조합론적 게임 이론
(Combinatorial Game Theory)

신촌 연합 겨울캠프 중급 11회차

UNIST 한동규 (queued_q)

(경제학적) 게임 이론 vs 조합론적 게임 이론

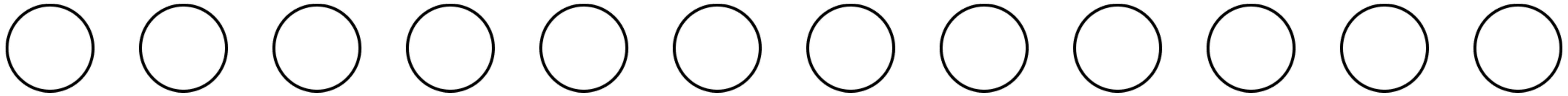
- (경제학적) 게임 이론
  - 자신의 이익을 최대화하려는 이성적인 참가자 (Rational agent)가 어떤 결정과 상호작용을 하는지에 대한 분석
  - 불완전한 정보 / 확률적 요소 등 존재
  - 제로섬/비제로섬 게임, 죄수의 딜레마, Nash equilibrium
- 조합론적 게임 이론
  - 플레이어 간에 차례가 있는 완전 정보 게임에 대한 분석
  - Nim, Sprouts, 체스, 바둑 등의 게임
  - Nimber, Surreal number, Minimax, Monte Carlo tree search

조합론적 게임 이론의 개념들

- **Impartial game:** 현재 어떤 플레이어의 차례인지에 관계 없이 가능한 행동이 동일한 게임
  - Impartial game: Nim, 배스킨라빈스 31 게임, Sprouts, Dots and Boxes
  - Partisan game: 체스, 바둑, 오목, Hex
  - 플레이어의 차례에 관계 없이 게임의 상태만 가지고 분석할 수 있어서 분석이 간단함
- **Normal play convention:** 두 사람이 차례를 번갈아 가지다가, 할 수 있는 행동이 없으면 지는 규칙
  - 또는 마지막으로 플레이한 사람이 이기는 규칙
  - 수학적으로 잘 분석되어 있음
- **Misère game:** 두 사람이 차례를 번갈아 가지다가, 할 수 있는 행동이 없으면 이기는 게임
  - 규칙을 조금 변형한 것 뿐인데도 분석하기 굉장히 어려움

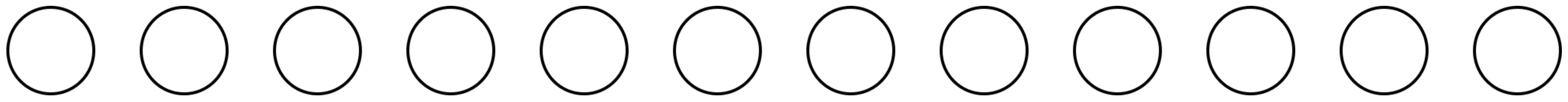
연습문제) BOJ 9660 돌 게임 6

- 각 플레이어의 차례에 돌 $N \le 10^{12}$ 개 중 {1, 3, 4} 개를 가져갈 수 있다.
- 마지막으로 돌을 가져간 사람이 이긴다.
- 완벽하게 플레이했을 때 이기는 사람은?



연습문제) BOJ 9660 돌 게임 6

- 완벽하게 / 최선의 전략으로 플레이한다는 것은 무슨 뜻일까?
- 최선의 전략은 항상 존재하는가?
- 항상 이기는 사람이 결정되는가? 비기는 경우는 없는가?



Zermelo's Theorem: 승리 전략은 존재한다

- (Zermelo, 1913) 유한 차례 안에 끝나고, 게임이 끝났을 때 비기는 상태가 없는 완전 정보 게임에서는, 항상 둘 중 한 명의 플레이어에게 승리 전략이 있다!
- 어떤 게임 상태에서 승리 전략이 존재한다는 말은?
    - 전략에 따라 특정 행동을 하고 나면, 상대가 어떤 행동을 하든 대응되는 승리 전략이 존재하는 상태
    - 재귀적으로 정의할 수 있다.

Zermelo's Theorem: 승리 전략은 존재한다

- (Zermelo, 1913) 유한 차례 안에 끝나고, 게임이 끝났을 때 비기는 상태가 없는 완전 정보 게임에서는, 항상 둘 중 한 명의 플레이어에게 승리 전략이 있다!
- 어떤 게임 상태에서 승리 전략이 존재한다는 말은?
  - 전략에 따라 특정 행동을 하고 나면, 상대가 어떤 행동을 하든 대응되는 승리 전략이 존재하는 상태
  - 재귀적으로 정의할 수 있다.
- 각 상태에 승리 전략이 존재하는지에 따라 이름을 붙여 보자.
- 승리 포지션: 현재 차례를 진행하는 플레이어에게 승리 전략이 존재하는 상태
- 패배 포지션: 현재 차례를 진행하는 플레이어가 어떤 행동을 하든 상대에게 승리 전략이 존재하는 상태
- Zermelo’s theorem = 게임의 모든 상태에 승리 포지션와 패배 포지션 둘 중 하나를 지정할 수 있다.

Zermelo's Theorem: 승리 전략은 존재한다

승리/패배 상태의 재귀적인 정의

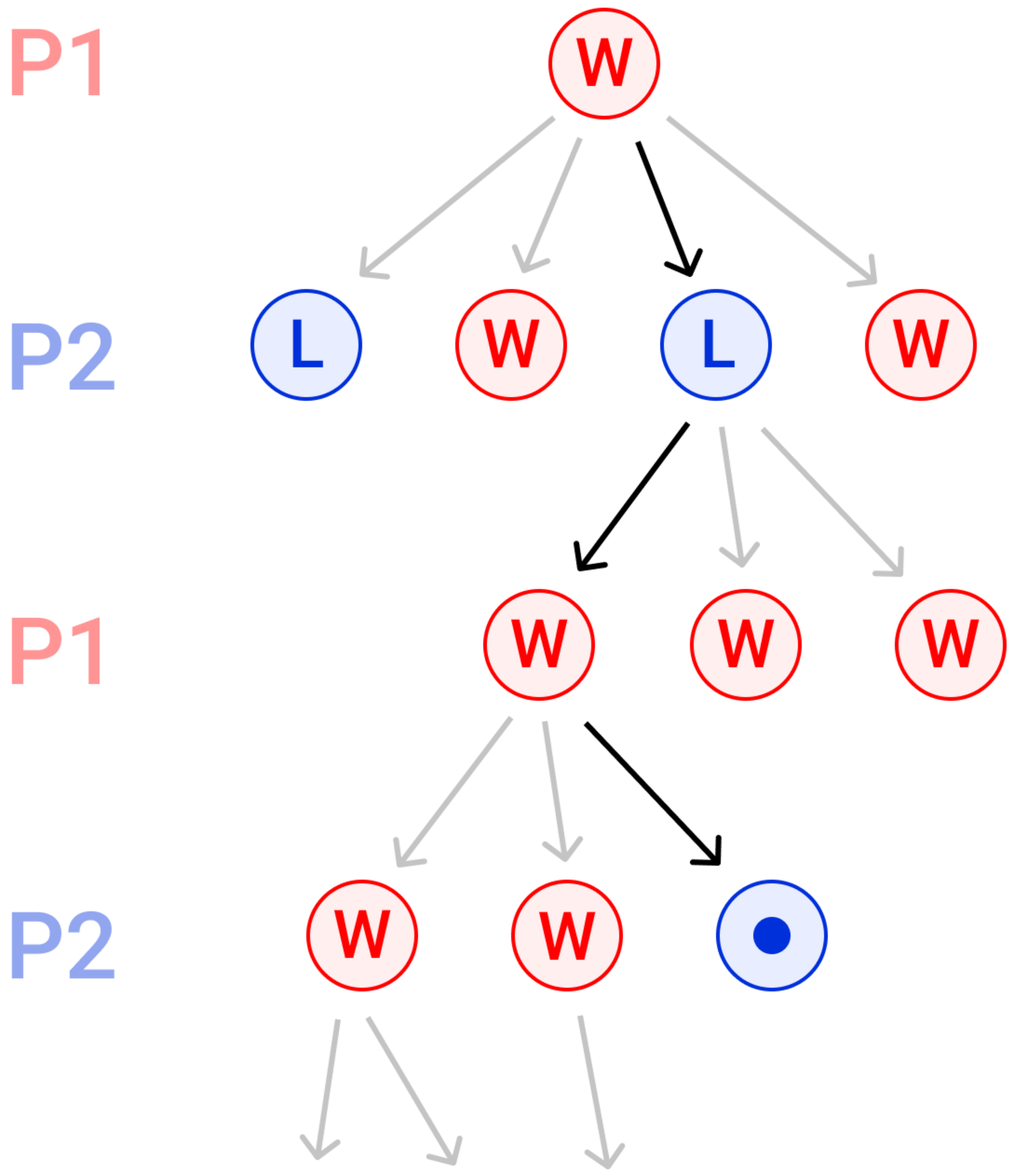
- (Base case) **게임이 종료되어** 승리/패배가 결정된 상태 → 승리/패배 포지션
- 이동할 수 있는 다음 상태 중 패배 포지션이 **존재함** → 승리 포지션
  - 해당 상태로 이동하여 상대 플레이어를 강제적으로 패배하게 만들 수 있으므로
- 이동할 수 있는 다음 상태 **모두가** 승리 포지션임 → 패배 포지션
  - 어떻게 행동하든 상대 플레이어에게 승리하는 전략이 있으므로

Zermelo's Theorem: 승리 전략은 존재한다

승리/패배 상태의 재귀적인 정의

- (Base case) **게임이 종료되어** 승리/패배가 결정된 상태 → 승리/패배 포지션
- 이동할 수 있는 다음 상태 중 패배 포지션이 **존재함** → 승리 포지션
    - 해당 상태로 이동하여 상대 플레이어를 강제적으로 패배하게 만들 수 있으므로
- 이동할 수 있는 다음 상태 **모두가** 승리 포지션임 → 패배 포지션
    - 어떻게 행동하든 상대 플레이어에게 승리하는 전략이 있으므로
- 이동할 수 있는 다음 상태에 패배 포지션도 존재하지 않고, 전부 승리 포지션인 것도 아님 → ?
- 게임의 남은 최대 턴 수를 기준으로 수학적 귀납법을 적용하면, 모든 상태에 승리/패배 포지션을 지정할 수 있음을 보이는 것이 가능하다.

Zermelo's Theorem: 승리 전략은 존재한다



Zermelo's Theorem: 승리 전략은 존재한다

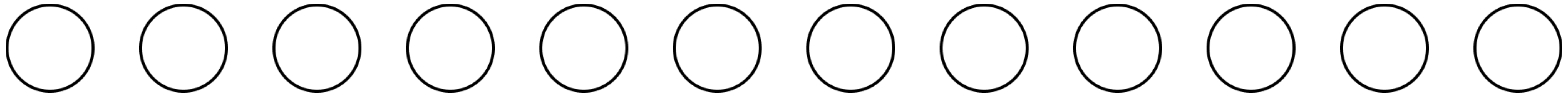
- *Note 1.* 어떤 플레이어의 차례인지가 승패에 영향을 미치지 않는 impartial game에서는,
    - 승리 포지션을 N-position 이라고 표기하며, Next player 가 승리하는 상태라는 의미이다.
    - 패배 포지션을 P-position 이라고 표기하며, Previous player 가 승리하는 상태라는 의미이다.
- *Note 2.* 상태의 수가 무한하지만 유한 차례 안에 끝나는 게임도 있다.
    - 테이블에 동전 놓기 게임

연습문제) BOJ 9660 돌 게임 6

- 완벽하게 / 최선의 전략으로 플레이한다는 것은 무슨 뜻일까?
  - 다음 상태에 (상대의) 패배 포지션이 있다면 패배 포지션으로만 이동한다.
  - 다음 상태에 패배 포지션은 없지만 무승부 포지션이 있다면 무승부 포지션으로 이동한다.
- (둘 중 한 명이 승리하는) 최선의 전략은 항상 존재하는가?
  - 무승부 포지션이 없다면.
- 항상 이기는 사람이 결정되는가? 비기는 경우는 없는가?
  - 돌의 개수 유한, 단조 감소하는 정수 → 유한 차례 안에 종료
  - 게임이 종료되었을 때 (돌이 0개 남았을 때) 차례를 갖는 플레이어가 짐 → 무승부 종료 상태 없음
  - Zermelo’s theorem → 최선의 전략으로 플레이했을 때, 두 플레이어 중 승자가 존재한다!

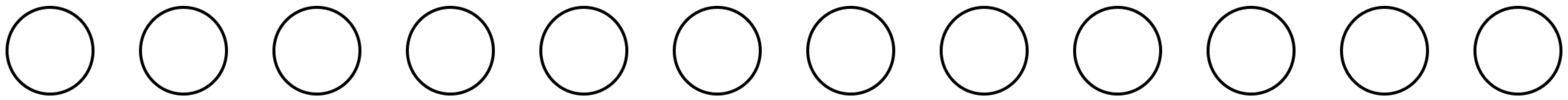
연습문제) BOJ 9660 돌 게임 6

- 재귀적 정의 → DP 식을 세워 보자.
- `D[i]` : 돌이 `i`개 남았을 때, 현재 차례의 플레이어가 이기는 전략이 있는지 (승리 포지션인지)
  - `D[0] = false`
  - `D[i] = !(D[i-1] && D[i-3] && D[i-4])`
- N이 너무 크다…



연습문제) BOJ 9660 돌 게임 6

- `D[i] = !(D[i-1] && D[i-3] && D[i-4])`
- 관찰: DP 식이 직전 네 값의 패턴에만 영향을 받는다.
    - 두 위치 `i`, `j`에서 직전 네 값의 패턴이 동일하면 앞으로의 DP 값들도 동일하게 채워진다.
    - 네 값의 패턴의 경우의 수: $2^4 = 16$
    - DP 값은 16 이하의 주기를 갖고 반복된다. (비둘기집의 원리)
    - `D[0]`부터 `D[3]`까지의 값은 범위를 벗어나는 인덱스 때문에 식이 달라지므로, 주기에 포함되지 않을 수 있음에 주의하자.



연습문제) BOJ 9660 돌 게임 6

- 이전 네 값의 패턴이 일치하는 부분을 찾아서, 직접 주기를 구해 보면 7의 주기를 갖는 것을 알 수 있다.
- 우연의 일치로, `D[0]`부터 `D[3]`까지도 주기에 포함되어 반복된다.
- `D[i] = i%7 in {1, 3, 4, 5, 6}`
- `D[N]`이 참일 때 선공 승리 (N), 거짓일 때 후공 승리 (P)

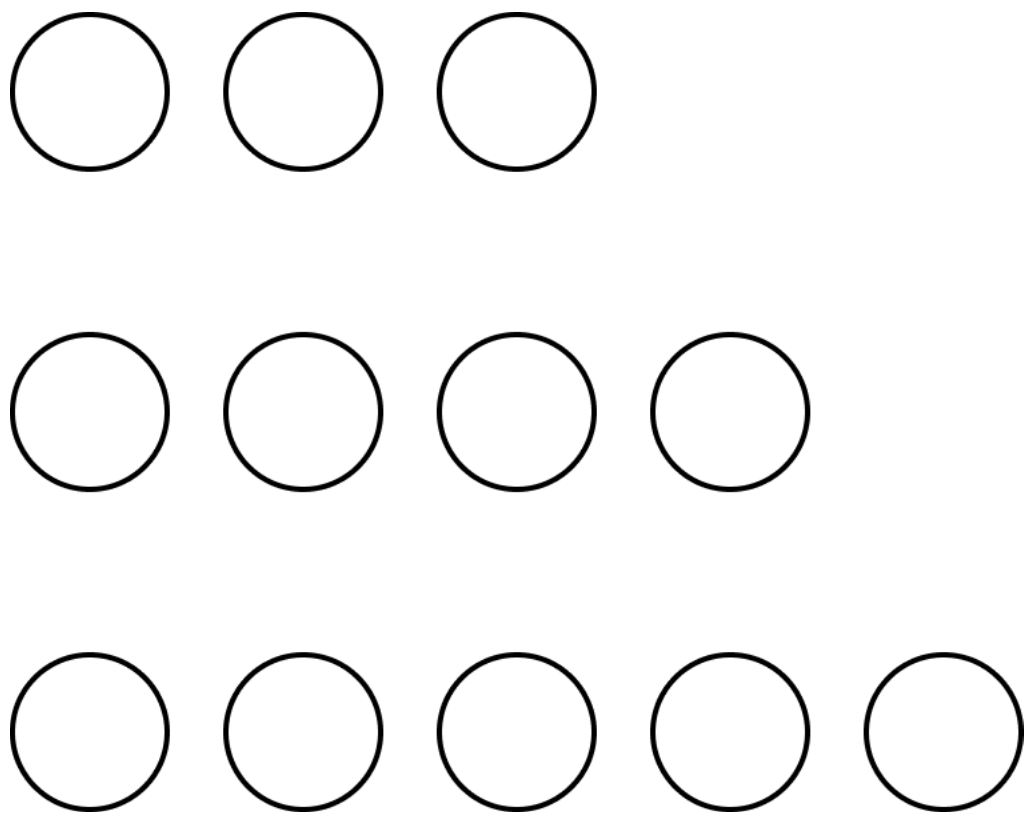
0	1	2	3	4	5	6	7	8	9	10	11
0	1	0	1	1	1	1	0	1	0	1	1

Nim 게임

(Nim Game)

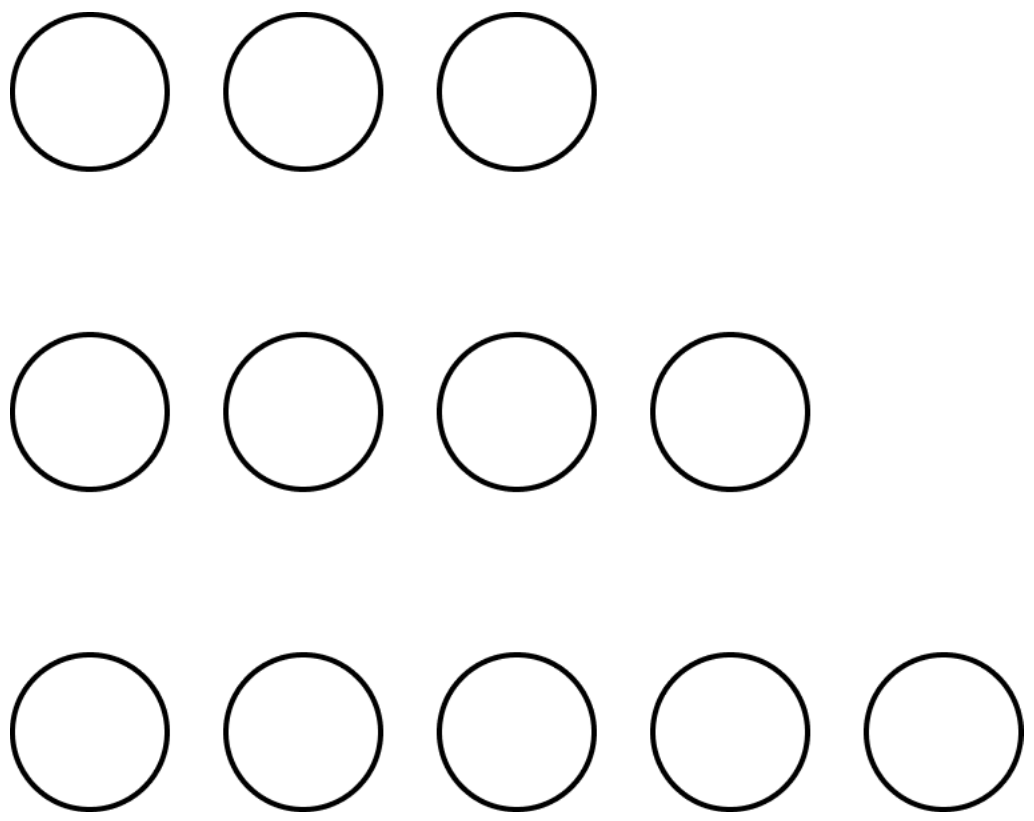
연습문제) BOJ 11868 님 게임 2

- N개의 돌 더미 (힙)이 있고, 각 돌 더미마다 x_i개의 돌이 있다.
- 각 차례에 플레이어는 비어 있지 않은 돌 더미를 골라서 하나 이상의 돌을 제거한다.
- 전체 돌 더미에서 마지막으로 돌을 제거하는 사람이 이긴다.
- 최적의 방법으로 플레이했을 때 이기는 사람은?



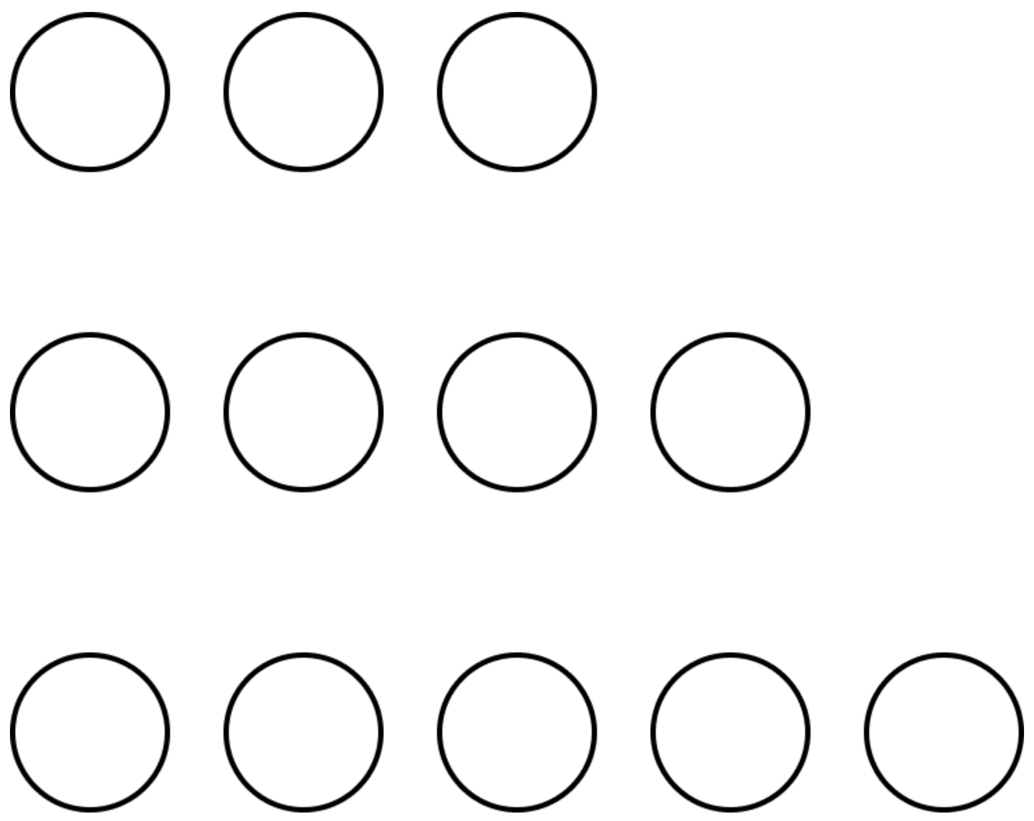
연습문제) BOJ 11868 님 게임 2

- 돌 더미가 하나인 경우: 첫 번째 플레이어가 이긴다.
- 돌 더미가 두 개인 경우: ?



연습문제) BOJ 11868 님 게임 2

- 돌 더미가 하나인 경우: 첫 번째 플레이어가 이긴다.
- 돌 더미가 두 개인 경우:
  - $x_1 = x_2$: 두 번째 플레이어가 이긴다.
  - $x_1 \neq x_2$: 첫 번째 플레이어가 이긴다.
- 돌 더미가 세 개인 경우: ?
  - 생각해내기 어려우니 알려져 있는 결과를 이용하자 😅



연습문제) BOJ 11868 님 게임 2

- (Bouton, 1901) $s = x_1 \wedge x_2 \wedge \ldots \wedge x_N$ 이 0이면 후공, 그 외에는 선공이 이긴다. (^는 xor 연산)
- $s = 0$: 돌을 어떻게 가져가든 이 값은 0이 아닌 정수가 된다.
- $s \neq 0$: 돌을 가져가서 이 값을 0으로 만드는 방법이 존재한다.
- 전략
  - A에게 주어진 게임 상태가 $s \neq 0$인 경우 이 값을 0으로 만든다.
  - A는 항상 $s \neq 0$인 상태를, B는 항상 $s = 0$인 상태를 받는다.
  - 돌을 전부 가져간 상태는 $x_1 = x_2 = \ldots = x_N = 0$ 이므로 $s = 0$ 이다.
  - 즉, A는 돌을 전부 가져간 상태를 받을 수 없고, 지는 것이 불가능하다. A가 승리한다.
  - A에게 주어진 게임 상태가 $s = 0$인 경우 B가 $s \neq 0$인 상태를 받으므로 A가 패배한다.

$$
\begin{array}{rrl}
 & 11000 & x_1 \\
 & 10110 & x_2 \\
\wedge & 01011 & x_3 \\
\hline
 & 00101 & s
\end{array}
$$

연습문제) BOJ 11868 님 게임 2

- k번째 힙의 돌을 가져가고 남은 돌의 수가 y라고 하자. 또한, 돌을 가져간 뒤의 s 값을 t라고 하자.
- $s = x_1 \wedge \dots \wedge x_N$, $t = x_1 \wedge \dots \wedge y \wedge \dots \wedge x_N$
- 그러면 $t = s \wedge x_k \wedge y$ 이다.

$$
\begin{array}{rrl}
 & 11000 & x_1 \\
 & 10110 & x_2 \\
\wedge & 01011 & x_3 \\
\hline
 & 00101 & s
\end{array}
$$

연습문제) BOJ 11868 님 게임 2

- k번째 힙의 돌을 가져가고 남은 돌의 수가 y라고 하자. 또한, 돌을 가져간 뒤의 s 값을 t라고 하자.
- $s = x_1 \wedge \ldots \wedge x_N$, $t = x_1 \wedge \ldots \wedge y \wedge \ldots \wedge x_N$
- 그러면 $t = s \wedge x_k \wedge y$ 이다.
- s = 0: 돌을 어떻게 가져가든 t는 0이 아닌 정수가 된다.
  - $x_k \neq y$ 이므로 $t = s \wedge x_k \wedge y = x_k \wedge y \neq 0$
- s ≠ 0: 돌을 가져가서 t를 0으로 만드는 방법 (적당한 k와 y)가 존재한다.
  - 이 경우가 조금 어려움

```
  11000  x1
  10110  x2
^ 01011  x3
-------
  00101  s
```

연습문제) BOJ 11868 님 게임 2

- $s \neq 0$: 돌을 가져가서 t를 0으로 만드는 방법 (적당한 k와 y)가 존재한다.
  - s에서 0이 아닌 가장 왼쪽 비트의 위치를 d라고 하자.
  - 모든 d번째 비트의 xor 값이 1이므로 d번째 비트가 1인 x_k가 존재한다.
  - **x_k ^ s 의 값은 x_k보다 작다.**
    - x_k의 d번째보다 왼쪽에 있는 비트가 그대로 유지되고,
    - x_k의 d번째 비트가 1에서 0으로 감소하므로
  - 따라서 k번째 힙의 돌을 가져가서 y = x_k ^ s 개의 돌로 만든다.
  - t = s ^ x_k ^ y = s ^ x_k ^ (x_k ^ s) = 0

```
  11000 x1
  10110 x2
^ 01011 x3
---------
  00101 s
```

```
  10110 x2
^ 00101 s
---------
  10011 y
```

연습문제) BOJ 11868 님 게임 2

- 결론: x_1 ^ x_2 ^ … ^ x_N = 0이면 후공 승리 (P), 아니면 선공 승리 (N)

스프라그-그런디 정리
(Sprague-Grundy Theorem)

연습문제) BOJ 16881 행렬 게임

- 자연수로 구성된 N×M 행렬이 주어진다.
- 각 플레이어는 자기 차례에 0이 아닌 수가 존재하는 행을 하나 고르고, 0이 아닌 제일 왼쪽 수를 K라고 했을 때 그 수를 1 이상 K 이하만큼 감소시킨다.
- 모든 수가 0이 되면 마지막으로 수를 감소시킨 사람이 승리한다.
- 최적의 방법으로 플레이했을 때 이기는 사람은?

1	5	2	3
4	1	1	2
1	3	1	1

연습문제) BOJ 16881 행렬 게임

- 각각의 행을 별개의 게임으로 생각하면, 행렬 게임을 여러 게임이 합쳐진 새로운 게임으로 볼 수 있다.
- nim 게임도 사실 비슷하다. 각각의 힙을 하나의 게임으로 생각한다.
  - 힙에 돌이 0개보다 많이 있으면 무조건 선공이 이기는 자명한 게임
  - 하지만 힙이 여러 개가 되어 복잡한 전략이 탄생
  - 각각의 게임에 대해 승패를 판별하는 것 이상의 정보가 필요하다.

1	5	2	3
4	1	1	2
1	3	1	1

연습문제) BOJ 16881 행렬 게임

- 각각의 행을 별개의 게임으로 생각하면, 행렬 게임을 여러 게임이 합쳐진 새로운 게임으로 볼 수 있다.
- nim 게임도 사실 비슷하다. 각각의 힙을 하나의 게임으로 생각한다.
  - 힙에 돌이 0개보다 많이 있으면 무조건 선공이 이기는 자명한 게임
  - 하지만 힙이 여러 개가 되어 복잡한 전략이 탄생
  - 각각의 게임에 대해 승패를 판별하는 것 이상의 정보가 필요하다.
- 게임 합: 게임 A, B가 있을 때, 게임 A+B를 다음과 같이 정의한다.
  - 게임의 상태는 A의 상태와 B의 상태의 순서쌍이다. (두 게임이 독립적으로 진행된다.)
  - 각 플레이어는 자기 차례에 A에서 할 수 있는 행동 또는 B에서 할 수 있는 행동 둘 중 하나를 선택해 수행한다.
  - 두 게임 모두에서 할 수 있는 행동이 없으면 패배한다.

1	5	2	3
4	1	1	2
1	3	1	1

Sprague–Grundy Theorem

- (Sprague, 1935 / Grundy, 1939) 모든 impartial game은 하나의 nim 힙으로 생각할 수 있다!
- Grundy number: 게임에 대응되는 nim 힙의 크기
- nim 힙으로 생각할 수 있다는 게 무슨 의미? → 게임 합 연산, 승리/패배 포지션에 대해 동등하다.
- 그런디 수가 a인 게임을 크기 a인 nim 힙으로 대체해도 동일한 결과가 나오게 만들자!
    - 크기 0인 nim 힙은 패배 포지션 (P)이다.
    - 그런디 수가 0인 게임은 패배 포지션 (P)이면 좋겠다.
    - 크기 a, b인 nim 힙 두 개는 크기 a ^ b인 nim 힙으로 대체할 수 있다.
    - 그런디 수가 a, b인 두 게임을 합하면 그런디 수가 a ^ b가 되면 좋겠다.

Sprague–Grundy Theorem

- Grundy 수는 어떻게 구할까?
- g(G) = mex { g(G’) | G’은 G의 다음 상태 }
    - mex (Minimum EXcluded value): 집합에 포함되지 않은 가장 작은 0 이상의 정수
    - mex {0, 1, 2, 4, 7} = 3
- 왜 하필 mex?

Sprague–Grundy Theorem

- Grundy 수는 어떻게 구할까?
- $g(G) = \text{mex}\ \{ g(G') \mid G'$은 G의 다음 상태 $\}$
  - mex (Minimum EXcluded value): 집합에 포함되지 않은 가장 작은 0 이상의 정수
  - $\text{mex}\ \{0, 1, 2, 4, 7\} = 3$
- 왜 하필 mex? Bouton의 전략이 성립하는 핵심적인 이유를 생각해 보자.
  - $s = 0 \rightarrow \forall\ t \neq 0$: nim 힙의 새 크기가 원래 크기와 같아질 수 없기 때문
  - $s \neq 0 \rightarrow \exists\ t = 0$: nim 힙의 원래 크기보다 작은 모든 크기로 줄이는 것이 가능하기 때문
  - mex: 다음 상태와 같아질 수 없으면서, 더 작은 모든 값에 도달할 수 있는 유일한 수

Sprague–Grundy Theorem

- Bouton의 전략을 응용하면 $g(A+B) = g(A) \wedge g(B)$ 임을 알 수 있다.
- 게임 $G_1 + \ldots + G_N$ 을 이기는 전략
  - $g(G_1) \wedge \ldots \wedge g(G_N) \neq 0$: 다음 상태의 Grundy 수가 0이 되도록 어떤 게임 하나에서 행동을 한다.
    - 항상 Grundy 수가 0보다 큰 게임을 받으므로, mex의 성질에 의해 더 작은 Grundy 수를 가지는 다음 상태가 존재한다. 즉, 패배하지 않는다 (승리한다).
  - $g(G_1) \wedge \ldots \wedge g(G_N) = 0$: 항상 상대에게 Grundy 수가 0보다 큰 게임을 넘겨주게 된다.
    - 항상 Grundy 수가 0인 게임을 받다가, 언젠가 모든 게임 $G_1, \ldots, G_N$에서 가능한 행동이 없는 상태를 받는다. 즉, 패배한다.

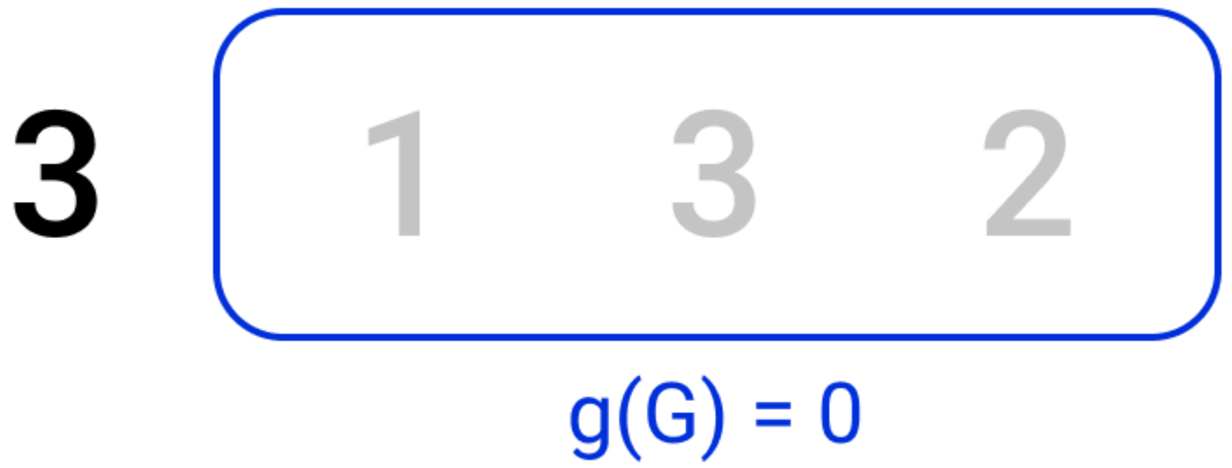
연습문제) BOJ 16881 행렬 게임

- 목표: 각 행을 별개의 게임으로 보고 Grundy 수를 구한 뒤, 전체 게임의 Grundy 수를 구하는 것
- 행 게임 [a_1 a_2 a_3 … a_M] 의 Grundy 수를 구해 보자.
- 마지막 열만 남은 경우: nim 힙과 동일, g([a]) = a
- 행 게임 [1 G] 에서 갈 수 있는 상태: 행 게임 G
- 행 게임 [a G] 에서 갈 수 있는 상태: 행 게임 G, [1 G], … , [a-1 G]

1	5	2	3
4	1	1	2
1	3	1	1

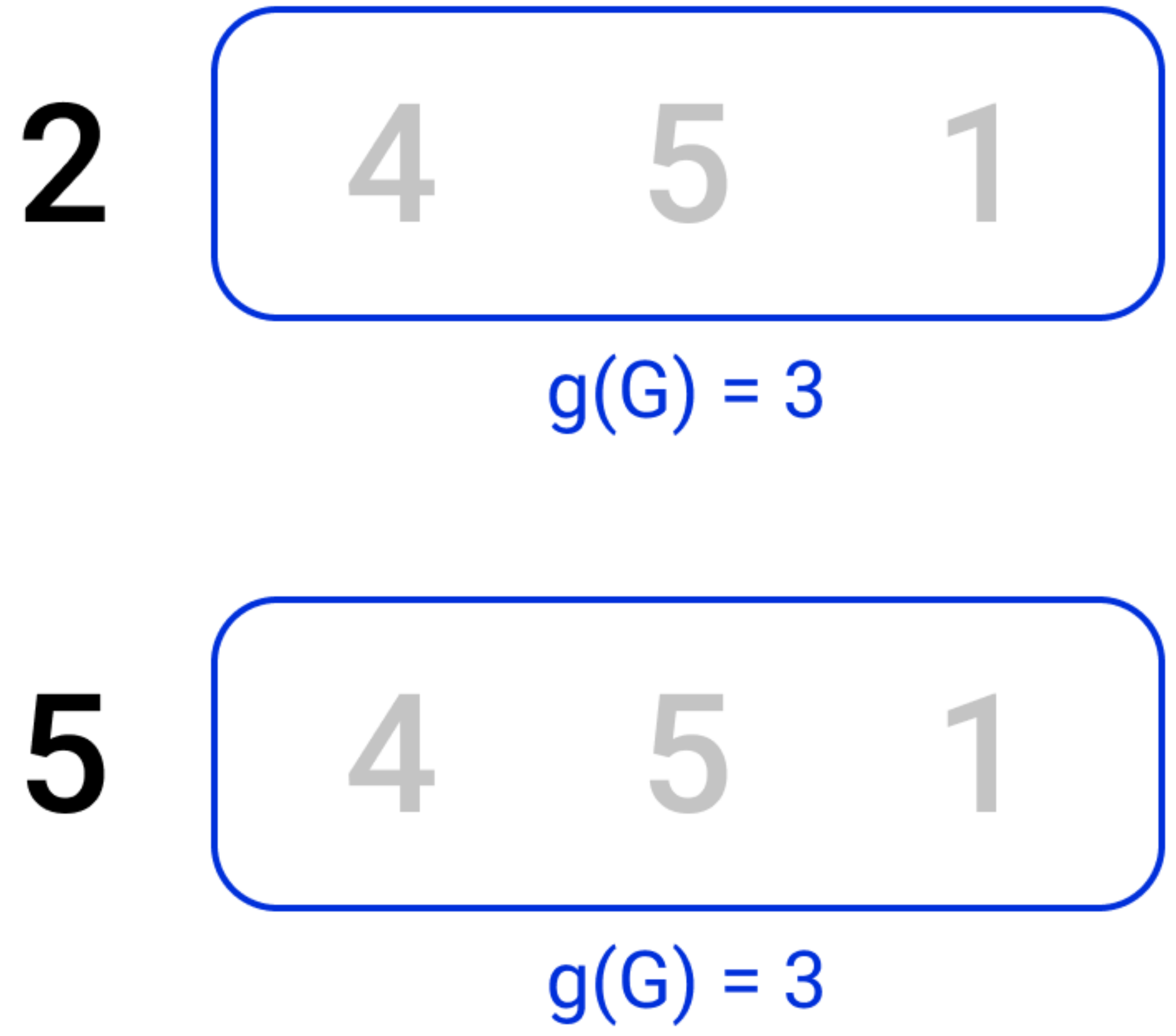
연습문제) BOJ 16881 행렬 게임

- g(G) = 0:
  - g([1, G]) = mex {0} = 1
  - g([2, G]) = mex {0, 1} = 2
  - …
  - g([a, G]) = mex {0, 1, …, a-1} = a



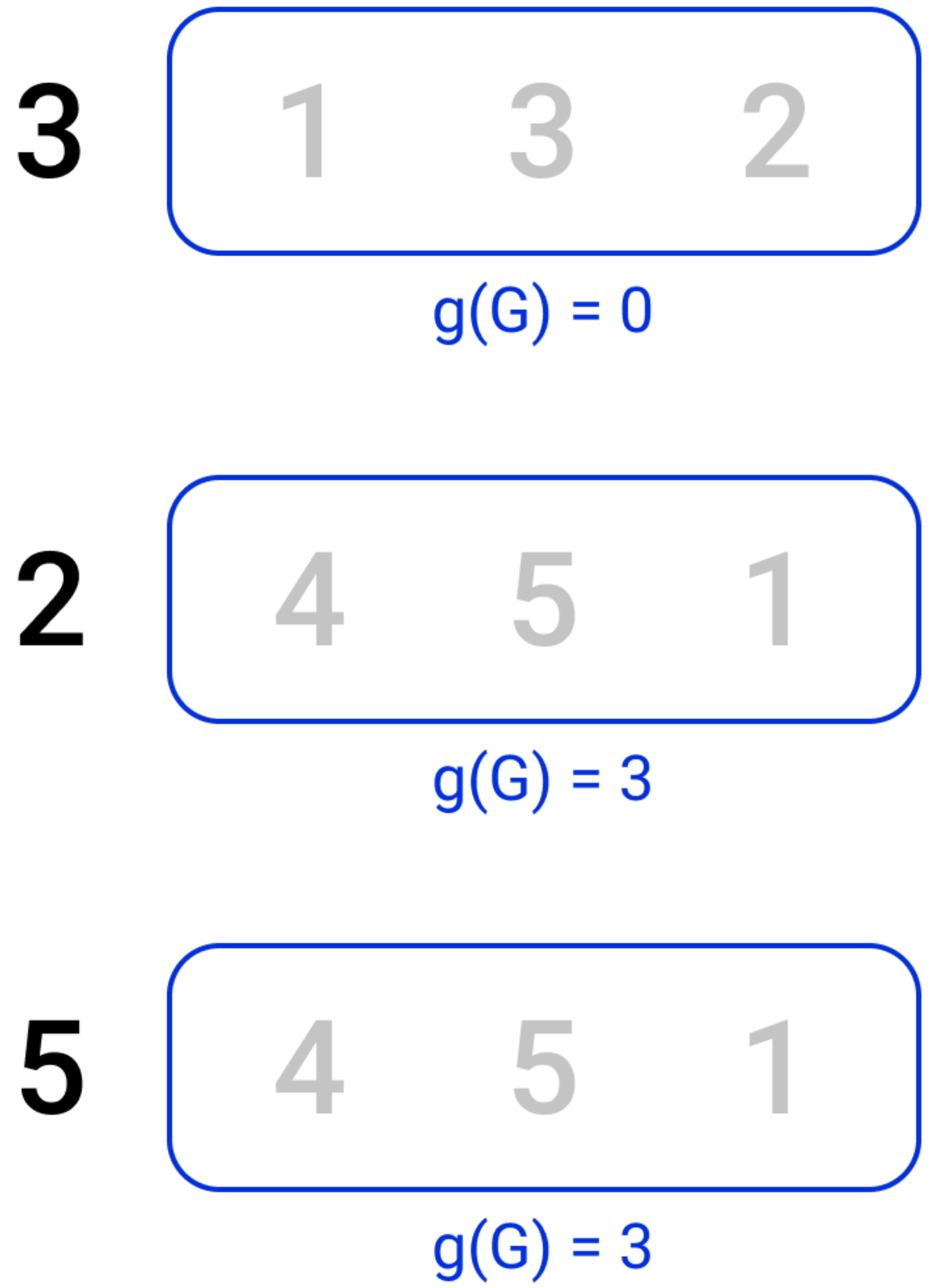
연습문제) BOJ 16881 행렬 게임

- $g(G) = k\ (> 0)$:
  - $g(\ [1, G]\ ) = \text{mex}\ \{k\} = 0$
  - $g(\ [2, G]\ ) = \text{mex}\ \{k, 0\} = 1$
  - …
  - $g(\ [k, G]\ ) = \text{mex}\ \{k, 0, 1, \ldots, k\text{-}2\} = k\text{-}1$
  - $g(\ [k+1, G]\ ) = \text{mex}\ \{k, 0, 1, \ldots, k\text{-}2, k\text{-}1\} = k+1$
  - …
  - $g(\ [a\ (> k), G]\ ) = \text{mex}\ \{k, 0, 1, \ldots, k\text{-}1, k+1, \ldots, a\text{-}1\} = a$



연습문제) BOJ 16881 행렬 게임

- 행의 값들이 `A[1..N]` 에 저장되어 있을 때,
- `D[i]` : 부분 게임 `A[i..N]` 의 Grundy 수
  - `D[i+1] = 0 → D[i] = A[i]`
  - `D[i+1] = g` (> 0)
    - `A[i] ≤ g → D[i] = A[i] - 1`
    - `A[i] > g → D[i] = A[i]`

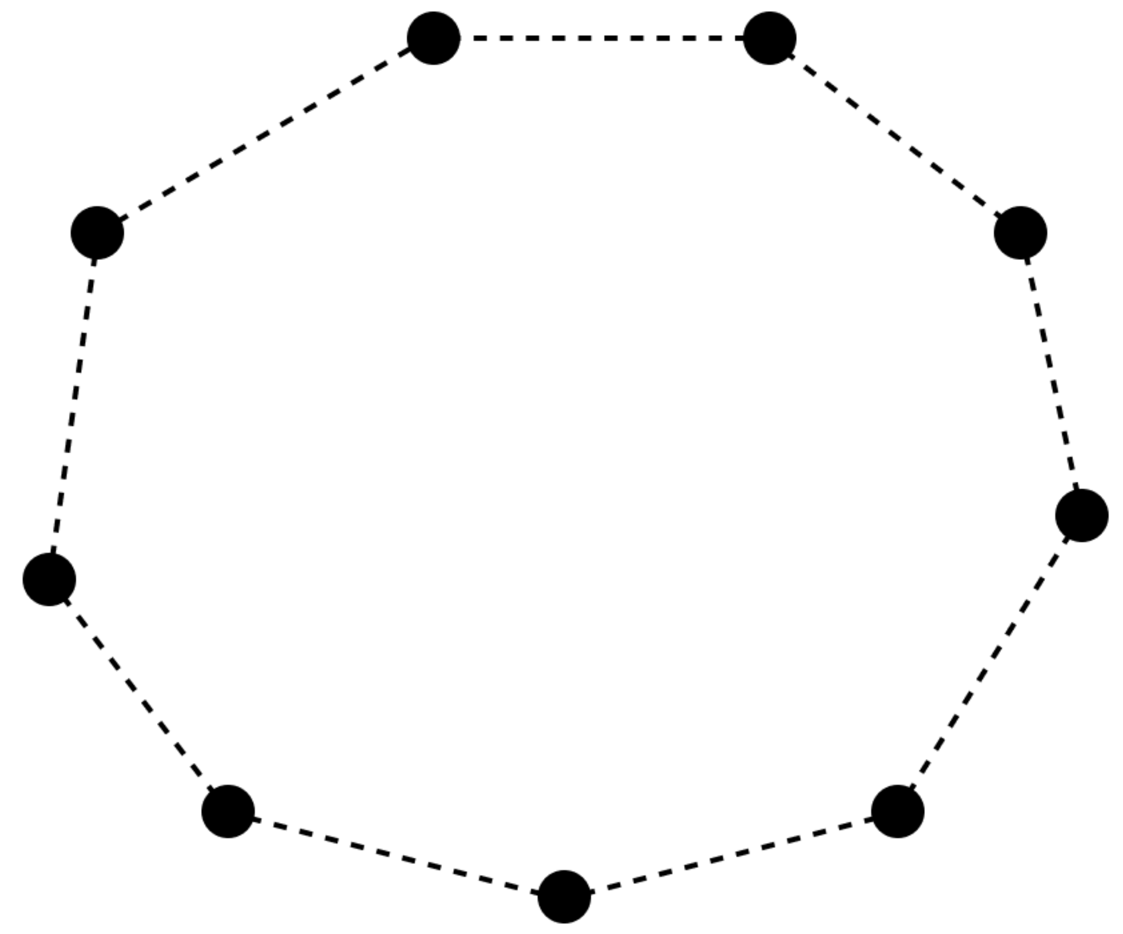


연습문제) BOJ 16881 행렬 게임

- 행의 값들이 `A[1..N]` 에 저장되어 있을 때,
- `D[i]` : 부분 게임 `A[i..N]` 의 Grundy 수
  - `D[i+1] = 0 → D[i] = A[i]`
  - `D[i+1] = g` (> 0)
    - `A[i] ≤ g → D[i] = A[i] - 1`
    - `A[i] > g → D[i] = A[i]`
- 각 행 게임의 Grundy 수 `D[1]`을 구하면, xor을 취해서 전체 게임의 Grundy 수를 구한다.
  - 전체 게임의 Grundy 수가 0인 경우: 후공 승리
  - 전체 게임의 Grundy 수가 0이 아닌 경우: 선공 승리

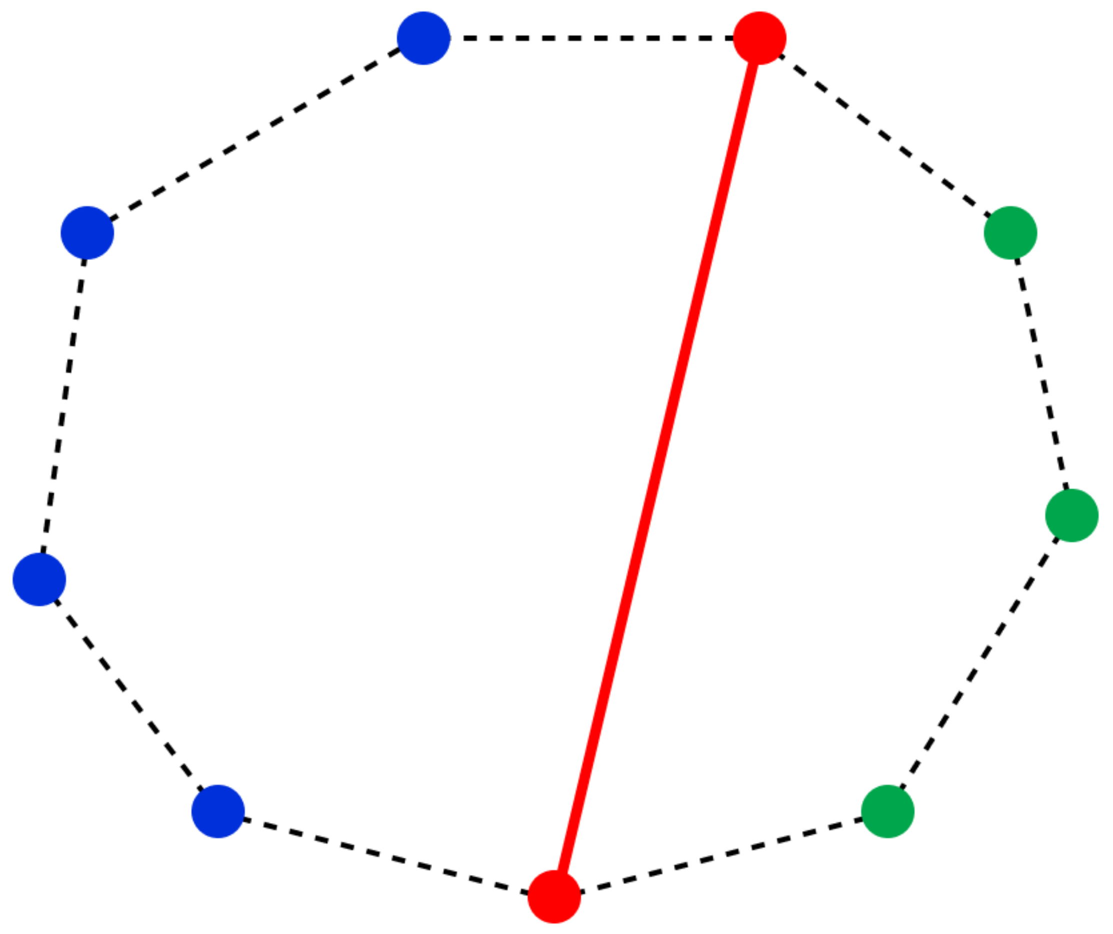
연습문제) BOJ 13034 다각형 게임

- N개의 꼭짓점으로 이루어진 볼록다각형이 있다.
- 각 플레이어는 두 꼭짓점을 골라 선분을 긋는다. 단, 이전에 그린 선분과 만나면 안 된다.
- 더 이상 선분을 그릴 수 없는 사람이 패배한다.
- 최적의 방법으로 플레이했을 때 이기는 사람은?



연습문제) BOJ 13034 다각형 게임

- 게임의 개수가 플레이어의 행동에 따라 변할 수도 있다!
- 선분을 그으면 게임이 둘로 나눠진다.
- `D[i]` : 꼭짓점이 `i`개인 게임의 Grundy 수
  - `D[0] = D[1] = 0`
  - `D[i] = mex { D[j] ^ D[i-j-2] | 0 ≤ j ≤ i-2 }`
- 집합을 0/1 배열로 관리하고, 반복문을 통해 직접 mex를 구해 주면 된다.
- `D[N] = 0`: 후공 승리, 그 외 선공 승리



연습 문제

- 9660 - 돌 게임 6
- 11868 - 님 게임 2
- 16881 - 행렬 게임
- 13034 - 다각형 게임

- 16884 - 나이트 게임
- 11694 - 님 게임

출석 문제

- 11062 - 카드 게임
- 9662 - 돌 게임 8

- 16877 - 핌버
- 16895 - 님 게임 3
- 3596 - 크로스와 크로스
- 11717 - Wall Making Game
- 16831 - Nim without Zero

Misère Game

- Misère game: 마지막으로 플레이한 사람이 지는 게임
- Normal game에 비해 이론이 잘 정립되지 않았다.
  - Genus theory, Indistinguishability quotient
- Dawson’s Kayles: N개의 볼링 핀 중 이웃한 두 개를 번갈아 제거하는 게임
  - Normal rule: Grundy 수 0–15에 대응되는 총 16가지 상태
  - Misère rule: 미해결 난제 / 힙의 크기를 33개 이하로 제한해도, 638가지 서로 다른 상태 존재
- 특수 케이스 — Misère nim은 해결된 문제!

연습문제) BOJ 11694 님 게임

- 문제: Misère nim 게임의 승자는?
- 크기가 1보다 큰 힙이 하나보다 많을 때는 Nim과 동일한 전략을 취한다. 즉, 상대에게 항상 xor 결과가 0인 게임 상태만 준다.
- 크기가 1보다 큰 힙이 하나 남으면,
  - 크기가 1인 힙이 짝수 개라면 크기가 1보다 큰 힙에서 돌을 한 개만 남긴다.
  - 크기가 1인 힙이 홀수 개라면 크기가 1보다 큰 힙의 돌을 전부 가져간다.
- 크기가 1보다 큰 힙이 하나인 상황이 상대한테 갈 수 있을까?
  - 크기가 1보다 크다 = 2^1의 자리 혹은 더 큰 자리의 비트가 1로 설정되어 있다.
  - 크기가 1인 힙이 아무리 많아도 xor 연산을 취하면 0번째 비트만 변하므로 xor 결과는 0이 아니다.
  - 상대에게 항상 xor 결과가 0인 게임 상태를 주므로 불가능