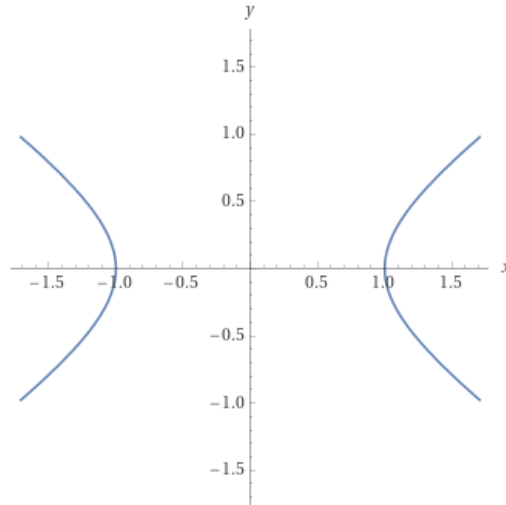


그림 3. 펠 방정식  $x^2 - 2y^2 = 1$  그래프 개형

(관찰한 점)

(i) 펠 방정식  $x^2 - 2y^2 = 1$ 의 그래프는 쌍곡선(hyperbola)의 형태로 존재한다.

(ii) 점근선  $x = \sqrt{2}y$ ,  $x = -\sqrt{2}y$ 이 존재 한다.<sup>[2]</sup>

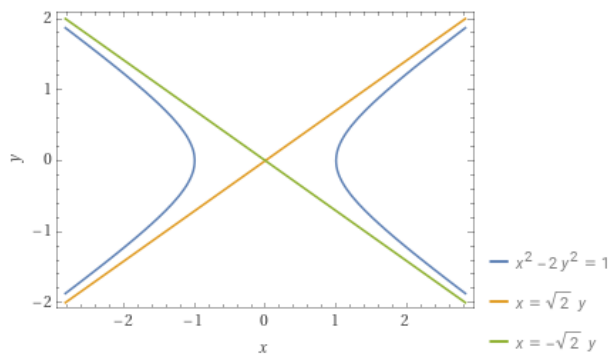


그림 4.  $x^2 - 2y^2 = 1$ 의 그래프와 점근선

(iii)  $(x, y)$ 가 펠 방정식의 해 이면,  $(x, y), (x, -y), (-x, y), (-x, -y)$ 가 모두 해 이다.

[2]일반적으로 펠방정식  $x^2 - dy^2 = 1$  또한 쌍곡선의 형태로 존재하며, 점근선  $x = \sqrt{d}y$ ,  $x = -\sqrt{d}y$ 이 존재한다.  
(증명)

$$\begin{aligned} x^2 - dy^2 &= 1 \\ x^2 &= 1 + dy^2 \\ x &= \pm \sqrt{dy^2} \\ \therefore x &= \pm \sqrt{d}y \end{aligned}$$

(주의)  $x^2 = 1 + dy^2$ 에서  $\lim_{(x,y) \rightarrow \infty} x^2 - dy^2 - 1 = d(-\infty) + \infty$ 이므로 1은 생략할 수 있다. ■

(iv) 펠 방정식  $x^2 - 2y^2 = 1$ 는  $(\pm 1, 0)$ 을 해로 가진다.<sup>[3]</sup> 이를 자명해(trivial solution)라 한다.

(v)  $x^2 - dy^2 = 1$ 에서  $d = c^2$  ( $c > 0$ )이면, 방정식의 해는 유일하다.

(증명)

$$\begin{aligned} x^2 - dy^2 &= 1 \\ x^2 - c^2y^2 &= 1 \\ (x - cy)(x + cy) &= 1 \\ x - cy &= x + cy = \pm 1 \\ \therefore x &= 1, y = 0 \end{aligned} \quad \blacksquare$$

## 5. 라그랑지 정리(Lagrange 1768)

펠 방정식  $x^2 - dy^2 = 1$ 에서  $d$ 가 완전제곱수가 아닌 자연수일 때,  $x^2 - dy^2 = 1$ 은 무수히 많은 해를 가진다. 왜냐하면 주어진 자연수  $d$ 에 대해  $x^2 - dy^2 = 1$ 의 해를 구해내는 것은  $1 + dy^2$ 가 완전제곱수가 되는  $y$ 를 찾는 것과 동치이기 때문이다. ( $\because x^2 - dy^2 = 1 \Leftrightarrow dy^2 + 1 = x^2$ )

(예시)

(i)  $x^2 - 2y^2 = 1$  ( $0 < y \leq 15$ )

표 1. 라그랑지 정리의 시각화

$y$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$2y^2 + 1$	3	9	19	33	51	73	99	129	163	201	243	289	339	393	451
제곱수	X	O	X	X	X	X	X	X	X	X	X	O	X	X	X

(결과)  $y = 2, y = 12$ 일 때  $x = 3, x = 17 \Rightarrow (x, y) = (2, 3), (12, 17)$

(주의) 앞서 말한 펠 방정식에서 발견한 성질<sup>[4]</sup>에 따라,  $(x, y), (-x, y), (x, -y), (-x, -y)$  모두 해 이다. 하지만 이번 예시에서는  $\{(x, y) | x, y \in \mathbb{N}\}$ 만 다루도록 하겠다.

[3] 일반적인 펠 방정식  $x^2 - dy^2 = 1$ 에서도 성립하는 자명한 해 이다. 만약  $d$ 가 완전제곱수라면 펠 방정식  $x^2 - dy^2 = 1$ 의 해는  $(x, y) = (1, 0)$ 으로 유일하다.

[4] (4.펠 방정식에 대한 기본적인 이해)-(관찰한 점)-(iii)을 참고하여라.

(ii)  $x^2 - 3y^2 = 1$  ( $0 < y \leq 15$ )

표 2. 라그랑지 정리의 시각화

$y$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$3y^2 + 1$	4	13	28	49	76	109	148	193	244	301	364	433	508	589	676
제곱수	O	X	X	O	X	X	X	X	X	X	X	X	X	X	O

(결과)  $y = 1, y = 4, y = 15$  일 때  $x = 2, x = 7, x = 26 \Rightarrow (x, y) = (1, 4), (4, 7), (15, 26)$

(결론) 펠 방정식  $x^2 - dy^2 = 1, d \neq c^2 (c \in \mathbb{Z})$  에서,  $y$ 가 끝없이 커질 때,  $dy^2 + 1$ 이 완전제곱수가 되는 경우는 무수히 많이 존재한다. 따라서 해  $(x, y)$ 는 무수히 많이 존재한다.

### III. 다각수와 펠 방정식의 관계

#### 1. 다각수(Polygonal number)

다각수는 삼각수와 정사각수를 임의의 정다각형에까지 일반화하여 얻는 평면 도형수이다.

(예시)

표 1.  $n, k$  값의 변화에 따른  $k$ 번째 삼각수 ( $n = 3$ )






$n \backslash k$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	
$n = 3$	$k$ 번째 $n$ 각형					
	$k$ 번째 $n$ 각수	1	3	6	10	15
		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5

표 2.  $n, k$  값의 변화에 따른  $k$ 번째 사각수 ( $n = 4$ )



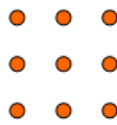

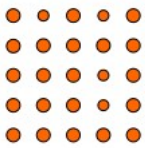


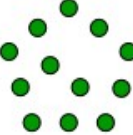
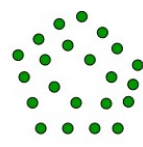

$n \backslash k$		$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$n=4$	$k$ 번째 $n$ 각형					
	$k$ 번째 $n$ 각수	1	4	9	16	25
		1	1+3	1+3+5	1+3+5+7	1+3+5+7+9

표 3.  $n, k$  값의 변화에 따른  $k$ 번째 오각수 ( $n=5$ )

$n \backslash k$		$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$n=5$	$k$ 번째 $n$ 각형					
	$k$ 번째 $n$ 각수	1	5	12	22	35
		1	1+4	1+4+7	1+4+7+10	1+4+7+10+13

(정의 3.1)  $n$ 번째 삼각수를  $T_n$ ,  $n$ 번째 사각수를  $S_n$ ,  $n$ 번째 오각수를  $P_n$  이라 정의한다.

(관찰한 점)  $T_n = 1+2+3+4+\dots+n$  이므로 공차가 1인 등차수열의 합과 같다. 따라서  $T_n$ 은 다음과 같이 정의할 수 있다.

$$T_n = \sum_{k=1}^n k = \frac{n(n+1)}{2}$$

$S_n = 1+3+5+7+\dots+2n-1$  이므로 공차가 2인 등차수열의 합과 같다. 따라서  $S_n$ 은 다음과 같이 정의할 수 있다.

$$S_n = \sum_{k=1}^n (2k-1) = n^2$$

$P_n = 1+4+7+10+\dots+3n-2$  이므로 이는 공차가 3인 등차수열의 합과 같다. 따라서  $P_n$ 은 다음과 같이 정의할 수 있다.

$$P_n = \sum_{k=1}^n (3k-2) = \frac{n(3n-1)}{2}$$

## 2. 삼각-사각수(Square triangular number)

삼각수 이면서 사각수인 자연수를 **삼각-사각수(Square triangular number)** 라고 한다.  $k$ 번째 삼각-사각수,  $N_k$  란  $T_n = S_m$  을 만족하는 수를 말한다.

(정리 3.3) 삼각-사각수들과 펠 방정식  $x^2 - 2y^2 = 1$  의 자연수 해들 사이에 일대일 대응이 존재한다.<sup>[5]</sup>

(증명)

$$T_n = S_m$$

앞선 (정의 3.1)에 의해,

$$\frac{n(n+1)}{2} = m^2$$

이다. 마저 정리 하면,

$$\begin{aligned} n^2 + n &= 2m^2 \\ 4n^2 + 4n &= 8m^2 \\ \therefore (2n+1)^2 - 2(2m)^2 &= 1 \end{aligned} \quad \blacksquare$$

(결론) 삼각-사각수를 찾는 것과  $x^2 - 2y^2 = 1$ 의 자연수 해를 찾는 것이 동치이다.<sup>[6]</sup>

## IV. 펠 방정식의 해를 찾는 여러 가지 방법

### 1. 기존의 해를 이용해서 새로운 해 찾기

앞선 연구에서 증명한 라그랑지 정리로 펠 방정식  $x^2 - dy^2 = 1$  에서 주어진  $d$ 가 완전제곱수가 아닐 때 무수히 많은 해가 존재함을 보였다. 이번에는 이미 알고 있는 기존의 해를 이용하여 새로운 해를 찾고자 한다. 예시를 통해 간단한 관찰로 알아낼 수 있는 성질에서부터 일반적인 펠 방정식에서도 적용되는지 증명하였다.

[5]  $x^2 - 2y^2 = 1$ 의 자연수 해는  $x \equiv 1 \pmod{2} \Rightarrow x^2 \equiv 1 \pmod{4}$ ,  $y^2 \equiv 0 \pmod{2} \Rightarrow y \equiv 0 \pmod{2}$ 를 만족한다.

[6] 같은 방법으로 사각-오각수와  $x^2 - 6y^2 = 1$ 의 자연수 해들 사이에 일대일 대응이 존재함을 보일 수 있다. 증명 방법이 정확히 같기 때문에 본 논문에서는 다루지 않았다.

(관찰 4.1.1)  $x^2 - 2y^2 = 1$ 의 해들 중  $(x, y) = (3, 2), (17, 12)$ 가 있음을 알고 있다. 이때,  $17 + 12\sqrt{2} = (3 + 2\sqrt{2})^2 = 9 + 8 + 12\sqrt{2}$  임을 관찰 할 수 있다. 이를 통해 기존의 해  $(3, 2)$ 를 제곱하면 또 다른 해  $(17, 12)$ 를 얻을 수 있다는 것을 발견하였다.

(관찰 4.1.2) 이번에는  $3 + 2\sqrt{2}$ 를 세제곱 해보자,

$$(3 + 2\sqrt{2})^3 = (17 + 12\sqrt{2})(3 + 2\sqrt{2}) = 51 + 48 + 36\sqrt{2} + 34\sqrt{2} = 99 + 70\sqrt{2}$$

이다. (관찰 4.1)에 의해  $(99, 70)$ 은 펠 방정식의 새로운 해가 될 것이다.

(관찰 4.1.3) 이번에는  $3 + 2\sqrt{2}$  네제곱 해보자.

$$(3 + 2\sqrt{2})^4 = (17 + 12\sqrt{2})^2 = 577 + 408\sqrt{2}$$

이다. (관찰 4.1)에 의해  $(577, 408)$ 은 펠 방정식의 새로운 해가 될 것이다.

같은 방법으로 다섯제곱을 하면, 같은 방법으로  $(3313, 2378)$ 은 새로운 해가 될 것이다.

$$(3 + 2\sqrt{2})^5 = (17 + 12\sqrt{2})^2(3 + 2\sqrt{2}) = 3313 + 2378\sqrt{2}$$

(관찰 4.2.1) 펠 방정식  $x^2 - 3y^2 = 1$ 의 해들 중  $(x, y) = (2, 1), (7, 4), (26, 15)$ 가 있음을 알고 있다. 이때, (관찰 4.1.1), (관찰 4.1.2)와 같은 방법으로 새로운 해를 찾으면, 다음과 같다.

$$(2 + \sqrt{3})^2 = 4 + 3 + \sqrt{3} = 7 + 4\sqrt{3}$$

$$(2 + \sqrt{3})^3 = (7 + 4\sqrt{3})(2 + \sqrt{3}) = 14 + 12 + 8\sqrt{3} + 7\sqrt{3} = 26 + 15\sqrt{3}$$

위 관찰들로 기존의 해로 새로운 해를 찾는 방법을 알아내었다.

(정리 1)  $(x', y'), (x'', y'')$ 이 펠 방정식  $x^2 - dy^2 = 1$ 의 두 해이면,  $(x' + y'\sqrt{d})(x'' + y''\sqrt{d})$ 의 계수들이  $x^2 - dy^2 = 1$ 의 새로운 해이다.

(증명)  $(x' + y'\sqrt{d})(x'' + y''\sqrt{d})$ 의 계수들은 다음과 같다.

$$(x' + y'\sqrt{d})(x'' + y''\sqrt{d}) = x'x'' + dy'y'' + (x'y'' + x''y')\sqrt{d} \dots \star$$

따라서 ★ 식의 계수들을 펠 방정식  $x^2 - dy^2 = 1$ 에 대입하면,

$$(x'x'' + dy'y'')^2 - d(x'y'' + x''y')^2 = 1$$

$$x'^2x''^2 + d^2y'^2y''^2 + 2dx'x''y'y'' - dx'^2y''^2 - dx''^2y'^2 - 2dx'x''y'y'' = 1$$

$$x'^2x''^2 + d^2y'^2y''^2 - dx'^2y''^2 - dx''^2y'^2 = 1$$

$$x''^2(x'^2 - dy'^2) - dy''^2(x'^2 - dy'^2) = 1$$

$$(x''^2 - dy''^2)(x'^2 - dy'^2) = 1 \quad \blacksquare$$

**(따름정리 1.1)** (정리 1)에 의해  $(x_1, y_1)$ 이  $x^2 - dy^2 = 1$ 의 해 이면, 모든 정수  $k (k \in \mathbb{Z})$ 에 대해,  $x_k + y_k \sqrt{d} = (x_1 + y_1 \sqrt{d})^k$  인 정수들  $(x_k, y_k)$ 도  $x^2 - dy^2 = 1$ 의 해 이다. 따라서  $x^2 - dy^2 = 1$ 이 한 개의 자연수 해를 가지면, 무한히 많은 자연수 해가 존재한다.

**(증명)** (정리 1)에 의해 모든 자연수  $k$ 에 대해  $(x_k, y_k)$ 는  $x^2 - dy^2 = 1$ 의 해이다.

$$(x_1 + y_1 \sqrt{d})^{-k} = \frac{1}{(x_1 + y_1 \sqrt{d})^k}$$

(정리 1)에 의해,

$$= \frac{1}{x_k + y_k \sqrt{d}} = \frac{x_k - y_k \sqrt{d}}{(x_k + y_k \sqrt{d})(x_k - y_k \sqrt{d})}$$

이다.

$$= \frac{x_k - y_k \sqrt{d}}{x_k^2 - dy_k^2} = x_k - y_k \sqrt{d} \quad \blacksquare$$

**(주의)**  $(x_k, y_k)$ 가 해 이므로,  $(x_k, -y_k)$ 도 해이다. 또한,  $k = 0$ 인 경우,  $(x_1 + y_1 \sqrt{d})^0 = 1$  이다. \blacksquare

**(예시)**

$$(3 + 2\sqrt{2})^4 = 577 + 408\sqrt{2}$$

$$(3 + 2\sqrt{2})^{-4} = 577 - 408\sqrt{2}$$

**(cf.)**  $(5 + 2\sqrt{2}) = \frac{5 - 2\sqrt{2}}{17}$

2. 모든 해를 찾는 방법.

(보조정리 1) 만약,  $(x, y)$ 가  $x^2 - dy^2 = 1$ 의 해 이고,  $x + y\sqrt{d} > 1$  이면,  $x \geq 2, y \geq 1$ <sup>[7]</sup> 이다. 이때,  $d$ 는 완전제곱수가 아닌 정수이다.

(증명)

$$0 < x - y\sqrt{d} = \frac{1}{x + y\sqrt{d}} < 1$$

이때,

$$0 < x - y\sqrt{d} < 1 < x + y\sqrt{d}$$

이다. 따라서

$$\begin{aligned} 2y\sqrt{d} &> 0 \\ \therefore y &> 0 \text{ 또는 } y \geq 1 \end{aligned} \quad \blacksquare$$

$x - y\sqrt{d} \geq 0$  이고,  $y$ 는 정수 이므로,

$$x - y\sqrt{d} > 0$$

이다.

$$\begin{aligned} x &> y\sqrt{d} \geq \sqrt{d} > 1 \\ \therefore x &\geq 2 \end{aligned} \quad \blacksquare$$

(보조정리 2)  $x^2 - dy^2 = 1, a^2 - db^2 = 1$ 을 만족하는 음이 아닌 정수  $a, b, x, y$ 에 대해, 다음과 같은 부등식에서 두 개의 크기조건을 만족한다.

$$a + b\sqrt{d} < x + y\sqrt{d}$$

(1.↔)  $a < x$  이고,  $b < y$

(2.↔)  $a < x$  또는  $b < y$

(증명) (1.↔)은 자명하다. (1.→)임을 보이자.

•  $a \geq 0$  이기 때문에  $a = 0$ 을 대입하면 다음과 같다.

$$0^2 - db^2 = 1$$

하지만 위 식을 만족하는 정수  $b$ 는 존재하지 않으므로

---

[7] (참고) 펠 방정식  $x^2 - 3y^2 = 1$ 은 해 중 (2,1)이 존재 하므로 최적의 부등식이다.

$$a \geq 1$$

이다. 같은 이유로  $x \geq 1$  이다.

•  $a + b\sqrt{d} < x + y\sqrt{d}$  의 역수를 취하면,

$$x - y\sqrt{d} < a - b\sqrt{d}$$

이다. 두 개의 부등식을 더하면,

$$\begin{aligned} (a+x) + (b-y)\sqrt{d} &< (a+x) + (y-b)\sqrt{d} \\ 0 &< 2(y-b)\sqrt{d} \end{aligned}$$

이다.

$$\therefore y > b, y^2 > b^2 (\because y \geq 0, b \geq 0)$$

또한,

$$\begin{aligned} a^2 = 1 + db^2 &< 1 + dy^2 = x^2 \\ \therefore a, x &\geq 0, a < x \end{aligned}$$

■

(2.↔)도 같은 방법으로 증명할 수 있기에 증명과정은 생략한다.

**(정리 2)**  $x^2 - dy^2 = 1$ 이 해를 가지고, 자연수 해들 중  $(x_1, y_1)$ 이  $y$  값이 가장 작은 해라 하면 다음이 성립한다.

(i)  $x^2 - dy^2 = 1$ 의 임의의 정수해는  $\pm(x_1 + y_1\sqrt{d})^k$  ( $k \in \mathbb{Z}$ )의 계수이다.

(ii)  $x^2 - dy^2 = 1$ 의 임의의 자연수 해는  $(x_1 + y_1\sqrt{d})^k$  ( $k \in \mathbb{N}$ )의 계수이다.

**(증명)**  $(x_1 + y_1\sqrt{d})^k$ 의 계수들이  $x^2 - dy^2 = 1$ 의 해가 되는 것은 이미 (따름정리 1.1)으로 알고 있다. 자연수 쌍  $(x, y)$ 가  $x^2 - dy^2 = 1$ 을 만족한다고 가정하자.  $x + y\sqrt{d} > 1$ 이고,  $\{(x_1 + y_1\sqrt{d})^n\}_{n=0}^{\infty}$ <sup>[8]</sup>은  $\infty$ 으로 발산하는 증가수열 이므로, 결론은 다음과 같다.

$$(x_1 + y_1\sqrt{d})^k \leq x + y\sqrt{d} < (x_1 + y_1\sqrt{d})^{k+1} \text{ 인 음이 아닌 정수 } k(k \geq 0) \text{ 가 존재한다.}$$

다음으로,  $1 \leq (x + y\sqrt{d})(x_1 - y_1\sqrt{d})^k < x_1 + y_1\sqrt{d} \dots$  (\*) 이 성립하므로,

$$(x + y\sqrt{d})(x_1 - y_1\sqrt{d})^k = a + b\sqrt{d} \quad (a, b \in \mathbb{Z}) \text{ 라 하면, } a^2 - db^2 = 1 \text{ 이다.}$$

[8]  $\{(x_1 + y_1\sqrt{d})^n\}_{n=0}^{\infty}$  는  $n$ 을 0에서 무한대 까지 모든 음이 아닌 정수에 대해서 바꾸면서 해당하는 숫자들을 다 모은 집합을 뜻한다.

(\*)을 정리 하면,

$$1 \leq a + b\sqrt{d} < x_1 + y_1\sqrt{d}$$

$$a > b\sqrt{d}$$

이면, (보조정리 1), (보조정리 2)에 의해  $a, b > 0$  이고  $a < x_1, b < y_1$  이다. 이것은  $y_1$  선택에 모순이다. 따라서  $a + b\sqrt{d} = 1$  즉,  $x + y\sqrt{d} = (x_1 + y_1\sqrt{d})^k$  이다. 일반적으로  $x, y$  가 정수해인 경우도 비슷한 논리로 증명되므로 생략하겠다. ■

(예시) (정리 2)의 예시들을 살펴보자.

(주의)  $(x_1, y_1)$ 이  $x^2 - dy^2 = 1$ 의 자연수 해 중  $y_1$ 이 최소라 하면, 이 자연수 해는  $x_1$ 도 최소이다.  $(x_1 + y_1\sqrt{d})^k$ 의 두 계수들은  $k$ 에 대한 증가함수이다.

(i)  $x^2 - 2y^2 = 1$ 의 자연수 해 중  $y$  값이 가장 작은 해는 (3,2)이다. 따라서 모든 자연수 해는  $(3 + 2\sqrt{2})^n$  ( $n > 0$ )의 계수들이다.

(ii)  $x^2 - 3y^2 = 1$ 의 자연수 해 중  $y$  값이 가장 작은 해는 (2,1)이다. 따라서 모든 자연수 해는  $(2 + \sqrt{3})^n$  ( $n > 0$ )의 계수들이다.

(iii)  $x^2 - 5y^2 = 1$ 의 자연수 해 중  $y$  값이 가장 작은 해는 (9,4)이다. 따라서 모든 자연수 해는  $(9 + 4\sqrt{5})^n$  ( $n > 0$ )의 계수들이다.

## V. 고차원의 펠 방정식

펠 방정식  $x^2 - dy^2 = 1$ 은 이차원 평면상의 방정식이다. 하지만 나는 고차원의 펠 방정식은 어떻게 표현할 수 있고, 또 어떠한 성질을 가지는지 궁금했다. 그래서 나는 3차원의 펠 방정식 중 한 가지 경우에서 무한한 해의 존재성을 증명하였다.

### 1. 그래프의 개형

3차원의 펠 방정식은 쌍곡선의 형태가 아닌 쌍곡면의 형태로 존재한다.

(예시)

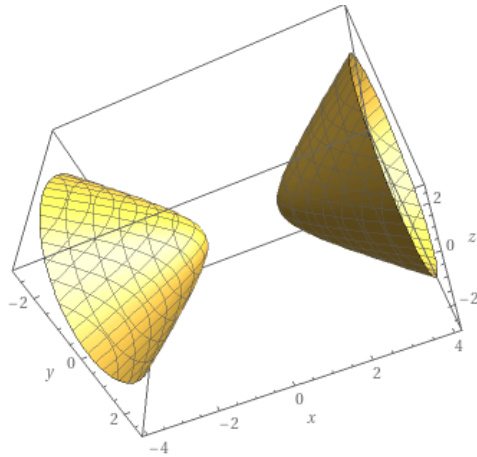


그림 5.  $x^2 - 2y^2 - 2z^2 = 1$  의 그래프 개형

## 2. 3차원의 펠 방정식의 해

목표는 3차원 공간상의 펠 방정식은 무수히 많은 해가 존재 하는지 증명하는 것이다. 즉, 그래프 상에서 보았을 때, 쌍곡면에서 무수히 많은 격자점을 가진다는 것을 증명해볼 것이다. 먼저 해를 쉽게 구해보기 위해서 *Python*을 통해 코딩하였다.

```
import math
def issquare(n):
    if math.sqrt(n) == int(math.sqrt(n)):
        return 1
    else:
        return -1
N=100
d=2
for i in range(1,N):
    for j in range(i,N):
        if issquare(1+d*i*i+d*j*j)==1:
            print(int(math.sqrt(1+d*i*i+d*j*j)), i, j)
```

그림 6.  $x^2 - 2y^2 - 2z^2 = 1$ 의 해를 찾기 위해 작성한 코드포  
그림 6을 통해 수많은 해를 구해내었다.

9	2	6
51	2	36
19	6	12
35	6	24
105	6	74
33	12	20
51	12	34
81	12	56
51	20	30
81	24	52
129	24	88
73	30	42
105	34	66
145	36	96
99	42	56
147	52	90
129	56	72
163	72	90

그림 7.  $x^2 - 2y^2 - 2z^2 = 1$  해의 일부

펠 방정식  $x^2 - 2y^2 - 2z^2 = 1$ 의 해  $(x, y, z)$  몇몇의 특정한 순서쌍에는 특정한 규칙이 있다는 것을 발견하였다. 규칙들은 크게 네 가지 정도 소개하겠다.

몇몇의 순서쌍  $(x, y, z)$ 에 대해 다음 규칙이 성립한다.

(규칙 1)  $x = y + z + 1$ 을 만족한다.

(규칙 2) (규칙 1)을 만족하는 순서쌍  $(x, y, z)$ 의  $y$  값과, 바로 다음으로 (규칙 1)을 만족하는 순서쌍  $(x, y, z)$ 의  $z$  값이 같다. 연쇄적이다.

(예시)  $(x, y, z) = (9, 2, 6) \rightarrow (x, y, z) = (19, 6, 12) \rightarrow (x, y, z) = (33, 12, 20) \dots$

(규칙 3)  $(x, y, z)$  중  $y$  값만 나열한 수열을  $\{A_n\}$ 이라 정의할 때, 수열  $\{A_n\}$ 은 계차수열이다.

(예시)  $\{A_n\} = \{2, 6, 12, 20, 30, 42, 56, \dots\}$

(규칙 4) 순서쌍  $(x, y, z)$  중  $y$ 만 나열하면 다음과 같다.

$$2, 6, 12, 20, 30, 42, 56 \dots$$

$$= 1 \times 2, 2 \times 3, 3 \times 4, 4 \times 5, \dots$$

연속한 두 수의 곱으로 이루어져 있음을 알 수 있다.

### 3. 3차원의 펠 방정식 $x^2 - 2y^2 - 2z^2 = 1$ 무한한 해의 존재성 증명

앞선 연구에서 3차원 펠 방정식  $x^2 - 2y^2 - 2z^2 = 1$ 의 해들을 코딩으로 찾아보고, 규칙들을 발견해 정리해보았다. 이젠 발견한 규칙들을 이용해 일반화 시켜 무한한 해의 존재성을 증명 하겠다.

(증명) 앞서 말한 규칙을 만족하는  $n$ 번째 순서쌍을  $(x, y, z)_n$  이라 하자.

(규칙 1), (규칙 4)에 의해  $(x, y, z)_n$ 은 다음과 같다.

$$(x, y, z)_n = (n(n-1) + n(n+1) + 1, n(n-1), n(n+1))$$

$x^2 - 2y^2 - 2z^2 = 1$ 에 대입하면,

$$\begin{aligned} & (2n^2 + 1)^2 - 2(n^2 + n)^2 - 2(n^2 - n)^2 \\ &= 4n^4 + 4n^2 + 1 - 2n^4 - 4n^3 - 2n^2 - 2n^4 + 4n^3 - 2n^2 \\ &= 4n^4 + 4n^2 + 1 - 2(2n^4 + 2n^2) = 1 \end{aligned} \quad \blacksquare$$

(결론) 3차원 펠 방정식  $x^2 - 2y^2 - 2z^2 = 1$ 의 해는 무수히 많다.

## VI. 결론

이번 연구를 통해 펠 방정식과 여러 성질을 소개하고 증명해 보았다. 먼저 펠 방정식과 다각수의 관계를 보이기 위해 다각수와 삼각-사각수를 정의하고, 삼각-사각수와 펠 방정식 사이에 일대일 대응이 존재함을 증명하였다.

펠 방정식의 해를 구하는 방법은 여러 가지가 있다. 그중 가장 유명한 방법은 라그랑지 정리이다. 펠 방정식의 성질과 더불어 라그랑지 정리의 기본적인 이해를 바탕으로 기존에 알고 있는 해를 이용해서 새로운 해를 찾는 방법과 일반적인 펠 방정식에서 모든 해를 찾는 방법을 소개하고 증명하였다.

본 연구에서는 특정 고차원의 펠 방정식의 해에서 규칙들을 발견하여 무한한 해의 존재성을 증명하였다. 덧붙여서,  $\sqrt{d}$ 의 새로운 유리수 근사를 고차원의 펠 방정식을 통해 얻을 수 있는지 추가로 연구할 계획이다.

## VII. 참고문헌

1. Edward J. Barbeau. (2003). Pell's Equation.
2. Martin Aigner. (2013). Markov's Theorem and 100 Years of the Uniqueness Conjecture
3. Roger B. Nelsen. (2018). Nuggets of Number Theory: A Visual Approach
4. [https://en.wikipedia.org/wiki/Pell%27s\\_equation](https://en.wikipedia.org/wiki/Pell%27s_equation)
5. <https://www.wolframalpha.com/>

수학B

## 06\_쥐와 고양이 게임의 기하학



# 쥐와 고양이 게임과 기하학

정우주 (여수 삼일중학교 3학년)

지도교수 : 권명기 (전북대학교 수학교육과 前순천대학교)

## 초록

수학에서는 문제를 해결하기 위해 여러 제약 조건을 사용하여 다각도로 문제에 접근한다. 쥐와 고양이 게임에서는 평면 상의 원으로 주어지는 초기 제약 조건 하에서 쥐가 고양이를 피할 수 있는 방법을 기하학적으로 탐구한다. 본 연구에서는 이를 확장시켜, 평면 상의 원에서의 이동 방법을 증명하고 다양한 평면도형, 입체도형에서 점의 이동 방법을 증명한다. 연구 가설과 이에 대한 증명을 지오지브라와 같은 작도 프로그램을 활용하여 시각적으로 검증하고, 다양한 도형에서의 공통된 특징을 찾아내어 이동 방법의 일반적인 규칙성을 추론한다. 이 논문에서는 각속도와 점의 속력을 기반으로, 도형 내의 점이 도형 외부로 나가는 과정에서 어떠한 기하학이 담겨있는지에 대해 고찰한다.

## I. 서론 (또는 연구의 필요성 및 목적)

쥐 고양이 게임[4]은 도형 내부의 쥐가 도형 외부의 고양에게 잡히지 않고, 외부로 이동하는 방법을 다룬다. 이 과정에서 핵심 요소는 점의 이동거리와 이동시간으로 이러한 요소가 문제 해결을 위한 조건이 된다. 쥐와 고양이 게임을 원에서 해결하고, 이 문제와 유사하게 다른 문제에 제약 조건을 변화시켜 이동 방법에 대해 생각해 보고, 수학적 요소를 이용한 계산과, 추론을 통한 연구 방법을 계획하고, 문제를 해결한다.

## II. 이론적 배경

### 1. 쥐와 고양이 게임

쥐와 고양이 게임에서는 문제 해결을 위해서 문제에 몇 가지 재미있는 설정을 추가한다. <sup>1</sup>도형 내부의 점을 *point 'Mouse'*(쥐)로 도형 겉면 상의 점을 *point 'Cat'*(고양이)로 설정한 것이다. 이것이 바로 쥐와 고양이 게임이 문제의 이름인 이유이다. <sup>2</sup>다음으로 쥐와 고양이의 속도의 비율이다. 다양한 비율의 설정이 가능하지만, 해당 영상에서는 쥐 : 고양이의 속도 비를 1:4로 제시하였다. <sup>3</sup>마지막으로 매우 전통적으로 고양이는 쥐를 잡고 싶어하고, 이에 따라 계속해서 쥐를 쫓아 움직인다. 쥐는

고양이를 피해 연못(연구의 핵심인 다양한 도형)으로 들어갔고, 연못 밖에서는 쥐가 고양이보다 빠르므로 쥐는 고양이를 피해 연못 밖으로만 나가면 된다.

## 2. 쥐와 고양이 게임의 확장

원이라는 평면 도형에서의 탈출 방법(도형 내의 한 점(쥐)이 겹면까지 이동하는 경우를 ‘탈출’이라고 명명하겠다.)만을 제시한 것은 쥐와 고양이 게임에서 가장 쉽게 발전의 여지를 찾을 수 있는 부분이다. 이외에도, 쥐와 고양이의 속도의 비를 조절하는 방법 또는 쥐와 고양이의 수를 변하게 하는 방법도 있을 것이다. 혹은 아예 새로운 조건을 추가하여 문제를 변화시킬 수도 있겠지만, 본 연구의 목표는 문제 해결 방법의 기하학적 원리와 규칙성을 찾아보는 것이므로, 다양한 평면도형, 입체도형에서의 탈출 방법을 알아보기로 하였다.

## 3. 각속도

각속도는 강제 운동(도형의 모양을 변형하지 않는 평면의 변환. 회전 이동과 평행 이동)을 대표하는 물리량 중 하나로, 기준이 되는 축 주위로 얼마나 빠르게 도는지를 나타낸다. 각속도와 회전 수는 비례 관계이고 따라서 각속도를 회전력이라고 하기도 한다.[6]

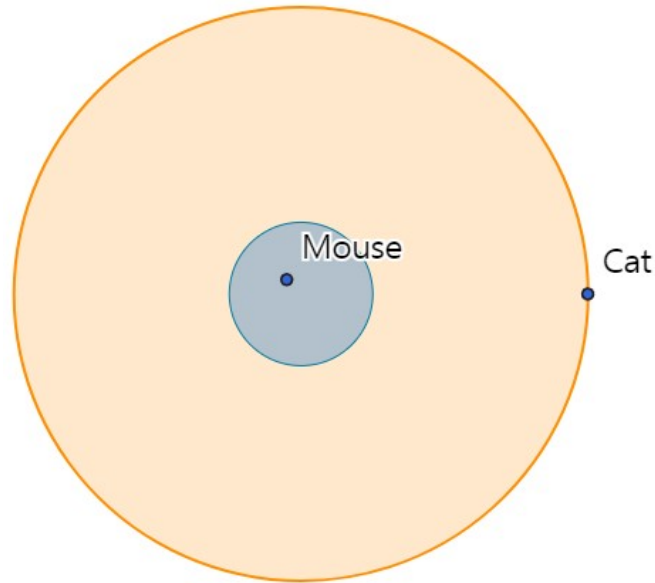
## 4. 측지선과 대원

측지선이란 직선의 개념을 굽은 공간으로 일반화한 것이고, 대원은 구의 단면 중 가장 큰 것을 뜻한다. 해당 구의 반지름과 같은 반지름인 원이며, 구면기하학에서 직선에 해당한다. 구면 상의 측지선은 대원에 해당한다고 알려져있다.[5]

## 5. 문제 설정 및 가설

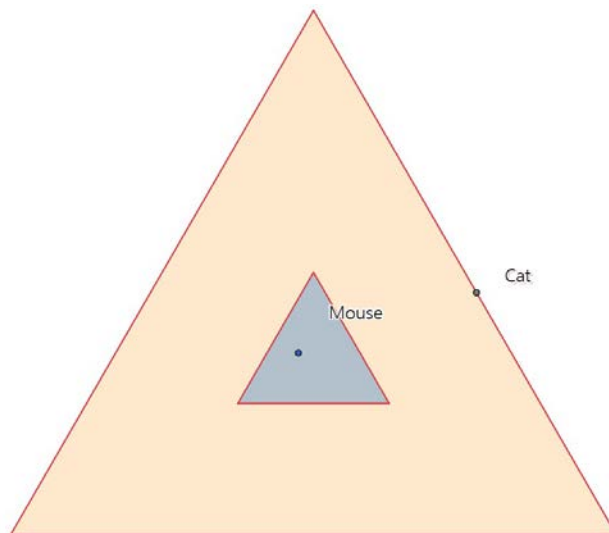
몇 가지 평면도형 및 입체도형에서의 문제 설정 및 가설은 다음과 같다.

① 쥐와 고양이 게임 (원)



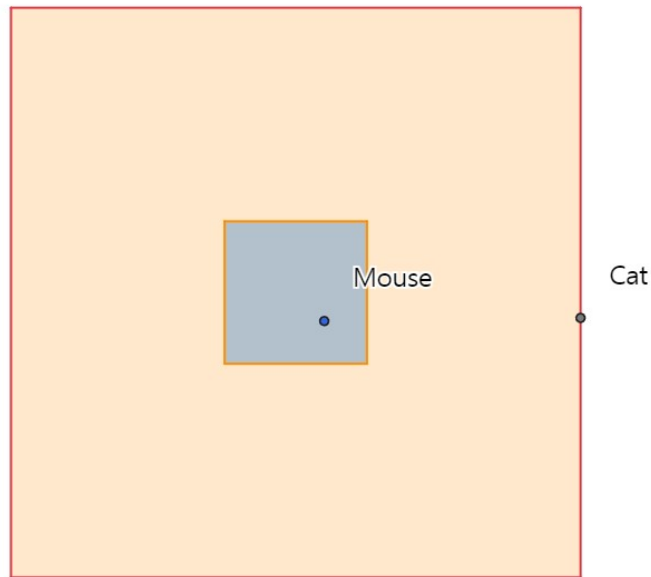
반지름이 노란색 영역의  $\frac{1}{4}$ 인 파란색 영역 내부를 쥐가 계속해서 돌게됨으로써, 각 속도 차이에 의해 쥐와 고양이 사이의 각이 벌어지게 되고, (파란색 영역) - (초록색 영역)에서 쥐가 계속해서 회전 운동을 반복하다가, 쥐-고양이 사잇각이  $180^\circ$ 가 될 때 쥐가 노란색 원의 원주를 향해 이동하면 탈출할 수 있다.

② 쥐와 고양이 게임 (정삼각형)



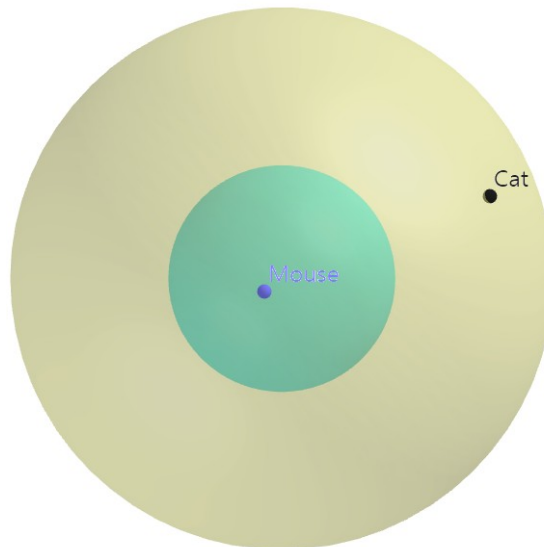
실제 거리의 최솟값과 차이가 나는 부분에서도, 이후 쥐의 이동으로 인한 쥐-고양이 사이 거리 최소화를 위해 고양이는 쥐를 쫓게 된다. 이때 쥐가 위치한 파란색 삼각형의 한 변은 고양이가 변의 위에 있는 노란색 삼각형의 한 변의  $\frac{1}{4}$  이므로 각 속도 차이에 의해 쥐-고양이 사잇각이 벌어지게 된다.

③ 쥐와 고양이 게임 (정사각형)



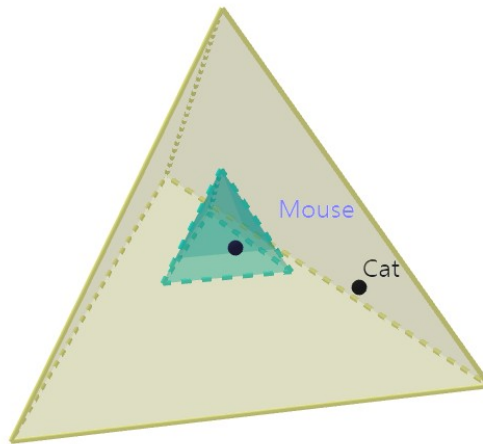
정삼각형과 마찬가지로 실제 거리의 최솟값과 차이가 나는 부분에서도, 이후 쥐의 이동으로 인한 쥐-고양이 사이 거리 최소화를 위해 고양이는 쥐를 쫓게 되고, 이때 쥐가 위치한 파란색 사각형의 한 변의 길이는 고양이가 변의 위에 있는 노란색 사각형의 한 변의  $\frac{1}{4}$  이므로 각속도 차이에 의해 쥐-고양이 사잇각이 벌어지게 된다.

④ 쥐와 고양이 게임 (구)



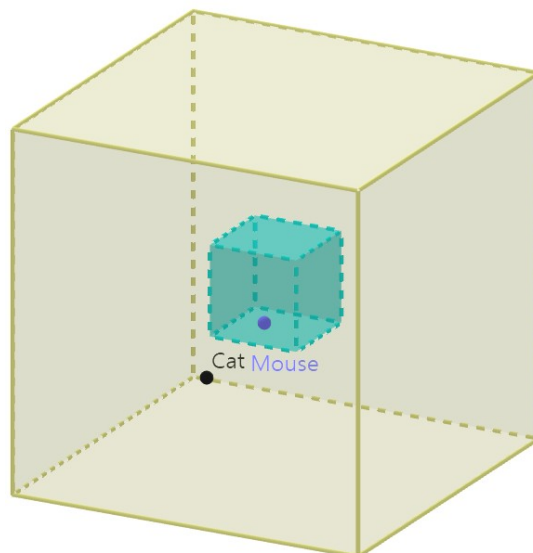
구에서 본 연구의 풀이 전략은 입체도형에 대한 문제를 평면도형의 경우로 변형시키는 것이다. 이를 위해 고양이, 쥐, 구의 중심을 지나는 평면을 생각하고, 이 평면과 구가 만나서 생기는 원을 고려한다. 이 원을 기존의 쥐와 고양이 게임에서 해결한 원에서와 같은 문제로 만들어서 해결한다.

⑤ 쥐와 고양이 게임 (정사면체)



입체도형인 정사면체 역시 어떻게 해야 입체도형을 평면도형의 기준을 이용해 판단할 수 있는지가 중요하다. 이를 위해 정사면체에서는 내부에 닮음비가 4:1인 정사면체를 그리는 기존의 방식과 전개도를 이용해서 고양이의 속도×시간과 쥐의 속도×시간의 차이를 이용하는 방식을 함께 사용했다.

⑥ 쥐와 고양이 게임 (정육면체)



정육면체도 마찬가지로 입체도형인 정육면체를 평면도형인 정사각형과 같이 생각하고 문제를 풀어야 한다. 정사면체와 마찬가지로 정육면체에서도 내부에 닮음비가 4:1인 정육면체를 그리는 기존의 방식과 전개도를 이용해서 고양이의 속도×시간과 쥐의 속도×시간의 차이를 이용하는 방식을 함께 사용했다.

III. 연구방법

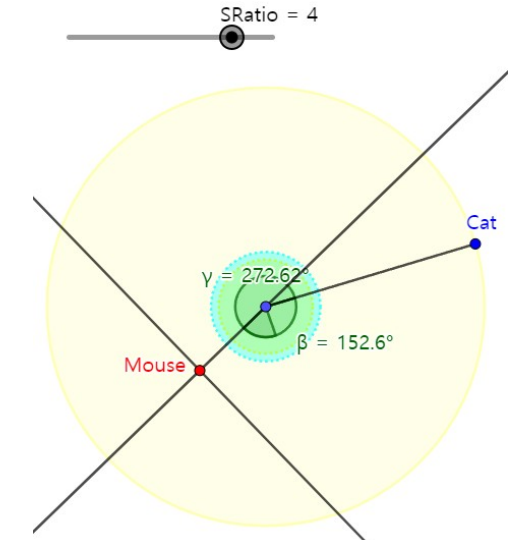
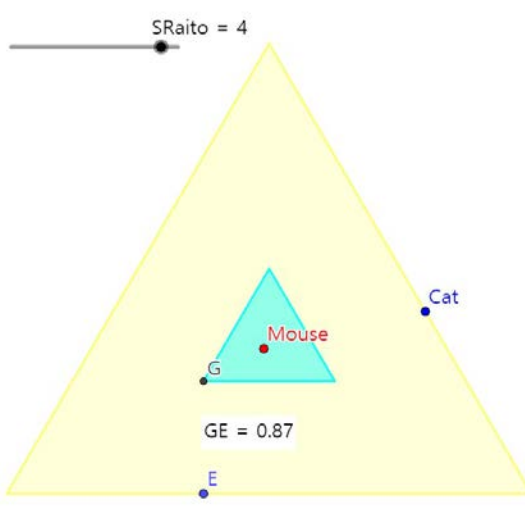
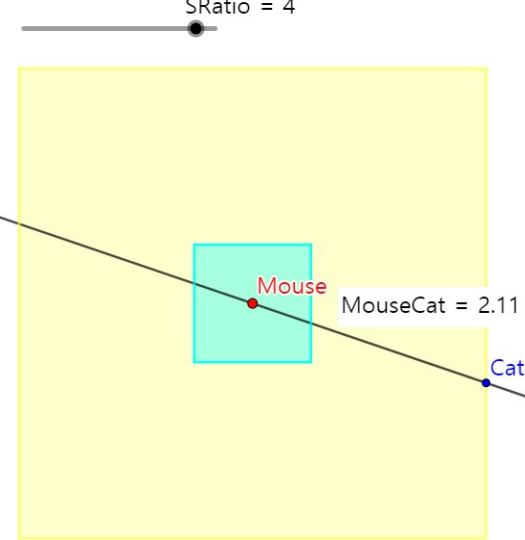
1. 탈출 시작 구간 추론

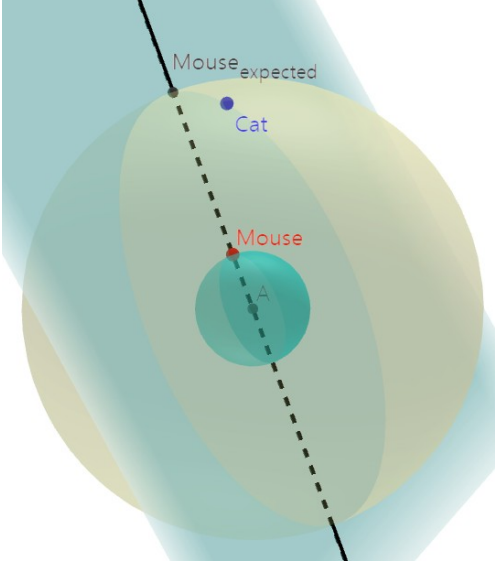
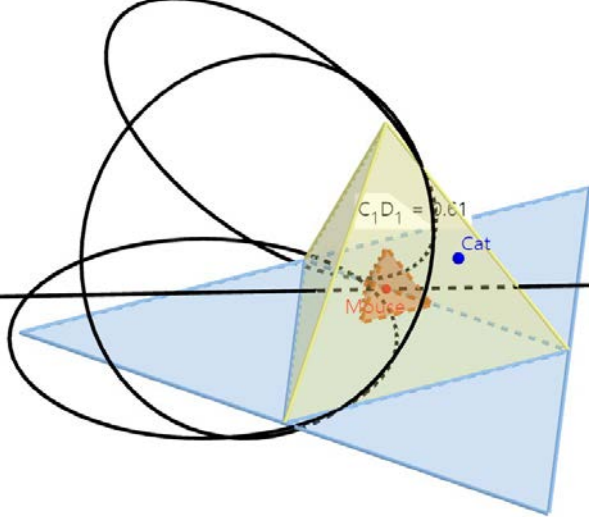
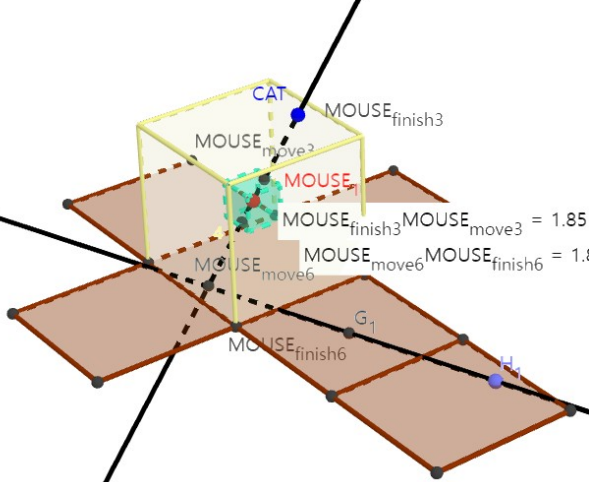
연구를 수행하기 위해 가장 먼저 기존의 쥐와 고양이 게임을 지오지브라를 통해서 현해 보고, 이때 쥐가 탈출하는 방법의 핵심이 무엇인지 생각해본다. 또한 정삼각형과 정사각형에서 쥐와 고양이 게임의 원과 같이 쥐가 고양이를 피해 탈출할 수 있는 안전지대를 추론하고 이를 정 $n$ 각형 상에 적용해 본다. 이후 속력, 시간, 각속도를 고려한 수학적 계산을 통하여, 쥐가 실제로 고양이를 피해 탈출할 수 있는지를 검토해본 뒤 이를 3차원 도형에 적용해본다.

## 2. 3차원 공간의 적용

또한 위의 개념을 3차원 도형에 적용할 때는 최단거리 경로를 활용한 구를 먼저 구현해 본다. 이를 이용해서, 정육면체와 정사면체를 작도한 뒤 (비록 곡선은 아니지만) 최단거리 경로의 개념을 정 $n$ 면체에 적용하여 2차원 도형과 같이 쥐가 고양이를 피해 탈출할 수 있는 안전지대를 추론하고, 작도한 도형에 표현해 본다.

추가적으로, 3차원 도형인 정 $n$ 면체의 경우에는 그 전개도를 생각해 볼 수 있다. 두 점이 전개도 내부를 벗어나지 않는 직선으로 이어진다면, 그 직선 경로를 다시 정다면체 위에서 그렸을 때(측지선)가 최단 경로가 될 것이다. 따라서 평면과 정 $n$ 면체와의 교선이 전개도를 통해 얻은 최단 거리 경로와 같게 되는 경우가 있는지를 연구한다. 이 연구를 위해 지오지브라를 활용해서 정 $n$ 면체를 작도하여 가설을 검증해보고, 지오지브라에서 가설에 대한 반례가 나온다면, 왜 가설이 잘못되었는지 수학적으로 이해해보고, 충분히 많은 경우에 대한 반례가 등장하지 않는다면, 가설이 참일 것이라고 예상하고 수학적 증명을 시도해본다.

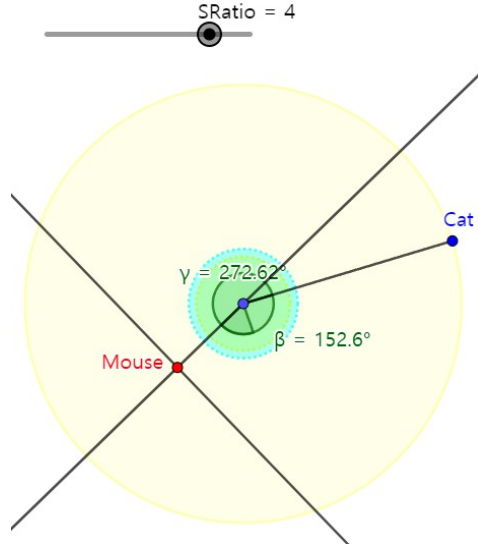
과제번호	사진	설명
1		<p>쥐와 고양이 게임 (원)</p>
2		<p>쥐와 고양이 게임 (정삼각형)</p>
3		<p>쥐와 고양이 게임 (정사각형)</p>

4		<p>쥐와 고양이 게임 (구)</p>
5		<p>쥐와 고양이 게임 (정사면체)</p>
6		<p>쥐와 고양이 게임 (정육면체)</p>

## V. 연구결과

M 은 점 쥐를 나타내고, C 는 점 고양이를 나타낸다.

### ① 과제 1



지오지브라로 구현한 쥐와 고양이 게임 (원)

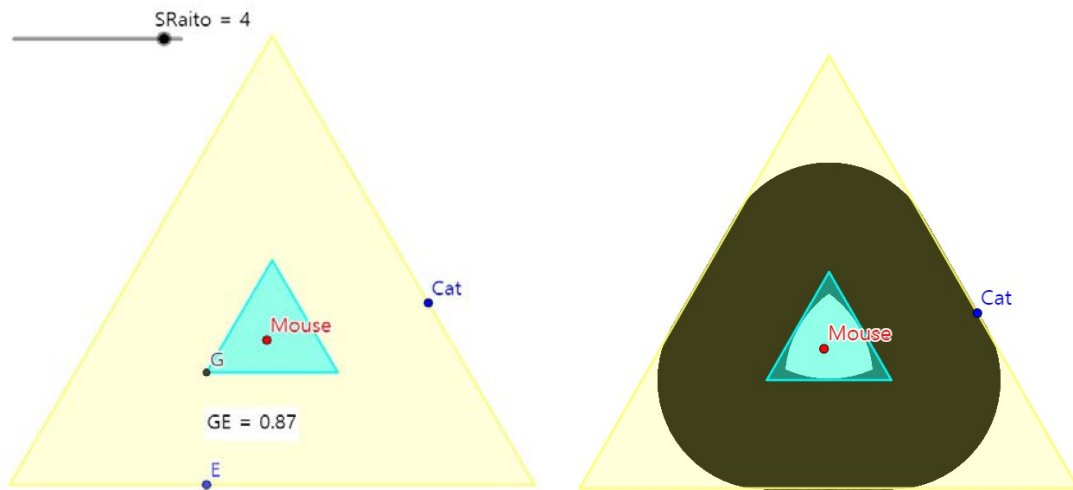
쥐의 속도는 1, 고양이의 속도는 4이다. 이때 내부의 파란색 원의 반지름은 외부의 노란색 원의 반지름의  $\frac{1}{4}$  이므로, 쥐가 파란색 원 속에서 계속해서 돈다면 각속도 차이에 의해서  $\angle MOC$  의 크기는 점점 커지게 된다. 쥐와 고양이의 사잇각의 크기가  $180^\circ$  가 되었을 때 쥐가 파란색 원에서 노란색 원 밖으로 이동하면, 쥐의 이동거리는 3, 고양이의 이동거리는  $3\pi$ 이므로  $3 \times 4 < 3\pi$ 이고 쥐는 탈출이 가능하다.

위의 증명으로 해당 도형에서 쥐의 탈출이 가능함을 밝혀내었다. 해당 도형을 작도하는 방법은 다음과 같다.

- ① 중심을 점  $O$ 로 하고 반지름의 길이가 4인 원  $c$  작도한다.
- ② 중심을 점  $O$ 로 하고 반지름의 길이가 1인 원  $c'$ 를 작도한다.
- ③ 원  $c'$ 의 원주 상의 한 점  $Mouse$ 를 잡고, 원  $c$ 의 원주 상의 한 점  $Cat$ 를 잡는다.
- ④  $\angle MOC$ 의 크기를 나타낸다.(그림 상에서는  $\beta$ )
- ⑤ 점  $Mouse$ 와 점  $Cat$ 을 애니메이션을 이용해서 회전시킨다.(이때 원  $c'$ 의 반지름의 길이가 1 이라면 각의 크기 변화가 0이므로 1 보다 약간 작게 해준다.)

결론:  $\angle MOC$ 의 크기가 점점 증가하는 것을 확인할 수 있고, 이로 인해 각  $\angle MOC$ 의 크기가  $180^\circ$  가 되었을 때 쥐가 탈출할 수 있다.

② 과제 2



지오지브라로 구현한 쥐와 고양이 게임 (정삼각형)

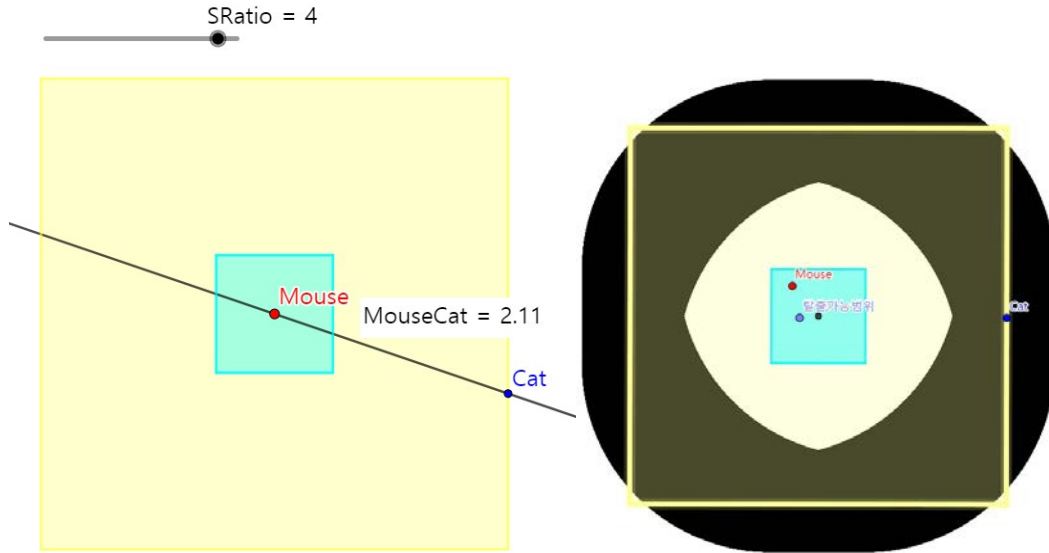
쥐의 속도는 1, 고양이의 속도는 4이다. 이때 내부의 파란색 삼각형의 한 변의 길이는 외부의 노란색 삼각형의 한 변의 길이의  $\frac{1}{4}$ 이므로, 쥐가 파란색 삼각형 속에서 계속해서 돈다면 각속도 차이에 의해서  $\angle MGC$ 의 크기는 점점 커지게 된다. 쥐와 고양이의 사잇각의 크기가  $180^\circ$ 가 되었을 때 쥐가 파란색 삼각형의 한 변에서 노란색 삼각형의 한 변으로 이동하면, 쥐의 이동거리는 0.87, 고양이의 이동거리는 6이므로  $0.87 \times 4 < 6$ 이므로 쥐는 탈출이 가능하다. 내부의 파란색 삼각형에서 노란색 삼각형의 변 위의 지점까지의 거리는 파란색 삼각형의 꼭짓점에서 차이가 발생하나 (오른쪽 그림의 색칠되지 않은 부분), 파란색 삼각형을 계속해서 회전하는 쥐를 잡기 위해, 고양이도 계속해서 회전한다는 조건을 추가했다.

위의 증명으로 해당 도형에서 쥐의 탈출이 가능함을 밝혀내었다. 해당 도형을 작도하는 방법은 다음과 같다.

- ① 한 변의 길이를 4로 하는 정삼각형  $t$ 를 작도한다.
- ② 정삼각형  $t$ 의 무게중심  $G$ 를 잡는다.
- ③ 정삼각형  $t$ 와 무게중심을 공유하고, 한 변의 길이를 1로 하는 정삼각형  $t'$ 를 작도한다.
- ④  $\angle MGC$ 의 크기를 나타낸다.
- ⑤ 점  $Mouse$ 와 점  $Cat$ 을 애니메이션을 이용해서 회전시킨다.(이때 정삼각형  $t'$ 의 한 변의 길이가 1이라면 각의 크기 변화가 0이므로 1보다 약간 작게 해준다.)

결론:  $\angle MGC$ 의 크기가 점점 증가하는 것을 확인할 수 있고, 이로 인해  $\angle MGC$ 의 크기가  $180^\circ$ 가 되었을 때 쥐가 탈출할 수 있다.

③ 과제 3



지오지브라로 구현한 쥐와 고양이 게임 (정사각형)

쥐의 속도는 1, 고양이의 속도는 4이다. 이때 내부의 파란색 사각형의 한 변의 길이는 외부의 노란색 사각형의 한 변의 길이의  $\frac{1}{4}$  이므로, 쥐가 파란색 사각형 속에서 계속해서 돈다면 각속도 차이에 의해서  $\angle MOC$ 의 크기는 점점 커지게 된다. 쥐와 고양이의 사잇각의 크기가  $180^\circ$ 가 되었을 때 쥐가 파란색 사각형의 한 변에서 노란색 사각형의 한 변으로 이동하면, 쥐의 이동거리는 15, 고양이의 이동거리는 8이므로  $1.5 \times 4 < 8$ 이므로 쥐는 탈출이 가능하다. 하지만, 각 꼭짓점에서는 쥐의 이동거리가 약 2.12, 고양이의 이동거리는 8이므로 사각형의 꼭짓점에서는 쥐의 탈출이 불가능하다. 따라서 쥐의 탈출이 불가능한 범위는 초록색 사각형으로부터 2 이상 떨어져 있는 부분이다. 우측의 그림은 탈출 가능 범위를 반지름이 2인 원을 이용해서 나타낸 것이다. 사각형에서도 삼각형에서도 마찬가지로, 내부의 파란색 사각형에서 노란색 사각형의 변 위의 지점까지의 거리는 파란색 사각형의 꼭짓점에서 차이가 발생하나(탈출 불가능 범위), 파란색 사각형을 계속해서 회전하는 쥐를 잡기 위해, 고양이도 계속해서 회전한다는 조건을 추가했다.

위의 증명으로 해당 도형에서 쥐의 탈출이 가능함을 밝혀내었다. 해당 도형을 작도하는 방법은 다음과 같다.

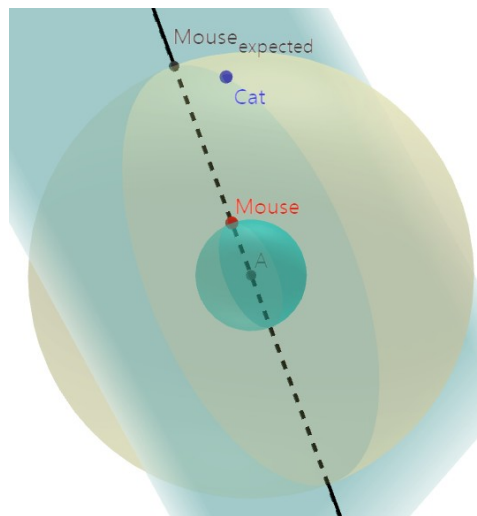
- ① 한 변의 길이를 4로 하는 정사각형  $s$ 를 작도한다.
- ② 정사각형  $s$ 의 중심  $O$ 를 잡는다.
- ③ 정사각형  $s$ 와 중심을 공유하고, 한 변의 길이를 1로 하는 정삼각형  $s'$ 를 작도한다.

- ④  $\angle MOC$ 의 크기를 나타낸다.
- ⑤ 점 *Mouse*와 점 *Cat*을 애니메이션을 이용해서 회전시킨다.(이때 정사각형  $s'$ 의 한 변의 길이가 1이라면 각의 크기 변화가 0이므로 1보다 약간 작게 해준다.)

결론:  $\angle MOC$ 의 크기가 점점 증가하는 것을 확인할 수 있고, 이로 인해  $\angle MOC$ 의 크기가  $180^\circ$ 가 되었을 때 쥐가 탈출할 수 있다.

과제 2과 과제 3은 또한 Leonidas J. Guibas • John Hershberger의 접근 방법[3]과 같이 평면 도형의 최단 이동 거리의 알고리즘화를 통해서 해결할 수도 있었다. 해당 알고리즘을 통해서 쥐와 고양이 각각의 경로를 구하고, 경로의 길이에 속도의 비를 적용하여 쥐의 탈출이 가능함을 증명할 수 있다.

④ 과제 4



지오지브라로 구현한 쥐와 고양이 게임 (구)

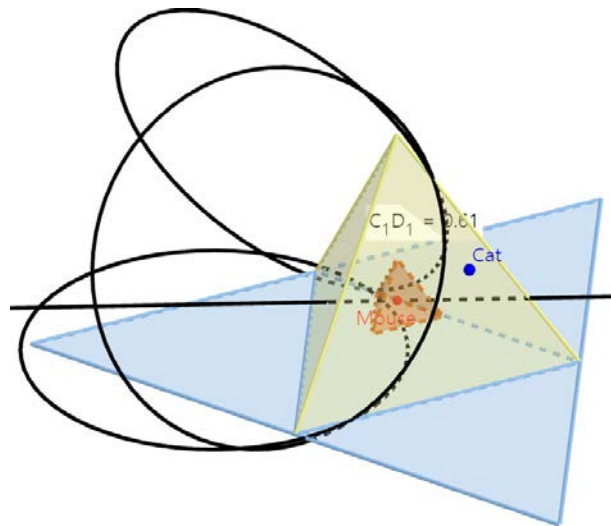
입체도형인 구에서의 탈출 방법을 탐구하기 위해서, 구면에서의 최단거리경로가 대원으로 나타난다는 사실을 이용한다. 즉 구의 중심과 쥐, 고양이를 지나는 평면을 그린 뒤, 해당 평면을 과제 ①의 쥐와 고양이 게임 (원)과 같이 생각한다. 그렇게 되면, 원과 마찬가지로 쥐의 속도는 1, 고양이의 속도는 4이고 내부의 파란색 원의 반지름은 외부의 노란색 원의 반지름의  $\frac{1}{4}$ 이므로, 쥐가 파란색 원 속에서 계속해서 돈다면 각속도 차이에 의해서  $\angle MOC$ 의 크기는 점점 커지게 된다. 쥐와 고양이의 사잇각의 크기가  $180^\circ$ 가 되었을 때 쥐가 파란색 원에서 노란색 원 밖으로 이동하면, 쥐의 이동거리는 3, 고양이의 이동거리는  $3\pi$ 이므로  $3 \times 4 < 3\pi$ 이므로 결국 내부의 파란색 구에서 외부의 노란색 구로 이동할 수 있게 되고 쥐는 탈출이 가능하다.

위의 증명으로 해당 도형에서 쥐의 탈출이 가능함을 밝혀내었다. 해당 도형을 작도하는 방법은 다음과 같다.

- ① 중심을 점  $O$ 로 하고 반지름의 길이가 4인 구  $c$ 를 작도한다.
- ② 중심을 점  $O$ 로 하고 반지름의 길이가 1인 구  $c'$ 를 작도한다.
- ③ 구  $c'$ 의 겉면 상의 한 점  $Mouse$ 를 잡고, 구  $c$ 의 겉면 상의 한 점  $Cat$ 를 잡는다.
- ④ 구의 중심  $O$ , 점  $Mouse$ , 점  $Cat$ 을 지나는 평면을 그린다.

결론: 측지선의 개념을 이용하여 입체도형인 구를 평면도형인 원의 풀이 방법을 이용하여 과제 ①의 쥐와 고양이 게임 (원)과 같은 결과를 얻어낼 수 있다.

⑤ 과제 5



지오지브라로 구현한 쥐와 고양이 게임 (정사면체)

입체도형인 정사면체에서의 탈출 방법을 탐구하기 위해서, 전개도를 생각해 보자. Jim Stewart의 최단 이동 거리 알고리즘화에 수록된 그림[4]과 같이 전개도화를 통해서 입체도형에서의 최단 이동 거리 문제를 해결할 수 있었다. 먼저 정사면체를 전개도화 하여 평면도형으로 생각한다. 그리고 과제 ②의 쥐와 고양이 게임 (정삼각형)에서의 증명을 이용하여, 쥐와 고양이가 각속도 차이를 이용하여 정사면체의 정반대편에 위치할 수 있다는 점을 이용하여 쥐와 고양이를 각각 정반대편에 위치시킨다. 다음으로 내부의 정사면체에 위치한 쥐의 위치에서 외부의 정사면체까지 거리가 가장 짧은 지점 (쥐의 예상 탈출 지점)를 측정하고, 전개도를 이용하여, 고양이의 설정위치에서 쥐의 예상 탈출 지점까지 거리를 계산한다. 이때, 쥐의 이동거리는 그림과 같이 0.61이고, 고양이의 이동거리는 6이다.  $0.61 \times 4 < 6$ 이므로 쥐는 탈출이 가능하다. 다른 방법으로 쥐의 이동거리를 역으로 이용해보았다. 첫 번째 방법과 마찬가지로 쥐는 내부의 정사면체의 한 면의 중심에 위치해 있고, 고양이는 외부의 정사면체 위의 점이다. 이때 쥐의 예상 이동거리는 2.44이고, 쥐의 예상 탈출 지점

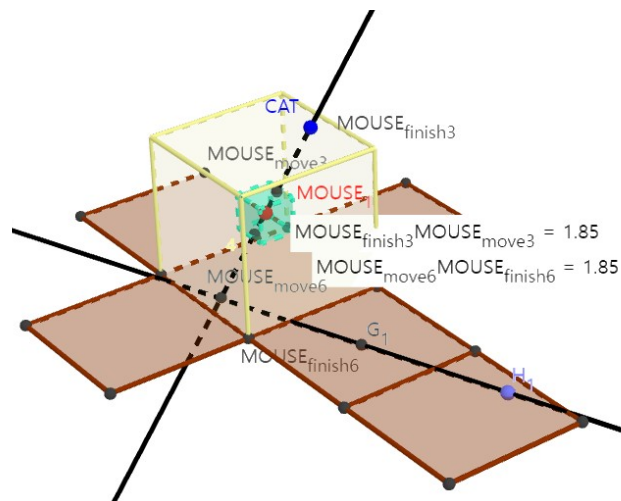
에서 반지름 2.44의 원을 작도한다. 이때 원이 정사면체의 한 면을 벗어난 부분은 원주와 정사면체의 교점을 이용해서 옆의 삼각형에 표현했다.(그림의 3개의 원) 이 도형 또한 내부의 주황색 정사면체에서 외부의 노란색 정사면체의 면 위의 지점까지의 거리는 주황색 정사면체의 꼭짓점에서 차이가 발생하나, 주황색 정사면체를 계속해서 회전하는 쥐를 잡기 위해, 고양이도 계속해서 회전한다는 조건을 추가했다.

위의 증명으로 해당 도형에서 쥐의 탈출이 가능함을 밝혀내었다. 해당 도형을 작도하는 방법은 다음과 같다.

- ① 한 변의 길이를 4로 하는 정사면체  $t$ 를 작도한다.
- ② 무게중심  $G$ 를 잡고, 무게중심을 공유하고, 한 변의 길이를 1로 하는 정사면체  $t'$ 를 작도한다.
- ③ 정사면체  $t'$ 의 겉면 상의 한 점  $Mouse$ 를 잡고, 정사면체  $t$ 의 겉면 상의 한 점  $Cat$ 를 잡는다.
- ④ 정사면체  $t$ 의 전개도를 작도한다.
- ⑤ 쥐의 예상 탈출 지점을 정사면체  $t$ 에 표시하고, 반지름이 2.44인 원을 그린다.
- ⑥ 원주와 정사면체의 교점을 잡고, 이 점들을 지나고, 반지름이 2.44인 원을 그린다.

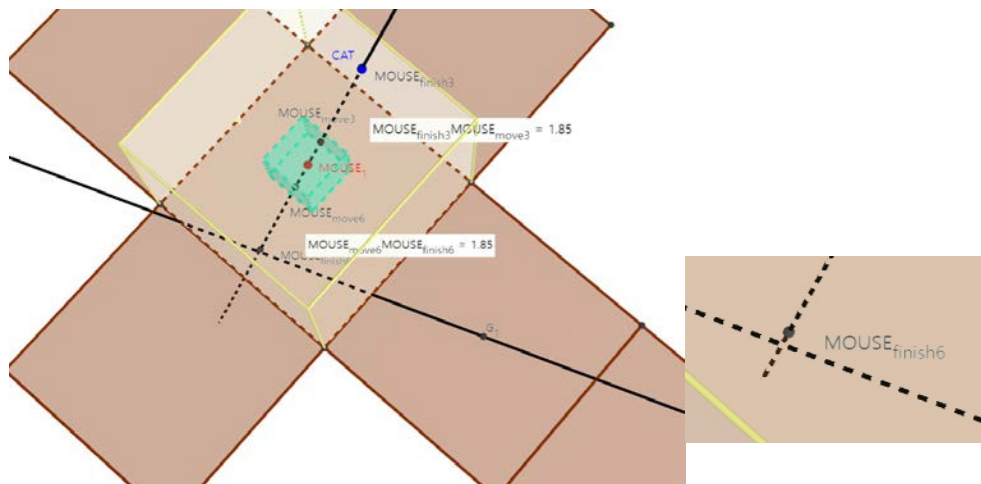
결론: 전개도를 이용하여 입체도형인 정사면체를 평면도형인 정삼각형의 풀이 방법을 이용하여 탈출 방법을 모색해보았고, 예상 탈출 지점의 추론과, 속도 비를 이용한 거리 계산으로 고양이의 위치에 따른 쥐의 탈출 가능 여부도 판단해 볼 수 있다.

## ⑥ 과제 6



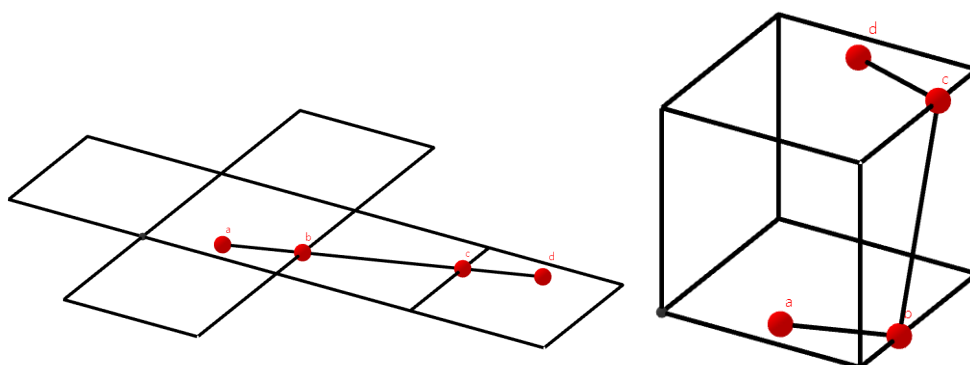
지오지브라로 구현한 쥐과 고양이 게임 (정육면체)

입체도형인 정육면체에서의 탈출 방법을 탐구하기 위해서, 정사면체와 마찬가지로 전개도를 생각해 보자. 먼저 정육면체를 전개도화 하여 평면도형으로 생각한다. 그리고 과제 ③의 쥐와 고양이 게임 (정사각형)에서의 증명을 이용하여, 쥐와 고양이가 각속도 차이를 이용하여 정육면체의 정반대편에 위치할 수 있다는 점을 이용하여 쥐와 고양이를 각각 정반대편에 위치시킨다. 다음으로 내부의 정육면체에 위치한 쥐의 위치에서 외부의 정육면체까지 거리가 가장 짧은 지점(쥐의 예상 탈출 지점)를 측정하고, 전개도를 이용하여, 고양이의 설정위치에서 쥐의 예상 탈출 지점까지 거리를 계산한다. 이때, 쥐의 이동거리는 1.5이고, 고양이의 이동거리는 8이다.  $1.5 \times 4 < 8$ 이므로 쥐는 탈출이 가능하다. 다른 방법으로 쥐의 도착지점을 추론해 보았다. 측지선의 개념을 이용하여 쥐의 예상 탈출 지점, 고양이의 위치, 쥐와 고양이 사이의 한 면의 중심( $G_1$ )을 지나는 평면을 그리고, 고양이  $\rightarrow$  평면과 모서리의 교점1  $\rightarrow$  평면과 모서리의 교점2  $\rightarrow$  쥐 까지의 경로가 최소일 것이라고 판단하고, 지오지브라 상에서 구현해 보았다. 하지만 결과는 예상과 달리 최단거리가 될 수 없었다.



(전개도 반례 사진)

새롭게 발생한 문제를 해결하기 위해서, 먼저 전개도에 선분을 작도한 뒤, 전개도를 접는 방법을 생각해 보았다. 그리고 이전 방법에서의 오류를 찾게 되었다.



내가 계획한 전개도이다. 쥐의 탈출위치를  $a$ 로 고양이의 위치를  $d$ (4번째 빨간 점)으로 설정하고, 선분  $\overline{bc}$ 의 중점을 지나는 점과  $a, d$ 를 지나는 평면을 생각했던 것이다. 하지만 실제로는 오른쪽의 그림처럼 전개도를 접었을 때, 이 네 점들은 한 평면 위에 있지 않다. 이것이 오류가 발생한 원인이었고, 이를 도입하여 문제를 해결해보면, 쥐의 이동거리는 1.5이고, 고양이의 이동거리는  $\frac{8\sqrt{10}}{3}$ 이다.  $1.5 \times 4 < \frac{8\sqrt{10}}{3}$ 이므로 쥐는 탈출이 가능하다. 이 도형 또한 내부의 초록색 정육면체에서 외부의 노란색 정육면체의 면 위의 지점까지의 거리는 초록색 정육면체의 꼭짓점에서 차이가 발생하나, 초록색 정육면체를 계속해서 회전하는 쥐를 잡기 위해, 고양이도 계속해서 회전한다는 조건을 추가했다.

위의 증명으로 해당 도형에서 쥐의 탈출이 가능함을 밝혀내었다. 해당 도형을 작도하는 방법은 다음과 같다.

- ① 한 변의 길이를 4로 하는 정육면체  $s$ 를 작도한다.
- ② 중심  $G$ 를 잡고, 중심을 공유하고, 한 변의 길이를 1로 하는 정육면체  $s'$ 를 작도한다.
- ③ 정육면체  $s'$ 의 겉면 상의 한 점 *Mouse*를 잡고, 정육면체  $s$ 의 겉면 상의 한 점 *Cat*를 잡는다.
- ④ 정육면체  $s$ 의 전개도를 작도한다.
- ⑤ 쥐의 예상 탈출 지점을 정육면체  $s$ 에 표시하고,  $\overline{MouseCat}$ 과 전개도의 교점을 잡는다.

결론: 전개도를 이용하여 입체도형인 정육면체를 전개도화해서, 두 점 사이의 최단 경로를 구해보고, 다른 과제들과는 달리 탐구 과정의 오류를 발견하고, 이를 해결하며 더 다양한 작도의 방법에 대해 깊이 있게 알게 되었다.

과제 5와 과제 6은 또한 천진동·이제한의 접근 방법[1]과 같이 다면체의 최단 이동 거리 알고리즘화를 통해서 해결할 수도 있었다. 해당 알고리즘을 통해서 쥐와 고양이 각각의 경로를 구하고, 경로의 길이에 속도의 비를 적용하여 쥐의 탈출이 가능함을 증명할 수 있다.

Jim Stewart의 최단 이동 거리 알고리즘화에 수록된 그림[2]과 같이 전개도화를 통해서 과제 5와 과제 6을 해결할 수 있었다.

## VI. 결론 및 제언

취과 고양이 게임은 겹보기에는 단순한 게임이지만 실제로는 두 점 사이의 거리를 구하는 다양한 방법에 대해 논의할 수 있는 문제이다. 확장하여 생각해보면, 이 문제를 해결하기 위해서 각 차원의 도형을 단순화하고, 낮은 차원에서 환원해보는 것은 수학의 효과적인 접근법이다.

나는 이 문제를 탐구하기 위해 다양한 도형(원 외 6개 도형)을 작도해보고, 각 속도를 적용하여, 이 문제들의 특징을 일반화해보았다. 또한,  $\text{시간} = \frac{\text{거리}}{\text{속력}}$  을 이용하여 쥐의 탈출 지점까지 거리를 예측하고 이를 고양이라는 다른 대상에게 적용하며, 문제를 뒤집어서 생각해보기도 하고, 전개도를 이용하여 문제를 새로운 방법으로 해결하기 위한 시도 또한 해보았다.

도형의 작도의 어려움으로 인해 포물선, 타원과 같은 이차곡선에서의 탐구가 어려웠고, 정다각형과 같은 특수한 도형에 대한 탐구만을 진행하며, 자연에서 보이는 유연하고 변형된 도형에 대한 탐구는 진행할 수 없었다. 하지만 내가 한 연구를 바탕으로 특정한 도형에 대한 이해도를 높일 수 있었고, 더 나아가 이차곡선과 변형된 도형에도 적용할 수 있는 일반화된 특징을 찾는데 활용해보고자 한다.

## VII. 참고문헌

- [1] 천진동·이제한 (1990), Shortest paths on a Polyhedron, Department of Computer Science University of Kentucky Lexington
- [2] Jim Stewart (1992), Shortest Path Problems, Stanford University
- [3] Leonidas J. Guibas · John Hershberger (1987), Optimal Shortest Path Queries in a Simple Polygon, Computer Science Department, Stanford University
- [4] Numberphile (2019), Game of Cat and Mouse
- [5] Pressley (2010), Elementary Differential Geometry, Christian Bär Cambridge University Press
- [6] Richard N. Zare (1988), angular momentum, Wiley-Interscience



수학B

07\_수의 패턴이 그리는 그림 수 바람개비의 특징  
연구



# 수의 패턴이 그리는 그림: 수 바람개비의 특징 연구

김예진 (용문초등학교 6학년)  
손민권 (남문초등학교 6학년)  
심민재 (대치초등학교 6학년)  
옥지훈 (안민초등학교 6학년)  
주민영 (동성초등학교 6학년)

지도교수 : 이동환 (부산교육대학교 수학교육과)

**초록 :** 본 연구의 목적은 일정한 규칙에 따라 수의 패턴을 그림으로 나타냈을 때 발견할 수 있는 특징을 탐구하는 것이다. 한 점에서 시작한 그림이 다시 그 점으로 돌아오고 그 뒤로 같은 그림이 반복해서 만들어지는 현상에 주목하여, 이러한 그림을 수 바람개비라고 정의했다. 수 바람개비가 만들어지는 수의 패턴을 탐구하고 그 규칙을 일반화하였다. 그 과정에서 여러 가지 주기의 수열을 컴퓨터 프로그램을 이용해서 탐구해보았고, 그 결과 주기가 4의 배수가 아닌 수열은 바람개비 그림이 그려졌지만, 주기가 4의 배수인 수열은 대부분 바람개비 그림이 그려지지 않는 것을 알 수 있었다.

## I. 서론

고대 피타고라스 시대부터 도형과 수를 관련시키려는 시도가 있었다. 그러한 시도는 대개 특정한 어떤 도형에 특정한 어떤 수를 대응시키는 것인데, 도형에 대응시킨 수를 도형수라고 한다(박교식, 2004). 예를 들어, 정삼각형 모양의 배열을 만드는 점의 수로 이루어진 수열 1, 3, 6, 10, 15, ...을 삼각수라고 하는데, 삼각수도 일종의 도형수이다. 도형수가 도형에 수를 대응시키면서 수의 패턴을 만들어내는 활동이라면, 수의 패턴을 먼저 만들고 이에 대응하는 도형이나 그림을 만드는 활동도 가능할 것이다. 그래서 본 연구는 수의 패턴에 대응하는 그림을 만들고 그 특징을 살펴보고자 하였다.

3의 배수에서 각 자리의 숫자들을 더해서 한 자리의 수로 나타내는 규칙이 있다. 예를 들어, 12의 경우  $1+2=3$ 으로 나타내고 24는  $2+4=6$  그리고 39는  $3+9=12$ ,  $1+2=3$ 으로 나타내는 규칙에 따라 3의 배수를 나타내보면 <표 1>과 같이 3, 6, 9가 반복적으로 나타나는 것을 확인할 수 있다.

3단	3	6	9	12	15	18	21	24	27	30	33	36	39	...
수 패턴	3	6	9	3	6	9	3	6	9	3	6	9	3	...

<표 1> 3의 배수에서 찾은 수의 패턴

이런 규칙에 따라 2의 배수, 4의 배수 등을 나타내면서 수의 패턴을 발견할 수 있다. 이러한 수의 패턴에 따라 그림을 그리는 방법을 추가하여 그림으로 나타낼 수 있다. 본 연구는 특정한 수의 패턴을 만들고 이를 그림으로 나타냈을 때 어떠한 특징을 발견할 수 있는지 탐구하고자 한다.

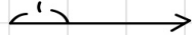

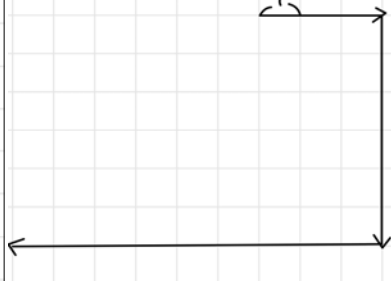
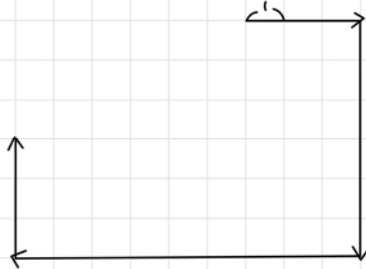

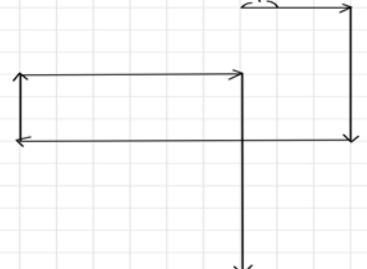
## II. 수 바람개비

### 가. 수의 패턴으로 그림 그리는 방법

방향	→	↓	←	↑	→	↓	←	↑	→	...
수 패턴	3	6	9	3	6	9	3	6	9	...

<표 2> 수의 패턴과 방향

수의 패턴에 4가지 방향(→, ↓, ←, ↑)을 적용하여 그림을 그릴 수 있다(<표 3> 참고).

→ 방향으로 3	↓ 방향으로 6	← 방향으로 9
		
↑ 방향으로 3	→ 방향으로 6	↓ 방향으로 9
		

<표 3> 수의 패턴으로 그림 그리는 방법

## 나. 수 바람개비의 정의

<표 3>의 그림들은 <표 2>를 그림으로 나타낸 것이며 그림이 완성되는 과정 또한 나타낸 것이다. 표를 사용해 그림을 그리며 우리에게 익숙한 모양이 나오게 놀랐고 이 그림 또한 반복되는 패턴과 규칙이 있음에 놀랐다. 한 점에서 시작한 그림이 다시 그 점으로 돌아오고 그 뒤로 같은 그림을 반복해서 만들어지며, 이 그림은 마치 바람개비처럼 생겼으며 수를 이용하여 만든 그림이기에 수 바람개비라고 정의했다.

## Ⅲ. 연구 결과

### 가. 수 바람개비가 만들어지는 조건

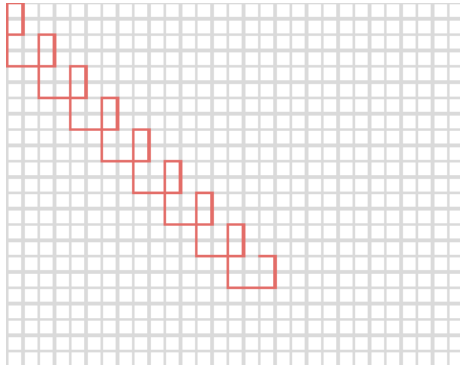
먼저 이 연구를 시작하게 만든 아이디어인 수의 배수들에 나타나는 수를 쓴 후 그 수의 각자리 수의 합을 한 자릿수가 나올 때까지 계속해서 나온 수를 이용해서 그림을 그려보았다. (예를 들어 12는  $1+2=3$ , 198은  $1+9+8=18$ ,  $1+8=9$ 이다) 2의 배수부터 9의 배수까지의 수를 이용해서 그림을 그려본 결과 모두 수 바람개비 그림이 그려지는 것을 확인할 수 있었다. 또한, 2의 배수부터 9의 배수까지는 각자리 수의 합이 1, 3, 9주기로 반복되는 것을 확인할 수 있었다.

2의 배수	2	4	6	8	10	12	14	16	18	20	22	24	...	
수 패턴	2	4	6	8	1	3	5	7	9	2	4	6	...	
3의 배수	3	6	9	12	15	18	21	24	27	30	33	36	...	
수 패턴	3	6	9	3	6	9	3	6	9	3	6	9	...	
4의 배수	4	8	12	16	20	24	28	32	36	40	44	48	...	
수 패턴	4	8	3	7	2	6	1	5	9	4	8	3	...	
5의 배수	5	10	15	20	25	30	35	40	45	50	55	60	...	
수 패턴	5	1	6	2	7	3	8	4	9	5	1	6	...	
6의 배수	6	12	18	24	30	36	42	48	54	60	66	72	...	
수 패턴	6	3	9	6	3	9	6	3	9	6	3	9	...	
7의 배수	7	14	21	28	35	42	49	56	63	70	77	84	...	
수 패턴	7	5	3	1	8	6	4	2	9	7	5	3	...	
8의 배수	8	16	24	32	40	48	56	64	72	80	88	96	...	
수 패턴	8	7	6	5	4	3	2	1	9	8	7	6	...	
9의 배수	9	18	27	36	45	54	63	72	81	90	99	108	...	
수 패턴	9	9	9	9	9	9	9	9	9	9	9	9	...	

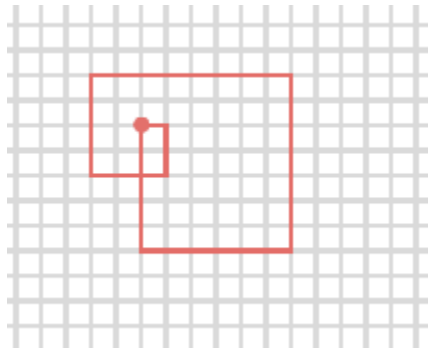
<표 4> 배수에서 나타나는 수 패턴

그다음은 숫자를 바꾸거나 주기를 바꾸는 등 여러 가지에 변화를 주어서 그림을 그려본 결과 주기가 {1, 2, 3, 5, 7} 등 4의 배수가 아니면 수 바람개비가 항상 그려지고, 주기가 4, 8 등 4의 배수이면 수 바람개비 그림이 그려지지 않는 경우가 많았다. 특히 주기가 4의 배수이고 수 바람개비 그림이 그려지지 않았을 경우 모두 한 쪽 방향으로 동일한 그림이 반복적으로 그려지는 현상이 있었다.

이를 확인한 후 스크래치 블록 코딩을 통해 수많은 주기를 빠른 속도로 확인해볼 수 있었다. 주기가 4의 배수가 아닌 경우에는 그림이 원점으로 돌아온다는 것은 변함없었지만, 주기가 4의 배수일 때는 그림이 원점으로 돌아오는 경우가 종종 생겼다. 주기가 4의 배수일 때 그림이 원점으로 돌아오는 경우를 알아보기 위해 주기가 1부터 5000까지 각각의 경우를 무작위로 지정하여 여러 번 돌려본 결과 원점으로 돌아오지만 4의 배수인 주기가 0, 1, 2개씩 생겼다. 하지만 4의 배수이면서 원점으로 돌아오는 주기는 수열을 바꿀 때마다 달라지는 것을 확인했다. 예를 들어 주기가 {1, 2, 3, 4}가 반복되는 경우에는 왼쪽 대각선 위로 사각형 모양이 반복적으로 나타나고, 주기가 {1, 2, 3, 4, 8, 7, 6, 5}가 반복되는 경우에는 사각형 모양의 수 바람개비 그림이 그려지는 것을 확인할 수 있었다.



[그림 1] 주기가 {1, 2, 3, 4}인 경우

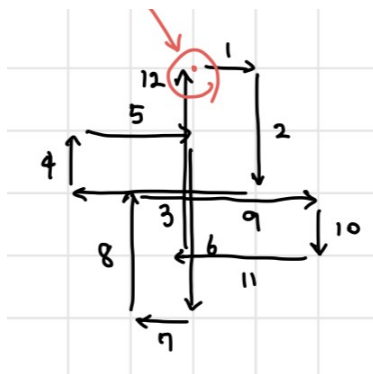


[그림 2] 주기가 {1, 2, 3, 4, 8, 7, 6, 5}인 경우

#### 나. 수 바람개비가 만들어지지 않는 이유

스크래치라는 프로그램을 통해 코딩으로 주기가 4의 배수일 때, 바람개비 그림이 그려지지 않는 것을 확인하였다. 그리고 그 이유를 찾아보았다.

반복되는 그림이 나타나기 위해서는 결국 여러 번을 거쳐 다시 원래 시작되었던 지점으로 와야 한다. 그렇지 않으면 바람개비 그림은 완성되지 않고 그림의 크기가 커지기만 할 것이다.



[그림 3] 3의 배수 주기의 그림에서 원래 시작되었던 지점으로 오는 과정

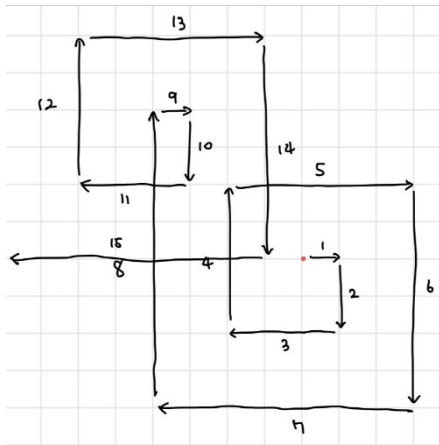
결국 주기가 4의 배수일 때 바람개비 그림이 그려지지 않는다는 것은 주기가 4의 배수일 때 원래 시작했던 지점으로 돌아올 수 없다는 것과 같다는 것이다. 그림을 그리는 방향도  $\rightarrow, \downarrow, \leftarrow, \uparrow$ 로 4가지인데, 숫자들의 주기도 4의 배수이면, 한 숫자에 모든 방향이 대응되지 않고, 특정한 숫자에 몇몇 방향만 대응되기 때문에 원래 시작되었던 지점으로 돌아오기 어렵다.

① 주기가 4의 배수인 경우

$\rightarrow$	$\downarrow$	$\leftarrow$	$\uparrow$
1	2	3	4
5	6	7	8
1	2	3	4
5	6	7	8

<표 5>

<표 5>에서  $\rightarrow$ 방향에 대응하는 수 1, 5와  $\leftarrow$  방향에 대응하는 수 3, 7이 서로 다르므로 원래 시작되었던 지점으로 돌아올 수 없다.



[그림 4]

실제로 <표 5>의 패턴을 [그림 4]와 같이 표현해보면, 원래 시작되었던 지점으로 돌아오지 않고, 그림의 크기만 무한대로 커지는 것을 알 수 있다.

② 주기가 4의 배수가 아닌 경우(5의 배수)

→	↓	←	↑
1	2	3	4
5	1	2	3
4	5	1	2
3	4	5	1
2	3	4	5

<표 6>

<표 6>에서 →, ↓, ←, ↑의 4개 방향에 모두 1이 대응되어 원점으로 돌아오게 되는 것을 확인할 수 있다. 또한 →, ↓, ←, ↑의 4개 방향에 모두 2가 대응한다. 숫자 3, 4, 5도 마찬가지로 모든 방향에 대응된다. 이와 같은 방식으로 →, ↓, ←, ↑의 4개 방향에 모두 같은 수가 대응하기 때문에 <표 6>의 패턴은 원점으로 돌아온다고 할 수 있다.

그래서 주기가 4의 배수이면 바람개비 그림이 나타나지 않으며, 원래 시작되었던 지점으로 돌아오기까지의 횟수는 무한대이다.

#### 다. 수 바람개비를 만드는 규칙의 일반화

위의 연구에서 찾은 규칙을 일반화해서 적용해보자.

##### ① 주기가 1인 수열

$$a, a, a, a, a, a, \dots$$

앞의 규칙대로 →, ↓, ←, ↑을 차례대로 적용하면, X축은 홀수 번째에 있는 숫자들이 더해지기 때문에  $a-a+a-a+\dots$ 가 반복된다. X축에 해당하는 숫자들의 합이 0이 되면 원점으로 돌아오는데, X축의 숫자의 개수가 2의 배수가 되면 합이 0이 되어 원점으로 돌아오게 된다. Y축도 생각해 보면, Y축도 같은 이유로 짝수일 경우 합이 0이 되어 원점으로 돌아온다.

##### ② 주기가 2인 수열

$$a, b, a, b, a, b, a, b, \dots$$

X축은 홀수 번째 숫자이므로,  $a-a+a-a+\dots$ 가 반복되고, 위와 같은 방식으로 짝수일 경우 합이 0이 되어 원점으로 돌아온다. Y축은  $b-b+b-b+\dots$ 가 반복되어 짝수개일 경우 합이 0이 되어 원점으로 돌아온다.

③ 주기가 3인 수열

$$a, b, c, a, b, c, a, b, c, \dots$$

X축은 홀수 번째 숫자이므로, X축에 해당하는 숫자들의 합은  $a-c+b-a+c-b+\dots$ 가 반복된다. 합이 처음으로 0이 되려면 X축 숫자 기준으로는 6번째 숫자여야 하고, X축 숫자의 개수가 6의 배수가 될 때마다 합이 0이 된다. Y축은 짝수 번째 숫자이므로, Y축에 해당하는 숫자들의 합은  $-b+a-c+b-a+c-\dots$ 가 반복되고, X축과 동일하게 6번째 숫자가 왔을 때 합이 0이 되어 원점으로 돌아올 수 있다. 따라서 X축과 Y축이 모두 원점으로 돌아오려면 주기가 3인 수열이 4번 반복되어 12번째 숫자가 왔을 때 원점으로 돌아온다는 것을 알 수 있다.

④ 주기가 4인 수열

$$a, b, c, d, a, b, c, d, \dots$$

X축은 홀수 번째 숫자이므로, X축에 해당하는 숫자들의 합은  $a-c+a-c+\dots$ 가 반복된다. 따라서 같은 숫자가 계속해서 더해지고, 빼지기 때문에 합이 0이 될 수 없다. Y축도 같은 방식으로 하면  $b-d+b-d+\dots$ 가 반복되어 0이 될 수 없다.

⑤ 주기가 5인 수열

$$a, b, c, d, e, \dots$$

위에서와 같은 방식으로 X축에는 홀수 번째 숫자들이 들어간다. 따라서 숫자들의 합은  $a-c+e-b+d-a+c-\dots$ 가 반복된다. 이때는 X축 숫자가 두 번째로 모두 들어오게 되는 횟수인 10번째 숫자가 합해질 때 합이 0이 될 수 있다. Y축도 같으므로 총 20번째의 숫자가 반복되어야 함을 알 수 있다.

⑥ 원점으로 돌아오는 최소 횟수

4의 배수의 주기에서는 특정한 경우를 제외하면 원점으로 돌아올 수 없고, 주기가 N인 수열에서 원점으로 돌아오는 최소 횟수는 4와 N의 최소공배수인 것을 알 수 있다. 그 이유를 생각해보면 아래와 같이 증명할 수 있다.

주기는 다음과 같이 정의하였다.

$n_1, n_2, n_3, n_4 \dots$ : 1주기 $n_1, n_2, n_3, n_4 \dots$ : 2주기 $\vdots$ 주기 = n개의 숫자가 반복되는 1순환
---

n개의 숫자가 반복되는 수열이 있다고 가정하자.

원점에서 출발한 뒤에 다시 원점으로 돌아오려면 X축과 Y축 모두 원점으로 돌아와야 한다.

$n=4k-3(k \geq 1$ 인 자연수)일 때의 X축을 생각해보자.

X축에는 홀수 번째의 숫자들이 더해지고, 숫자들이 더해질 때 +와 -가 반복되어 더해진다.

X축에 들어오는 숫자가 최소 2번은 반복되어야 합이 0이 될 수 있다. 주기가 1번 지나면  $2k-1$ 개의 숫자가 들어오게 되고, 2번째 주기에서는 1번 주기에서 더해지지 않았던 남은  $2k-2$ 개의 숫자가 더해진다. 그다음 3번째 주기에서는 1주기에서 더해진 숫자가 다시 반복되는데,  $4k-3$ 은 홀수이므로 +,-가 반복되었을 때 1주기와 3주기에서는 서로 다른 부호가 앞에 붙게 된다. 따라서  $n=4k-3$ 일 때, x축은 4주기를 지나면 반드시 합이 0이 된다. y축도 같은 방식으로 더해지기 때문에  $n=4k-3$ 일 때, 숫자가 4주기를 지난 뒤인  $4 \times (4k-3)$ 번째에는 x축과 y축의 합이 모두 0이 되어 원점으로 돌아온다.

같은 방식으로  $n=4k-1$ ( $k \geq 1$ 인 자연수)일 때의 X축을 생각해보자. 이 경우에도 x축에 들어오는 숫자가 최소 2번은 반복되어야 합이 0이 될 수 있다. 주기가 1번 지나면  $2k$ 의 수가 합해지고, 2번째 주기에서는 첫 번째 주기에서 더해지지 않은 새로운  $2k-1$ 의 숫자가 더해진다. 다음 3번째 주기에서는 1주기에서 더해진 숫자가 다시 반복되는데,  $4k-1$ 은 홀수이므로 1주기와 3주기에서는 서로 다른 부호가 앞에 붙게 되어 4주기를 지나면 반드시 합이 0이 된다. y축도 같은 방식으로 더해지기 때문에  $n=4k-1$ 개의 숫자가 반복되는 수열은  $4 \times (4k-1)$ 개의 숫자가 반복되는 4주기가 지나면 x축과 y축 모두 합이 0이 되어 원점으로 돌아온다.

같은 방식으로 다시  $n=4k-2$ ( $k \geq 1$ 인 자연수)일 때의 X축을 생각해보자. 이 경우의 특징은  $n$ 이 짝수이기 때문에 홀수 번째와 짝수 번째의 숫자들이 변하지 않는다는 것이다. 따라서 앞선 경우와 다르게 2주기가 지나게 되면 x축과 y축에서 각각 숫자가 2번씩 반복된다. 또한, 1주기에서  $2k-1$ 개의 숫자가 더해지는데,  $2k-1$ 은 홀수이기 때문에 다음번 주기에서는 +,-가 바뀐 상태로 더해지게 되어  $n=4k-2$ 개의 개수가 반복되는 수열에는 2주기만 반복되어도 원점으로 돌아오게 된다. 즉,  $2 \times (4k-2)$ 번째에 원점으로 돌아온다.

마지막으로  $n=4k$ ( $k \geq 1$ 인 자연수)일 때의 X축을 생각해보자. 바로 앞의  $4k-2$ 와 같이  $n$ 이 짝수이기 때문에 홀수와 짝수 번째의 숫자들이 변하지 않는다. 또한 1주기에 들어오는 숫자는  $2k$ 개인데,  $2k$ 는 짝수이기 때문에 부호 또한 바뀌지 않고 반복된다. 따라서 앞선 경우들과 다르게 합이 0이 될 수 없다.

위 경우들을 표로 정리하면 다음과 같다.

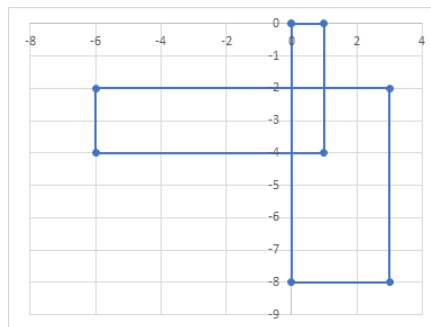
n개의 숫자가 반복되는 수열	$n=4k-3$	$n=4k-2$	$n=4k-1$	$n=4k$
최소 주기	4주기	2주기	4주기	-
최소 번째 숫자	$4 \times (4k-1)$	$2 \times (4k-2)$	$4 \times (4k-1)$	-

앞에서는 숫자가 반복되는 수열에서 원점으로 돌아오는 경우를 알아보았다. 하지만 특정한 경우에 숫자가 반복되지 않을 때, 숫자들이 특정한 합을 이룬다면 원점으로 돌아올 수 있다.

주기가 없는 수열이자 유한개의 숫자를 가진 수열인  $a, b, c, d, e, f, g, h$ 가 X축, Y축 각각 4번째 숫자가 나오면 원점으로 돌아온다고 가정해보자. X축, Y축에 대한 식은 아래와 같다.

$$a-c+e-g=0, -b+d-f+h=0$$

위 식을 양변으로 정리하면, X축은  $a+e=c+g$ , Y축은  $b+f=d+h$ 일 때 원점으로 돌아올 수 있다. 예를 들어, 1, 4, 7, 2, 9, 6, 3, 8의 경우,  $1+9=7+3$ ,  $4+6=7+3$ 이다. 이를 그려 보면 아래와 같이 원점으로 돌아오는 것을 확인할 수 있다.



따라서 숫자가 반복되지 않더라도 8번째 숫자가 나왔을 때 원점으로 돌아오도록 한다면, 위와 같은 합의 규칙을 만족하면 된다.

이는 짝수가 아닌 홀수개의 숫자가 있더라도 가능하다. 이번에는 총 9개의 숫자가 지난 후에 원점으로 돌아온다고 가정해보자,  $a, b, c, d, e, f, g, h, I$ 의 숫자가 원점으로 돌아오려면 아래와 같은 식을 만족해야 한다.

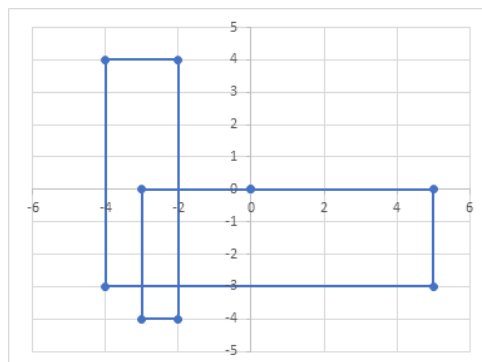
$$a-c+e-g+i=0, -b+d-f+h=0$$

위 식을 다시 정리하여 합에 관한 공식으로 정리하면 아래와 같다.

$$a+e+i=c+g, d+h=b+f$$

위 식을 만족하면 반복되지 않은 9개의 수도 원점으로 돌아올 수 있다.

예를 들어 수열 5, 3, 9, 7, 2, 8, 1, 4, 3도 같은 방식으로 그림을 그리면 원점으로 돌아오는 것을 확인할 수 있다.



즉, 위의 두 경우를 보았을 때, 유한개의 숫자들이 무작위로 배열되어 있더라도, 특정한 조건을 만족하면 원점으로 돌아올 수 있다는 것을 알 수 있다. 특정한 조건은 각 숫자의 합이 서로 같아야 한다는 것인데, 이를 일반화 시키면 아래와 같다.

아래와 같은 수열이 있다고 가정하자. (단,  $k \geq 1$ 인 자연수)

$$a_1, a_2, a_3, a_4, a_5, a_6, \dots, a_{4k-3}, a_{4k-2}, a_{4k-1}, a_{4k}, \dots$$

X축의 합이 0이 되어야 원점으로 돌아올 수 있으므로, 이를 식으로 쓰면 아래와 같다.

$$a_1 - a_3 + a_5 - a_7 + \dots + a_{4k-3} - a_{4k-1} + \dots = 0$$

Y축도 같은 방식으로 식을 쓰면 아래와 같다.

$$-a_2 + a_4 - a_6 + a_8 + \dots - a_{4k-2} + a_{4k} - \dots = 0$$

덧셈에 관한 식으로 두 식을 정리하면 아래와 같다.

$$a_1 + a_5 + \dots + a_{4k-3} + \dots = a_3 + a_7 + \dots + a_{4k-1} + \dots$$

$$a_2 + a_6 + \dots + a_{4k-2} + \dots = a_4 + a_8 + \dots + a_{4k} + \dots$$

위 두 식을 기호를 이용하여 일반화시키면 아래와 같이 쓸 수 있다.

$$\sum_{k=1} a_{4k-3} = \sum_{k=1} a_{4k-1}, \sum_{k=1} a_{4k-2} = \sum_{k=1} a_{4k}$$

각 합이 위와 같은 식을 만족한다면 수열의 반복 여부와 숫자 개수와 관계없이 원점으로 돌아오는 수열을 만들 수 있다.

#### IV. 결론 및 제언

본 연구를 통해 특정한 수의 패턴을 만들고 이를 그림으로 나타냈을 때 어떠한 특징을 발견할 수 있는지 탐구 해 보았다.

눈에 쉽게 보이지 않는 규칙들을 그림으로 나타내어 보니 특정한 주기의 숫자를 오른쪽, 아래쪽, 왼쪽, 위쪽 순서로 몇 번씩 반복하면 반복되는 문양을 나타낼 수 있는 것을 알 수 있다. 이러한 수의 패턴을 수 바람개비라고 정의하고 2단부터 9단 까지 수를 이용해서 그림을 그려본 결과 모두 “바람개비 그림”이 그려지는 것을 확인함으로써 수 바람개비가 만들어지는 조건을 알 수 있었다. 또한, 2단부터 9단까지는 각자리 수의 합이 1, 3, 9주기로 반복되는 것을 확인할 수 있었다. 구구단의 규칙과 방향의 규칙, 두 개의 규칙만으로도 수많은 규칙이 있는 그림을 만들 수 있고 그 수 많은 규칙 들 중에서도 구구단과 방향 규칙의 주기, 합이 일정하다는 점이 그림 안에서도 발견되었다.

여기에서 더 나아가 숫자를 바꾸거나 주기를 바꾸는 등 여러 가지에 변화를 주어서 그림을 그려본 결과 주기가 1, 2, 3, 5, 7 등 4의 배수가 아니면 “바람개비 그림”이 항상 그려지고, 주기가 4, 8 등 4의 배수이면 “바람개비 그림”이 그려지지 않는 경우가 많았다. 특히 주기가 4의 배수이고 “바람개비 그림”이 그려지지 않았을 경우

모두 한 쪽 방향으로 같은 그림이 반복적으로 그려지는 현상을 발견할 수 있었다. 하지만 주기가 4의 배수 개여도 수 바람개비를 성립하는 그림이 있었다. 그 경우는 오른쪽으로 가는 수와 왼쪽으로 가는 수의 합이 아래쪽으로 가는 수와 위쪽으로 가는 수의 합이 같은 경우이다.

그리하여 본 연구를 통해 스크래치 블록 코딩을 통해 수많은 주기와 수를 빠른 속도로 확인해볼 수 있었고 주기가 4의 배수가 아닌 경우에는 그림이 원점으로 돌아온다는 것은 변함없었지만, 주기가 4의 배수일 때는 그림이 원점으로 돌아오는 경우가 종종 생겼는데 4의 배수이면서 원점으로 돌아오는 주기는 수열을 바꿀 때마다 달라지는 것을 확인하였다.

마지막으로 수 바람개비를 성립하는 규칙을 일반화해보기 위해서 주기가 1인 수열부터 주기가 5인 수열까지 탐구해보니 원점으로 돌아오는 최소 횟수는 4의 배수의 주기에서는 특정한 경우를 제외하면 원점으로 돌아올 수 없고, 주기가  $N$ 인 수열에서 원점으로 돌아오는 최소 횟수는 4와  $N$ 의 최소공배수인 것을 알 수 있었다.

## V. 참고 문헌

박교식 (2004). 도형에 대응시킨 수. 서울: 경문사

수학B

## 08\_최소 힌트의 개수를 갖는 변형스도쿠 만들기



# 최소 힌트의 개수를 갖는 변형 스도쿠 만들기

김태은 (삼정자중학교 2학년)

박경민 (대곡중학교 2학년)

이예정 (거제제일중학교 2학년)

임재윤 (거제고현중학교 2학년)

지도교수 : 황혜정 (경상국립대학교 수학과)

## 요약(초록)

우리는 일반적으로 보는 정사각형으로만 이루어진 스도쿠를 접했으며 정사각형이 아닌 7자 또는 L자와 같은 여러 가지 모양으로 이루어진 스도쿠를 만들었을 때 이 스도쿠에 처음 주어지는 힌트의 위치와 그 개수의 최소성에 대한 궁금증을 갖고 연구를 시작하게 되었다.

$n \times n$ 개의 칸으로 이루어진 큰 정사각형이  $n$ 개의 작은 정사각형 모양으로 만들어진 다양한 모형  $n$ 개를 갖는 스도쿠를 변형  $n$ -스도쿠라 하고 이 스도쿠의 최소 힌트의 개수  $n-1$ 을 갖는 스도쿠를 만들어 보기로 하였으며 먼저 변형 4-스도쿠를 21개 만들었고 각각에 변형 4( $k$ )라는 번호를 붙여서 최소힌트의 개수를 찾았다.

그렇게 찾은 스도쿠들 중에서  $k=4, 5, 9, 10, 11, 12, 20$ 의 크기를  $n$ 으로 확대한 변형  $n(k)$  스도쿠를 만들었고 이 스도쿠의 힌트의 위치 또한 일반화했을 때 변형  $n(k)$  스도쿠를 완성할 수 있다는 결론을 얻게 되었다.

마지막으로 모형  $T_1$ 을 포함하는 변형 4( $k$ ) 스도쿠를 이어 붙여서 만든 스도쿠에 대하여 1부터 4까지의 숫자를 채워가는 변형 4( $k$ )의 최소힌트의 개수가 어떻게 변하는지를 연구하였다. 변형 4( $k$ )에서 제공되는 힌트의 개수가 최소힌트의 개수보다 많고 그 위치가 다르더라도 이어 붙여진 변형 4( $k$ ) 스도쿠를 완성할 수 있음을 확인하였다.

## I. 연구 동기 및 연구 목적

### 1) 연구 동기

우리가 재미로 하던 스도쿠 게임은 작은 정사각형들로 이루어진 큰 정사각형 모양(표준 스도쿠)이거나  $n$ 개의 연속된 블록으로 채워진 큰 정사각형 모양(변형 스도쿠)에 몇 개의 힌트를 바탕으로 나머지 칸을 일정한 숫자를 중복 없이 한 번씩 채워가는 것이다. 스도쿠에서 제공되는 힌트의 개수를 최소로 하는 스도쿠를 만들어 보기 위하여 이 연구를 시작하게 되었다.

### 2) 연구 목적

2018년에 연구된 결과에서  $n \times n$  스도쿠에서 제공되는 힌트의 개수는 적어도  $(n-1)$ 개 이상임을 밝혔고 우리는 이 결과를 바탕으로 가로 세로의 칸이 4개로 이루어진 스도쿠에 대하여 최소 힌트의 개수가  $(n-1)$ 개를 갖는 스도쿠를 찾고 그 찾은 스도쿠에 대하여 스도쿠를 크게 만들었을 때 힌트의 개수와 그 힌트의 위치에 대한 규칙을 찾는데 목적을 두고 연구를 시작하였다. 이런 스도쿠들을 이어 붙여가면서 새로운 스도쿠를 만들고 이 경우에는 힌트의 개수가 어떻게 변하는지를 관찰하였다.

## II. 용어의 정의와 기호

### 1) 스도쿠

스도쿠는 숫자 퍼즐로  $n \times n$  으로 나타낼 때 가로  $n$ 칸 세로  $n$ 칸으로 구성되어 있으며 가로, 세로 칸에 숫자들을 겹치지 않고 가로와 세로 그리고 각  $n$ 개의 칸들로 연결된  $n$ 개의 블록의 칸에도 숫자가 겹치지 않도록 숫자를 칸에 넣는 것이다.

### 2) 표준 $n$ -스도쿠

가로  $n$ 줄, 세로  $n$ 개의 줄(즉,  $n^2$ 칸으로 이루어진) 정사각형 모양에 칸의 개수가  $n$ 개인  $\sqrt{n} \times \sqrt{n}$  작은 정사각형  $n$ 개로 이루어진  $n \times n$  정사각형에 스도쿠의 규칙에 따라 1부터  $n$ 까지의 자연수를 겹치지 않게 채우는 퍼즐을 표준  $n$ -스도쿠라 한다.

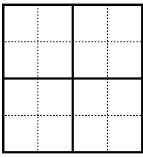
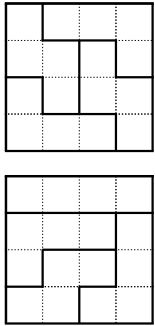
### 3) 변형 $n$ -스도쿠

3 이상인 자연수  $n$ 에 대하여 표준  $n$ -스도쿠가 아닌  $n \times n$  스도쿠를 변형  $n$ -스

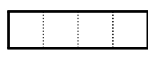


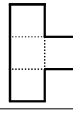
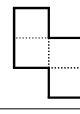
도쿠라 한다. 즉,  $n$ 개의 칸을 변끼리 연결하여 만든  $n$ 개의 서로 다른 블록으로 채워진  $n \times n$  정사각형의 스도쿠를 변형  $n$ -스도쿠라 한다.

4) 스도쿠의 좌표  $(i, j)$  : 위에서부터  $i$ 번째 행과 왼쪽에서부터  $j$ 번째 열이 만나는 칸의 좌표를  $(i, j)$ 라 한다. ([그림 1-3]참고)

예를 들면, [그림1-1]은 표준 4-스도쿠이고 [그림1-2]는 표준 4-스도쿠가 아닌 변형 4-스도쿠이다.

		<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>(1,1)</td><td>(1,2)</td><td>...</td><td>(1,j)</td><td>...</td><td>(1,n)</td></tr> <tr><td>(2,1)</td><td>(2,2)</td><td>...</td><td>(2,j)</td><td>...</td><td>(2,n)</td></tr> <tr><td>⋮</td><td>⋮</td><td>⋱</td><td>⋮</td><td>⋱</td><td>⋮</td></tr> <tr><td>(i,1)</td><td>(i,2)</td><td>...</td><td>(i,j)</td><td>...</td><td>(i,n)</td></tr> <tr><td>⋮</td><td>⋮</td><td>⋱</td><td>⋮</td><td>⋱</td><td>⋮</td></tr> <tr><td>(n,1)</td><td>(n,2)</td><td>...</td><td>(n,j)</td><td>...</td><td>(n,n)</td></tr> </table>	(1,1)	(1,2)	...	(1,j)	...	(1,n)	(2,1)	(2,2)	...	(2,j)	...	(2,n)	⋮	⋮	⋱	⋮	⋱	⋮	(i,1)	(i,2)	...	(i,j)	...	(i,n)	⋮	⋮	⋱	⋮	⋱	⋮	(n,1)	(n,2)	...	(n,j)	...	(n,n)
(1,1)	(1,2)	...	(1,j)	...	(1,n)																																	
(2,1)	(2,2)	...	(2,j)	...	(2,n)																																	
⋮	⋮	⋱	⋮	⋱	⋮																																	
(i,1)	(i,2)	...	(i,j)	...	(i,n)																																	
⋮	⋮	⋱	⋮	⋱	⋮																																	
(n,1)	(n,2)	...	(n,j)	...	(n,n)																																	
[그림1-1] 표준 4-스도쿠	[그림1-2] 변형 4-스도쿠	[그림1-3] 스도쿠의 좌표 $(i, j)$																																				

5) 작은 블록에 이름 붙이기 : 작은 정사각형 4개로 만들어진 모양을 아래와 같이 이름을 붙인다. (단, 뒤집거나 돌리기를 해서 같은 모양이 나오면 같은 걸로 간주한다.)

				
$T_1$	$T_2$	$T_3$	$T_4$	$T_5$

6) 변형 4( $k$ ) 스도쿠

5개의 블록  $T_1, T_2, T_3, T_4, T_5$ 을 이용해서 만들 수 있는 표준 4-스도쿠를 제외한 21개의 변형 4-스도쿠에 다음과 같이 번호를 부여한다. ([그림 1-4] 참고)

7) 변형 4( $k$ )의 확장 스도쿠들 ([그림 1-5] 참고)

변형 4( $k$ ) 스도쿠의 스도쿠들을 이어 붙여서 만든 스도쿠들이다.

변형 4(1)	변형 4(2)	변형 4(3)	변형 4(4)	변형 4(5)	변형 4(6)
변형 4(7)	변형 4(8)	변형 4(9)	변형 4(10)	변형 4(11)	변형 4(12)
변형 4(13)	변형 4(14)	변형 4(15)	변형 4(16)	변형 4(17)	변형 4(18)
			[그림1-4] 변형 4(k) 스토쿠		
변형 4(19)	변형 4(20)	변형 4(21)			

변형 4(5)	변형 5(5)	변형 6(5)	
변형 4x8(11,12)		변형 4x8(12,12)	

[그림1-5] 변형 4(k)의 확장 스토쿠

① 변형  $(4(k), m \times n)$  스토쿠

변형  $4(k)$  스토쿠  $mn$ 개를 가로  $m$ 개, 세로  $n$ 개를 이어 붙여 만든 스토쿠

② 변형  $(4(k), m \times n)^*$  스토쿠

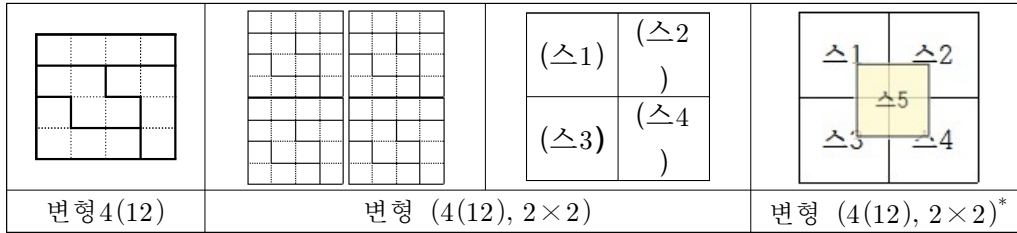
변형  $(4(k), m \times n)$  스토쿠를 구성하고 있는 변형  $4(k)$  스토쿠에서 한 개의 행을 제외한 변형  $4(k)^*$  스토쿠  $mn$ 개를 가로  $m$ 개, 세로  $n$ 개를 이어 붙여 만든 스토쿠

③ 이어붙이고 덮기 :  $(스k)$

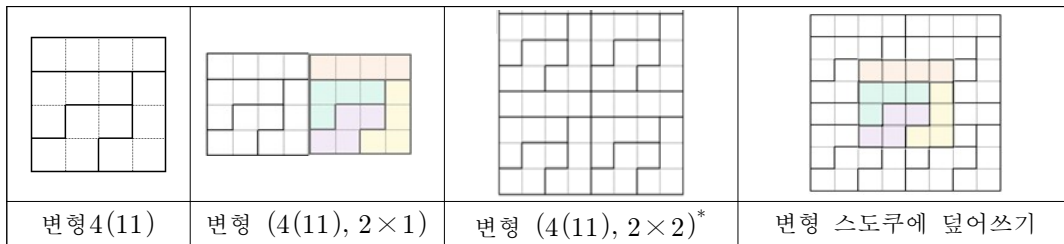
이어 붙여서 만든 변형  $(4(k), m \times n)$  스토쿠에서  $mn$ 개의 변형  $4(k)$  스토쿠를 맨 위줄 왼쪽에서부터 순서대로  $(스1), (스2), \dots, (스mn)$ 이라 하고, 변형  $(4(k), m \times n)$

스도쿠 위에 덮은 1개의 스도쿠를 스( $mn+1$ )이라 하자.

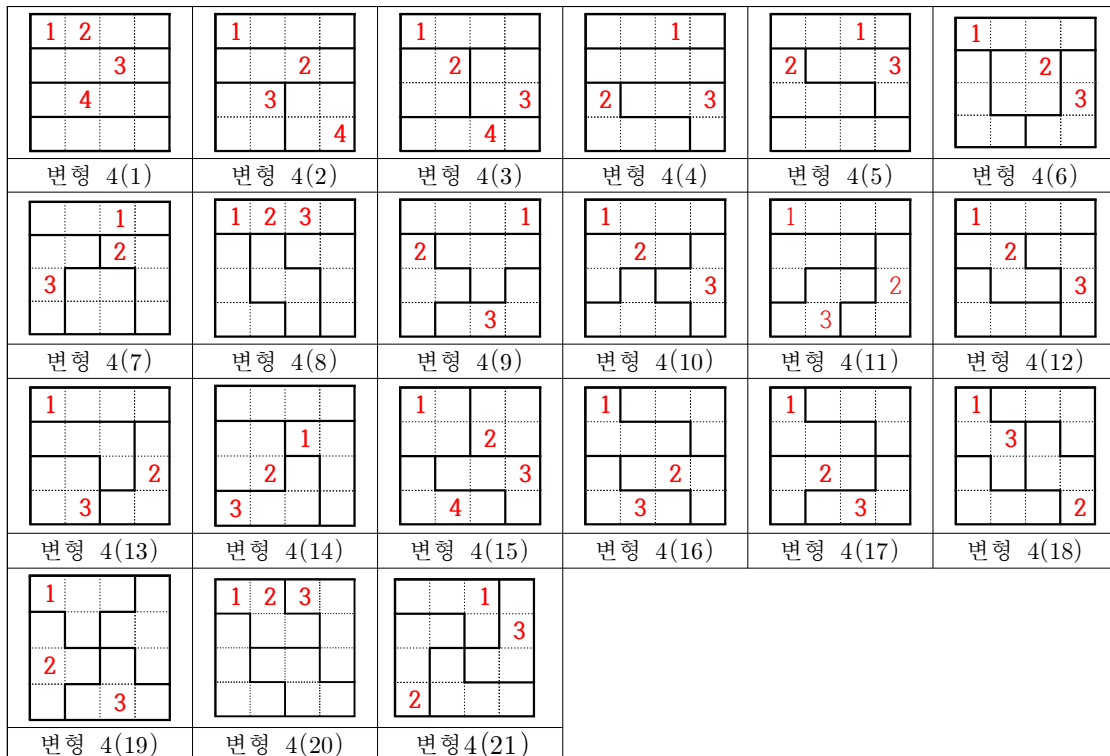
예를 들면, (1) 변형4(12) 스도쿠 4개를 가로와 세로에 각각 2개씩 이어 붙여서 만든 변형 ( $4(k), 2 \times 2$ ) 스도쿠 위에 올린 경우 다섯 번 스도쿠를 (스5)라 한다.



(2) 다음 그림은 변형 4(11)를 이어 붙여 만든 스도쿠에 이름을 붙인 것이다. 좌우 2개씩 이어 붙여서  $8 \times 8$ 을 만든 다음 유색 변형 4(11)를 덮은 그림이다.



아래 그림은 앞에서 언급한 21개의 변형 4-스도쿠에 대하여 이 스도쿠를 중복 없이 한 번씩만 사용하여 모든 칸을 채울 수 있게 제공되는 힌트가 들어가는 위치를 나타낸 것이다.

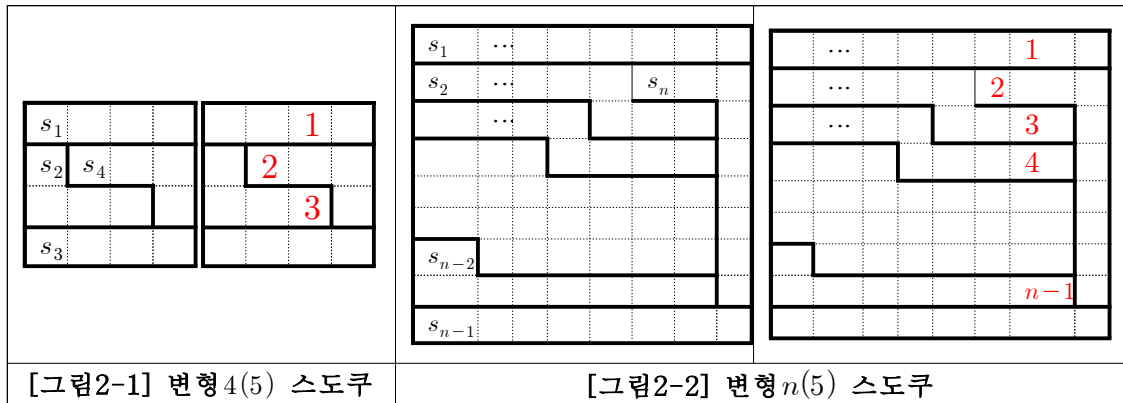


### III. 연구 결과

**정리 1** (1) 변형 4(5) 스도쿠에서 최소 힌트의 개수는 3개다.

(2) 변형  $n(5)$  스도쿠에서 최소 힌트의 개수는  $(n-1)$ 개다.

**증명** (1)  $n \times n$  스도쿠에서 최소 힌트의 개수는 적어도  $(n-1)$ 개이므로 변형 4(5) 스도쿠에서 힌트 3개만으로도 나머지 빈 칸들을 1부터 4까지 숫자를 한 번씩 중복 없이 채울 수 있음을 보이면 된다. 먼저 변형 4(5) 스도쿠에서 4칸으로 이루어진 모형을 순서대로  $s_1, s_2, s_3, s_4$ 라 한다.([그림2-1] 참고) 여기서, 3개의 힌트를 (1,3), (2,2), (3,3)의 위치에 1, 2, 3을 준다. 그러면 제 3열에서 2는 (2,3)에 들어갈 수 없으므로 (4,3)에 넣을 수 있으며 4는 (2,3)에 들어가야 한다.  $s_4$ 와 제 4행을 비교해 보면 숫자 2는 (1,4)에, 마지막으로 (3,1)에 들어가야 한다.  $s_2$ 에서 4는 (2,1)에 들어갈 수 없으므로 (3,2)에 들어가게 된다. 4열에서 4가 들어갈 수 있는 칸인 (4,4)에, 마지막으로 (1,1)에 넣는다. 1행에서 (1,2)에는 3을 넣을 수 있다.  $s_4$ 에서 3은 (3,4)에 들어갈 수 없으므로 (2,4)에, 마지막으로 (4,1)에 넣는다. 마지막으로 남은 빈 칸 (2,1), (3,4), (4,2)에는 숫자 1을 채워 넣으면 스도쿠를 완성할 수 있다.



(2) 변형  $n(5)$  스도쿠에서 [그림2-2]와 같이  $n$ 칸으로 이루어진 모형을 순서대로  $s_1, s_2, s_3, \dots, s_n$ 라 하자.  $n \times n$  스도쿠의 최소 힌트의 개수는  $(n-1)$ 개 이상이므로  $(n-1)$ 개의 힌트만으로 변형  $n(5)$  스도쿠를 중복없이 빈 칸을 채울 수 있음을 보이기로 하자.  $(n-1)$ 개의 힌트 1, 2, 3,  $\dots, n-1$ 을 각각 (1,  $n-1$ ), (2,  $n-2$ ), (3,  $n-1$ ), (4,  $n-1$ ),  $\dots$ , ( $n-1, n-1$ )에 주기로 한다. 그러면 (2,  $n-1$ )에 2가 들어갈 수 없으므로 ( $n, n-1$ )에 2를 넣고, 자동으로  $n$ 은 (2,  $n-1$ )에 들어가야 한다. 2는  $s_n$ 과  $n$ 행에 들어갈 수 없으므로 (1,  $n$ )에 2가 들어가야 한다. 2는  $s_2$ 에서 2행과 (3,  $n-2$ )에 들어갈 수 없으므로 (3,  $n-3$ )에, 2는  $s_3$ 에서 3행과 (4,  $n-3$ )에 들어갈 수 없으므로 (4,  $n-4$ )에 넣는다. 그리고 이와 같은 방법으로 (4,  $n-4$ ), (5,  $n-5$ ),  $\dots$ , ( $n-1, 1$ )에 2를 넣을 수 있다. 숫자  $n$ 은  $n$ 열에서  $s_n$ 에 들어갈 수 없으므로 ( $n, n$ )에 넣는다.  $s_2$ 에서

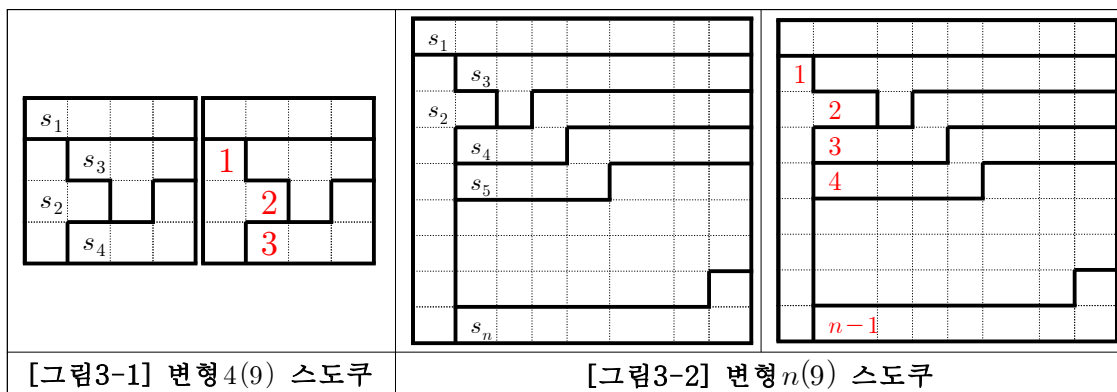
2행에 들어갈 수 없으므로  $(3, n-2)$ 에 들어가게 되고  $s_3$ 에서는 3행과  $(4, n-2)$ 에 들어갈 수 없으므로  $(4, n-3)$ 에 넣는다. 그리고 이와 같은 방법으로 나머지  $n$ 이 들어갈 자리는  $(5, n-4), (6, n-5), \dots, (n-1, 2)$ 에, 마지막으로  $(1, 1)$ 에 넣는다. 숫자 3은  $s_3$ 에서 3행에 들어갈 수 없으므로  $(4, n-2)$ 에,  $s_4$ 에서 4행과  $(5, n-2)$ 에 들어갈 수 없으므로  $(5, n-3)$ 에 들어간다. 이와 같은 방법으로 3을  $(6, n-4), \dots, (n-1, 3)$ 에 넣고  $s_n$ 에선  $(2, n)$ 에, 1행에선  $(1, 2)$ 에, 마지막으로  $(n, 1)$ 에 넣는다. 나머지 숫자들도 앞과 같은 방법으로 숫자  $k$  ( $k$ 는  $4, 5, \dots, (n-2), 1$ 인 자연수)는  $(k+1, n-2), (k+2, n-3), \dots, (n, k-1), (k-1, n), (1, k-2), (2, k-3), \dots, (k-2, 1)$ 에 채워 넣으면 이 스도쿠를  $n$ 개의 모든 숫자를 빠짐없이 중복되지 않게 모든 칸을 채울 수 있다.



**정리 2** (1) 변형 4(9) 스도쿠에서 최소 힌트의 개수는 3이다.

(2) 변형  $n(9)$  스도쿠에서 최소 힌트의 개수는  $(n-1)$ 이다.

**증명** (1) 변형 4(9) 스도쿠에서 4칸으로 이루어진 모형을 순서대로  $s_1, s_2, s_3, s_4$ 라 한다.([그림3-1]참고) 변형 4(9) 스도쿠에서 3개의 힌트가 주어졌을 때 모든 칸을 중복없이 모두 채울 수 있음을 보이면 된다. 3개의 힌트 1, 2, 3을 순서대로  $(2, 1), (3, 2), (4, 2)$ 에 넣는다. 그러면 2열에서 1은  $(2, 2)$ 에 들어갈 수 없으므로  $(1, 2)$ 에 넣고 4는  $(1, 2)$ 에 넣는다. 2행에는 1이 들어갈 수 없으므로  $s_3$ 의  $(3, 3)$ 에, 마지막으로  $(4, 4)$ 에 1이 들어간다.  $s_2$ 에 2가 있으므로 1열에서 2는 들어갈 수 있는 칸은  $(1, 1)$ 에,  $s_4$ 에서 들어갈 수 있는 칸은  $(4, 3)$ 뿐이다. 마지막으로  $(2, 4)$ 에 숫자 2를 넣는다. 제 4행과  $s_4$ 에서 남은 빈 칸에는 숫자 4가 들어가고 남은 4는  $(1, 3)$ 에 넣고 남은 빈 칸에는 숫자 3을 채우므로써 스도쿠를 완성할 수 있다.



(2) 변형  $n(9)$ 스도쿠에서  $n$ 칸으로 이루어진 모형을 순서대로  $s_1, s_2, s_3, \dots, s_n$ 라 한다.([그림3-2] 참고)  $n \times n$  스도쿠의 최소 힌트 개수는  $(n-1)$ 이상이므로  $(n-1)$ 개의 힌트를  $1, 2, \dots, (n-1)$ 를 순서대로  $(2, 1), (3, 2), (4, 2), \dots, (n, 2)$ 에 넣는다. 그러면

숫자 1은 2열에서 (2,2)에 들어갈 수 없으므로 (1,2)에 넣게 되고 2열의 빈 칸 (2,2)에 숫자  $n$ 을 넣는다.  $s_2$ 에 이미 2가 있으므로 1열에서 2는 (1,1)에 넣어야 한다. 그 다음으로 1은 2행에 있으므로  $s_3$ 에서 1이 들어갈 좌표는 (3,3)뿐이다. 그러면 3행에 1이 있으므로  $s_4$ 에서는 (4,4)에,  $s_5, s_6, \dots, s_n$ 에서는 각각 (5,5), (6,6), (7,7),  $\dots$ ,  $(n,n)$ 에 숫자 1을 넣을 수 있다. 숫자 2는 (1,1), (3,2)에 들어 있으므로  $s_4$ 와 제 3행이 겹친 부분에는 2를 넣을 수 없으므로 (4,3)에 넣을 수 있고,  $s_5$ 와 제 4행이 겹친 부분에도 2를 넣을 수 없으므로 (5,4)에 넣을 수 있다. 이런 방법으로 하면 나머지 2는  $s_6, \dots, s_n$ 에서 (6,5), (7,6),  $\dots$ ,  $(n,n-1)$ 에 넣을 수 있고, 마지막으로 2는 (2, $n$ )에 넣을 수 있다.

숫자 3은 4행에 있으므로  $s_5$ 에서 3이 들어 갈 수 있는 칸은 (5,3)이고, 5행과  $s_6$ 을 비교하면 (6,4)에 넣을 수 있다. 이와 같은 규칙을 적용하면 남은 3은 순서대로  $s_7, \dots, s_n$ 에서 들어 갈 수 있는 칸은 (7,5),  $\dots$ ,  $(n,n-2)$ 에 넣는다. 제 1열에서 3이 들어있는 열들과 비교해 보면 남은 칸 (3,1)에 넣을 수 있고,  $s_3$ 에서는 (2, $n-1$ )에,  $n$ 열에서 (1, $n$ )에 넣을 수 있다. 이와 같은 방법으로 남은 숫자 4부터  $n-1$  사이의 숫자  $k$ 가 들어갈 수 있는 위치는  $(k+2,k), (k+3,k+1), \dots, (n,n-k+1)$ 에 넣은 후  $k$ 가 들어간 열들과 2행을 비교하면 (2, $n-k+2$ ), (1, $n-k+3$ ), (3, $n-k+4$ ),  $\dots$ ,  $(k-1,n)$ 에 넣고 마지막에는 제 1열에서 ( $k,1$ )에 숫자  $k$ 를 넣을 수 있다. 그러면 변형  $n(9)$  스도쿠의 모든 칸을 빈틈없이 중복없이 채울 수 있다. ■

**따름정리 3** (1) 변형  $n(4)$  스도쿠에서 최소 힌트의 개수는  $(n-1)$ 이다.

(2) 변형  $n(10)$  스도쿠에서 최소 힌트의 개수는  $(n-1)$ 이다.

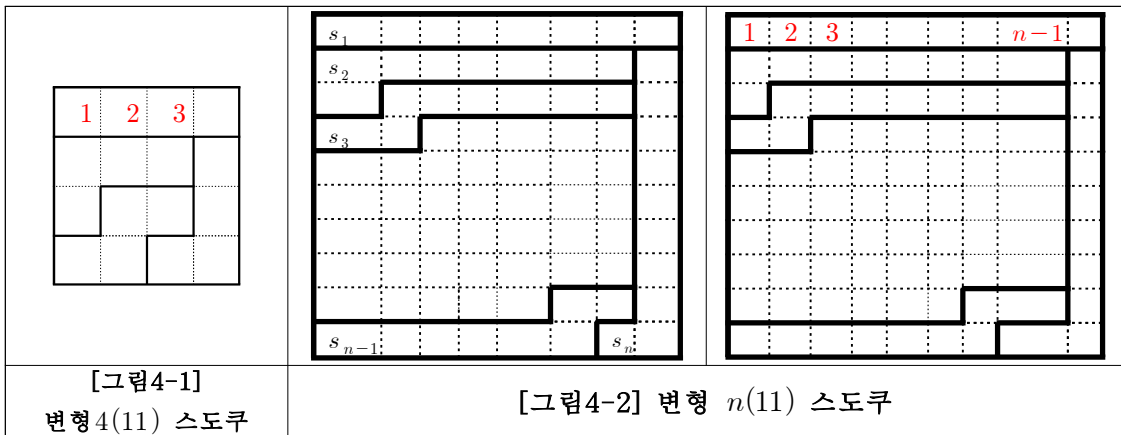
**증명** (1) 변형  $n(4)$  스도쿠는 변형  $n(5)$  스도쿠에서 제  $n$ 행을 맨 위로 올린 모양이므로  $n-1$ 개의 힌트 1, 2,  $\dots$ ,  $(n-1)$ 을 각각 변형  $n(4)$  스도쿠에서 (2, $n-1$ ), (3, $n-2$ ), (4, $n-1$ ),  $\dots$ ,  $(n,n-1)$ 에 넣으면 변형  $n(4)$  스도쿠의 모든 칸을 한 줄을 채울 수 있다.

(2) 변형  $n(10)$  스도쿠는 변형  $n(9)$  스도쿠에서  $s_1$ 을 제외한 나머지  $s_2, s_3, \dots, s_n$ 을 180° 돌린 모양이다. 따라서  $n$ 칸으로 이루어진 모형을 변형  $n(9)$  스도쿠와 동일하게 정한다. 그러면 변형  $n(9)$  스도쿠에서 준 힌트를 180° 돌린 위치에 넣으면 남은 빈 칸을 변형  $n(9)$  스도쿠와 비스한 방법으로 채울 수 있다. ■

정리 4 (1) 변형 4(11) 스토쿠에서 최소 힌트의 개수는 3이다.

(2) 변형  $n(11)$  스토쿠에서 최소 힌트의 개수는  $(n-1)$ 이다. ( $n \geq 5$ )

증명 (1) 3개의 힌트로 변형 4(11)스도쿠를 빈틈없이 모든 칸을 채울 수 있음을 보이면 된다. 세 개의 힌트를 스토쿠의 (1,1), (1,2), (1,3)에 1,2,3을 주기로 하자. 그러면 (1,4)에는 4가 들어가게 되고 제 4열에는 4가 들어가지 못하므로 (4,3)에는 4를, (3,2)와 (2,1)에도 4가 들어가게 된다. (4,4), (3,3), (2,2)에는 같은 수가 들어가야 하고 열을 비교하면 2,3,4는 넣을 수 없으므로 반드시 그 칸에는 1이 들어가야 한다. 그러면 제 3열의 (2,3)에는 2가 들어가고 (3,4)와 (4,1)에 2가 들어가야 한다. 따라서 남은 세 빈 칸 (2,4), (3,1), (4,2)에 3을 채움으로써 스토쿠를 완성할 수 있다.



(2) 변형  $n(11)$  스토쿠에서  $(n-1)$ 개의 힌트가 주어졌을 때 남은 빈 칸을 중복없이 빈틈없이 채울 수 있음을 보이기로 하자. 변형  $n(11)$  스토쿠에서  $n$ 칸으로 이루어진  $n$ 개의 모형을 순서대로  $s_1, s_2, \dots, s_n$ 이라 하자. ([그림4-2]참고) 변형  $n(11)$  스토쿠에서  $(n-1)$ 개의 힌트를 (1,1), (1,2),  $\dots$ , (1, $n-2$ ), (1, $n-1$ )에 각각 1, 2,  $\dots$ ,  $n-1$ 를 넣자. 그러면 이 스토쿠의 제 1행의 빈 칸 (1, $n$ )에 숫자  $n$ 을 넣고 나면  $s_n$ 에서  $n$ 이 들어갈 수 있는 칸은  $(n-1,n)$ 이다. 제  $n$ 행에  $n$ 이 있으므로  $n$ 은  $s_{n-1}$ 에서  $(n-1,n-2)$ 에,  $s_{n-2}$ 에서  $(n-2,n-3)$ 에, 남은  $n$ 은  $(n-3,n-4), \dots, (3,2), (2,1)$ 에 넣어야 한다. 그런 다음  $s_n, s_{n-1}, \dots, s_3, s_2$ 의  $(n,n), (n-1,n-1), \dots, (3,3), (2,2)$ 에는 같은 숫자가 들어가야 하는데 2열부터  $n$ 열까지 비교하면 숫자 2, 3,  $\dots, n$ 은 들어갈 수 없으므로  $(n,n), (n-1,n-1), \dots, (3,3), (2,2)$ 에는 숫자 1이 들어가야 한다.

$s_n, s_{n-1}, \dots, s_3, s_2$ 의 모양을 비교하면  $(n-1,n), (n-2,n-1), \dots, (3,4), (2,3)$ 에는 같은 숫자가 들어가야 하는데 3열부터  $n$ 열까지 비교하면 숫자 3, 4,  $\dots, n$ 은 들어갈 수 없고 (2,2)에는 숫자 1이 있으므로  $(n-1,n), (n-2,n-1), \dots, (3,4), (2,3)$ 에는 숫자 2가 들어가야 한다. 마지막 제  $n$ 행과 열을 비교해 보면  $(n,1)$ 에는 숫자 2가 들어가야 한다. 이와 같은 방법으로 3부터  $n-1$ 까지의 숫자는 순서대로 다음과 같은 위치

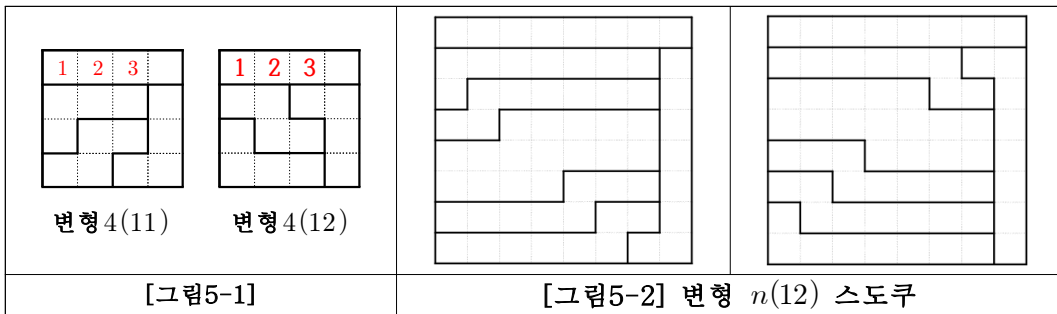
에 넣으면 된다. 숫자  $k$ 는

$(n-k+1, n), (n-k, n-1), \dots, (3, k+2), (2, k+1)$ 와  $(n, k-1), (n-1, k-2), \dots, (n-k+2, 1)$ 에 들어가야 한다. 그러면 모든 칸을 중복없이 빈틈없이 1부터  $n$ 까지 숫자를 채워서 이 스도쿠를 완성할 수 있다. ■

**따름정리 5** (1) 변형 4(12) 스도쿠에서 최소 힌트의 개수는 3이다.

(2) 변형  $n(12)$  스도쿠에서 최소 힌트의 개수는  $(n-1)$ 이다. ( $n \geq 5$ )

증명



(1) 변형 4(12) 스도쿠는 변형 4(11) 스도쿠의 2, 3, 4행으로 만들어진 직사각형 모양을 아래로  $180^\circ$  돌린 것과 같으므로 3개의 힌트 1, 2, 3을  $(1,1), (1,2), (1,3)$ 에 넣으면 숫자 4는  $(1,4), (2,3), (3,2), (4,1)$ 에 넣고 나면  $(2,4), (3,3), (2,3)$ 에 같은 숫자가 들어가야 하고 열을 비교하면 그 칸에는 2, 3, 4를 넣을 수 없으므로 숫자 1을 넣어야 한다. 그런 다음 숫자 2는  $(2,1), (3,4), (4,3)$ 에, 숫자 3은  $(2,2), (3,1), (4,4)$ 에 넣으므로써 스도쿠를 완성할 수 있다.

(2) 변형  $n(12)$  스도쿠는 변형  $n(11)$  스도쿠의 2, 3,  $\dots, n$ 행으로 만들어진 직사각형 모양을 아래로  $180^\circ$  돌린 것과 같으므로  $n-1$ 개의 힌트 1, 2,  $\dots, n-1$ 을 순서대로  $(1,1), (1,2), \dots, (1, n-1)$ 에 주면 숫자  $n$ 은  $(1, n), (2, n-1), \dots, (n-1, 2), (n, 1)$ 에 넣을 수 있다.  $(2, n), (3, n-1), \dots, (n-1, 3), (n, 2)$ 에는 같은 숫자가 들어가야 하고 열을 비교하면 2,  $\dots, n-1, n$ 을 넣을 수 없으므로 그 자리에는 숫자 1을 넣어야 한다.

다음에는 2부터  $n-1$ 까지의 숫자는 순서대로 채워져야 하고 이 사이의 숫자  $k$ 가 들어가는 자리는 다음과 같다.

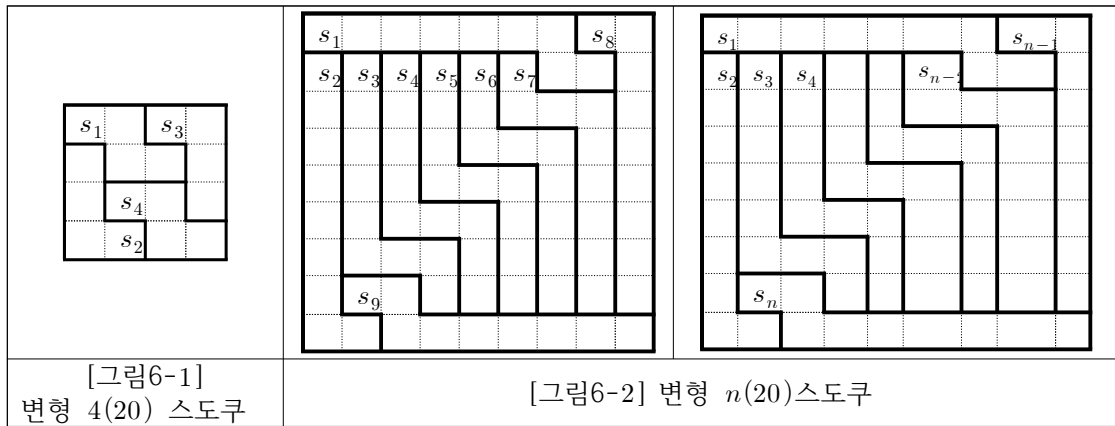
숫자 $k$	숫자 $k$ 를 넣어야 하는 좌표
2	$(3, n), (4, n-1), \dots, (n-1, 4), (n, 3) \rightarrow (2, 1)$
3	$(4, n), (5, n-1), \dots, (n-1, 5), (n, 4) \rightarrow (3, 1), (2, 2)$
$k$	$(k+1, n), (k+2, n-1), \dots, (n-1, k+2), (n, k+1)$
$(4 \leq k \leq n-1)$	$\rightarrow (k, 1), (k-1, 2), \dots, (2, k-1)$

이와 같은 순서대로 빈 칸을 채워 가면 이 스도쿠를 완성할 수 있다. ■

**정리 6** (1) 변형 4(20) 스토쿠의 최소힌트 개수는 3개이다.

(2) 변형  $n(20)$  스토쿠에 필요한 힌트의 개수는  $n-1$ 이다.

**증명** (1) 변형 4(20) 스토쿠에 세 개의 힌트 1, 2, 3을 (1,1), (1,2), (1,3)에 차례대로 주면 1열에 남은 자리는 4가 들어가야 한다. 그 다음  $s_1$ 에서 숫자 3이 들어갈 수 있는 칸은 (2,2)뿐이고,  $s_4$ 에는 숫자 3이 있으므로 제 4열의 남은 자리 (4,4)에, 마지막으로 (3,1)에 3을 넣어야 한다. 숫자 4는  $s_1$ 의 빈 칸 (2,3)에,  $s_4$ 에서는 (3,2)에 넣고 나머지는 (4,1)에 숫자 4를 넣으면 된다. 숫자 1은 제 1열에 있으므로  $s_2$ 의 (4,2)에 넣고 차례대로 (3,3), (2,4)에 넣으면 된다. 그리고 남은 빈 칸 (2,1), (3,4), (4,3)에는 숫자 2를 채움으로써 3개의 힌트로 이 스토쿠를 완성할 수 있다.



(2) 변형  $n(20)$  스토쿠에 최소 힌트 개수는  $n-1$ 개임을 증명하기 위해 (1,1), (1,2), ..., (1,  $n-1$ )에 차례대로  $n-1$ 개의 힌트 1, 2, ...,  $n-1$ 을 주자. 그러면 (1,  $n$ )에는 숫자  $n$ 이 들어가야 한다.

숫자  $n-1$ 은  $s_1$ 과 제  $(n-1)$ 열을 비교하면 (2,  $n-2$ )에,  $s_{n-1}$ 에서 제  $(n-2)$ 열과 제  $(n-1)$ 열이 겹치는 부분을 제외한 (3,  $n-3$ )에, 이와 비슷한 이유에 의해서 (4,  $n-4$ ), (5,  $n-5$ ), ..., (  $n-1$ , 1)에 넣고 마지막으로 제  $n$ 열에서는 (  $n$ ,  $n$ )에 채워 넣어야 한다.

숫자  $n$ 은  $s_1$ 의 (2,  $n-1$ )에,  $s_{n-1}$ 에서 제  $(n-1)$ 열을 제외한 (3,  $n-2$ )에, 이와 비슷한 이유에 의해서 (4,  $n-3$ ), (5,  $n-4$ ), ..., (  $n-1$ , 2), (  $n$ , 1)에 넣어야 한다. 숫자 1은 제 1열에 있으므로  $s_2$ 의 빈 칸 (  $n$ , 2)에 넣고,  $s_n$ 과 제  $n$ 행을 비교하면 숫자 1을 넣으면 된다.  $s_3$ 에는 (  $n-2$ , 4)에,  $s_4$ 에는 (  $n-3$ , 5)에, ...,  $s_{n-2}$ 에는 (3,  $n-1$ )에, 마지막으로  $s_{n-1}$ 의 (2,  $n$ )에 남은 숫자 1을 넣으면 된다.  $s_3$ 에서 숫자 2가 들어갈 칸은 (  $n-1$ , 4)이다.

이 스토쿠의 위치 (  $n$ ,  $n-1$ )에 들어갈 숫자를 ★이라 하면 ★은 제  $n-1$ 열을 포함하는  $s_{n-2}$ 에서는 (2,  $n-3$ )에 넣어야 하고  $s_{n-2}$ 에 포함된 제  $n-3$ 열에는 ★이 들어갈 수 없으므로 제  $n-3$ 열을 포함한  $s_{n-4}$ 중에서 (3,  $n-5$ ), (4,  $n-5$ )중 하나에 ★이 들

어가야 한다.  $(3, n-5)$ 에 ★이 들어간다고 하면 남은  $n-5$ 열에 들어갈 수 없으니  $s_{n-6}$ 의 상단에 이런 방법으로 교차하면서 ★이 들어가야 하는 자리는  $s_{n-2}, s_{n-4}, \dots, s_4$ 의 하단,  $s_{n-3}, s_{n-5}, \dots, s_5$ 의 상단에 들어가야 한다. 그러므로 ★은  $n-1, n, 1, n-2, n-4, \dots, 5, n-3, n-5, \dots, 4$ 가 될 수 없고 제 3열에는 ★이 들어갈 자리가 없게 되므로 모순이다. 따라서 ★은  $(4, n-5)$ 에 들어가야 한다. 나머지 ★이 들어갈 자리는  $(3, n-4), (4, n-5), \dots, (n-2, 1)$ 과 마지막으로  $(n-1, n)$ 이 되고 이 자리의 열을 비교하면 ★은  $n-1, 1, 2, \dots, n-3, n$ 이 될 수 없으므로 ★은  $n-2$ 이다.

$(n, n-2)$ 에 들어갈 숫자를 ♥이라 하면 유사한 방법으로 ♥은  $(2, n-4), (3, n-5), \dots, (n-3, 1)$ 과 마지막으로  $(n-1, n-1), (n-2, n)$ 이 되고 이 자리의 열을 비교하면 ♥은  $n-2, n-4, \dots, 2, 1, n-1, n$ 이 될 수 없으므로 ♥은  $n-3$ 이다.

이 과정을 정리해 보면  $(n, n-k)$ 에 들어갈 숫자와 같은 숫자를 채울 수 있는 위치는  $(2, n-k-2), (3, n-k-3), \dots, (n-k-1, 1)$ 과 마지막으로  $(n-1, n-k+1), (n-2, n-k+2), \dots, (n-k, n)$ 이 되고 이 자리의 열을 비교해 보면  $n-k, n-k-2, n-k-3, \dots, 2, 1, n-k+1, n-k+2, \dots, n$ 이 될 수 없으므로 그 위치에는  $n-k-1$ 이다. 여기서  $k=n-2, n-3, \dots, 2$ 이다.

이렇게 행과 열에 겹치는 숫자 없이 완성되었기 때문에 스도쿠가 풀렸다. 따라서 변형  $n(20)$  스도쿠에 최소 힌트 개수는  $n-1$ 개임을 알 수 있다.

[표] 변형  $n(20)$  스도쿠에 같은 숫자가 들어가는 위치

	같은 숫자를 넣을 위치	채울 수 없는 숫자	채울 수 있는 숫자
$(1, n-1)$	$(2, n-2), (3, n-3), (4, n-4), \dots, (n-1, 1), (n, n)$		$n-1$
$(1, n)$	$(2, n-1), (3, n-2), \dots, (n-1, 2), (n, 1)$		$n$
$(1, 1)$	$(n, 2), (n-1, 3), (n-2, 4), \dots, (3, n-1), (2, n)$		1
$(n, n-k)$	$(2, n-k-2), (3, n-k-3), \dots, (n-k-1, 1), (n-1, n-k+1), (n-2, n-k+2), \dots, (n-k, n)$ (여기서, $k=n-2, n-3, \dots, 2$ )	$n-k, n-k-2, n-k-3, \dots, 2, 1, n-k+1, n-k+2, \dots, n$	$n-k-1$



1부터  $n$ 까지의 숫자를 채워가는 변형  $n(k)$  스도쿠의 최소힌트의 개수가  $(n-1)$ 이상 이라는 사실로부터 다양한 스도쿠를 만들어 보았다. 다음 정리를 이렇게 만든 스도쿠들을 이용해서 이어 붙여서 만든 스도쿠에 대하여 최소 힌트의 개수가 어떻게 변하는지를 연구하였다. 다음 정리 7에서 언급한 스도쿠는 1부터 4까지의 숫자를 채워가는 변형  $4(12), 2 \times 2$ 를 이어 붙여 만든 스도쿠에 대하여 최소 힌트의 개수가 어떻게 변하는지를 연구한 결과이다.

**정리 7** 변형  $(4(12), 2 \times 2)^*$ 와 그 스도쿠에서 5번째 스도쿠(스5)를 겹쳤을 때 다음의 결과를 얻을 수 있다.

- (1) 변형  $(4(12), 2 \times 2)^*$ 의 최소 힌트의 개수는 6이다.
- (2) 그 겹쳐지는 위치가 (스1)과 (스3) 또는 (스2)와 (스4)에서만 겹쳐지면 최소 힌트의 개수는 6개이다.
- (3) 겹쳐지는 위치가 변형  $(4(12), 2 \times 2)^*$ 의 정중앙에 있으면 최소힌트의 개수는 3개다.

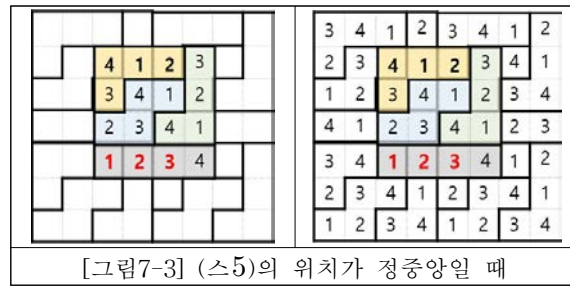
**증명** (1) 변형  $(4(12), 2 \times 2)^*$ 스도쿠의 좌측과 우측 모두 4행을 공유하는 변형  $4(12)$ 스도쿠의 모양으로 만들어진 스도쿠이므로 따름정리 5(1)에 의해서 변형  $4(12)$ 스도쿠의 최소힌트의 개수는 3이므로 그 힌트의 위치를 이용해서 (4,1), (4,2), (4,3)과 (4,5), (4,6), (4,7)에 힌트 1, 2, 3을 채워주면 [그림7-1]과 같이 모든 칸을 중복없이 채울 수 있음을 알 수 있다.

1	2	3	4				
2	3	4	1				
3	4	1	2				
4	1	2	3				
4	1	2	3	4	1	2	3
3	4	1	2	3	4	1	2
2	3	4	1	2	3	4	1
1	2	3	4	1	2	3	4
4	1	2	3	4	1	2	3
3	4	1	2	3	4	1	2
2	3	4	1	2	3	4	1

[그림7-1] 변형 $(4(12), 2 \times 2)^*$

[그림7-2] (스5)의 위치가 (스1)과 (스3)와 겹쳐질 때							
4	1	2	3				
3	4	1	2				
2	3	4	1	1	2	3	
1	2	3	4				
1	2	3	4	4	1	2	3
4	1	2	3	3	4	1	2
3	4	1	2	2	3	4	1
2	3	4	1	1	2	3	4
1	2	3	4	4	1	2	3
4	1	2	3	3	4	1	2
3	4	1	2	2	3	4	1

(2) [그림7-2]와 같이 (스5)가 변형  $(4(12), 2 \times 2)^*$ 스도쿠의 (스1)과 (스3)의 부분과 겹친 스도쿠라고 하자. 그러면 (스5)는 세 개의 힌트로 완전히 채울 수 있게 되고 그 완성된 그림은 (스1)과 (스3)과 각각 9칸, 4칸이 겹쳤으므로 따름 정리5(1)의 최소힌트의 개수보다 많으므로 (스1)과 (스3)의 남은 칸을 채울 수 있다. 또한 변형  $(4(12), 2 \times 2)^*$ 스도쿠의 (스2)과 (스4) 부분도 (4,5), (4,6), (4,7)에 힌트 1, 2, 3을 채워주면 [그림7-2]과 같이 모든 칸을 중복없이 채울 수 있음을 알 수 있다. 따라서 (스5)가 겹쳐진 변형  $(4(12), 2 \times 2)^*$ 스도쿠의 최소 힌트의 개수는 6이다.



(3) [그림7-3]와 같이 (스5)가 변형  $(4(12), 2 \times 2)^*$  스토쿠의 정중앙에 겹쳐졌다고 가정하자. 그러면 변형  $(4(12), 2 \times 2)^*$  스토쿠의 위치인 (6,3), (6,4), (6,5)에 세 개의 힌트 1, 2, 3를 주면 (스5)를 완전히 채울 수 있게 되고 이 완성된 (스5)는 (스1), (스2), (스3), (스4)과 각각 6개, 6개, 4개, 4개씩 겹쳤으므로 (스1), (스2), (스3), (스4)의 남은 칸을 빈틈없이 채울 수 있게 된다. ([그림7-3]참고) 따라서 (스5)가 변형  $4(12)$  스토쿠  $(2,2)^*$ 의 정중앙에 있으면 3개의 힌트만으로도 이 스토쿠를 완성할 수 있다. ■

앞에서 언급한  $T_1$ 을 포함하는 변형  $4(k)$ 스토쿠를 이어 붙여 만든 변형  $(4(k), 2 \times 2)^*$ 에 대해서도 정리 7과 같은 결과를 예측할 수 있다.

#### IV. 참고문헌

[2018년 경상국립대학교 과학영재교육원 수학사사] 스토쿠의 최소 힌트의 개수

수학B

09\_진법탐구



# 진법 탐구

이시형 (남춘천중학교 2학년)

최은찬 (강원중학교 2학년)

지도교수 : 서승현 (강원대학교 수학교육과)

## 초록

이진수 뿔셈에서의  $\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}$  패턴을 해결하기 위해 특정 규칙과 과정을 도출했다. 먼저, 일정한 규칙을 통해 특정 패턴을 계산하는 방법을 발견했다. 1과 0이 번갈아 가며 나타나는 패턴에서 1은 0으로, 0은 1로 바뀌는 규칙을 이용하여 계산할 수 있었다.

또한, 0과 1로 이루어진 이진수의 특정 패턴이 반복되는 경우, 이를 재구성하여 계산을 단순화할 수 있는 방법을 제시했다.

또한 파스칼의 삼각형과 시어핀스키 삼각형 사이에 흥미로운 관련성을 발견하였다. 파스칼의 삼각형의 항 중에서 홀수를 나타내는 위치와 시어핀스키 삼각형의 형태 사이에 공통점이 존재하는데, 이것은 일정한 조건을 충족하는 경우 두 삼각형이 유사한 패턴을 공유한다는 것을 보여준다. 특히, 파스칼의 삼각형에서 홀수가 나오는 위치를 결정하는데 있어서 다음과 같은 조건들을 도출하였다.

## I. 서론 (또는 연구의 필요성 및 목적)

시계를 보다가 우리가 일상생활에서 가장 많이 쓰는 10을 기준으로 하지 않고 왜 시계의 시침은 12를 기준으로 하였을까 궁금하여 조사해보니 10보다 12는 약수가 많아 나눌 수 있는 수가 많기 때문에 효율이 높다는 이유에서였다. 이렇게 12를 기준으로 표현하는 방식을 12진법이라고 한다는 것을 깨달았고, 이런과정에서 여러 가지 진법을 찾아보았다. 가장 대표적인 2진법을 먼저 찾아보았는데, 컴퓨터가 2진법을 사용한다는 것을 알았다. 왜 컴퓨터는 2진법을 사용하는 이유가 궁금하여 검색해보니 덧셈, 뺄셈의 경우의 수가 10진법보다 적어서 오류가 적다는 이유에서였다. 2진법의 덧셈과 뺄셈을 어떤 방식으로 계산하고, 보다 수월하게 계산하는 방법은 무엇이 있는지 궁금하여 첫 번째 연구를 진행하였다.

또 3진법의 예시를 찾아보다가 칸토르 집합(칸토어 집합)이라는 집합이 있다는 것을 알게되었다. 이 집합에 대해 자세히 조사해보니 이 집합은 프랙탈이라는 것을 알게되었다. 또 다른 프랙탈의 종류는 무엇이 있을까 궁금하여 자료조사해보니 시어핀스키 삼각형도 프랙탈이고, 파스칼 삼각형의 홀수는 칠하고 짝수는 칠하지 않으면 시어핀스키 삼각형 모양이 나와 프랙탈이라는 것을 알게 되었다. 파스칼삼각형의 홀수/짝수로 만든 시어핀스키 삼각형과 이진법이 어떻게 연관이 되어있는지 궁금하여 파스칼, 시어핀스키 삼각형과 이진법의 관계에 대해 자세히 알아보고 싶어 두 번째 연구를 시작하게 되었다.



### III. 연구 방법

#### 1. 이진수의 덧셈과 뺄셈

이진수의 덧셈은 나올수 있는 경우의 수가 0+0, 0+1, 1+0, 1+1 4가지 밖에 없다. 0+0=0, 0+1=1, 1+0=1, 1+1은 받아올림을 해 앞자리에 1을 더해주는 받아올림을 하면 되므로 덧셈은 딱히 문제가 되지 않는다.

뺄셈에서도 0-0, 1-0, 1-1, 0-1 4가지 경우의 수가 존재한다. 0-0, 1-0, 1-1 패턴은 계산하기 번거로운 패턴이 없으므로 수월히 계산이 가능하지만 0-1패턴의 계산은 까다롭다. 0-1 패턴을 어떻게 해결할까 고민해 보았다. 계산하기 수월한 0-0, 1-0, 1-1 패턴으로만 이루어진 구간을 A, 1-0패턴으로 시작하여 0-1패턴으로 끝나는 구간을 B라고 해보자. B패턴을 계산하는 법을 알기만 한다면 0-1패턴도 수월하게 계산할 수 있다. 단, B패턴에서  $\frac{1}{0}$ 패턴은 중간에  $\frac{1}{0}$ 패턴이 나오면 패턴이 끊기기 때문에 한번만 나와야한다. 이제 B패턴을 계산하는 방법을 살펴보자.

먼저 맨윗줄이 10000000...인 경우이다.

1000000에서는 빼는수 0101101<sub>(2)</sub>에다가 1010010을 더하면 111111이 나온다.  

$$\begin{array}{r} 1000000 \\ - 0101101 \end{array}$$

$2^m - 1 = 1 + 2 + 3 + 4 + \dots + 2^{m-1}$ 이라는 공식을 이용하면

$$111111_{(2)} = 1 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 2^6 - 1$$

가 된다. 더해서  $2^6$ 가 되는 수를 찾는 것이므로 1010010에다가 1을 더하면 나오는 1010011이 1000000의 계산의 결과값이 된다.

$$\begin{array}{r} 1010011 \\ - 0101101 \end{array}$$

여기서 나오는 결과값 1010011은 빼는 수 0101101에서 0이면 1, 1이면 0이고 위 • 아래 모두 1이라는 규칙으로 계산하면 쉽게 계산할 수 있다.

두 번째로 맨 윗줄이 10000...이 계속 반복되는 경우나 아닌 경우이다. (중간에  $\frac{1}{0}$ 패턴이 나오면 끊기므로  $\frac{1}{0}$ 패턴은 나오지 않고 오직  $\frac{1}{1}, \frac{0}{1}, \frac{0}{0}$ 패턴만 나오는 경우이다.)

이 경우에는 1-1의 결과값은 0이고 0-0의 결과값도 0이므로  $-\frac{1}{1} \Leftrightarrow \frac{0}{-0}$ 로 바꿀 수 있다. 즉 맨 윗줄을 1000...의 형태로 바꿀 수 있다.

$\begin{array}{r} 1011011011100000111010 \\ - 0010001101000110010001 \end{array}$ 의 경우처럼 B블록이 여러번 존재하는 상황에서도 이런 방

식을 이용한다면 쉽게 계산할 수 있다.  $\begin{array}{r} 1011011011100000111010 \\ - 0010001101000110010001 \end{array}$ 에서

$\frac{1}{0}$ 으로 시작하여  $\frac{0}{1}$ 패턴으로 끝나는 부분을 B, 나머지부분을 A라고 하여 계산하면 쉽게 계산이 가능하다.

## 2. 이항계수의 합동

파스칼의 삼각형은 이항계수를 삼각형의 모양으로 나타낸 것이다. 파스칼의 삼각형의 각 항은 모두  ${}_n C_k$ 의 형태로 나타낼 수 있다. 이때 파스칼의 삼각형에 홀수만 칠하게 된다면 그 도형은 시에르핀스키 삼각형과 매우 유사한 도형이 된다. 이때 홀수가 나오는 자리의 특징은  ${}_n C_k$ 가 홀수이다. 그렇다면 그 조건은 무엇일까?

<정리 1>  $p$ 가 소수이고,  $1 \leq k \leq p-1$ 이면  $\binom{p}{k}$ 는  $p$ 의 배수이다.

[증명]  $\binom{p}{k} = \frac{p!}{(p-k)!k!}$ 에서  $1 \leq k \leq p-1$ 이므로  
분자의  $p$ 는 소거되지 않음

$a \equiv b \pmod{p} \Leftrightarrow a-b$ 가  $p$ 의 배수

다항식  $f(x), g(x)$ 에 대하여  $f(x) \equiv g(x) \pmod{p}$

$\Leftrightarrow$  모든 차수의 계수가  $p$ 에 대하여 합동

<정리 2>  $p$ 가 소수이면  $(x+1)^p \equiv x^p + 1 \pmod{p}$

[증명]  $(x+1)^p = \binom{p}{0}x^p + \binom{p}{1}x^{p-1} + \dots + \binom{p}{p}x^0$   
 $= x^p + \binom{p}{1}x^{p-1} + \dots + x + 1$   
 $= x^p + 1 \pmod{p}$

<정리 3>  $p$ 가 소수이고  $k$ 가 자연수이면  $(x+1)^{p^k} \equiv x^{p^k} + 1 \pmod{p}$

[증명]  $(x+1)^{p^k} = ((x+1)^p)^{p^{k-1}}$   
 $\equiv (x^p + 1)^{p^{k-1}} \pmod{p}$   
 $= ((x^p + 1)^p)^{p^{k-2}}$   
 $\equiv (x^{p^2} + 1)^{p^{k-2}} \pmod{p}$   
 $\vdots$   
 $\equiv (x^{p^k} + 1)^{p^0} \equiv x^{p^k} + 1 \pmod{p}$

<정리 4>  $p$ 가 소수이고,  $n \geq m$  은 음이 아닌 정수 이 때, 다음이 성립한다.

$$\binom{n}{m} \equiv \binom{a_0}{b_0} \binom{a_1}{b_1} \dots \binom{a_k}{b_k} \pmod{p}$$

단,  $n = \sum_{i=0}^k a_i p^i = (a_k a_{k-1} \dots a_0)_p$

$$m = \sum_{i=0}^k b_i p^i = (b_k b_{k-1} \dots b_0)_p$$

$$\begin{aligned}
[\text{증명}] \quad & \sum_{m=0}^n \binom{n}{m} x^m \\
&= (1+x)^n \quad (\because \text{이항정리}) \\
&= (1+x)^{a_0+a_1p+a_2p^2+\dots+a_kp^k} \\
&= (1+x)^{a_0}(1+x)^{a_1p}(1+x)^{a_2p^2} \dots (1+x)^{a_kp^k} \\
&\equiv (1+x)^{a_0}(1+x^p)^{a_1}(1+x^{p^2})^{a_2} \dots (1+x^{p^k})^{a_k} \pmod{p} \\
&= \left( \sum_{c_0=0}^{a_0} \binom{a_0}{c_0} x^{c_0} \right) \times \left( \sum_{c_1=0}^{a_1} \binom{a_1}{c_1} x^{c_1} \right) \times \left( \sum_{c_2=0}^{a_2} \binom{a_2}{c_2} x^{c_2} \right) \times \dots \times \left( \sum_{c_k=0}^{a_k} \binom{a_k}{c_k} x^{c_k} \right)
\end{aligned}$$

$m$ 의  $p$ 진수 전개가  $(b_k b_{k-1} \dots b_0)_p$ 라 할 때,

$x^m$ 의 계수는  $\binom{a_0}{b_0} \binom{a_1}{b_1} \dots \binom{a_k}{b_k}$ 이므로 이는 범  $p$ 에 대하여  $\binom{n}{m}$ 과 같다.

#### IV. 연구 결과

##### 1. 받아내림과 덧셈을 하지 않는 이진수의 뺄셈

두 이진수  $M, N$ 에 대하여  $M-N$ 을 할 때,  $M > N$ 이면 반드시  ${}_1^0$  패턴을 해결하는  ${}_0^1$  패턴이 존재한다. 십진수의 뺄셈에 이 방식을 이용해보겠다.  $a < b$ 인  ${}_b^a$  패턴을 가장 가까운  $c > d$ 인  ${}_d^c$  패턴이 해결한다.  $426087435106$ 의 앞에서 2번째 블록 2-5의 계산값과 0-3의 계산값이 같으므로  ${}_3^0$ 으로 바꿀 수 있다. 이런 방식으로 블록들을 계산하기 편하게 바꾸면  $400000000000$ 로 바꿀 수 있다. 일의 자리는 합쳐서 10이 되게 만들어주고 나머지 자리수는 합쳐서 9가 되게 만든다면 결과값은 168987956839가 된다.

##### 2. 파스칼 삼각형에서의 홀/짝의 성질

앞에서 다룬 4개의 정리로부터  $\binom{n}{k}$ 가 언제  $p$ 의 배수인지 알 수 있다.

$ex) \binom{47}{16}$ 이 3의 배수인가?

$$47 = 1 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 2 \cdot 3^0 = (1202)_3$$

$$16 = 1 \cdot 3^2 + 2 \cdot 3^1 + 1 \cdot 3^0 = (0121)_3$$

$$\therefore \binom{47}{16} \equiv \binom{2}{1} \binom{0}{2} \binom{2}{1} \binom{1}{0} \pmod{3}$$

$$\equiv 2 \cdot 0 \cdot 2 \cdot 1 \equiv 0 \pmod{3}$$

$$\Rightarrow \binom{47}{16} \text{은 } 3 \text{의 배수}$$

특히  $p$ 가 2인 경우

$$\binom{5}{2} \equiv \binom{1}{0} \binom{0}{1} \binom{1}{0} \equiv 1 \cdot 0 \cdot 1 \equiv 0 \pmod{2} \therefore \binom{5}{2} \text{는 짝수}$$

$$\binom{5}{4} \equiv \binom{1}{0} \binom{0}{0} \binom{1}{1} \equiv 1 \cdot 1 \cdot 1 \equiv 1 \pmod{2} \therefore \binom{5}{4} \text{는 홀수}$$

짝수가 되는 경우 0을 곱할 때 즉,  $\binom{0}{1}$ 이 있을 때

$$\Rightarrow \binom{0}{1} \text{이 없으면 모두 홀수}$$

$\therefore$  모두 홀수가 되는  $n$ 은  $2^\alpha - 1$  꼴은  $1 = 2^1 - 1$   $3 = 2^2 - 1$   $7 = 2^3 - 1 \dots$  등이 된다.

## V. 결론 및 제언

연구 결과를 종합해보면, 이진수 뺄셈에서의  $\binom{0}{1}$  패턴을 해결하기 위해 특정 규칙과 과정을 도출했다. 먼저, 일정한 규칙을 통해 특정 패턴을 계산하는 방법을 발견했다. 1과 0이 번갈아 가며 나타나는 패턴에서 1은 0으로, 0은 1로 바뀌는 규칙을 이용하여 계산할 수 있었다.

또한, 0과 1로 이루어진 이진수의 특정 패턴이 반복되는 경우, 이를 재구성하여 계산을 단순화할 수 있는 방법을 제시했다. 이러한 방법은 반복되는 패턴이 나타나는 부분을 특정 형태로 변환하여 계산을 용이하게 하는 것을 목표로 했다. 또한, 이러한 방법은 십진수 뺄셈에도 적용 가능하며, 더 큰 수의 뺄셈을 간편하게 해결할 수 있음을 입증하였다.

이러한 방법들을 종합하면, 이진수 뺄셈 및 십진수 뺄셈에서의 특정 패턴을 다루는 새로운 계산 방법을 제시하고, 이를 통해 더 큰 수의 뺄셈을 보다 간편하게 수행할 수 있음을 확인하였다.

연구를 통해 파스칼의 삼각형과 시에르핀스키 삼각형 사이에 흥미로운 관련성을 발견하였다. 파스칼의 삼각형의 항 중에서 홀수를 나타내는 위치와 시에르핀스키 삼각형의 형태 사이에 공통점이 존재하는데, 이것은 일정한 조건을 충족하는 경우 두 삼각형이 유사한 패턴을 공유한다는 것을 보여준다.

특히, 파스칼의 삼각형에서 홀수가 나오는 위치를 결정하는데 있어서 필요한 정리의 증명을 수행하였다.

이때, 특정한 소수  $p$ 의 배수를 확인하기 위한 조건들을 도출하였고, 이를 활용하여 주어진 수가  $p$ 의 배수인지 여부를 확인할 수 있다.

더불어, 특정한 경우에 파스칼의 삼각형의 홀수 위치와 시에르핀스키 삼각형의 형태가 유사하게 나타나는 것을 발견하였으며, 이를 통해 두 삼각형 사이에 규칙적인 관계가 존재함을 입증하였다.

향후, 다항계수  $\binom{n}{p, q, r}$ 꼴에 대한 홀수/짝수의 성질을 판별하고 3차원 시어핀스키 사면체와의 관계에 대하여 더 탐구하고 싶다.

## VI. 참고문헌

S. Wolfram, Geometry of Binomial Coefficients, American Mathematical Monthly, 91(9), 1984.

R.L. Graham, P. Erdos, Old and new problems and results in combinatorial number theory, L'Enseignement Mathematique; 1st edition, 1980.



수학B

10\_물체의 회전관성으로 유도한 Fibonacci  
수열들의 합



# 물체의 회전관성으로 유도한 Fibonacci 수열들의 합

오주하 (양정중학교 3학년)

이도현 (원촌중학교 3학년)

이우준 (중동중학교 3학년)

지도교수 : 이준복 (연세대학교 수학과)

## 논문 초록

본 논문에서는 물체의 회전관성으로 도출한 항등식과 이를 활용한 *Fibonacci* 수열의 성질 증명에 관하여 연구하였다. 밑도가 일정한 얇은 막대의 한 끝에 회전축을 배치하면 막대의 회전관성을 구할 수 있다. 이 막대를  $n$ 개의 조각으로 나눈 후, 막대 전체의 회전축에 대한 회전관성이 각 조각의 회전관성의 합과 같다는 점을 이용하여 새로운 항등식을 도출하였다. 새로운 항등식은 일반항과 닫힌 형태의 합이 정의되는 모든 수열에 대해 적용된다. 항등식은 다음과 같이 도출된다.

$$\sum_{j=1}^n (a_j^3 + 3a_j S_{j-1} S_j) = S_n^3$$

위 항등식의  $a_j$ 에  $j$ 을 대입하면 잘 알려져 있는 자연수의 5제곱의 합 공식

$$\sum_{j=1}^n j^5 = \frac{j^2(j+1)^2(2j^2+2j-1)}{12}$$

을 확인할 수 있다. 또한, 이 논문에서는  $a_j$ 에  $F_j$ 과  $F_j^2$ 도 대입하여 유의미한 결과가 나오는지 그 여부를 확인하였다. 결과적으로, 항등식의  $a_j$ 에 *Fibonacci* 수열의 일반항  $F_j$ 을 대입하면,

$$\sum_{j=1}^n F_j^3 = \frac{F_{n+1}^3 + F_n^3 + 3(-1)^{n-1} F_{n-1} + 2}{4}$$

이라는  $F_j^3$ 에 대한 관계식을 얻는다. 동일한 항등식의  $a_j$ 에 *Fibonacci* 수열의 일반항을 제곱한 값  $F_j^2$ 을 대입하면,

$$\sum_{j=1}^n F_j^6 = \frac{F_n^5 F_{n+3} + F_{2n}}{4}$$

이라는  $F_j^6$ 에 대한 관계식을 얻는다.

## I. 서론 (또는 연구의 필요성 및 목적)

최근 수학과 물리를 접목하여 함께 연구하는 사례들이 많아지고 있다. 복잡한 물리학 법칙일수록, 수학적 관점에서 바라보았을 때 더욱 뚜렷한 증명법을 얻을 수 있는 경우가 많기 때문이다. 영재원 프로그램이 본격적으로 시작하기 전인 작년(2022년) 말부터 우리는 각자 여러 논문 후보들 중 학습하고 싶은 논문을 선택하는 과정을 거쳤다. 그 결과 선택된 논문 [1]은 일반적 연구 방법인 “수학으로 물리학 법칙을 설명”하는 방식이 아닌, “물리학적 개념으로 수학적 항등식을 증명”하는 독특한 방법을 채택한다는 점을 알게 되었다. 구체적으로, 해당 논문은 수열에 “질량중심”의 개념을 도입하여 하나의 식

$$\sum_{j=1}^n (a_j^2 + 2a_j S_{j-1}) = S_n^2 \quad (1.1)$$

을 발견하고, 그 식에 다양한 일반항을 대입해서 관계식을 찾아 나간다. 우리는 이런 흔치 않은 접근에서 확장 가능성을 확인하였다.

선행 연구 논문 [1]은 직사각형의 질량중심을 이용하여 일반적인 수열에 관한 항등식  $\sum_{j=1}^n (a_j^2 + 2a_j S_{j-1}) = S_n^2$ 을 만들고,  $a_j$ 에 다양한 수열의 항을 대입함으로써 *Fibonacci* 수열의 여러 성질을 증명하였다. 이 논문의 방식에 착안하여 우리는 물체의 특정한 물리적 성질을 하나 정하고, 그 성질을 이용해서 새로운 항등식을 만들어 기존의 수식을 증명하거나 새로운 관계식을 도출하는 것을 연구 방향으로 삼았다. 물리적 성질 중 운동량, 운동에너지, 탄성력 등 다양한 물리량을 고려하였고, 결과적으로 “관성 모멘트”를 활용하는 것이 연구에 가장 적합할 것이라고 판단했다. 다음 항등식

$$\sum_{j=1}^n (a_j^3 + 3a_j S_{j-1} S_j) = S_n^3 \quad (1.2)$$

은 “밀도가 일정한 얇은 막대의 관성 모멘트”를 통해 도출한 항등식이자, 논문의 주 정리이다.

항등식 (1.2)는  $a_j$ 과  $S_j$ 이 정의되어 있는 임의의 수열에 대해 항상 성립하기 때문에 매우 유용하다. 이 점을 확인하기 위해 기존에 알려져 있는 자연수 5제곱의 합

$$\sum_{j=1}^n j^5 = \frac{j^2(j+1)^2(2j^2+2j-1)}{12}$$

이 항등식 (1.2)로부터 얻어짐을 확인할 수 있었다. 항등식 (1.2)의  $a_n$ 에 *Fibonacci* 수열의 일반항을 제공한 값  $F_n^2$ 를 대입했을 때, 기존에 알려져 있던 방식을 활용하는 증명법보다 더욱 간단한 방식으로  $F_n^6$ 에 대한 관계식을 유도할 수 있다. 또한, 동일한 항등식의  $a_n$ 에 *Fibonacci* 수열의 일반항  $F_n$ 을 대입했을 때,  $F_n^3$ 의 합에 대한 완전히 새로운 항등식이 도출된다는 사실도 발견하였다.

## II. 이론적 배경

*Fibonacci* 수열은  $F_1 = 1, F_2 = 1$ 이고, 각 항 사이의 관계식이  $F_{n+2} = F_{n+1} + F_n$ 을 만족하는 수열이다.

선행 연구 논문에서 사용하거나 밝혀낸 식들을 본 논문에서 *Fibonacci* 합 관계식을 증명하는 과정에서 다시 활용하였다. 다음 식들은 *Fibonacci* 수열의 대표적인 성질들이다.

$$\sum_{j=1}^n F_j = F_{n+2} - 1 \quad (2.1)$$

$$\sum_{j=1}^n F_j^2 = F_n F_{n+1} \quad (2.2)$$

$$\sum_{j=1}^n F_j F_{j+3} = F_{n+2}^2 - 1 \quad (2.3)$$

$$\sum_{j=1}^n F_{2j} = F_{2n+1} - 1 \quad (2.4)$$

$$\sum_{j=1}^n F_j \cdot (-1)^j = (-1)^n F_{n-1} - 1 \quad (2.5)$$

$$F_n (2F_{n+1} - F_n) = F_{2n} \quad (2.6)$$

등식 (2.1)에서 (2.6)는 모두 수학적 귀납법을 통해 보일 수 있으나, 여기서는 (2.1), (2.3), (2.5)식을 수학적 귀납법으로 증명해 보았다.

(2.1) 증명

$$n = 1 : F_1 = F_3 - 1 = 1 \rightarrow \text{참}$$

$$n = k : \sum_{j=1}^k F_j = F_{k+2} - 1 \text{ 성립함을 가정}$$

$$n = k + 1 : \sum_{j=1}^{k+1} F_j = F_{k+2} - 1 + F_{k+1} = F_{k+3} - 1$$

$$\therefore \sum_{j=1}^n F_j = F_{n+2} - 1$$

$$n = 1 : F_1 F_4 = F_3^2 - 1 = 3 \rightarrow \text{참}$$

$$n = k : \sum_{j=1}^k F_j F_{j+3} = F_{j+2}^2 - 1 \text{ 성립함을 가정}$$

$$n = k + 1 : \sum_{j=1}^{k+1} F_j F_{j+3} = \sum_{j=1}^k F_j F_{j+3} + F_{k+1} F_{k+4}$$

$$= F_{k+2}^2 - 1 + F_{k+1} F_{k+4}$$

$$= F_{k+2}^2 - 1 + (F_{k+3} - F_{k+2})(F_{k+3} + F_{k+2})$$

$$\begin{aligned}
&= F_{k+2}^2 - 1 + F_{k+3}^2 - F_{k+2}^2 \\
&= F_{k+3}^2 - 1 \\
&\therefore \sum_{j=1}^n F_j F_{j+3} = F_{n+2}^2 - 1
\end{aligned}$$

(2.5) 증명

$n$ 이 짝수일 때,  $n = 2k$ 라 하면

$$\begin{aligned}
\sum_{j=1}^{2k} F_j \cdot (-1)^j &= (F_2 - F_1) + (F_4 - F_3) + \dots + (F_{2k} - F_{2k-1}) \\
&= F_2 + F_4 + F_6 + \dots + F_{2k-2} \\
&= \sum_{j=1}^{k-1} F_{2j} = F_{2k-1} - 1
\end{aligned}$$

같은 방법으로,  $n$ 이 홀수일 때  $n = 2k+1$ 이라 하면

$$\begin{aligned}
\sum_{j=1}^{2k+1} F_j \cdot (-1)^j &= F_{2k-1} - 1 - F_{2k+1} = -F_{2k} - 1 \\
\therefore \sum_{j=1}^n F_j \cdot (-1)^j &= (-1)^n F_{n-1} - 1
\end{aligned}$$

선행 연구 논문에서 유도하거나 활용한 *Fibonacci* 수열에 대한 기본적인 성질들 외에도, 두 가지의 다른 관계식을 증명 과정에서 이용하였다.

$$F_{n-1}F_{n+1} - F_n^2 = (-1)^n \quad (2.7)$$

$$F_{n+1}F_{n+2} + (-1)^n = F_n F_{n+3} \quad (2.8)$$

두 식은 유사한 방법으로 성립함을 보일 수 있기 때문에, (2.8)에 대한 증명은 생략하고 *Cassini-Catalan* 식으로 알려져 있는 (2.7)을 수학적 귀납법을 통해 증명하겠다.

(2.7) 증명

$$n = 2: F_1 F_3 - F_2^2 = (-1)^2 \rightarrow \text{참}$$

$$n = k: F_{k-1} F_{k+1} - F_k^2 = (-1)^k \text{ 성립함을 가정}$$

$$n = k+1: F_{k-1} F_{k+1} + F_k F_{k+1} - F_k^2 - F_k F_{k+1} = (-1)^k$$

$$F_{k+1}(F_{k-1} + F_k) - F_k(F_k + F_{k+1}) = (-1)^k$$

$$F_{k+1}^2 - F_k F_{k+2} = (-1)^k$$

식의 양변에  $(-1)$ 을 곱하면

$$F_k F_{k+2} - F_{k+1}^2 = (-1)^{k+1}$$

$$\therefore F_{n-1} F_{n+1} - F_n^2 = (-1)^n$$

마지막으로, 이 논문의 주정리가 될 항등식을 도출하기 위해서는 관성모멘트와 평

행축 정리의 기본적인 개념에 대해 이해하고 있어야 한다.

관성모멘트란 회전 운동하는 물체가 운동을 유지하려는 정도를 나타내는 물리량으로, “회전 관성” 이라고도 불린다. 관성모멘트는 물체의 모양과 밀도, 질량의 영향을 받으며, 동일한 물체라도 회전축에 따라 그 값이 다르게 측정된다. 특히 회전축이 중앙에 있는 얇은 막대의 경우, 관성모멘트  $I_{cm}$  는  $\frac{1}{12}ML^2$  (단,  $M$ 은 질량,  $L$ 은 막대의 길이)으로 나타낼 수 있다. 증명은 다음과 같다.

증명) 물체의 질량을 길이와 밀도에 대한 관계식으로 나타내자.

$M = \rho L$ 이므로  $dm = \rho dr$ 이다. (단,  $\rho$ 는 밀도,  $r$ 은 회전축에서 질량중심까지 거리) 따라서

$$I_{cm} = 2 \int_0^{\frac{L}{2}} r^2 dm = 2\rho \int_0^{\frac{L}{2}} r^2 dr = \frac{1}{12}\rho L^3 = \frac{1}{12}ML^2$$

$$\therefore I_{cm} = \frac{1}{12}ML^2 \quad (2.9)$$

관성모멘트와 관련된 정리에는 평행축 정리와 수직축 정리가 있는데, 그 중 평행축 정리를 활용하였다. 평행축 정리는 한 물체의 서로 평행한 두 회전축에 대한 관성모멘트의 관계를 나타낸 식이다. 관성모멘트  $I = I_{cm} + Md^2$ 로 나타낼 수 있으며 증명은 다음과 같다.

증명) 두 회전축이  $z$ 축에 평행하다고 하자. 이때  $I_{cm}$ 은

$$I_{cm} = \sum_i M_i(x_i^2 + y_i^2)$$

와 같이 나타낼 수 있다. 질량 중심을 직교좌표계의 중심으로 놓고  $xy$ 평면 상에서 새로운 회전축의 좌표를  $(a, b)$ 라 놓으면  $d^2 = a^2 + b^2$ 이 되고 (단,  $d$ 는 회전축을 평행이동한 거리), 새로운 관성모멘트  $I$ 는

$$I = \sum_i M_i[(x_i - a)^2 + (y_i - b)^2]$$

$$= \sum_i M_i(x_i^2 + y_i^2) - 2a \sum_i M_i x_i - 2b \sum_i M_i y_i + (a^2 + b^2) \sum_i M_i$$

$$= I_{cm} + Md^2 \left( \because \sum_i M_i x_i = 0, \sum_i M_i y_i = 0 \right)$$

$$\therefore I = I_{cm} + Md^2 \quad (2.10)$$

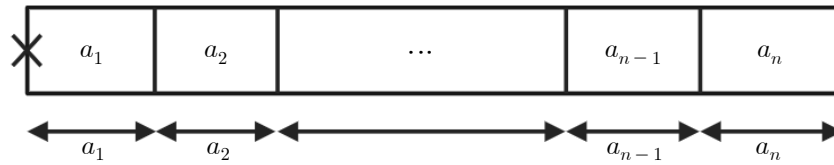
위에서 구한 회전축이 중앙에 있는 얇은 막대의 관성모멘트 식에 평행축 정리를 적용했을 때, 회전축이 한 끝에 있는 막대의 관성모멘트를 구할 수 있다. 즉,

$$I = \frac{1}{12}ML^2 + M \cdot \left(\frac{1}{2}L\right)^2 = \frac{1}{3}ML^2 \text{으로 나타난다.}$$

$$\therefore I = \frac{1}{3}ML^2 \quad (2.11)$$

### Ⅲ. 연구 방법

밀도가 1이고 길이가  $L$ 인 얇은 막대 하나를 생각해 보자. 이 막대를 임의로  $n$ 개의 조각으로 나누고, 왼쪽에서부터  $j$ 번째 조각의 길이와 질량을  $a_j$ 라 하자. 여기서, 첫 번째 조각의 길이 및 질량은  $a_1$ , 두 번째 조각의 길이 및 질량은  $a_2$ ,  $n$ 번째 조각의 길이 및 질량  $a_n$ , ... 등이다.  $a_1 + a_2 + a_3 + \dots + a_n = S_n$ 이라 정의한다.

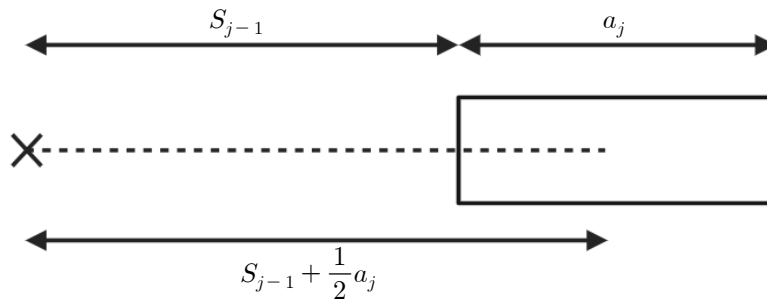


<그림1. 왼쪽 끝이 회전축인 밀도가 1인 막대>

막대를  $n$ 개의 조각으로 나누었을 때, 회전축에 대한 각 조각의 관성모멘트의 총합은 막대 전체의 관성모멘트 값과 필연적으로 같아야 한다. 이를 활용해 항등식을 유도해 보겠다.

이때, 막대 전체의 관성모멘트  $I = \frac{1}{12}a_j \cdot a_j^2$ 이다.

막대의 왼쪽 끝의 좌표를  $O$ 라 하고  $n$ 개의 조각으로 나누었을 때, 막대의 왼쪽 끝을 기준으로  $j$ 번째 조각의 질량중심의 좌표는  $S_{j-1} + \frac{1}{2}a_j$ 로 표현된다.



<그림2. 회전축과 막대의  $j$ 번째 조각>

따라서, 평행축 정리  $I_j = I_{cm} + Md^2$ 을 이용해  $j$ 번째 조각의 관성모멘트를 나타내면

$$\begin{aligned} I_j &= \frac{1}{12}a_j \cdot a_j^2 + a_j \left( S_{j-1} + \frac{a_j}{2} \right)^2 \quad (\text{단, } S_0 = 0) \\ &= \frac{1}{3}a_j^3 + a_j^2 S_{j-1} + a_j S_{j-1}^2 \\ &= \frac{1}{3}a_j^3 + a_j S_{j-1} S_j \end{aligned}$$

여기서, 각 조각의 관성모멘트를 모두 더한 값은

$$\sum_{j=1}^n I_j = \sum_{j=1}^n \left( \frac{1}{3} a_j^3 + a_j S_{j-1} S_j \right)$$

으로 도출된다.

정리 1.2)

$$\therefore \sum_{j=1}^n (a_j^3 + 3a_j S_{j-1} S_j) = S_n^3$$

#### IV. 연구 결과

관성모멘트를 이용해 구한 항등식 (1.2)의  $a_j$ 에 다양한 수열의 일반항을 대입하여 기존의 항등식을 새롭게 증명하거나, 새로운 항등식을 도출할 수 있다.

예제 3.1)

$$\sum_{j=1}^n j^5 = \left\{ \frac{j(j+1)}{2} \right\}^2 \cdot \frac{2n^2 + 2n + 1}{3} \quad (3.1)$$

항등식 (1.2)의  $a_j$ 에  $j$ 를 대입함으로써 1부터  $n$ 까지 자연수의 5제곱의 합 공식을 보이자.

이때,  $S_j$ 는  $\frac{j(j+1)}{2}$  꼴로 나타난다. 이를 식에 대입하면

$$\begin{aligned} \sum_{j=1}^n \left[ j^3 + 3j \cdot \frac{(j-1)j}{2} \cdot \frac{j(j+1)}{2} \right] &= \left( \sum_{j=1}^n j \right)^3 \\ \sum_{j=1}^n \left\{ \frac{3}{4} j^5 + \frac{1}{4} j^3 \right\} &= \left( \sum_{j=1}^n j \right)^3 \\ \rightarrow \sum_{j=1}^n \left( \frac{3}{4} j^5 \right) &= \left( \sum_{j=1}^n j \right)^3 - \frac{1}{4} \sum_{j=1}^n j^3 \\ \frac{3}{4} \sum_{j=1}^n j^5 &= \sum_{j=1}^n j^3 \cdot \left\{ \frac{n(n+1)}{2} - \frac{1}{4} \right\} \\ \therefore \sum_{j=1}^n j^5 &= \sum_{j=1}^n j^3 \cdot \left( \frac{2n^2 + 2n + 1}{3} \right) \end{aligned}$$

또한,

$$\sum_{j=1}^n j^3 = \left\{ \frac{n(n+1)}{2} \right\}^2$$

임을 위 식에 적용하면

$$\sum_{j=1}^n j^5 = \left\{ \frac{j(j+1)}{2} \right\}^2 \cdot \frac{2n^2 + 2n + 1}{3}$$

널리 알려져 있는 자연수 5제곱의 합 관계식을 확인함에 따라 항등식이 성립함을 입증하는 동시에, (1.1) 항등식으로는 보일 수 없는 식을 증명하였다.

**정리 3.2)**

$$\sum_{j=1}^n F_j^6 = \frac{1}{4} F_n^5 F_{n+3} + F_{2n} \quad (3.2)$$

항등식 (1.2)에 *Fibonacci* 수열을 적용해보자.

위 항등식의  $a_j$ 에  $F_j^2$ 를 대입한다.

$a_k = F_k^2$ 일 때,  $S_k = F_k F_{k+1}$  ( $\because$  (2.2))

$$\begin{aligned} F_n^3 F_{n+1}^3 &= \sum_{j=1}^n F_j^6 + 3 \sum_{j=1}^n F_j^2 \cdot F_j F_{j+1} \cdot F_{j-1} F_j \\ &= \sum_{j=1}^n F_j^6 + 3 \sum_{j=1}^n F_j^4 F_{j+1} F_{j-1} \end{aligned}$$

*Cassini - Catalan* 식을 이용해 정리하면

$$\begin{aligned} &= \sum_{j=1}^n F_j^6 + 3 \sum_{j=1}^n F_j^6 + 3 \sum_{j=1}^n F_j^4 \cdot (-1)^j \\ &= 4 \sum_{j=1}^n F_j^6 + 3 \sum_{j=1}^n F_j^4 \cdot (-1)^j \end{aligned}$$

이때,  $3 \sum_{j=1}^n F_j^4 \cdot (-1)^j$ 은 논문 [2]의 따름정리 3.6에 따라

$3 \cdot \frac{(-1)^n}{3} F_n F_{n+1} F_{n-2} F_{n+3}$ 으로 나타낼 수 있다. 즉,

$$F_n^3 F_{n+1}^3 = 4 \sum_{j=1}^n F_j^6 + (-1)^n F_n F_{n+1} F_{n-2} F_{n+3}$$

위 식을  $\sum_{j=1}^n F_j^6$ 에 대해 표현하기 위해 이항한 후 정리하면

$$\begin{aligned} \sum_{k=1}^n F_k^6 &= \frac{1}{4} F_n F_{n+1} \{F_n^2 F_{n+1}^2 - (-1)^n F_n F_{n+1} + 2\} \\ &= \frac{1}{4} F_n \{F_n^2 F_{n+1}^3 - (-1)^n F_n F_{n+1}^2 + 2F_{n+1}\} \\ &= \frac{1}{4} F_n [F_n^2 F_{n+1} \{F_n F_{n+2} - (-1)^{n+1}\} - (-1)^n F_n \{F_n F_{n+2} - (-1)^{n+1}\} + 2F_{n+1}] \quad (\because (2.7)) \\ &= \frac{1}{4} F_n \{F_n^3 F_{n+1} F_{n+2} + (-1)^n F_n^2 F_{n+1} - (-1)^n F_n^2 F_{n+2} + 2F_{n+1} - F_n\} \\ &= \frac{1}{4} F_n^3 \{F_n F_{n+1} F_{n+2} + (-1)^n F_{n+1} - (-1)^n F_{n+2}\} + \frac{1}{4} F_{2n} \\ &= \frac{1}{4} F_n^4 \{F_{n+1} F_{n+2} + (-1)^{n+1}\} + \frac{1}{4} F_{2n} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{4}F_n^4 F_n F_{n+3} + \frac{1}{4}F_{2n} (\because (2.8)) \\
&= \frac{1}{4}(F_n^5 F_{n+3} + F_{2n})
\end{aligned}$$

따라서, *Fibonacci* 수열 각 항의 6제곱의 합은 다음과 같이 나타난다.

$$\sum_{j=1}^n F_j^6 = \frac{1}{4}F_n^5 F_{n+3} + F_{2n}$$

논문 [3]에서 증명한 과정과는 전혀 다른 방법으로 문제에 접근하였고, 몇몇 보조정리들을 도입한 결과 (1.2) 항등식으로 매우 간결하게 증명을 완성하였다.

**정리 3.3)**

$$\sum_{j=1}^n F_j^3 = \frac{1}{4}\{F_{n+1}^3 + F_n^3 + 3(-1)^{n-1}F_{n-1} + 2\} \quad (3.3)$$

다음으로, 항등식 (1.2)의  $a_j$ 에 *Fibonacci* 수열의 일반항  $F_j$ 를 대입해보자.

$$a_j = F_j \text{이고, } S_j = F_{j+2} - 1 (\because (2.1))$$

(1.2) 항등식에 의해

$$\begin{aligned}
&\sum_{j=1}^n \{F_i^3 + 3F_i(F_{i+1} - 1)(F_{i+2} - 1)\} = (F_{n+2} - 1)^3 \\
&= \sum_{j=1}^n \{F_j^3 + 3F_j(F_{j+1}F_{j+2} - F_{j+3} + 1)\} \\
&= \sum_{j=1}^n \{F_j^3 + 3F_jF_{j+1}F_{j+2}\} - 3\sum_{j=1}^n F_jF_{j+3} + 3\sum_{j=1}^n F_j \\
&= \sum_{j=1}^n (F_j^3 + 3F_jF_{j+1}F_{j+2}) - 3(F_{n+2}^2 - 1) + 3(F_{n+2} - 1) (\because (2.1), (2.3)) \\
&\therefore \sum_{j=1}^n (F_j^3 + 3F_jF_{j+1}F_{j+2}) = F_{n+2}^3 - 1
\end{aligned}$$

이때,  $\sum_{j=1}^n (F_j^3 + 3F_jF_{j+1}F_{j+2})$ 은 다음과 같이 표현할 수도 있다.

$$\begin{aligned}
\sum_{j=1}^n (F_j^3 + 3F_jF_{j+1}F_{j+2}) &= \sum_{j=1}^n \{F_j^3 + 3F_{j+1}(F_{j+1}^2 + (-1)^{j+1})\} \\
&= \sum_{j=1}^n F_i^3 + 3\sum_{j=1}^n F_{j+1}^3 + 3\sum_{j=1}^n F_{j+1} \cdot (-1)^{j+1} \\
&= \sum_{j=1}^n F_j^3 + 3\left(\sum_{j=1}^{n+1} F_j^3 - 1\right) + \left\{3\sum_{j=1}^{n+1} F_j \cdot (-1)^i + 1\right\} \\
&= 4\sum_{j=1}^{n+1} F_i^3 + 3\sum_{j=1}^{n+1} F_i \cdot (-1)^j - F_{n+1}^3
\end{aligned}$$

$n+1$ 을 다시  $n$ 으로 바꿔 쓴 후 식을 정리하면

$$4 \sum_{j=1}^n F_j^3 + 3 \sum_{j=1}^n F_j \cdot (-1)^j = 4 \sum_{j=1}^n F_j^3 + \{3F_{n-1} \cdot (-1)^n - 1\} (\because (2.5))$$

$$= F_{n+1}^3 + F_n^3 - 1$$

따라서, *Fibonacci* 수열 각 항의 3제곱의 합은 다음과 같이 나타난다.

$$\sum_{j=1}^n F_j^3 = \frac{1}{4} \{F_{n+1}^3 + F_n^3 + 3(-1)^{n-1} F_{n-1} + 2\}$$

항등식의  $a_n$ 에  $F_n$ 을 대입함으로써 *Fibonacci* 수열의 일반항의 세제곱의 합을 공식화하였다. 기존의 알려져 있던 식이 아닌, 항등식 (1.2)을 통해 새롭게 만들어진 수열의 합 관계식이다.

## V. 결론 및 제언

결론적으로, 본 논문에서는 물리적인 성질을 도입하여 모든 수열에 대해 성립하는 간단한 형태의 항등식을 만들고, 여기에 *Fibonacci* 숫자와 관련된 일반항을 대입함으로써 *Fibonacci* 관계식 (3.1), (3.2), (3.3)을 새롭게 도출하고 증명하였다. 논문의 주정리로 활용되었던 항등식 (1.2)는 물리법칙을 이용하여 귀납법 등의 일반적인 방법으로는 다루기 어려운 고차의 일반항에 대한 관계식에 쉽게 접근하는 데 도움을 준다는 점에서도 활용도가 높다.

본 논문에서 다룬 내용은 다양한 측면에서 확장 가능성이 있다. 우선, 논문의 주정리로 활용되었던 항등식 (1.2)를 통해 우리가 본 논문에서는 다루지 않은 *Lucas* 수 (각 항 사이의 관계식  $a_{n+2} = a_{n+1} + a_n$ 은 *Fibonacci* 수와 동일하지만, 초기항이  $L_1 = 1, L_2 = 3$ 로 다른 수열)에 관한 관계식을 찾아볼 수 있을 것이다.

또한, 선행 연구 논문의 항등식 (1.1)과 본 논문의 항등식 (1.2)는 모두  $a_j$ 의 거듭제곱, 그리고  $a_j$ 와  $S_j$ 의 곱이 포함된 식이라는 점에서 형태가 유사하다는 점을 발견하였다. 이를 바탕으로 하여, 추후  $a_j$ 과  $S_j$ 이 포함된 4차 이상의 항등식을 물리적인 방법으로 모색하는 것도 흥미로워 보인다.

## VI. 참고문헌

- [1] David Treeby, Further physical derivations of Fibonacci summations, *The Fibonacci Quarterly*, 54.4(2016), 327 - 334.
- [2] Kunle Adegoke, Sums of fourth powers of Fibonacci and Lucas numbers, <https://arxiv.org/pdf/1706.00407.pdf>.
- [3] "A New Formula For The Sum Of The Sixth Powers Of Fibonacci

Numbers" H Ohtsuka, S Nakamura, *Congressus Numerantium* Vol. 201 (2010),  
Proceedings of the Thirteenth Conference on Fibonacci Numbers and their  
Applications



수학B

## 11\_함수와 수학적 모델링



# 함수와 수학적 모델링

김정윤 (수서중학교 3학년)  
김민후 (역삼중학교 2학년)  
어우경 (단국대학교 사범대학 부속 중학교 2학년)  
이성준 (대청중학교 2학년)  
황현우 (삼선중학교 2학년)

지도조교 : 박재현 (서울대학교 수학교육과)  
강우혁 (서울대학교 수학교육과)  
지도교수 : 석진명 (서울대학교 수학교육과)

## 초록

본 연구에서는 피식자-포식자의 생태계에 대해 다양한 조건에서 수학적 모델링을 해보고 탐구한다. 먼저, 인구증가이론과 새로운 조건을 추가한 로지스틱 모델에 대해 살펴본 후, 이를 바탕으로 피식자-포식자 모델의 식에서도 계수를 바꾸었을 때, 자기 경쟁이라는 상황이 생겼을 때, 조업 활동과 같은 외부요인이 발생하였을 때로 나누고 식으로 나타내었다. Python의 코딩을 활용해 그래프로 시각화하고 분석한 후 해석하였다. 한 종의 개체 수가 2 미만을 멸종이라고 가정하고 외부요인이 발생할 때 피식자보다 포식자가 먼저 멸종할 것이라는 말을 알 수 있었다. 또한 피식자와 포식자 모델의 식에서 계수를 바꾸었을 때, 그래프 개형에 각 계수가 어떤 영향을 주는지 확인하였다. 모델링 결과와 달리 TB Atwood(2020)에서 초식 동물이 육식 동물보다 더 먼저 멸종할 수 있을 것이라는 연구가 있다는 사실로부터 모델링을 할 때, 현실에 잘 부합하는 결론을 도출하기 위해 다양한 조건을 고려할 필요가 있다.

## I. 서론

우리는 오랫동안 COVID-19라는 유례없는 팬데믹 시기를 겪었다. 이를 극복하기 위해서 감염병 예측 모델을 통해 앞으로의 위기를 예측하고 다양한 정책들을 펼쳤다. 이처럼 수학적 모델링이 자연의 현상이나 실생활의 여러 가지 상황들을 이해하고 예측하는데 도움을 줄 수 있다는 점에서 수학의 가치를 깨달을 수 있었다.

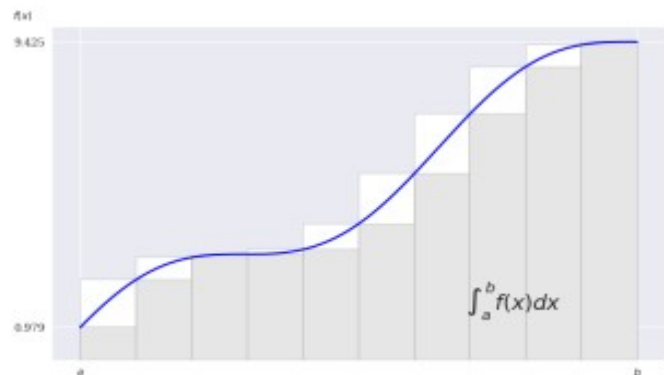
최근 기후 변화와 환경문제가 심각한 문제로 대두하였고 이러한 문제들이 피식자와 포식자가 존재하는 생태계에는 어떻게 영향을 주는지 탐구해보고 싶었다. 이러한 질문을 해결하기 위해 수학적 모델링을 하기로 하였다. 수학적으로 모델링하기 위한 방법을 찾아보던 중 연속적인 변화에 대한 모델링에 사용되는 미분방정식이 적합하다고 생각하였다. 본 연구에서는 미분방정식을 활용해 시스템을 모델링하여 어떤 변수가 영향을 미치고 있는지 파악해보고 변수를 제어하여 결과를 바꾸거나 이후에 일어날 현상을 예측해보고자 다음과 같은 연구 질문을 설정하였다.

1. 피식자-포식자 모델에서 서로 경쟁하는 조건을 추가하였을 때 어떠한 해석을 할 수 있는가?
2. 피식자-포식자 모델에서 외부요인이 추가되었을 때 어떠한 해석을 할 수 있는가?
3. 피식자-포식자 모델에서 계수를 변경하여 그래프의 개형이 바뀌었을 때, 이를 어떻게 해석할 수 있는가?

## II. 이론적 배경

### 1. 적분과 미분

적분이란 정의된 함수의 그래프와 그 구간으로 둘러싸인 도형의 넓이를 구하는 것이다. 주로  $\int f(x)dx$  와 같이 표기한다. 적분은 크게 부정적분과 정적분으로 나뉜다. 부정적분은 미분의 역연산으로  $f(x)$ 의 원시함수  $F(x)$ 를 찾는 과정이다. 정적분은 주어진 도형을 작게 나눈 기본 도형의 넓이나 부피의 합으로 근사값을 구한 다음, 그 근사값의 극한으로서 주어진 도형의 넓이나 부피를 구하는 방법이다. 넓이를 구할 때 주어진 도형을 작게 나눈 기본 도형의 넓이나 부피의 합으로 근사값을 구한 다음, 그 근사값의 극한으로써 주어진 도형의 넓이나 부피를 구하는 방법을 구분구적법이라 한다. 그리고 구분구적법을 통해 정적분을 정의할 수 있다. 상합과 하합을 이용하여 넓이를 근사할 수 있는데 등분한 수를 점점 늘려나가면 오차를 줄일 수 있다. 그래프의 구간을 일정하게 나눌 때 상합은 주어진 구간에서 함수를 위에서 근사적으로 정적분하는 방법 중 하나이다. 하합은 반대로 주어진 구간에서 함수를 아래에서 근사하는 방법이다. 상합과 하합 계산을 코딩하여 그래프로 아래의 그림과 같이 나타낼 수 있다.



[그림 II-1] 상합과 하합 계산을 코딩하여 나타낸 그래프

구분구적법, 정적분의 응용으로 입체도형, 평면도형의 넓이를 근사하고 구할 수 있다. 예를 들어 원의 넓이를 구할 때 두 가지 방법을 사용할 수 있는데, 한 가지는

원에 외접하는 다각형과 내접하는 다각형의 넓이를 구해 사잇값을 근사하면 원의 넓이를 근사할 수 있다. 다른 방법은 원의 중심에서 테두리까지 잘게 잘라 이어 붙여 직사각형을 만들어 넓이를 구하는 방법이다. 이와 같은 방법들로 각기둥, 각뿔의 부피 등도 구할 수 있다.

한편 미분은 어떤 함수의 순간 변화율 즉 한 점에서의 기울기를 구하는 작업이다. 함수가 연속이고 미분계수가 존재하는 경우에만 미분가능하다. 함수를 미분하면 그 함수의 도함수가 나온다. 주로  $f'(x)$ 나  $\frac{dy}{dx}$ 로 표기한다.

적분이라는 개념은 미분보다 훨씬 오래되었다. 고대에 아르키메데스 등 여러 수학자들이 포물선과 같이 곡선으로 이루어진 영역의 넓이를 삼각형으로 나눠 구하는 방법, 대상을 잘게 나눠서 원주율 구하기, 구나 원기둥의 넓이 구하기 등을 알아냈다. 적분은 여러 가지 도형의 넓이를 구하는 것에서 시작되었다.

반면 미분은 적분보다 발달이 늦었다. 곡선의 접선에 관한 문제와 함수의 최대 최소에 관한 문제로 인하여 창조되었고 뉴턴과 라이프니츠에 의해 많이 발전했다. 뉴턴과 라이프니츠는 서로 독자적인 방법으로 미분을 발견했다. 뉴턴은 기하에서의 접선, 곡률들을 이용해 발전시켰고 라이프니츠는 변화량을 이용했다. 그리고 나서 적분과 미분이 서로 역연산의 관계에 있다는 사실이 밝혀졌다.

## 2. 극대, 극소와 변곡점

### 가. 그래프의 개형 찾기

고등학교 미적분 교과서에서, 어떤 함수  $y=f(x)$ 의 그래프를 그리기 위해서는 일반적으로 다음 표Ⅱ-1 과 같은 사항들을 조사할 수 있다고 하였다.

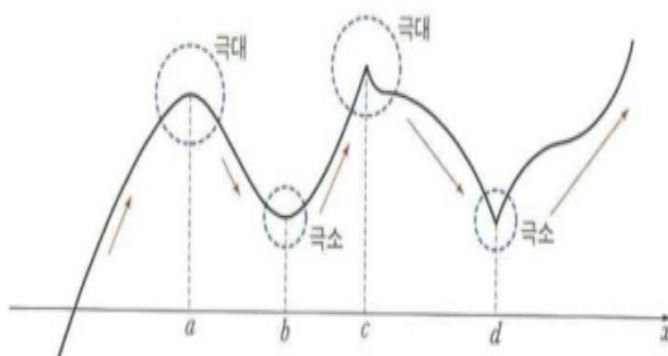
<ul style="list-style-type: none"> <li>① 함수의 정의역과 치역을 찾는다.</li> <li>② 대칭축과 주기의 유무를 확인한다.</li> <li>③ 좌표축과의 교점들을 구한다.</li> <li>④ <math>x \rightarrow \pm\infty</math>일 때의 <math>f(x)</math>의 값을 구해서 점근선의 유무를 확인한다.</li> <li>⑤ <math>f'(x)</math>를 구하여 함수의 증감과 극값을 조사한다.</li> <li>⑥ <math>f''(x)</math>서 변곡점과 함수의 오목, 볼록을 조사한다.</li> <li>⑦ 불연속점, 미분 불가능한 점들을 찾는다.</li> </ul>
---

[표Ⅱ-1]  $y=f(x)$ 의 그래프 개형 그리는 방법

이번 연구에서는 위의 7단계 중 ⑤,⑥,⑦을 중심으로 그래프를 그릴 것이다.

## 나. 극대와 극소의 정의

상수함수가 아닌 함수  $f(x)$ 에 대하여  $x=a$ 를 포함하는 어떤 열린구간에서  $f(a) \geq f(x)$ 이면 함수  $f(x)$ 는  $x=a$ 에서 극대, 그때의  $f(a)$ 를 극댓값이라고 부른다. 상수함수가 아닌 함수  $f(x)$ 에 대하여  $x=a$ 를 포함하는 어떤 열린구간에서  $f(a) \leq f(x)$ 이면 함수  $f(x)$ 는  $x=a$ 에서 극소, 그때의  $f(a)$ 를 극솟값이라고 부른다. 열린 구간은 구간의 양 끝점  $a, b$ 를 포함하지 않는 구간을 의미한다. 소괄호를 이용해서  $(a, b) = \{x \in \mathbb{R} \mid a < x < b\}$ ,  $(a, \infty) = \{x \in \mathbb{R} \mid a < x\}$ ,  $(-\infty, b) = \{x \in \mathbb{R} \mid x < b\}$ ,  $(-\infty, \infty) = \mathbb{R}$ 와 같이 나타낸다.



[그림 II-2] 그래프의 극대와 극소

## 다. 극대, 극소 구하기

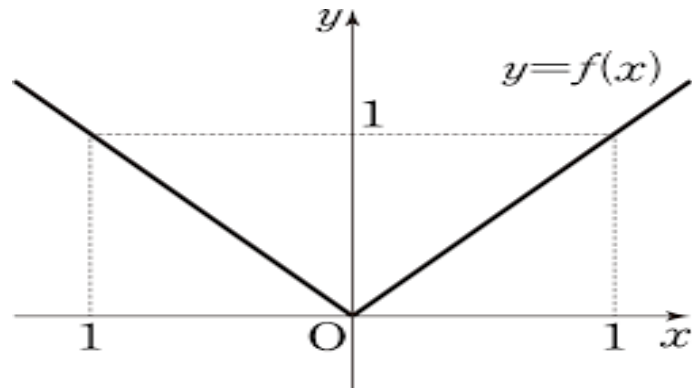
2번 미분가능한 함수에서 미분가능한 함수  $f(x)$ 에 대하여  $f'(a)=0$ 이고  $x=a$ 의 좌우에서  $f'(x)$ 의 부호가 양에서 음으로 바뀌면  $x=a$ 의 좌우에서  $f(x)$ 가 증가상태에서 감소상태로 바뀌므로  $f(x)$ 는  $x=a$ 에서의 극대라고 할 수 있다.

미분가능한 함수  $f(x)$ 에 대하여  $f'(a)=0$ 이고  $x=a$ 의 좌우에서  $f'(x)$ 의 부호가 음에서 양으로 바뀌면  $x=a$ 의 좌우에서  $f(x)$ 가 감소상태에서 증가상태로 바뀌므로  $f(x)$ 는  $x=a$ 에서의 극소라고 할 수 있다.

$f'(x)$ 의 부호가 음에서 양으로 변할 때  $f'(x)$ 가 증가하므로  $f''(a) > 0$ 이고  $f'(x)$ 의 부호가 음에서 양으로 변할 때  $f'(x)$ 가 감소하므로  $f''(a) < 0$ 이다. 이를 통해  $f''(a)$ 를 이용해서  $f(a)$ 가 극댓값인지 극솟값인지 판단할 수 있다. 예를 들어  $f(x) = x^4 - 2x^3$ 의 극대와 극소를 찾는다면,  $f'(x) = 4x^3 - 6x^2$ 이다.  $f'(x) = 0$ 을 만족하는  $x$ 의 값은  $x=0$  또는  $x = \frac{3}{2}$ 이고  $f''(x) = 12x^2 - 12x$ 이다. 따라서,  $f''(0) = 0$ 이고  $f''(\frac{3}{2}) = 9 > 0 = 0$ 이므로  $x=0$ 일 때는 극값을 갖지 않고  $x = \frac{3}{2}$ 일 때 극솟값을 가진다.

한편, 미분 불가능한 점이 있는 함수에서는 미분 불가능한 점이 있어도 극대, 극소를 가질 수 있다. 다음의 그림과 같은 예를 살펴보았을 때, 함수는  $x=0$ 에서 미분

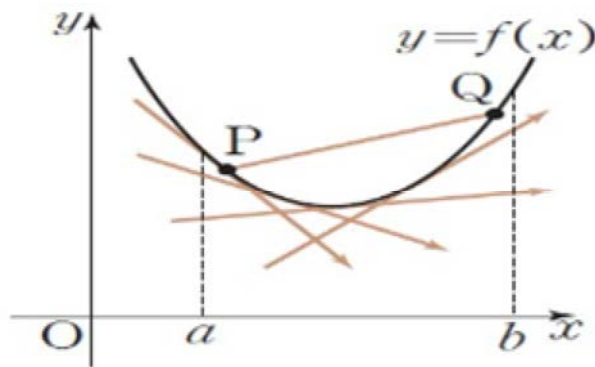
불가능하지만  $x=0$ 에서 극솟값을 가진다.



[그림 II-3] 미분 불가능하지만 극솟값을 갖는 예

라. 함수의 오목, 볼록과 변곡점

함수  $f(x)$ 가 어떤 구간에서  $f''(x) > 0$ 이면 곡선  $y=f(x)$ 는 이 구간에서 아래로 볼록하다. 마찬가지로, 함수  $f(x)$ 가 어떤 구간에서  $f''(x) < 0$ 이면 곡선  $y=f(x)$ 는 이 구간에서 위로 볼록하다.



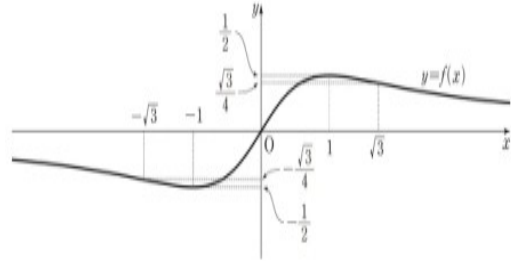
[그림 II-4] 볼록함수에서의 이계도함수

위 그림에서도 이계도함수가 양수이므로 도함수가 증가함수이고 함수는 아래로 볼록하다. 함수  $f(x)$ 가 열린 구간  $(a,b)$ 에서 정의된 두 번 미분가능한 함수라고 할 때, 열린 구간  $(a,b)$ 의 한 점  $c$ 에서  $f''(c)=0$ 이고  $x=c$  좌우에서  $f''(x)$ 의 부호가 바뀌면 점  $(c, f(c))$ 는 변곡점이 된다. 변곡점이 아닌 점에서 접선은 접점의 주변에서 함수의 그래프가 접선의 위 또는 아래의 어느 한쪽에 위치하게 된다. 그러나 변곡점에서 그래프의 접선은 변곡점에서 그래프의 개형이 바뀌므로 변곡점의 주변에서 함수의 그래프가 접선의 위와 아래 모두에 위치하게 된다. 즉, 함수의 변곡점에서의 접선은 그래프와 한 점에서 만난다.

마. 함수의 증감표

고등학교 미적분 교과서에서, 증감표는 함수의 증가와 감소를 나타낸 표로 특정한 함수의 그래프의 개형을 파악하기 위하여 함수의 증가와 감소, 변곡점을 나타낸 표이다. 예시는 다음과 같다.

$x$	...	$-\sqrt{3}$	...	$-1$	...	$0$	...	$1$	...	$\sqrt{3}$	...
$y'$	-	-	-	0	+	+	+	0	-	-	-
$y''$	-	0	+	+	+	0	-	-	-	0	+
$y$	↘	$-\frac{\sqrt{3}}{4}$ (변곡점)	↘	$-\frac{1}{2}$ (극소)	↗	$0$ (변곡점)	↗	$\frac{1}{2}$ (극대)	↘	$\frac{\sqrt{3}}{4}$ (변곡점)	↘



[표 II-1] 함수  $f(x) = \frac{x}{x^2+1}$ 의 증감표    [그림 II-5] 함수  $f(x) = \frac{x}{x^2+1}$ 의 그래프

위의 예시는  $f(x) = \frac{x}{x^2+1}$ 의 그래프를 증감표를 통해 극값과 변곡점을 찾아서 그래프의 개형을 그린 것이다. 위와 같은 방법으로 그래프의 특징들을 조사하여 그래프의 형태를 그릴 수 있다.

2. 미분방정식과 수학적 모델링

수학적 모델링은 현실 세계의 현상에서 발생하는 것들을 단순화해 수학의 언어로 바꾸어 표현하는 것을 말한다. 수학적 모델링을 통해 그 현상의 원리, 미래 추세, 그 현상에 대한 다양한 시나리오를 예측해 볼 수 있어 많은 분야에서 사용되고 있다. 예를 들어, 자연과학 분야에서는 지구 온난화의 원인과 영향을 이해하고, 미래의 기후 변화를 예측하기 위해 수학적 모델이 사용되고 또한 경제학 분야에서 시장의 동향과 경제 정책의 효과를 분석하기 위해 수학적 모델이 사용된다.

수리생물의 영역에서는 다양한 영역에서 모델을 만들어 연구를 한다. 이런 영역들은 모두 시간에 따른 현상의 변화를 연구하지만 시간과 공간 속에서의 변화를 연구하는 영역이 있으며 여기서는 편미분방정식이 수학 모델의 핵심이 된다(이원재, 2012).

3. 인구증가모델

영국의 경제학자 맬서스가 제안한 인구 증가 모델은 인구 변화량이 현재 인구 수 자체에 비례한다는 것을 바탕으로 만들어진 모델링이다. 따라서,  $y(t)$ 는 시간  $t$ 에서의 인구수,  $y(0)$ 는 현재의 인구수,  $R$ 은 인구 증가율이라고 했을 때 앞에 나온 사실

에 의해  $y'(t) = R \times y(t)$ 가 성립한다. 여기서  $y'(t)$ 는  $t$ 시간의 인구 증가율을 의미한다. 박테리아 증식 과정을 통해 이를 증명해볼 수 있다. 박테리아에게 충분한 영양분을 주고 페트리 접시에 증식한다고 했을 때, 1일 안에는 모든 박테리아가 2배로 증식한다고 한다. 박테리아가 하루 안에 증식하는데 정확히 언제 증식하는지는 알 수 없으므로 하루를 절반으로 나누어 하루의  $\frac{1}{2}$ 까지는 박테리아의 절반이 증식한다. 그러면, 하루의  $\frac{1}{2}$ 까지는 박테리아가 원래의  $\left(1 + \frac{1}{2}\right)$ 가 된다. 그런데 하루의  $\frac{1}{2}$ 이후에도 하루의  $\frac{1}{2}$ 전에 증식한  $\left(1 + \frac{1}{2}\right)$ 이 또 증식하므로 하루가 지나면 박테리아가  $\left(1 + \frac{1}{2}\right)^2$ 이 된다.

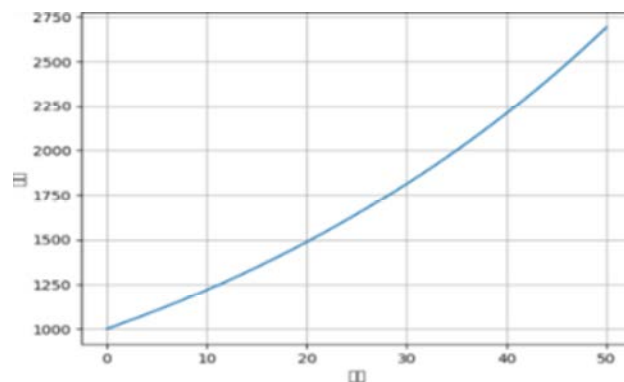
이렇게 하루를 2번으로 쪼개는 것이 아니라 3번, 4번...  $n$ 번으로 쪼개면  $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$ 가 된다.  $y(t)$ 가  $t$ (하루 단위)가 지났을 때 생기는 박테리아 양이라고 했을 때,

$$y\left(t + \frac{1}{n}\right) = \left(1 + \frac{1}{n}\right)y(t)$$

$$y\left(t + \frac{1}{n}\right) - y(t) = \frac{1}{n}y(t)$$

$$\frac{y\left(t + \frac{1}{n}\right) - y(t)}{\frac{1}{n}} = y(t)$$

$y'(t) = y(t)$ 이므로  $y(t) = y(0) \times e^t$ 라고 할 수 있다. 그리고 Python을 활용해 다음과 같이 코드를 통해 인구 증가 모델 그래프를 작성할 수 있다.



[그림 II-6] 인구증가모델 그래프

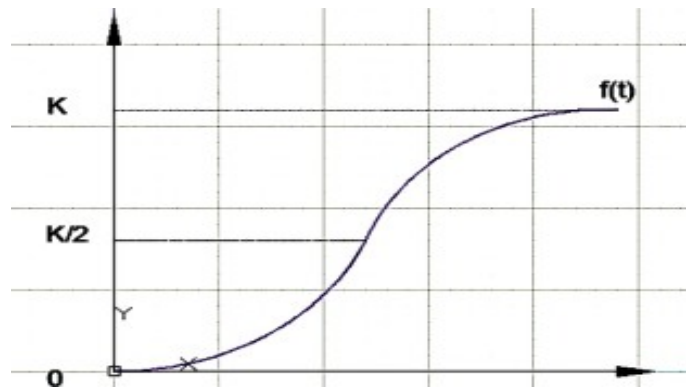
인구 증가를 수학적으로 모델링하는 것은 인구 정책 수립, 자원 배분, 경제 발전 등 다양한 분야에 활용될 수 있다. 예를 들어, 인구 증가 모델을 사용하여 인구 증가 속도를 예측할 수 있다. 이를 통해 인구 증가에 따른 사회·경제적 변화를 미리 대비할

수 있다. 하지만 맬서스의 인구 증가 모델은 한계점이 많다. 맬서스 인구 증가 모델은 여러 사회적 요인을 고려하지 못했다. 또한 맬서스 인구 증가 모델은 여러 단순화된 가정에 기반하며, 현실 세계의 복잡성을 완벽하게 반영하지 못한다는 한계가 있다.

#### 4. 로지스틱 인구 모델

다음으로 기존 인구 증가 모델에서 서로 죽이며 인구가 조절될 때를 생각해 보았다. 기존 맬서스 인구 증가 모델이 단순히 인구 증가율이 기존 인구에 비례한다고 가정했다면, 로지스틱 모델은 자원 경쟁으로 인해 인구가 조절되는 경우까지 고려한 모델링이다. 조절되는 인구수를 나타내는 항을 추가하면 다음과 같다.

$y' = ay - y^2 = y(a - by)$ 이고  $y = 0$  일 때  $y' = 0$ 임을 알 수 있다. 이계도함수도 구해보면,  $y'' = ay' - 2byy'$ 이고  $y = \frac{a}{2b}$ 일 때 변곡점임도 확인할 수 있다. 여기서  $K = \frac{a}{b}$ 로 두고 그래프 개형을 유추해보면 아래의 그림과 같다.



[그림 II-7] 로지스틱 인구모델 그래프 개형

위 그림처럼  $\frac{a}{b}$ 에는 결국 도달하지 못할 것으로 유추해볼 수 있다. 실제로  $\frac{a}{b}$ 를 넘지 못할지는 방정식을 풀어보아야 하는데 다음과 같은 과정을 통해 풀 수 있다.

$$\begin{aligned} \frac{dy}{dt} &= ay - by^2 \\ \int_0^T \frac{1}{y(a-by)} \frac{dy}{dt} \times dt &= \int_0^T 1 dt \\ \int_{y(0)}^{y(T)} \frac{1}{y(a-by)} dy &= \int_0^T 1 dt \\ \frac{1}{a} \int_{y(0)}^{y(T)} \frac{1}{y} dy + \frac{b}{a} \int_{y(0)}^{y(T)} \frac{1}{a-by} dy \\ &= \frac{1}{a} [\ln x]_{y(0)}^{y(T)} + \frac{b}{a} \left[ \left(-\frac{1}{b}\right) \ln(a-by) \right]_{y(0)}^{y(T)} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{a} \ln \frac{y(T)(a-by(0))}{y(0)(a-by(T))} \\
&\therefore \frac{y(T)(a-by(0))}{y(0)(a-by(T))} = e^{aT} \\
y(T) &= \frac{ae^{aT}y(0)}{a+b(e^{aT}-1)y(0)} = \frac{ay(0)}{by(0)+(a-by)e^{-aT}} < \frac{a}{b} \text{ 이므로 } \frac{a}{b} \text{ 를 넘지 못함을 확인할 수 있다.}
\end{aligned}$$

### III. 연구 방법

#### 1. 연구 방법

기존의 연구에서 이원재(2012)의 연구에서는 인구증가모형을 엑셀로 계산하고 시각화하였다. 이에 덧붙여 연립으로 주어진 미분방정식의 그래프를 시각화하고자 Python에서 제공하는 상미분방정식 풀이 모듈인 Scipy에서 제공하는 odeint를 활용하였다. Python을 이용하면 시간에 따른 그래프와  $x$ 와  $y$ 를 각각 축으로 하는 그래프도 그릴 수 있어 엑셀에서 제공하는 것보다 더 다양한 탐구가 가능하다. 또한 최근 생성형 인공지능을 활용하면 코딩에 대해 깊게 탐구하지 않더라도 필요한 기능을 만들어 낼 수 있는 새로운 환경이라는 점도 고려하였다.

본 연구에서는 기본 피식자-포식자 모델에 대해 다양한 조건에서 식을 찾고 Python으로 코드를 작성하여 그래프로 나타내었다. 그리고 이를 바탕으로 그래프를 해석하고 각 계수가 그래프에 주는 영향에 대해서 탐구하였다. 이후, 생물 멸종에 대한 가설을 세우고 식을 변경해가면서 포식자와 피식자의 멸종 가능성에 대해 탐구하였다.

#### 2. 연구 과정

##### 가. 기본 피식자-포식자 모델

피식자-포식자 모델의 기본 식을 구하는 과정은 다음과 같다.

$$\begin{cases} \frac{dx}{dt} = ax - bxy \\ \frac{dy}{dt} = -cy + dxy \end{cases}$$

위 식의 퀘도방정식을 풀어서  $\frac{y^a \times x^c}{e^{by} \times e^{dx}} = k$ 라는 식을 도출할 수 있었다. 또한,

$f(x) = \frac{x^c}{e^{dx}}$ 라고 하고  $g(y) = \frac{y^a}{e^{by}}$ 라고 한다면  $x$ 의 값이 작을 때는 다항함수의 증가율이 지수함수의 증가율보다 크지만  $x$ 의 값이 커짐에 따라 지수함수의 증가율이 월등히

커지기 때문에  $f(x)$ 의 최댓값을  $M_f$ 라고 하고,  $g(y)$ 의 최댓값을  $M_g$ 라고 한다면 최댓값을 기준으로 그전까지는 증가, 그 후에는 감소하는 개형을 가지게 된다.

이를 통하여 상수  $k$ 의 값과  $M_f$ 와  $M_g$ 의 곱을 비교하면 해의 개수에 기반한 그래프를 Python을 이용해 나타내었다.

나. 조건을 추가한 경우에 대한 탐구

(1) 각 개체가 만나 경쟁하는 경우의 피식자-포식자 모델

피식자-포식자 모델에서 각각의 개체가 만나 경쟁하는 모델을 다음과 같이 수식으로 표현할 수 있다.

$$\begin{cases} \frac{dx}{dt} = ax - bxy - ex^2 \\ \frac{dy}{dt} = -cy + dxy - fy^2 \end{cases}$$

여기서  $ex^2$ 와  $fy^2$ 가 각 개체가 만나 경쟁하여 죽는 경우를 의미하는 것이다. 피식자끼리 서로 만나는 경우를  $x^2$ 으로, 만나서 피식자끼리의 경쟁에서 사망할 확률을  $e$ 라고 놓으면 피식자의 개체 수는  $ax - bxy$ 에서  $ex^2$ 을 뺀 값으로 나타낼 수 있다.

또한, 포식자끼리 서로 만나는 경우를  $y^2$ 으로, 만나서 피식자끼리의 경쟁에서 사망할 확률을  $f$ 라고 놓으면 피식자의 개체 수는  $-cy + dxy$ 에서  $fy^2$ 을 뺀 값으로 나타낼 수 있고 Python을 이용해 그래프로 나타내었다.

(2) 외부요인이 추가된 피식자-포식자 모델

외부요인 중 조업 활동을 예시로 구상해보았다. 단위시간 당 조업 활동으로 잡아들이는 어종의 양은 개체 수에 비례할 것이라고 가정한다고 하고 비례상수  $\varepsilon$ 는 어종에 관계없이 동일하므로  $x$ 와  $y$ 의 증가율에 각각  $-\varepsilon x(t)$ ,  $-\varepsilon y(t)$ 를 추가로 더해주면 된다. 따라서 조업 활동을 고려한 모델은 다음과 같다.

$$\begin{cases} \frac{dx}{dt} = (a - \varepsilon)x - bxy \\ \frac{dy}{dt} = -(c + \varepsilon)y + dxy \end{cases}$$

여기서  $x$ ,  $y$ 의 평형점은 각각  $\bar{x} = \frac{c + \varepsilon}{d}$ ,  $\bar{y} = \frac{a - \varepsilon}{d}$ 가 됨을 알 수 있다. 이를 바탕으로 Python을 이용해 그래프로 나타내었다.

(3) 계수를 변경한 경우에 대한 탐구

각각의 계수  $a, b, c, d, e, f$ 에 대해  $a$ 는 피식자의 자연 증가율,  $b$ 는 포식자가 피식자를

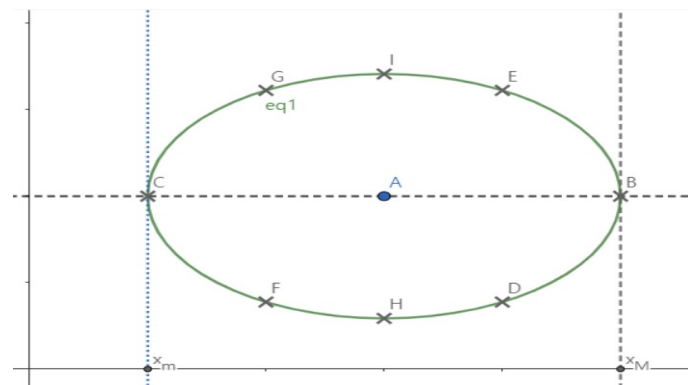
먹는 비율,  $c$ 는 포식자의 자연 감소율,  $d$ 는 포식자가 피식자를 먹어서 증가하는 비율,  $e$ 는 피식자들이 경쟁해 사망하는 비율,  $f$ 는 포식자들이 경쟁해 사망하는 비율이라는 의미를 가지고 있다. 또한 각각의 계수들은 단위시간을 변경하면 정수값으로도 표현이 가능함을 확인하였다.

계수를 변경하였을 때, 그래프에 어떠한 영향을 줄 것인가에 대해 계수를 하나씩만 변경하고 나머지는 유지하여 각각의 계수가 그래프에 어떤 변화를 주는지 확인하였다.  $a, b, c, d$ 를 각각 3, 0.5, 2, 0.2로 두고 탐구한 후  $c$ 를 2에서 4로 변경하여 다시 탐구하였다.

#### IV. 연구 결과

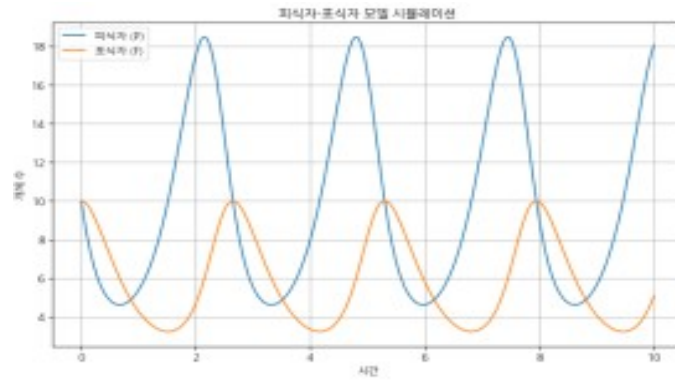
##### 1. 기본 피식자-포식자 모델

$k > M_f \times M_g$ 일 때는 해가 존재하지 않고,  $k = M_f \times M_g$ 일 때는 평형점이 유일한 해가 되며  $k < M_f \times M_g$ 일 때는 그림 IV-1과 같은 궤도가 나올 것이다.



[그림 IV-1] 기본 피식자-포식자 모델 그래프

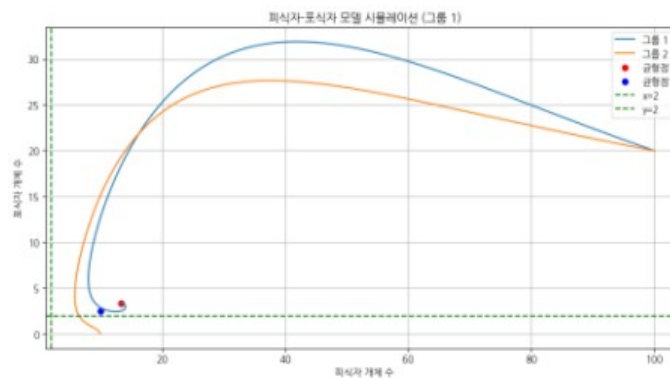
그림 IV-1는 피식자-포식자 모델의 기본 식을 그래프로 만든 것이며, 이를 통해 궤도를 그려내는 식을 프로그래밍하여 궤도를 그리면 위와 같은 그래프를 얻을 수 있다. 그림 IV-2는 시간의 흐름에 따른 피식자와 포식자의 개체 수를 각각 표현한 것이며 이를 통해 일정한 주기로 반복하여 개체 수가 늘어나고 줄어든다는 것을 확인할 수 있다.



[그림 IV-2] 기본 피식자-포식자 모델 시뮬레이션

## 2. 각 개체가 만나 경쟁을 하는 경우

초기 피식자 개체 수를 100마리, 초기 포식자 개체 수를 20마리로 두 그룹 모두 초깃값을 같게 설정하고 그룹2는 피식자 간 경쟁을 2배로 변경하여 다음과 같이 그래프를 작성했다. 또한, 이를 Python 코드를 활용해 그림으로 나타내면 다음과 같다.



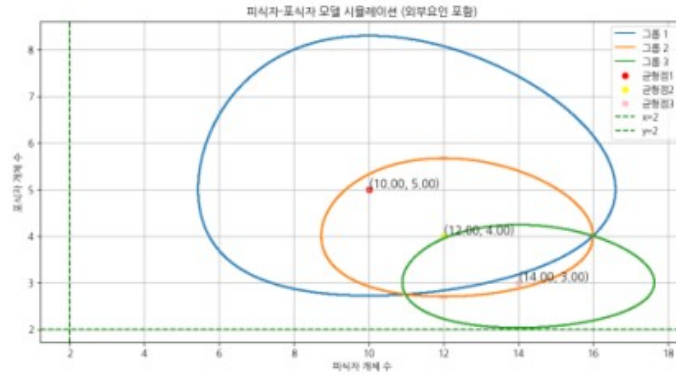
[그림 IV-3] 각 개체가 만나서 경쟁하는 경우의 피식자-포식자 모델 그래프

본 연구에서는 한 종의 개체가 2마리 이하일 때는 번식이 불가능하다고 판단하여 녹색 점선으로 표기하였다.

두 그래프를 통해 피식자 간 경쟁을 2배로 변경해 피식자가 위험에 빠지기 쉬웠음에도 오히려 포식자가 먼저 멸종하였다는 것을 알 수 있다. 여기서 피식자를 초식 동물, 포식자를 육식 동물이라 한다면 육식 동물이 초식 동물보다 멸종할 가능성이 높다는 것을 알 수 있다.

### 3. 외부요인을 추가한 경우

피식자 16마리, 포식자 4마리라는 같은 초깃값에서 외부요인이 없는 경우, 외부요인이 1인 경우, 외부요인이 2인 경우를 각각 그래프로 표현했다.



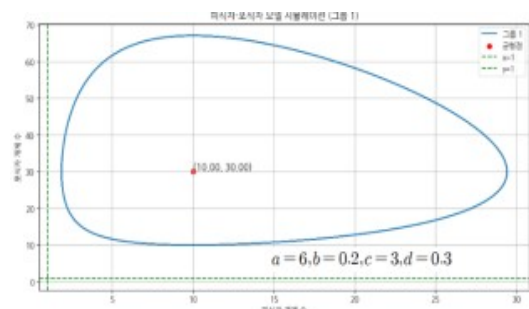
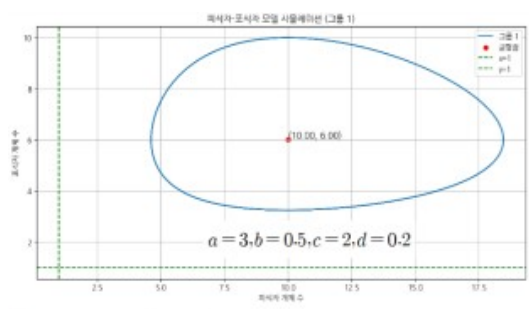
[그림 IV-4] 외부요인을 추가한 경우의 피식자-포식자 모델 그래프

위의 그래프에서 평형점이 포식자는 5마리, 4마리, 3마리로 감소하는 방향으로 이동하지만 피식자는 10마리, 12마리, 14마리로 늘어나는 방향으로 이동하고 있다. 이 결과는 외부요인이 포식자보다 피식자에게 더 유리함을 보여준다. 이를 바꾸어 말하면 외부요인이 평소보다 영향이 늘어나면 피식자보다 포식자가 멸종할 가능성이 커진다고 볼 수 있다. 또한, 최근 환경오염, 온난화와 같이 종에 무관하게 작용하는 외부요인도 조업 활동과 동일하게 적용할 수 있다고 판단하여, 기후 위기에 따라 포식자가 피식자보다 멸종할 가능성이 높다는 추론을 할 수 있었다.

### 4. 계수를 변경한 경우

#### (1) a, b, c, d 계수를 모두 변경한 경우

아래에서 왼쪽의 그림 IV-5는  $a=3, b=0.5, c=2, d=0.2$ 로 설정했을 때의 궤도이고 오른쪽은 그림 IV-6은  $a=6, b=0.2, c=3, d=0.3$ 으로 설정했을 때의 궤도이다.

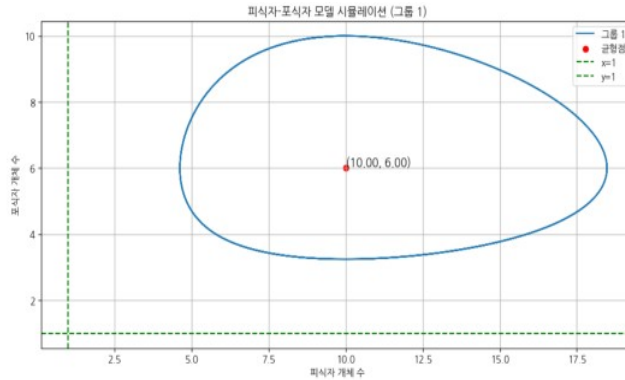


[그림 IV-5, 6] a,b,c,d 계수를 변경하기 전과 후의 그래프

이러한 그래프를 통해서  $a, b, c, d$ 의 값을 바꾸었을 때 평형점이 이동하고 그래프의 개형이 달라짐을 확인할 수 있었다.

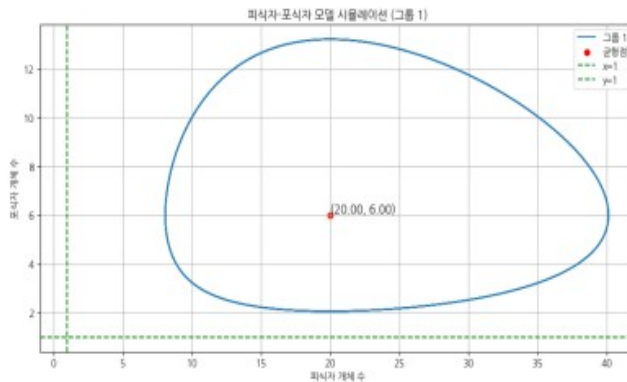
(2)  $c$  계수만 변경한 경우

그림 IV-7은  $c$  계수를 변경하기 전에 계수  $a, b, c, d$ 를 각각 3, 0.5, 2, 0.2로 지정하여 그린 것이다.



[그림 IV-7]  $c$  계수를 변경하기 전 피식자-포식자 모델 그래프

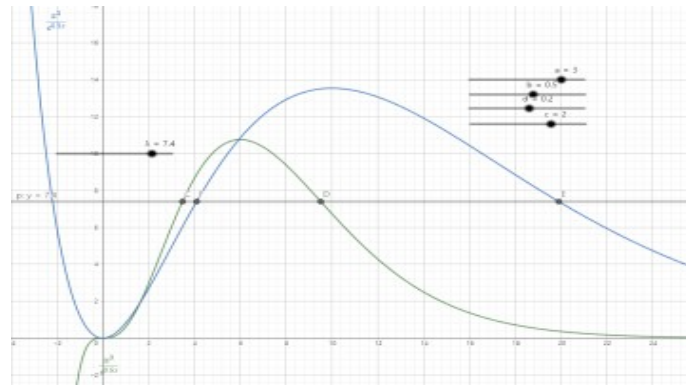
위 그래프의 평형점은 (10,6)이고 곡선형 궤도가 나타나는 것을 알 수 있다. 그림 IV-8은 계수  $a, b, c, d$  중  $c$ 를 2에서 4로 변경하여 그린 것이다.



[그림 IV-8]  $c$  계수를 변경한 후 피식자-포식자 모델 그래프

평형점은 (20,6)이고 곡선형 궤도가 나타났다.  $c$ 가 포식자의 자연 감소율이어서  $c$ 가 증가하면 포식자의 수가 감소하고 피식자의 개체 수가 위의 그래프보다 늘어난 것을 알 수 있다. 하지만 포식자의 자연 감소율이 증가했는데 포식자의 최대 개체 수가 늘어남을 확인할 수 있었다. 포식자의 최대 개체 수가 늘어남에 의문을 가지고 추가적인 탐구를 진행하였다.

$k < M_f \times M_g$  일 때  $M_f$ 보다 작은  $f(x)$ 의 값을  $\lambda$ 라고 하였을 때, 초기 개체 수인  $x, y$  모두 10으로 동일하게 설정하였다.  $f(x)g(y) = \lambda M_g$ 이고  $a, b, d$ 의 값은 3, 0.5, 0.2로 고정하였기 때문에  $M_g = g(6)$ 이고  $\lambda = \frac{f(10)g(10)}{g(6)}$ 이며  $\lambda$ 의 값이 같다고 가정했을 때, 그림 IV-9와 같이,  $c$ 의 값이 2배가 된 그래프의  $x$ 값의 간격이 넓은 것을 알 수 있다.



[그림 IV-9]  $\lambda$ 이 동일한 경우 각각의 그래프의 폭

따라서, 좌우 폭이 커지게 되는 것을 알 수 있다.

$k < M_f \times M_g$  일 때  $M_g$ 보다 작은  $g(y)$ 의 값을  $\lambda$ 라고 하자. 초기 개체 수인  $x, y$  모두 10으로 같게 설정하였고  $a, b, d$ 의 값은 3, 0.5, 0.2로 고정하였다.

$f(x)g(y) = \lambda M_f$  이라고 하였을 때,  $\lambda = \frac{f(10)g(10)}{M_f}$  이다.  $c$ 가 2일 때,

$\lambda_2 = \frac{f(10)g(10)}{f(10)} = g(10)$  이고  $c=4$ 일 때  $\lambda_4 = \frac{f(10)g(10)}{f(20)} < g(10) = \lambda_2$ 이므로,  $\lambda_2$ 가  $\lambda_4$ 보다

크다. 그리고 이를 그래프로 표현하면 그림 IV-9와 같다.

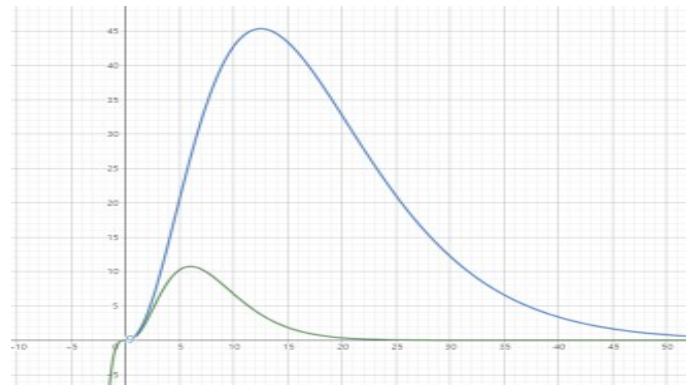


[그림 IV-9]  $\lambda_2$ 가  $\lambda_4$ 보다 큰 그래프

따라서, 궤도의 상하 폭이 커지게 되는 것을 알 수 있다.

우리가 구한 그래프가 상하 폭이 커지는 것이 옳다는 것을 확인한 후, 그 의미를 해석하기 위해 논의를 진행하였다. 논의 결과, 포식자의 평균인 균형점의 식이  $y = \frac{a}{b}$ 라는 것을 고려할 때, 포식자의 변화율에 관련 식의 계수는 포식자의 평균에 영향을 주지 못하고, 피식자의 평균  $x = \frac{c}{d}$ 에 영향을 준다. 따라서  $c$ (포식자의 자연 감소율)의 증가는 포식자가 일정한 상황에서 피식자 개체 수 평균의 증가를 가져온다고 볼 수 있다. 따라서 피식자가 최대치일 때에서 출발한다고 가정하면, 포식자의 입장에서는 포식자 대비 피식자가 늘어나 먹이가 풍부해져서 더 많이 증가할 수 있는 환경이 제공된다고 해석할 수 있다. 시간의 흐름에 따라 포식자가 최대가 되는 순간에서는 포식자가 지나치게 늘어나 더 급격히 감소하는 그래프를 보인다고 해석할 수 있었다.

추가로 피식자-포식자 모델의 그래프가 완벽한 타원형이 아닌 이유도 설명할 수 있게 되었다. 아래 그림과 같이  $f(x)$ ,  $g(y)$ 의 그래프가 완벽한 대칭이 아니기 때문에 피식자-포식자 모델의 그래프는 타원형이 될 수 없다.



[그림 IV-10]  $f(x)$ 와  $g(y)$ 의 그래프

마지막으로 계수  $c$ 뿐만 아니라  $a$ ,  $b$ ,  $d$ 의 값을 변화하였을 때 이와 같은 원리로 평형점, 그래프의 개형, 그래프의 상하좌우 폭이 달라질 수 있음을 알 수 있다.

## V. 결론 및 제언

수학적 모델링을 통해 사회현상, 자연현상을 단순화하여 보여줄 수 있고 미래나 어떤 조건을 바꾸었을 때의 결과를 예측할 수 있다. 복잡한 식을 Python을 이용하여 그래프를 그린 결과를 시각적으로 확인해봄으로써 직접 계산하지 않고도 그래프나 결과값을 쉽게 확인하고 다음과 같이 해석하여 결론을 내렸다.

첫째, 각 개체가 만나 경쟁을 하는 경우에는 포식자 또는 피식자가 죽을 확률에 따라 그래프의 모양이 변하고 오히려 포식자가 멸종할 가능성이 높다고 결론을 내렸다.

둘째, 조업 활동 및 환경오염과 같은 외부요인을 추가함으로써, 이러한 외부요인들이 실제로 생태계에 영향을 줄 수 있으며 멸종까지도 가능할 것이라는 결론을 내렸다.

이 두 사례를 통해 기본적인 모델링에서 변수를 추가해 만든다면 현실과 가까운 결과들을 도출하고 기본적인 모델링에 비해 좀 더 정교한 예측을 할 수 있다는 것을 알 수 있었고 이를 통해 수학적 모델링과 같은 수학의 유용성을 알 수 있었다.

셋째, 모델링에서 사용되는 변수와 각각의 계수들이 가지고 있는 의미를 그래프를 이용해 시각적으로 확인하고 해석함으로써 본 연구의 모델링한 식을 검증할 수 있었다. 또한, 수학적 검증을 통해 그래프가 바뀌는 원인을 이해하고 그래프를 정성적으로 재해석했을 때, 더 풍부한 논의가 가능함을 알 수 있었다.

하지만 2020년에 진행된 미국 유타주립대학의 연구는 초식 동물의 멸종 위험이 육식 동물보다 더 높다는 결과를 보여주었다. 우리 팀이 수학적 모델을 사용하여 탐구한 내용과는 상반된 결과가 도출된 것으로 볼 수 있다. 해당 연구에서는 초식 동물 중 25.5%가 멸종 위험에 처해 있다는 결과가 도출되었으며, 이는 육식 동물보다 높은 비율임을 나타내었다. 이러한 한계점을 통해 수학적 모델링의 한계와 현실 세계의 복잡성을 더 많이 고려할 필요가 있다는 것을 알 수 있다.

따라서 후속 연구자들은 이보다 더 많은 변수를 고려한 정교한 수학적 모델링을 통해 실생활과 가깝게 수학적 모델링을 하는 노력을 기울일 것을 제안하는 바이다.

## VI. 참고문헌

- 류희찬, 선우하식, 조정목, 이병만, 김용식, 임미선, 한명주, 남선주, 김명수, 정성윤(2019). *고등학교 미적분 교과서*. 천재 교과서 출판
- 에자키 타카히로(2020). 데이터 분석을 위한 수리모델 입문(김범준 역). 위키북스
- 이원재. (2012). 수리생물의 모델과 학교수업에서의 활용 방안 (국내석사학위논문).
- Hoppensteadt,F.(2006). *Predator-prey model*. Scholarpedia.1(10):1563. doi:10.4249/scholarpedia.1563
- Martin Braun(1992). *Differential equations and their applications(4th Edition)*. Springer.
- TB Atwood et al(2020). *Herbivores at the highest risk of extinction among mammals, birds, and reptiles*. Science advances 6(32). doi: 10.1126/sciadv.abb8458



수학B

12\_자연수  $n$ 의  $m$ -regular composition 개수와  
 $m$ -bonacci 수열



# 자연수 $n$ 의 $m$ -regular composition 개수와 $m$ -bonacci 수열

박지유 (목일중학교 3학년)

박찬욱 (서울과학고등학교 1학년)

지도교수 : 이준복 (연세대학교 수학과)

## 논문 초록(abstract)

본 논문은 모든  $m$ 에 대해  $m$ -regular composition의 개수가  $U_n$ 임을 밝히고 일반화하였다.  $U_n$ 은  $m$ -bonacci 수열 중 초깃값이 2의 거듭제곱 꼴인 수열이다.

자연수  $n$ 을 양의 정수의 합으로 나타내는 방법(compositions) 중에서 표현된 양의 정수가  $m$ 으로 나누어지지 않을 때의 composition을  $m$ -regular composition이라고 한다.  $m=2$ 일 때 2-regular composition의 개수는 피보나치수열의  $n$ 번째 항의 값이라는 것은 논문[1]에서 MacMahon bit sequence를 이용해 증명되었다.

논문[1]에서는  $m=3$ 일 때도 3-regular composition의 개수가 Tribonacci 수열의  $n$ 번째 항의 값과 같다는 것을 보인다. 증명은 Tribonacci identity와 MacMahon bit sequence를 이용하여 이루어진다. 논문[1]의 conjecture에서는 모든  $m$ 에 대하여  $m$ -regular composition의 개수가  $m$ -bonacci 수열의  $n$ 번째 항의 값이라는 가정을 제시한다.

본 논문에서는  $m$ 의 값에 따라  $T_n$ 과  $U_n$ 을 설정하였고  $m=4$ 일 때 4-regular compositions의 개수가 Tetranacci 수열의  $n$ 번째 값과 같은지 확인하였다. 그리고 확인한 사실을 기반으로 모든  $m$ 에서 conjecture가 성립함을 보였다.

## I. 서론 (또는 연구의 필요성 및 목적)

임의의 자연수  $n$ 을 자연수 몇 개로 적절히 분할하는 경우의 수는  $2^{n-1}$ 로 잘 알려져 있다.  $n$ 을 홀수들의 합으로 분할하는 경우의 수는  $F_n$  (피보나치 수)임도 쉽게 알 수 있다. 또한 선행 연구는  $n$ 을 3의 배수가 아닌 자연수들의 합으로 분할하는 경우의 수가 Tribonacci 수열 ( $a_n = a_{n-1} + a_{n-2} + a_{n-3}$ 을 점화식으로 가지는 수열)임을 귀납법으로 증명하였다. 어떤 자연수  $n$ 의 composition을 MacMahon bit sequence로 바꾸어 0의 개수가  $3k$  또는  $3k+1$  꼴의 두 가지 경우로 나누었다. 그리고 Tribonacci 수열  $T_n$ 과  $U_n$ 의 점화식과 Tribonacci identity는 다음과 같다.

$$T_n = T_{n-1} + T_{n-2} + T_{n-3}, T_0 = 0, T_1 = 1, T_2 = 1$$

$$U_n = U_{n-1} + U_{n-2} + U_{n-3}, U_0 = 0, U_1 = 1, U_2 = 2$$

여기서 우리는 다음과 같은 4개의 항등식을 얻을 수 있다.

$$(1) \sum_{k=0}^{\lfloor \frac{n}{3} \rfloor} U_{n-3k} = T_{n+1} \text{ if } n \equiv 1, 2 \pmod{3}$$

$$(2) 1 + \sum_{k=0}^{\lfloor \frac{n}{3} \rfloor} U_{n-3k} = T_{n+1} \text{ if } n \equiv 0 \pmod{3}$$

$$(3) U_{n-1} + U_{n-2} = T_n + T_{n-2} \text{ for } n \geq 2$$

$$(4) T_{n+1} - T_{n-2} = U_n \text{ for } n \geq 2$$

n	0	1	2	3	4	5	6	7	8	9	10
$T_n$	0	1	1	2	4	7	13	24	44	81	149
$U_n$	0	1	2	3	6	11	20	37	68	125	230

선행논문[1]에서 이러한 identities와 MacMahon sequence를 적용하여 3-regular composition의 개수가 Tribonacci 수열  $U$ 의  $n$ 번째 항의 값이라는 것을 귀납적으로 증명하였다.

자연스럽게, ‘임의의 자연수  $n$ 을  $m$ 의 배수가 아닌 자연수의 합으로 분할하는 경우의 수는 몇 개일까?’라는 생각을 할 수 있고 이는 논문[1]에서 다음과 같이 conjecture로 제시되어 있다.

정수  $m$ 과  $n$ 이  $2 \leq m < n$ 일 때,  $c(n, m)$ 은  $n$ 의  $m$ -regular composition의 개수라고 하자. 어떠한  $m$ 에 대하여

$$a(n, m) = \sum_{i=1}^m a(n-i, m) \text{ for } n \geq m$$

초깃값은  $a(0, m) = 0, a(j, m) = 2^{j-1}$  for  $1 \leq j \leq m-2, a(m-1, m) = 2^{m-2} - 1$ 이다.  
 그러면  $c(n, m) = a(n, m)$ 이다.

본 논문에서는  $U_n$ 과  $T_n$ 의 관계식을 이용하여 conjecture를 증명한다.

## II. 이론적 배경

음이 아닌 정수  $i$ 에 대해,  $F_i$ 을 피보나치수열이라고 하며 이 수열은  $F_0 = 0, F_1 = 1$ 이고 관계식  $F_{i+2} = F_{i+1} + F_i$ 을 만족한다고 하자. 이 피보나치수열 이외에 Tribonacci 수열, Tetranacci 수열 등  $m$ -bonacci 수열도 있다.

$m$ -bonacci 수열에서는  $T_m = T_{m-1} + T_{m-2} + \dots + T_{m-n+1}$ 이 성립한다.  $m$ -bonacci수열은 초깃값에 따라 수열의 항의 값이 달라진다. 일반적으로  $m$ -bonacci수열은 1, 1, 2로 시작하여  $m$ 번째 항의 값을 정의할 수 있을 때까지 2의 거듭제곱 꼴로 나타난다. 자연수  $n$ 을 양의 정수들의 합으로 나타낸 것의 집합에서 어떤 양의 정수도 짝수가 아닌 composition의 개수는 피보나치수열의  $n$ 번째 항인  $F_n$ 임을 알 수 있고 어떤 양의 정수도 3의 배수가 아닐 때의 composition의 개수는 트리보나치수열의 종류 중 하나인 수열  $U$ 의  $n$ 번째 항인  $U_n$ 임을 알 수 있다. composition의 모든 양의 정수가  $m$ 으로 나누어지지 않을 때 그 composition은  $m$ -regular라고 한다.

논문[2]에 소개된 MacMahon bit sequence는 어떤 자연수  $n$ 을 양의 정수의 합으로 나타낸 composition을 MacMahon graph로 나타낸 뒤에 0과 1로 변환하여 composition을 나타낸 bit sequence이다. 예를 들어, 13을 2+4+1+1+5로 나타낼 수 있다. 이것을 MacMahon graph로 바꾸면  $-- \cdot ----- \cdot - \cdot - \cdot -----$ 로 나타내어진다. 이 graph를 bit sequence로 바꾸려면 -오른쪽에 점이 없으면 0으로, 있으면 1로 나타낸다.

따라서  $-- \cdot ----- \cdot - \cdot - \cdot -----$ 의 MacMahon bit sequence는 01 0001 1 1 0000이 된다. 여기에서 한 가지 알 수 있는 사실이 있는데,  $n$ 의 composition을 MacMahon bit sequence는  $n-1$ 개의 0과 1로 이루어진다는 사실을 알 수 있다.

다음은 트리보나치 수열  $T_n$ 과  $U_n$ 의 관계식이다.

- (1)  $U_n = T_n + T_{n-1}$
- (2)  $\sum_{k=0}^{\lfloor \frac{n}{3} \rfloor} U_{n-3k} = T_{n+1}$  if  $n \equiv 1, 2 \pmod{3}$

$$(3) 1 + \sum_{k=0}^{\lfloor \frac{n}{3} \rfloor} U_{n-3k} = T_{n+1} \text{ if } n \equiv 0 \pmod{3}$$

$$(4) U_{n-1} + U_{n-2} = T_n + T_{n-2} \text{ for } n \geq 2$$

$$(5) T_{n+1} - T_{n-2} = U_n \text{ for } n \geq 2$$

이 identities는 귀납법으로 쉽게 증명된다.

논문[1]에서는 이러한 identity를 이용하여 3-regular composition의 개수가  $U_n$ 임을 증명하였다. 본 논문에서 4-regular에 해당하는 identity를 적용하여 증명한 후 일반적인 identity를 도출한다. 논문[1]에서 제시한 conjecture를 보이기 위해 임의의 자연수  $n$ 에 대한  $m$ -regular composition의 개수가  $m$ -bonacci 수열  $U$ 의  $n$ 번째 항임을 증명할 것이다.

또한  $m$ -regular composition이 되려면 composition을 MacMahon sequence로 바꾸었을 때 0의 길이  $\equiv n-1 \pmod{n}$ 이 되어야 한다.

### III. 연구방법

논문[1]에서 3-regular composition을 증명할 때 사용한 식을 4-regular에 맞게 변형한 뒤 4-regular에서도 성립함을 증명하였다. 이후 모든  $n$ 에서 식이 성립하도록 하는 규칙을 찾아 식을 일반화하였다. 그리고 모든  $n$ 에서  $n$ -regular composition의 개수가  $n$ -bonacci 수열의  $n$ 번째 항이라는 것을 일반화하였다.

**정리 3.1** 자연수  $n$ 을 홀수인 정수들로 나누는 composition의 개수는 피보나치 수열  $F$ 의  $n$ 번째 항이다.

**증명.** 귀납법으로 증명할 때 정리 1이  $n-1$ 일 때 성립한다고 하고 composition을 MacMahon bit sequence로 바꾸었을 때 1로 끝나는 sequence의 개수와 0으로 끝나는 sequence의 개수를 각각 구해 더하면  $F_n$ 이 나온다.

**정리 3.2.** 자연수  $n$ 의 composition 중 어떤 정수도 3으로 나누어지지 않을 때의 composition의 개수는 Tribonacci 수열  $U$ 의  $n$ 번째 항이다.

**증명.** composition을 MacMahon bit sequence로 바꾸었을 때 0의 개수가  $3k$  또는

$3k+1$ 의 꼴에서의 개수를 귀납법으로 증명하여 구한 뒤 더하면  $U_n$ 이 나온다.

**정리 3.3.** 자연수  $n$ 의 4-regular compositions의 개수는 Tetranacci 수열  $U$ 의  $n$ 번째 항이다.

**증명.** 정리 2와 같은 방법으로 하였을 때 0의 개수가  $4k, 4k+1, 4k+2$ 일 때의 개수를 다 더하면  $U_n$ 이다.

4-regular에서는 Tetranacci numbers를 이용한다.

다음은 4-regular 증명에 이용될 두 종류의 Tetranacci numbers이다.

n	0	1	2	3	4	5	6	7	8	9	10
$T_n$	0	1	1	2	4	8	15	29	56	108	208
$U_n$	0	1	2	4	7	14	27	52	100	193	372

$$T_n = T_{n-4} + T_{n-3} + T_{n-2} + T_{n-1}, \quad T_0 = 0, T_1 = 1, T_2 = 1, T_3 = 2$$

$$U_n = U_{n-4} + U_{n-3} + U_{n-2} + U_{n-1}, \quad U_0 = 0, U_1 = 1, U_2 = 2, U_3 = 4$$

앞에서 Tribonacci identity를 Tetranacci에 맞게 적용하여 Tetranacci identity를 얻었다.

$$(1) U_n = T_{n-2} + T_{n-1} + T_n$$

$$(2) U_{n-1} + U_{n-2} + U_{n-3} = T_n + T_{n-1} + T_{n-3} \quad \text{for } n \geq 3$$

$$(3) U_n = T_{n+1} - T_{n-3} \quad \text{for } n \geq 3$$

$$(4) \sum_{k=0}^{\lfloor \frac{n}{4} \rfloor} U_{n-4k} = T_{n+1} \quad \text{if } n \equiv 1, 2, 3 \pmod{4}$$

$$(5) 1 + \sum_{k=0}^{\lfloor \frac{n}{4} \rfloor} U_{n-4k} = T_{n+1} \quad \text{if } n \equiv 0 \pmod{4}$$

**보조정리 3.4.** Tribonacci sequence인  $U_n$ 과  $T_n$ 에 대하여

$$U_n = T_{n-2} + T_{n-1} + T_n$$

**증명.**

$n=1$ 일 때,  $U_2 = 2, T_2 + T_1 + T_0 = 2$ 이므로  $U_2 = T_2 + T_1 + T_0$ 이다.

$n=m-4$ 일 때,  $U_{m-4} = T_{m-4} + T_{m-5} + T_{m-6}$

$n=m-3$ 일 때,  $U_{m-3} = T_{m-3} + T_{m-4} + T_{m-5}$

$n=m-2$ 일 때,  $U_{m-2} = T_{m-2} + T_{m-3} + T_{m-4}$

$n = m - 1$  일 때,  $U_{m-1} = T_{m-1} + T_{m-2} + T_{m-3}$

$n$ 이 각각  $m-4, m-3, m-2, m-1$  일 때의  $U_n$  과  $T_n$  의 관계식을 모두 더하면

$$\begin{aligned} & U_{m-1} + U_{m-2} + U_{m-3} + U_{m-4} \\ &= (T_{m-1} + T_{m-2} + T_{m-3} + T_{m-4}) + (T_{m-2} + T_{m-3} + T_{m-4} + T_{m-5}) + (T_{m-3} + T_{m-4} + T_{m-5} + T_{m-6}) \end{aligned}$$

이므로

$$U_m = T_m + T_{m-1} + T_{m-2} \text{ 이다.}$$

**보조정리 3.5.**  $U_{n-1} + U_{n-2} + U_{n-3} = T_n + T_{n-1} + T_{n-3}$  for  $n \geq 3$

$$\begin{aligned} \text{증명. } & U_{n-1} = T_{n-1} + T_{n-2} + T_{n-3}, \\ & U_{n-2} = T_{n-2} + T_{n-3} + T_{n-4} \\ & U_{n-3} = T_{n-3} + T_{n-4} + T_{n-5} \end{aligned}$$

$$\therefore U_{n-1} + U_{n-2} + U_{n-3} = T_n + T_{n-1} + T_{n-3}$$

**보조정리 3.6**  $T_{n+1} - T_{n-3} = U_n$  for  $n \geq 3$

증명.

$$\begin{aligned} T_{n+1} - T_{n-3} &= T_n + T_{n+1} + T_{n-1} - T_n - T_{n-1} - T_{n-3} \\ &= U_{n+1} - (U_{n-1} + U_{n-2} + U_{n-3}) \\ &= U_n \end{aligned}$$

**보조정리 3.7.**  $\sum_{k=0}^{\lfloor \frac{n}{4} \rfloor} U_{n-4k} = T_{n+1}$  if  $n \equiv 1, 2, 3 \pmod{4}$

증명.

①  $n = 1$  일 때

$$\sum_{k=0}^{\lfloor \frac{1}{4} \rfloor} U_{1-4k} = U_1 = 1 = T_2$$

②  $n = 4m + 1$  일 때

$$\sum_{k=0}^{\lfloor \frac{4m+1}{4} \rfloor} U_{4m+1-4k} = U_{4m+1} + U_{4m-2} + \dots + U_1 = T_{4m+2} \text{ 임이 성립한다고 하자.}$$

③  $n = 4(m+1) + 1 = 4m + 5$  일 때

$$\begin{aligned} \sum_{k=0}^{\lfloor \frac{4m+5}{4} \rfloor} U_{4m+5-4k} &= U_{4m+5} + U_{4m+1} + \dots + U_1 = T_{4m+2} + U_{4m+5} \\ &= T_{4m+2} + T_{4m+3} + T_{4m+4} \\ &= T_{4m+5} \end{aligned}$$

임으로써 증명이 완료되었다.

**보조정리 3.8**  $1 + \sum_{k=0}^{\lfloor \frac{n}{4} \rfloor} U_{n-4k} = T_{n+1}$  if  $n \equiv 0 \pmod{4}$

증명. 귀납법으로 보조정리 1.4.와 같은 방식으로 증명이 된다.

**정리 3.3.**  $n$ 이 자연수라고 할 때,  $f(n)$ 은  $n$ 의 4-regular composition의 개수라고 하면  $f(n) = U_n$ 이다.

**증명.** 귀납법을 이용하여 증명하도록 하자. 이 정리는  $1 \leq n \leq 4$ 에서 성립한다. 논문 [1]에서와 같이 우리는 이 정리를 composition의 MacMahon bit sequence를 이용하여 증명할 것이다. 우선  $n+1$ 의 composition은  $n$ -bit sequence에 대응함을 알아두자. 그리고 composition의 어느 양의 정수도 4로 나누어지지 않는다는 것은 bit sequence의 어느 0들의 개수도 4로 나누었을 때 나머지가 3이 아니라는 의미이다. 이렇게 조건을 만족하는 bit sequence를 good bit sequence라고 하자. 여기서 good  $n$ -bit sequenced의 개수가  $U_{n+1}$ 을 보이도록 하자.

귀납법에 따라  $n-1$ 의 길이인 good bit sequence의 개수가  $U_n$ 이라고 하자. 이  $n-1$  good bit sequence에 가장 오른쪽에 1을 추가하면  $n$ -good bit sequence가 생성되고 개수는 마찬가지로  $U_n$ 이 될 것이다. 반대로 1로 끝나는  $n$ -good bit sequence에서 끝의 1을 빼면  $n-1$ -good bit sequence가 생성될 것이다. 이처럼 1로 끝나는  $n$ -good bit sequence와  $n-1$ -good bit sequence 사이에 일대일대응이 성립하므로 길이  $n$ 의 1로 끝나는 good bit sequence의 개수는  $U_n$ 이다.

그다음, 0으로 끝나는  $n$ -good bit sequence의 개수를 생각해야 한다. 0의 길이  $\equiv 3 \pmod{4}$ 임을 충족해야 하므로 0의 길이가  $4k, 4k+1, 4k+2$ 인 경우로 나누어서 증명하도록 하겠다. 증명과정에서 Tetranacci identity가 이용된다.

0의 길이가  $4k+1$ 이면 bit sequence의 개수는  $U_{n-4k-1}$ 이 될 것이다.  $n \not\equiv 1 \pmod{4}$ 일 때 identity (2)를 이용하고  $n$ 위치에  $n-1$ 를 대입하면 이 sequence의 개수는

$$\sum_{k=0}^{\lfloor \frac{n-1}{4} \rfloor} U_{n-1-4k} = T_n.$$

$n \equiv 1 \pmod{4}$ 일 때 identity (3)에  $n-1$ 을 대입해보자. 그러면 이 sequence의 개수는

$$1 + \sum_{k=0}^{\lfloor \frac{n-1}{4} \rfloor} U_{n-1-4k} = T_n.$$

이렇게 0의 길이가  $4k+1$ 인  $n$ -good bit sequence의 개수는  $T_n$ 이다.

다음으로, 0의 길이가  $4k+2$ 일 경우를 살펴보자. 이때의 bit sequence의 개수는  $U_{n-4k-2}$ 이므로  $n \not\equiv 2 \pmod{4}$ 일 때 identity (2)에  $n-2$ 를 대입하자. 그러면 sequence의 개수는

$$\sum_{k=0}^{\lfloor \frac{n-2}{4} \rfloor} U_{n-2-4k} = T_{n-1}.$$

$n \equiv 2 \pmod{4}$ 일 때 identity (3)에  $n-2$ 를 대입하여 sequence의 개수를 구하면

$$1 + \sum_{k=0}^{\lfloor \frac{n-2}{4} \rfloor} U_{n-2-4k} = T_{n-1}.$$

따라서 0의 길이가  $4k-2$ 인  $n$ -good bit sequence의 개수는  $T_{n-1}$ 이다.

마지막으로 0의 길이가  $4k$ 일 때를 생각해 보자. 이때 bit sequence의 개수는  $U_{n-4k}$ 이다.  $n \not\equiv 0 \pmod{4}$ 일 때 identity (2)에 따라서 sequence의 개수는

$$\sum_{k=1}^{\lfloor \frac{n}{4} \rfloor} U_{n-4k} = T_{n+1} - U_n.$$

$n \equiv 0 \pmod{4}$ 일 때 identity (3)에 따르면 sequence의 개수는

$$1 + \sum_{k=1}^{\lfloor \frac{n}{4} \rfloor} U_{n-4k} = T_{n+1} - U_n.$$

따라서 0의 길이가  $4k$ 일 때  $n$ -good bit sequence의 개수는  $T_{n+1} - U_n$ 이다.

결론적으로  $n$ -good bit sequence의 개수는

$$U_n + T_n + T_{n-1} + T_{n+1} - U_n = U_{n+1} \text{이다.}$$

#### IV. 연구결과

논문[1]에는 다음과 같은 일반화가 제시되어 있었다.

**conjecture 4.1**  $a(n, m) = \sum_{i=1}^m a(n-i, m)$  for  $n \geq m$

$$a(0, m) = 0, a(j, m) = 2^{j-1} \text{ for } 1 \leq j \leq m-2, a(m-1, m) = 2^{m-2} - 1$$

$$\text{Then } c(n, m) = a(n, m)$$

$c(n, m)$ 은  $m$ -regular composition의 개수이다. 이때  $m$ -bonacci 수열에서  $m$ -regular composition의 개수가  $c(n, m)$ 을 의미하는  $U_n$ 임을 증명하면 된다.

앞서 보인 자연수  $n$ 에 대하여 3-regular compositions와 4-regular compositions의 개수가 각각 Tribonacci, Tetranacci 수열인  $U_n$ 이라는 점을 생각하여 자연수  $n$ 에 대하여  $m$ -regular compositions의 개수를 구할 수 있었다.

$n$ 의  $m$ -regular composition의 개수를 구하자.

#### 정의 4.2

수열  $U$ 를 다음과 같이 정의하자.

$$U_0 = 0, U_n = 2^{n-1} (1 \leq n \leq m-1)$$

$$U_n = \sum_{i=1}^m U_{n-i} (n \geq m)$$

### 정의 4.3

수열 T를 다음과 같이 정의하자.

$$T_n = n (n=0,1), T_n = 2^{n-2} (2 \leq n \leq m-1)$$

$$T_n = \sum_{i=1}^m T_{n-i} (n \geq m)$$

### 보조정리 4.4

$$T_{n+2} - T_{n+1} = U_n$$

pf) n에 대한 수학적 귀납법으로 증명하자.

i)  $0 \leq n \leq m-1$  일 때

a.  $n=0$

$$T_2 - T_1 = 1 - 1 = 0 = U_0$$

b.  $1 \leq n \leq m-3$

$$T_{n+2} - T_{n+1} = 2^n - 2^{n-1} = 2^{n-1} = U_n$$

c.  $n = m-2$

$$\begin{aligned} T_m - T_{m-1} &= (T_0 + T_1 + T_2 + \dots + T_{m-1}) - T_{m-1} \\ &= T_0 + T_1 + \dots + T_{m-2} = 0 + 1 + 1 + 2 + 4 + \dots + 2^{m-4} = 2^{m-3} = U_{m-2} \end{aligned}$$

ii)  $n \leq k-1$  일 때 성립 가정,  $n = k$  일 때 보이자.

$$\begin{aligned} T_{k+2} - T_{k+1} &= \sum_{i=k-m}^{k-1} T_{i+2} - T_{i+1} (\because \text{귀납 가정}) \\ &= \sum_{i=k-m}^{k-1} U_i = U_k (\because U \text{ 점화식}) \end{aligned}$$

### 보조정리 4.5

$$U_n + T_{n-m+1} = T_{n+1}$$

pf)  $U_n + T_{n-m+1}$

$$\begin{aligned} &= U_n + T_{n+1} - (T_n + \dots + T_{n-m+2}) (\because T_{n+1} = T_n + \dots + T_{n-m+1}) \\ &= U_n - (T_{n+2} - T_{n+1}) + T_{n+1} \\ &= T_{n+1} (\because \text{보조정리 4.3}) \end{aligned}$$

### 보조정리 4.6

$$\sum_{i=0}^{\lfloor \frac{n}{m} \rfloor} U_{n-mi} = T_{n+1} (n \not\equiv 0 \pmod{m})$$

a.  $1 \leq n \leq m-1$

$$\sum_{i=0}^{\lfloor \frac{n}{m} \rfloor} U_{n-mi} = \sum_{i=0}^0 U_{n-mi} = U_n = 2^{n-1} = T_{n+1}$$

b.  $n \leq k-1$ 일 때 성립 가정,  $n=k$ 일 때 성립 보이자

$$\begin{aligned} \sum_{i=0}^{\lfloor \frac{n}{m} \rfloor} U_{n-mi} &= U_n + \sum_{i=0}^{\lfloor \frac{n-m}{m} \rfloor} U_{n-mi} \\ &= U_n + T_{n-m+1} = T_{n+1} (\because \text{보조정리 4.4}) \end{aligned}$$

#### 보조정리 4.7

$$1 + \sum_{i=0}^{\lfloor \frac{n}{m} \rfloor} U_{n-mi} = T_{n+1} (n \equiv 0 \pmod{m})$$

pf)

a.  $n=0$

$$1 + \sum_{i=0}^{\lfloor \frac{n}{m} \rfloor} U_{n-mi} = 1 + U_0 = 1 = T_1$$

b.  $n \leq k-1$ 일 때 성립 가정.  $n=k$ 일 때 보이자.

$$\begin{aligned} 1 + \sum_{i=0}^{\lfloor \frac{n}{m} \rfloor} U_{n-mi} &= U_n + 1 + \sum_{i=0}^{\lfloor \frac{n-m}{m} \rfloor} U_{n-mi} \\ &= U_n + T_{n-m+1} \\ &= T_{n+1} (\because \text{보조정리 4.4}) \end{aligned}$$

#### 정리 4.8

자연수  $n$ 의  $m$ -regular composition의 개수는  $U_n$ 개다.

pf)

a.  $0 \leq n \leq m-1$

$n$ 의 분할의 각 원소들은  $m-1$  이하이므로, (조건 없는)  $n$ 의 분할의 경우의 수와 같다.  $U_n = 2^{n-1}$  for  $0 \leq n \leq m-1$ 이므로 성립.

b.  $n \leq k$ 일 때 성립 가정,  $n=k+1$ 일 때 보이자.

$k+1$ 의 분할의 마지막 원소가  $\text{mod } m$ 으로  $i (\neq 0,1)$ 일 때

i)  $n \not\equiv i-1 \pmod{m}$

$$(\text{경우의 수}) = \sum_{j=0}^{\lfloor \frac{n-i+1}{m} \rfloor} U_{n-i+1-mj} = T_{n-i+1}$$

ii)  $n \equiv i-1 \pmod{m}$

$$(\text{경우의 수}) = 1 + \sum_{j=0}^{\lfloor \frac{n-i+1}{m} \rfloor} U_{n-i+1-mj} = T_{n-i+1}$$

따라서, 분할의 마지막 원소가  $i (\neq 0 \pmod{m})$ 일 때 경우의 수는  $T_{n-i+1}$ 이다.

$$(\text{k+1의 } m\text{-regular composition의 개수}) = \sum_{i=1}^{m-1} T_{n-i+2} = T_{n+3} - T_{n+2} = U_{n+1}$$

따라서, 수학적 귀납법에 의해  $n$ 의  $m$ -regular composition의 개수는  $U_n$ 개임을 알 수 있다.

## V. 결론 및 제언

본 연구에서, 임의의 자연수  $m$ 에 대해, 어떤 자연수  $n$ 의  $m$ -regular composition의 경우의 수가  $m$ -bonacci 수열로 나타나는 것을 밝혔다.

위 문제를 해결하여, 자연수의 분할에 관련된 여러 가지 문제에 응용할 수 있다. 본 연구를 확장하여, 분할의 원소들에 다른 조건에 의한 문제들이 있을 수 있다. 예를 들어, ‘임의의 자연수  $n$ 을  $m$ 의 거듭제곱이 아닌 수들로 분할하는 경우의 수는 몇 개인가’ 같은 문제가 가능하다.

## VI. 참고문헌

- [1] D. Marques, J. Sellers, and P. Trojovsky, On divisibility properties of certain Fibonomial coefficients by a prime  $p$ , *Fibonacci Quart.* 51(1) (2013), 78-83.
- [2] A. Sills, Compositions, partitions, and Fibonacci numbers, *The Fibonacci Quarterly*, 49.4 (2011), 384-354.

