

그리디 알고리즘

최백준 choi@startlink.io

그리디 알고리즘

그리디 알고리즘

Greedy Algorithm

3

- 결정해야 할 때, 그 순간에 가장 좋다고 생각하는 것을 선택하면서 답을 찾아가는 알고리즘
- 그 때 그 때는 최적일지도 모르지만, 최종적으로는 답이 최적일 아닐 수도 있다.

거스름돈 문제

Greedy Algorithm

- 1, 5, 10, 50, 100, 500원 동전을 매우 많이 가지고 있고
- 1000, 5000, 10000, 50000원 지폐를 매우 많이 가지고 있을 때
- N원을 거슬러주는 문제
- 이 때, 사용하는 지폐와 동전의 개수를 최소로 해야 한다.
- 가장 큰 액면가를 가진 지폐나 동전부터 거슬러 주자.

7570원

5000원 x 1개

2570원

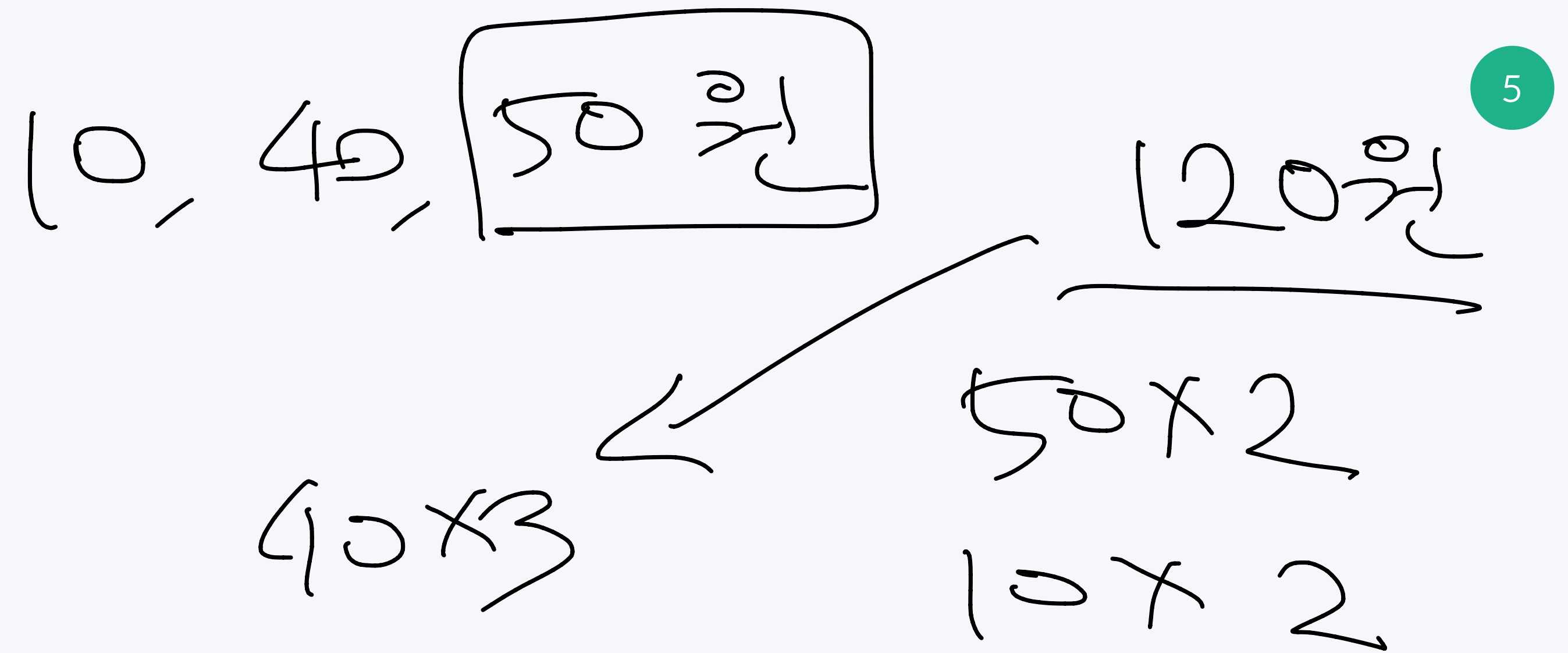
1000원 x 2개

570원

거스름돈 문제

Greedy Algorithm

- 7570원을 거슬러 받아야 한다
- 5000원 1장
- $7570 - 5000 = 2570$
- 1000원 2장
- $2570 - 2000 = 570$
- 500원 1개
- $570 - 500 = 70$
- 50원 1개
- $70 - 50 = 20$
- 10원 2개



거스름돈 문제

Greedy Algorithm

6

- 동전이 1, 4, 5원 있는 경우
- 12원을 거슬러주는 경우에는
- 5원 2개
- $12 - 10 = 2$
- 1원 2개
- 총 4개가 필요하다
- 하지만, 정답은 4원 3개이다

그리디 알고리즘

Greedy Algorithm

7

- 언제 그리디 알고리즘을 쓰나?
- 지금 이 순간 가장 좋은 경우를 선택하는 것이 항상 최적인 경우에
- 그래서 쉽다

그리디 알고리즘

Greedy Algorithm

- 언제 그리디 알고리즘을 쓰나?
- 지금 이 순간 가장 좋은 경우를 선택하는 것이 항상 최적인 경우에
- ~~그래서 쉽다~~
- 가장 어렵다
- 그것이 왜 최적이 되는지를 증명해야하기 때문 $\pi\pi$

동전 0

<https://www.acmicpc.net/problem/11047>

- 준규가 가지고 있는 동전은 총 N 종류이고, 각각의 동전을 매우 많이 가지고 있다.
- 동전을 적절히 사용해서 그 가치의 합을 K 로 만드려고 한다. 이 때 필요한 동전 개수의 최소값을 구하는 프로그램을 작성하시오.
- N 개의 줄에 동전의 가치 A_i 가 오름차순으로 주어진다. ($1 \leq A_i \leq 1,000,000$, $A_1 = 1$, $i \geq 2$ 인 경우에 A_i 는 A_{i-1} 의 배수)

동전 0

10

<https://www.acmicpc.net/problem/11047>

- 소스: <http://boj.kr/70f6b71fc12542998a71ef16d2b28590>

회의실 배정

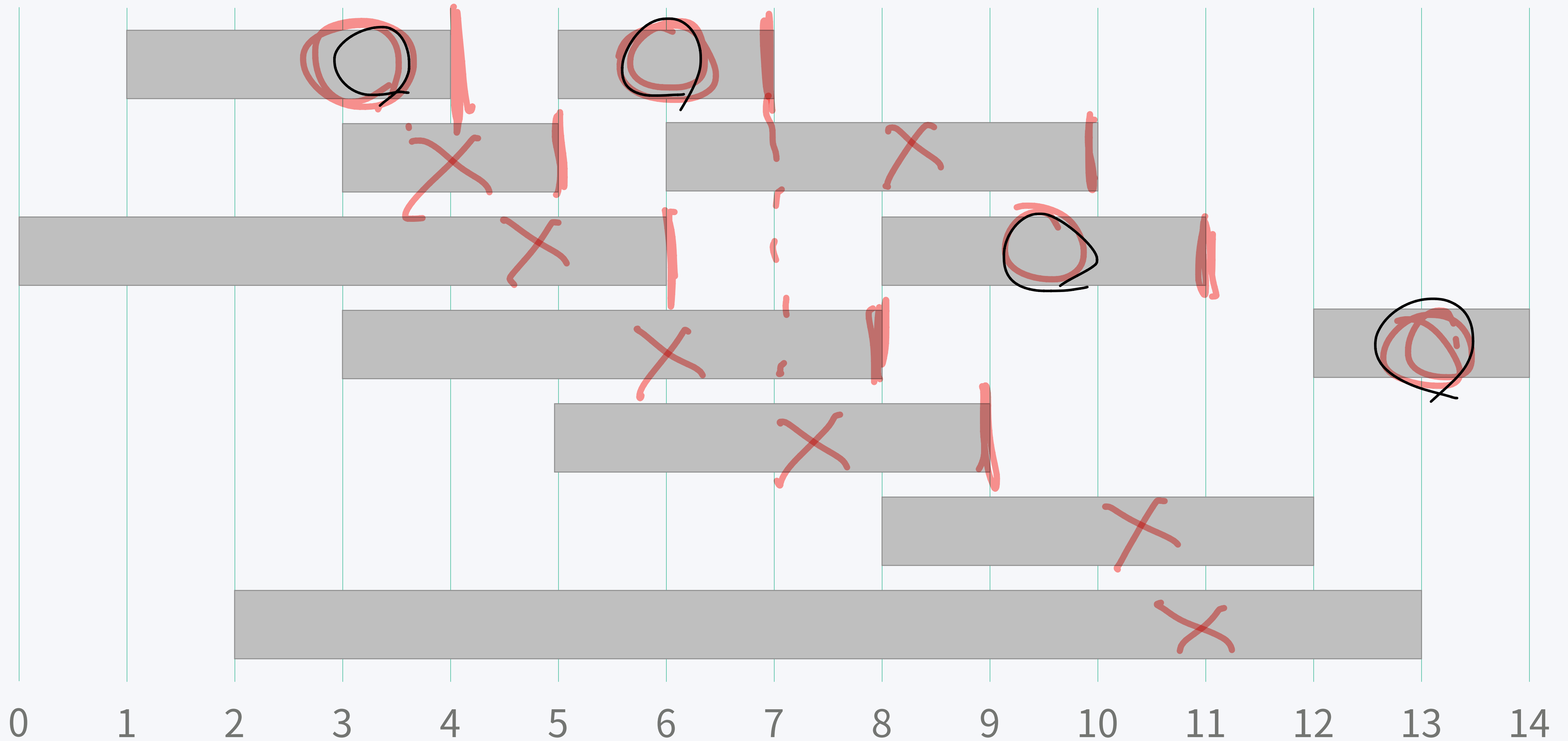
<https://www.acmicpc.net/problem/1931>

- 한 개의 회의실이 있는데 이를 사용하고자 하는 n 개의 회의들에 대하여 회의실 사용표를 만들려고 한다
- 각 회의 i 에 대해 시작시간과 끝나는 시간이 주어져 있고, 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 최대 수의 회의를 찾아라
- 단, 회의는 한번 시작하면 중간에 중단될 수 없으며 한 회의가 끝나는 것과 동시에 다음 회의가 시작될 수 있다
- 회의의 시작시간과 끝나는 시간이 같을 수도 있다
- 이 경우에는 시작하자마자 끝나는 것으로 생각하면 된다.

회의실 배정

12

<https://www.acmicpc.net/problem/1931>

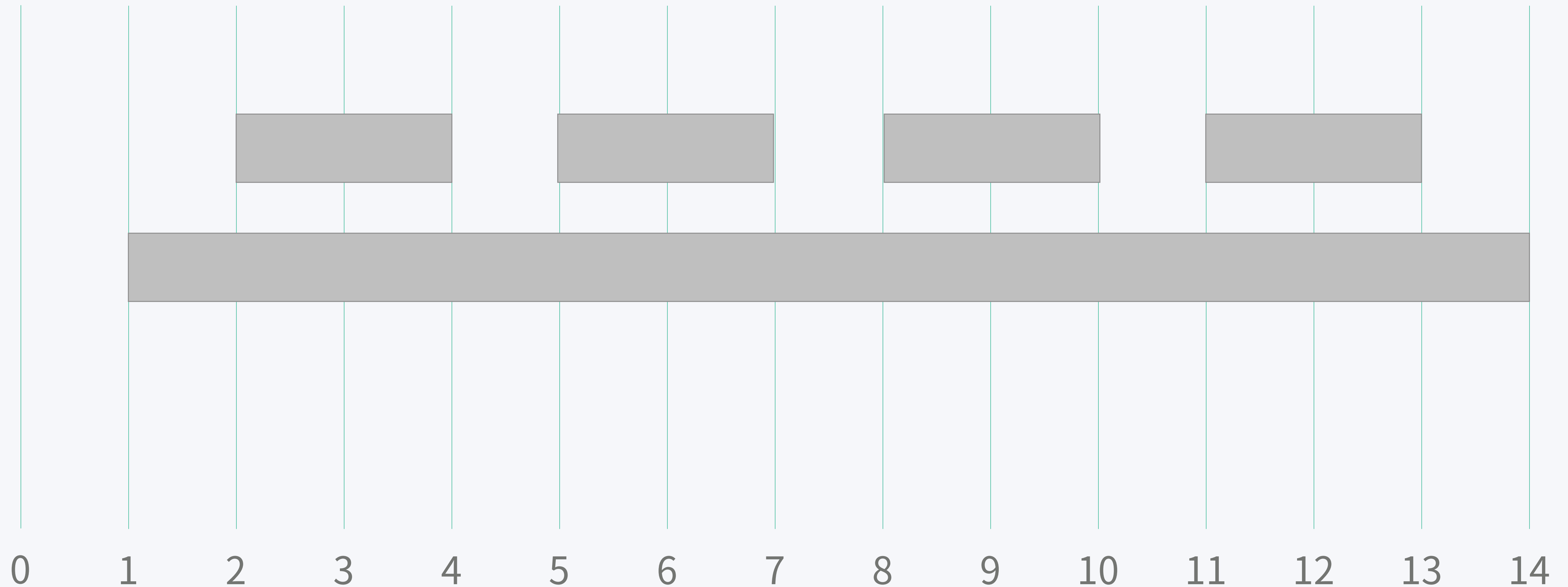


회의실 배정

13

<https://www.acmicpc.net/problem/1931>

- 일찍 시작하는 회의를 배정한다

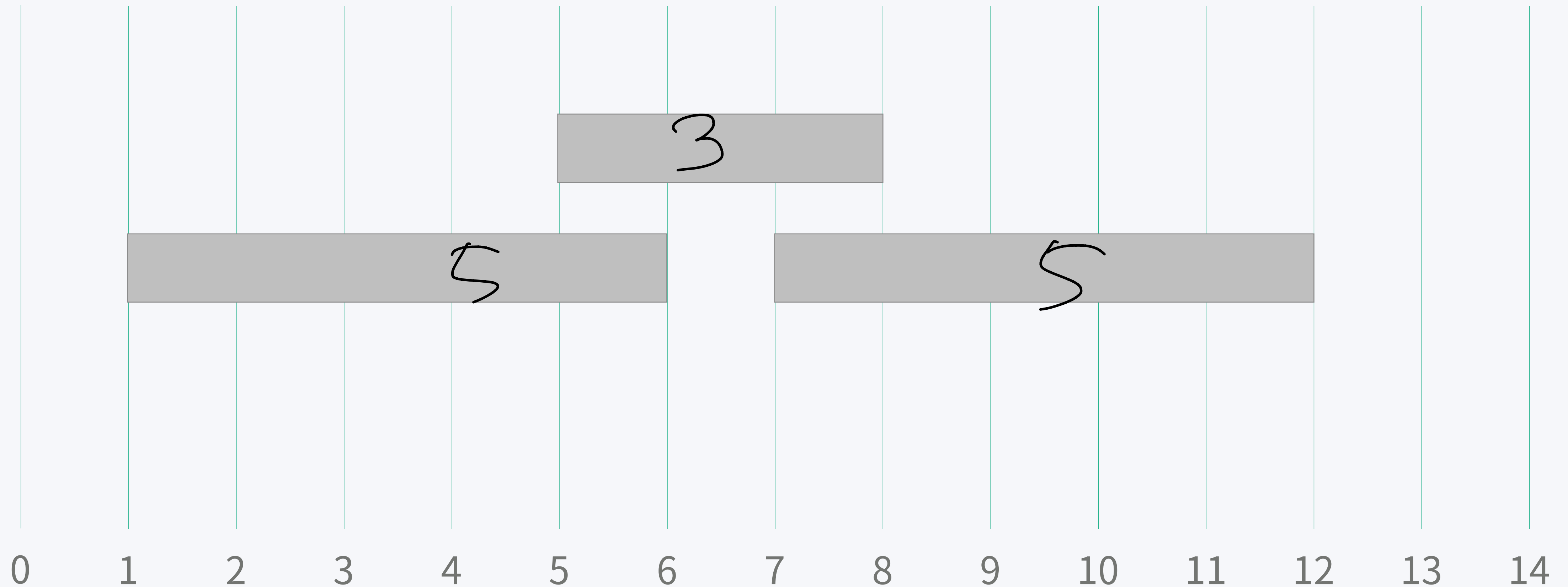


회의실 배정

14

<https://www.acmicpc.net/problem/1931>

- 짧은 회의를 먼저 배정한다.

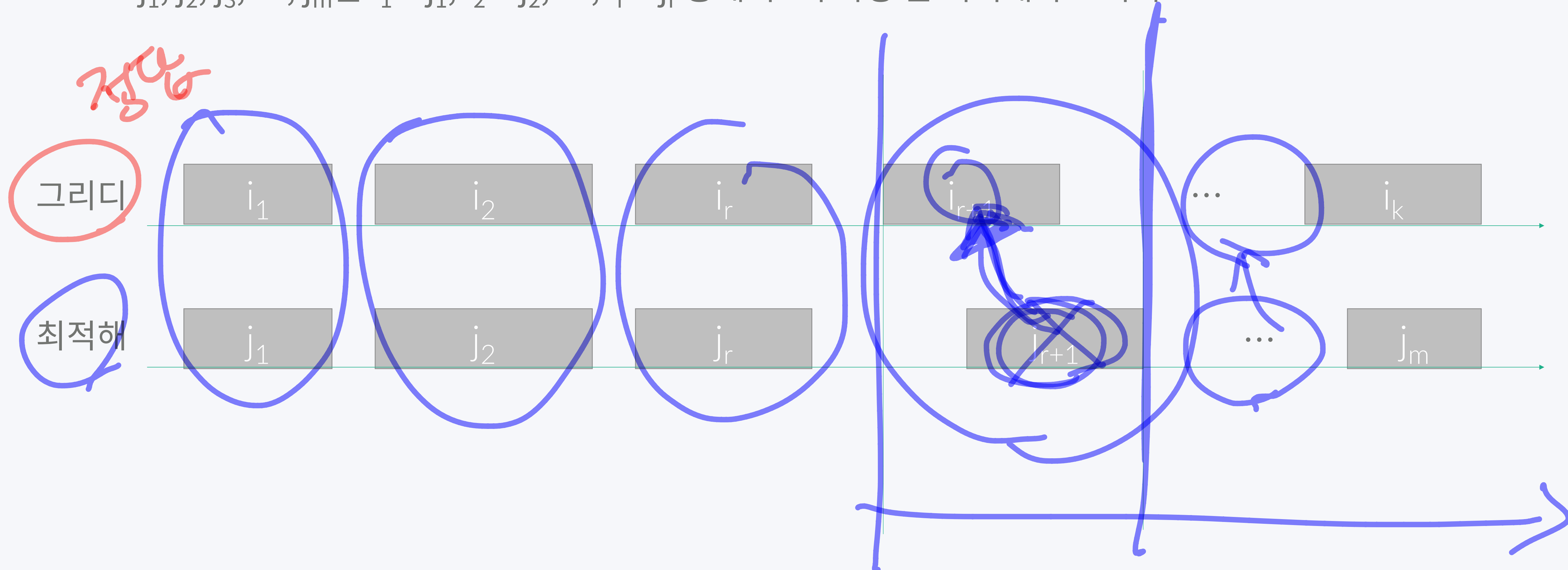


회의실 배정

15

<https://www.acmicpc.net/problem/1931>

- $i_1, i_2, i_3, \dots, i_k$ 를 그리디 알고리즘으로 선택한 정답
- $j_1, j_2, j_3, \dots, j_m$ 을 $i_1 = j_1, i_2 = j_2, \dots, i_r = j_r$ 중에서 r 이 가장 큰 최적해라고 하자



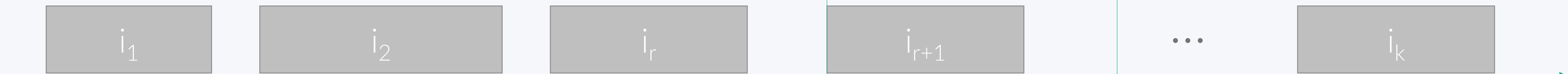
회의실 배정

16

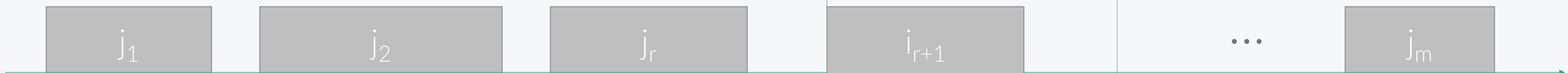
<https://www.acmicpc.net/problem/1931>

- j_{r+1} 보다 일찍 끝나는 i_{r+1} 이 있다고 했을 때 j_{r+1} 을 i_{r+1} 로 바꿔도 정답이다.

그리디



최적해



회의실 배정

<https://www.acmicpc.net/problem/1931>

- 소스: <http://boj.kr/74b997389d2345129d9811f8449b2ffe>

ATM

<https://www.acmicpc.net/problem/11399>

- 인하은행에는 ATM이 1대밖에 없다
- 지금 이 ATM앞에 N명의 사람들이 줄을 서있다
- 사람은 1번부터 N번까지 번호가 매겨져 있으며, i번 사람이 돈을 인출하는데 걸리는 시간은 P_i 분이다.

[41] p[5]

N=5

3, 1, 4, 3, 2

3 4 8 11 13

39분

1, 2, 3, 3, 4
1 3 6 9 13 = 32

3, 1, 2, 3, 4

3 4 6 9 13 = 35분

ATM

<https://www.acmicpc.net/problem/11399>

- 사람들이 줄을 서는 순서에 따라서, 돈을 인출하는데 필요한 시간의 합이 달라지게 된다
- 예를 들어, 총 5명이 있고, $P1 = 3$, $P2 = 1$, $P3 = 4$, $P4 = 3$, $P5 = 2$ 인 경우를 생각해보자.
- $[1, 2, 3, 4, 5]$ 순서로 줄을 선다면, 1번 사람은 3분만에 돈을 뽑을 수 있다.
- 2번 사람은 1번 사람이 돈을 뽑을 때 까지 기다려야 하기 때문에, $3+1 = 4$ 분이 걸리게 된다.
- 3번 사람은 1번, 2번 사람이 돈을 뽑을 때까지 기다려야 하기 때문에, 총 $3+1+4 = 8$ 분이 필요하게 된다.
- 4번 사람은 $3+1+4+3 = 11$ 분, 5번 사람은 $3+1+4+3+2 = 13$ 분이 걸리게 된다.
- 이 경우에 각 사람이 돈을 인출하는데 필요한 시간의 합은 $3+4+8+11+13 = 39$ 분이 된다.

ATM

<https://www.acmicpc.net/problem/11399>

- 줄을 서 있는 사람의 수 N 과 각 사람이 돈을 인출하는데 걸리는 시간 P_i 가 주어졌을 때, 각 사람이 돈을 인출하는데 필요한 시간의 합의 최소값을 구하는 문제

ATM

<https://www.acmicpc.net/problem/11399>

- 기다리는 시간이 짧은 사람부터 ATM을 인출하는 것이 좋다.

ATM

<https://www.acmicpc.net/problem/11399>

- $p_1, p_2, p_3, \dots, p_n$
- $p_1 \leq p_2 \leq p_3 \leq \dots \leq p_n$ 이 정답이라고 가정
- 총 돈을 인출하는데 걸리는 시간의 합
- $S = \underbrace{(p_1) + (p_1 + p_2) + (p_1 + p_2 + p_3) + \dots + (p_1 + p_2 + \dots + p_n)}$
- $S = n * p_1 + (n-1) * p_2 + \dots + p_n$

ATM

<https://www.acmicpc.net/problem/11399>

23

$$T < J$$

$$P_i \quad P_j$$

- $p_1 \leq p_2 \leq p_3 \leq \dots \leq p_n$ 이 정답이라면, 중간에 $i < j$ 인 p_i 와 p_j 의 순서를 바꿨을 때, 더 커져야 한다.

$$S \leq S' \quad S - S' \leq 0$$

- 총 돈을 인출하는데 걸리는 시간의 합
- $S = n * p_1 + \dots + (n-(i-1)) * p_i + \dots + (n-(j-1)) * p_j + \dots + p_n$
- p_i 와 p_j 의 순서를 바꿨을 때 시간의 합
- $S' = n * p_1 + \dots + (n-(i-1)) * p_j + \dots + (n-(j-1)) * p_i + \dots + p_n$

ATM

<https://www.acmicpc.net/problem/11399>

- S 가 정답이기 때문에, $S \leq S'$ 를 만족해야 한다.
- $S \leq S'$ 라면 $S - S' \leq 0$ 이 되어야 한다.

ATM

25

<https://www.acmicpc.net/problem/11399>

- $S - S' = (n-i+1) * p_i + (n-j+1) * p_j - (n-i+1) * p_j - (n-j+1) * p_i$
- $= (n-i+1-n+j-1)p_i + (n-j+1-n+i-1) * p_j$
- $= (-i+j)p_i + (-j+i)p_j$
- $= (j-i)p_i + (i-j)p_j$
- $= -(i-j)p_i + (j-j)p_j$
- $= \underbrace{(i-j)(p_i - p_j)}_{S-S' \leq 0} \leq 0 \quad ?$

$$\boxed{\begin{array}{l} \tau < j \quad p_i \leq p_j \\ (\tau - j) < 0 \\ p_j - p_i \geq 0 \\ \hline \leq 0 \end{array}}$$

ATM

<https://www.acmicpc.net/problem/11399>

- $S - S' = (i - j)(p_j - p_i)$
- 여기서 $i < j$ 이기 때문에, $i - j < 0$ 이다.
- $p_i \leq p_j$ 이기 때문에 $0 \leq p_j - p_i$ 이다.
- $i - j$ 는 음수이고, $p_j - p_i$ 는 양수 또는 0이기 때문에
- $S - S' \leq 0$ 이다.
- 따라서 오름차순이 정답이다.

non-decreasing
⑤

ATM

<https://www.acmicpc.net/problem/11399>

- 소스: <http://boj.kr/b37737afd7334d4fbab483b529f958a0>

잃어버린 괄호

28

<https://www.acmicpc.net/problem/1541>

- 식에 괄호를 적절히 쳐서 식의 값을 최소로 만드는 문제

+ -

- $1+2+3-4-(5+6+7+8)-(9+10)-11-(12+13+14+15)$

잃어버린 괄호

<https://www.acmicpc.net/problem/1541>

- 식에 괄호를 적절히 쳐서 식의 값을 최소로 만드는 문제
- -가 나오면, 항상 뒤의 식을 모두 -로 만들 수 있다.
- $1+2+3-4-5+6+7+8-9+10-11-12+13+14+15$
- $1+2+3-4-(5+6+7+8)-(9+10)-11-(12+13+14+15)$

읽어버린 괄호

30

<https://www.acmicpc.net/problem/1541>

- 소스: <http://boj.kr/1234a5ff1e354651a3b0dc25ae97d8d7>

수 묶기

<https://www.acmicpc.net/problem/1744>

- 길이가 N인 수열에서 두 수를 적절히 묶어서 수열의 합을 최대로 하는 문제
- 수의 순서를 바꿔도 상관없다
- 같은 위치에 있는 수를 묶는 것은 불가능하고
- 각 수는 단 한 번만 묶거나 묶지 않아야 한다
- 묶은 후에는 두 수의 곱이 수가 됨
- 이 때 최대 찾기

수 묶기

<https://www.acmicpc.net/problem/1744>

- 0, 1, 2, 4, 3, 5 인 경우
- 4와 5 를 묶고 2와 3을 묶으면
- $0 + 1 + (4*5) + (2*3) = 27$

수 묶기

<https://www.acmicpc.net/problem/1744>

- 양수는 큰 수끼리
- 음수는 작은 수끼리
- 0은 묶지 않는 것이 최대

수 묶기

<https://www.acmicpc.net/problem/1744>

- 주의해야할 점
- 1은 묶지 않는 것이 좋다
- 묶이지 않는 음수가 있는 경우 0을 이용할 수 있다

수 묶기

35

<https://www.acmicpc.net/problem/1744>

- 소스: <http://boj.kr/5271dc4cae604505a05a3f0dabfa9b6e>

대회 or 인턴

<https://www.acmicpc.net/problem/2875>

- 여학생 N명, 남학생 M명이 있다
- 1팀: 여2, 남1
- K명은 인턴에 참가해야 한다
- 몇 팀을 만들 수 있을까?

대회 or 인턴

<https://www.acmicpc.net/problem/2875>

- 한 팀을 만드려면 다음과 같은 조건을 만족해야 한다
- 여학생이 2명 이상
- 남학생이 1명 이상
- $M+N \geq K+3$
 - 팀은 3명이고, 인턴은 K명이 해야하기 때문

대회 or 인턴

38

<https://www.acmicpc.net/problem/2875>

- 소스: <http://boj.kr/9bc8a0b9047b462e9b8c7392d50c835c>

30

<https://www.acmicpc.net/problem/10610>

- 어떤 수 N 이 주어졌을 때, 숫자를 섞어 30의 배수로 만드는 문제
- 이 때, 가장 큰 수를 만들어야 함
- $N = 30$, 답 = 30
- $N = 102$, 답 = 210
- $N = 2931$, 답 = 불가능

N

2763

10, 3

120

0도 있고, 30의 배수

2831

14

30

<https://www.acmicpc.net/problem/10610>

- 30은 $2 \times 3 \times 5$ 이다
- 즉, N이 30으로 나누어 떨어지려면, 2, 3, 5로 나누어 떨어져야 한다
- 2로 나누어 떨어지는 수
 - 마지막 자리가 짝수
- 3으로 나누어 떨어지는 수
 - 자리의 합이 3으로 나누어 떨어져야 함
- 5로 나누어 떨어지는 수
 - 마지막 자리가 0 또는 5
- 30으로 나누어 떨어지는 수
 - 마지막 자리가 0이면서 자리의 합이 3으로 나누어 떨어져야 함

<https://www.acmicpc.net/problem/10610>

- 자리의 합은 변하지 않기 때문에, 마지막 자리만 0으로 만들어주면 되는 문제
- 가장 큰 수이기 때문에, 그냥 내림차순 정렬을 하면 되는 문제

<https://www.acmicpc.net/problem/10610>

- 소스: <http://boj.kr/0fbad6ee0749420086e1eb530f103fa0>

병든 나이트

<https://www.acmicpc.net/problem/1783>

43

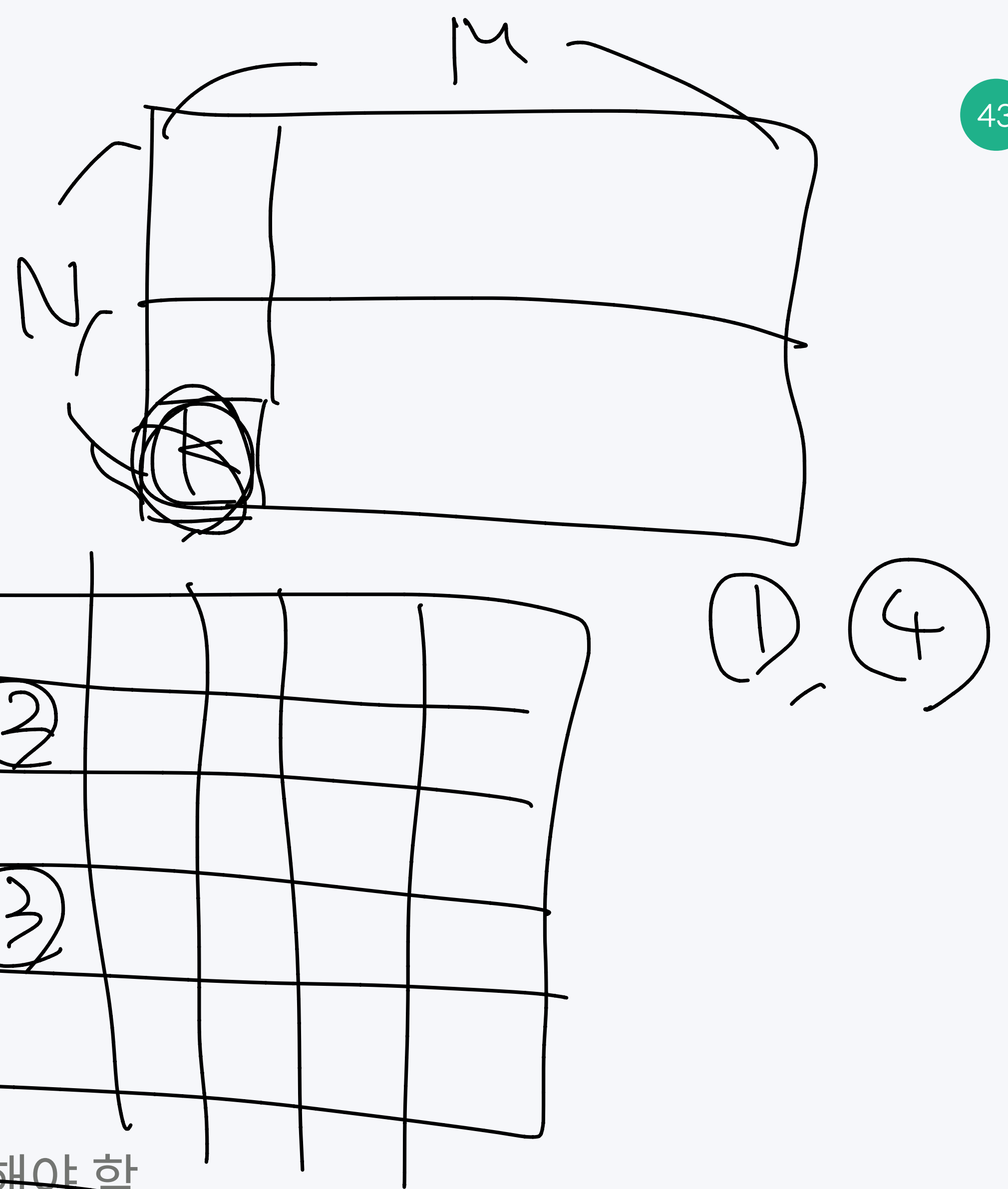
- $N \times M$ 크기의 체스판 가장 왼쪽 아래칸에 나이트가 있다
- $1 \leq N, M \leq 2,000,000,000$

• 이동할 수 있는 방법

1. 2칸 위로, 1칸 오른쪽
2. 1칸 위로, 2칸 오른쪽
3. 1칸 아래로, 2칸 오른쪽
4. 2칸 아래로, 1칸 오른쪽

• 이동 횟수가 4번 초과면 위의 방법을 모두 1번씩은 이용해야 함

• 이 때, 방문할 수 있는 칸의 최대 개수



병든 나이트

<https://www.acmicpc.net/problem/1783>

1. height = 1인 경우

2. height = 2인 경우

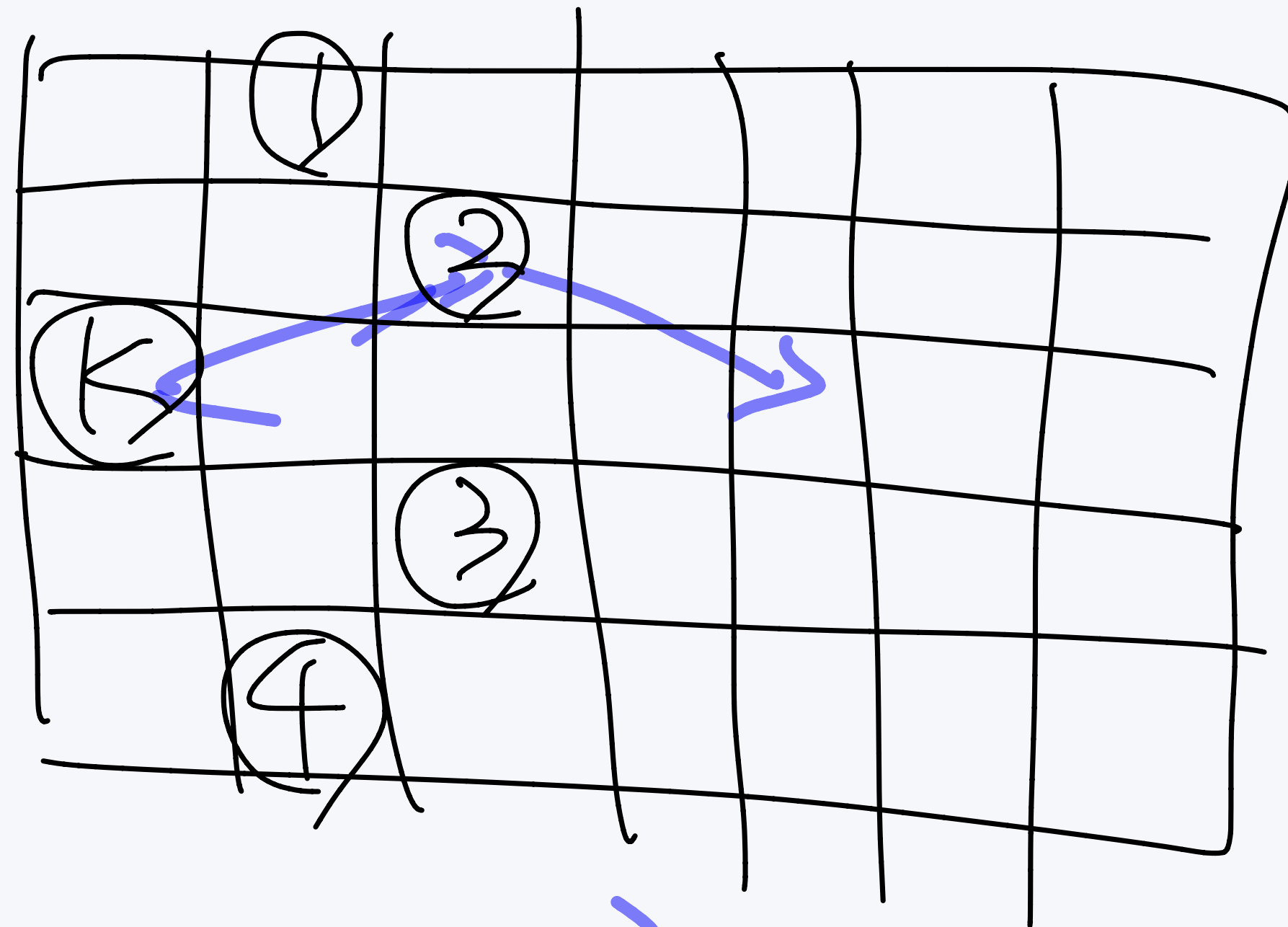
3. height ≥ 3 보다 큰 경우

2, 3

1, 2, 3, 4

> ①

min (4,)



병든 나이트

45

<https://www.acmicpc.net/problem/1783>

1. $\text{height} = 1$ 인 경우
 - 움직일 수 없기 때문에 정답은 1
2. $\text{height} = 2$ 인 경우
3. $\text{height} \geq 3$ 보다 큰 경우

병든 나이트

<https://www.acmicpc.net/problem/1783>

1. height = 1인 경우

- 움직일 수 없기 때문에 정답은 1

2. height = 2인 경우

- 두 가지 방법만 사용할 수 있다. (+2, +1), (+2, -1)
- 따라서, 정답은 $\min(4, (\text{width} + 1) / 2)$
- 이동 횟수 제한 때문에 4가 필요함

3. height ≥ 3 보다 큰 경우

병든 나이트

<https://www.acmicpc.net/problem/1783>

1. height = 1인 경우

- 움직일 수 없기 때문에 정답은 1

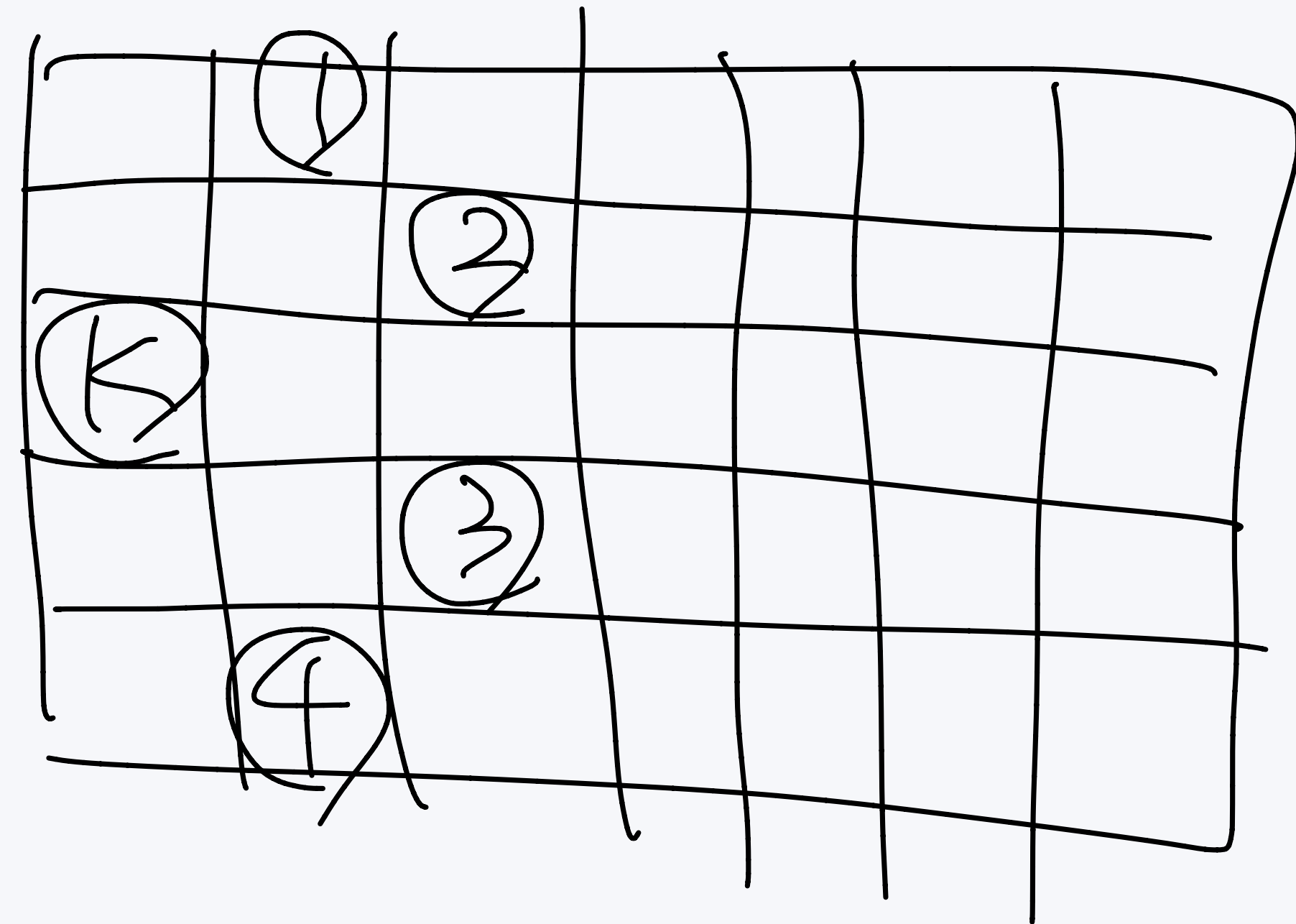
2. height = 2인 경우

- 두 가지 방법만 사용할 수 있다. $(+2, +1)$, $(+2, -1)$
- 따라서, 정답은 $\min(4, (\text{width} + 1) / 2)$
- 이동 횟수 제한 때문에 4가 필요함

3. height ≥ 3 보다 큰 경우

1. width ≥ 7

2. width < 7



병든 나이트

48

<https://www.acmicpc.net/problem/1783>

3. $\text{height} \geq 3$ 보다 큰 경우

1. $\text{width} \geq 7$

- 4가지 방법을 모두 사용하면 $(7, 1)$ 으로 이동한다
- 여기서부터 $(+1, +2)$ 와 $(+1, -2)$ 를 번갈아가면서 사용
- $\text{width} - 2$

2. $\text{width} < 7$

병든 나이트

<https://www.acmicpc.net/problem/1783>

3. $\text{height} \geq 3$ 보다 큰 경우

1. $\text{width} \geq 7$

2. $\text{width} < 7$

- 4가지 방법을 모두 사용할 수 없다
- $(+1, +2)$ 와 $(+1, -2)$ 를 번갈아 가면서 사용할 수 있다
- $\min(4, \text{width})$

병든 나이트

50

<https://www.acmicpc.net/problem/1783>

- 소스: <http://boj.kr/4e68062654bf430ea9ea2606f80e48bc>

NMK

$N=5, M=1, K=1$

가장 긴 증가하는 부분수열

51

<https://www.acmicpc.net/problem/1201>

- 1부터 N까지의 수를 한 번씩 이용해서 최대 부분 증가 수열의 길이가 M이고, 최대 부분 감소 수열의 길이가 K인 수열을 구하는 문제

$1 \sim N = 5$

$N=5, M=3, K=2$

2 1 5 3 4

2 1 4 5 3

NMK

<https://www.acmicpc.net/problem/1201>

52

$$\boxed{M+K-1} \leq N \leq ?$$

- 불가능한 경우 찾기

M증가 K감소

- 적어도 M개의 정수는 증가 수열에 포함되어야 하고
- 적어도 K개의 정수는 감소 수열에 포함되어야 한다
- 두 수열은 최대 정수 1개를 공유할 수 있기 때문에
- $N \geq M+K-1$ 이어야 한다



NMK

53

<https://www.acmicpc.net/problem/1201>

- 불가능한 경우 찾기
- 또, N 은 MK 를 넘을 수 없다
- $N = MK+1$ 인 경우에 길이가 $M+1$ 인 증가 수열이나 길이가 $K+1$ 인 감소 수열을 반드시 만들 수 있다.
- 비둘기집 원리로 증명할 수 있음
- Erdős-Szekeres Theorem
- <http://mathworld.wolfram.com/Erdos-SzekeresTheorem.html>



7/6/20

1 ~ N

$\{M\} \geq \frac{2}{3}$

72 (54)

$$N=13, \quad M=5, \quad K=4$$

-
- 1 2 3/4 5 6 7/8 9/10 11 12/13
- 3 2 1/2 6 5 4 9 8 12 11 10 13
- K27 K24 K27

<https://www.acmicpc.net/problem/1201>

1. 1부터 N 까지 수를 오름차순으로 적는다
2. 수를 M 등분 한다. 이 때, 그룹에 들어있는 수는 K 보다 작거나 같아야 하며, 적어도 한 그룹은 들어있는 수의 개수가 K 이어야 한다
3. 각 그룹에 들어있는 수의 순서를 뒤집는다
4. 끝

NMK

56

<https://www.acmicpc.net/problem/1201>

- $N = 13, M = 5, K = 4$ 인 경우

1. 1 2 3 4 5 6 7 8 9 10 11 12 13

2. [1 2 3] [4] [5 6 7 8] [9 10] [11 12 13]

3. [3 2 1] [4] [8 7 6 5] [10 9] [13 12 11]

- 사전 순으로 가장 앞서는 순열을 찾는 경우에는
- [1] [2] [5 4 3] [9 8 7 6] [13 12 11 10]
- 와 같이 나누어야 한다

NMK

57

<https://www.acmicpc.net/problem/1201>

- 소스: <http://boj.kr/49999b496b574266a199c493fcca4f76>

행렬

$N, M \leq 50$

$N \times M$ 행렬 A, B

$(N-2) \times (M-2)$

<https://www.acmicpc.net/problem/1080>

- 0과 1로만 이루어진 행렬 A와 행렬 B가 있다. 이 때, 행렬 A를 행렬 B로 바꾸는데 필요한 연산의 횟수의 최소값을 구하는 문제

이dea 3x3 $0 \rightarrow 1$
 $1 \rightarrow 0$

26번

- 행렬을 변환하는 연산은 어떤 3×3 크기의 부분 행렬에 있는 모든 원소를 뒤집는 것이다. ($0 \rightarrow 1$, $1 \rightarrow 0$)

A

0 0 0 0
0 0 1 0
0 0 0 0

2500

2

B
1 0 0 1
1 0 1 1
1 0 0 1

1 1 1 0
1 1 0 0
1 1 0 0

4

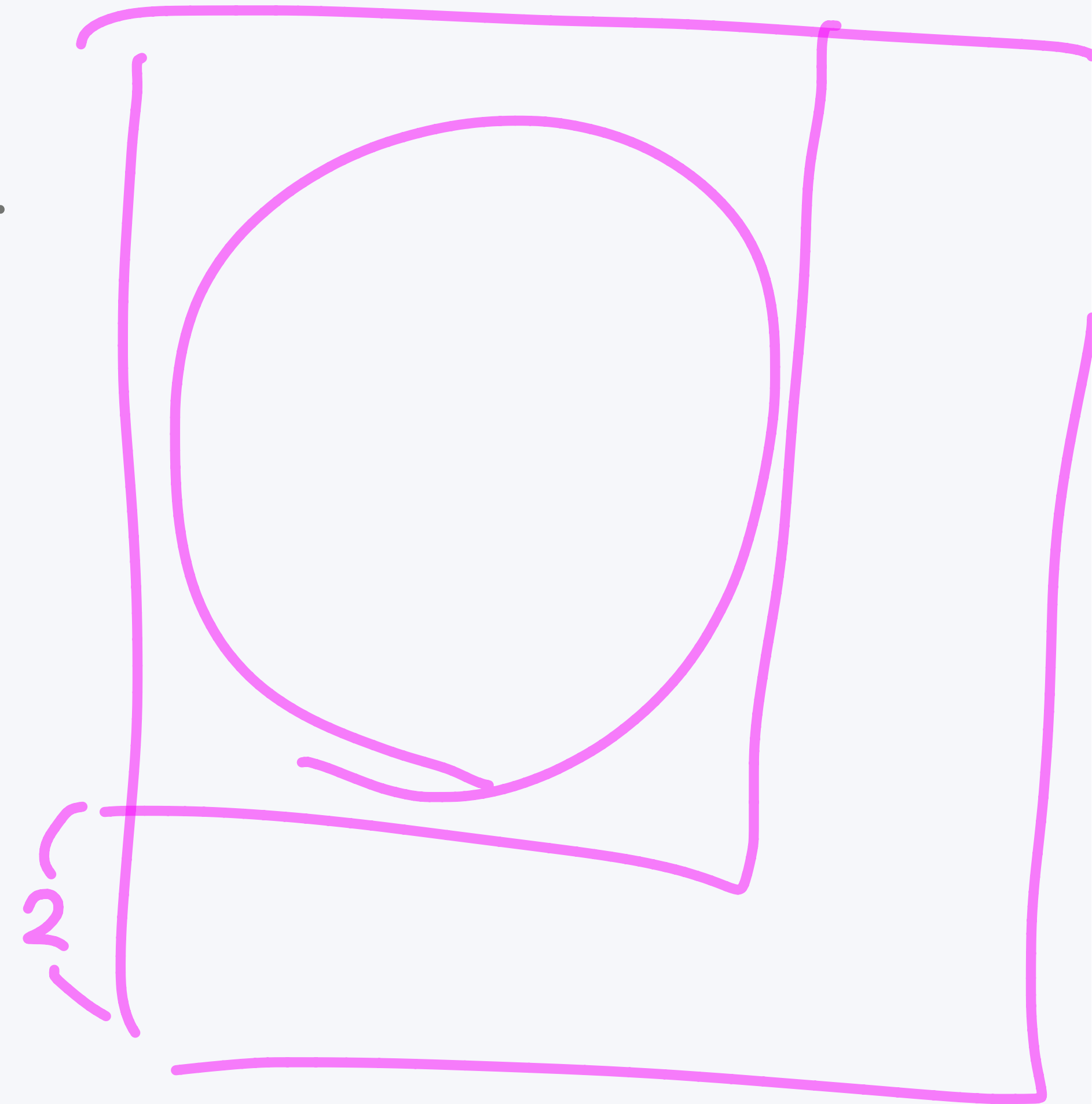
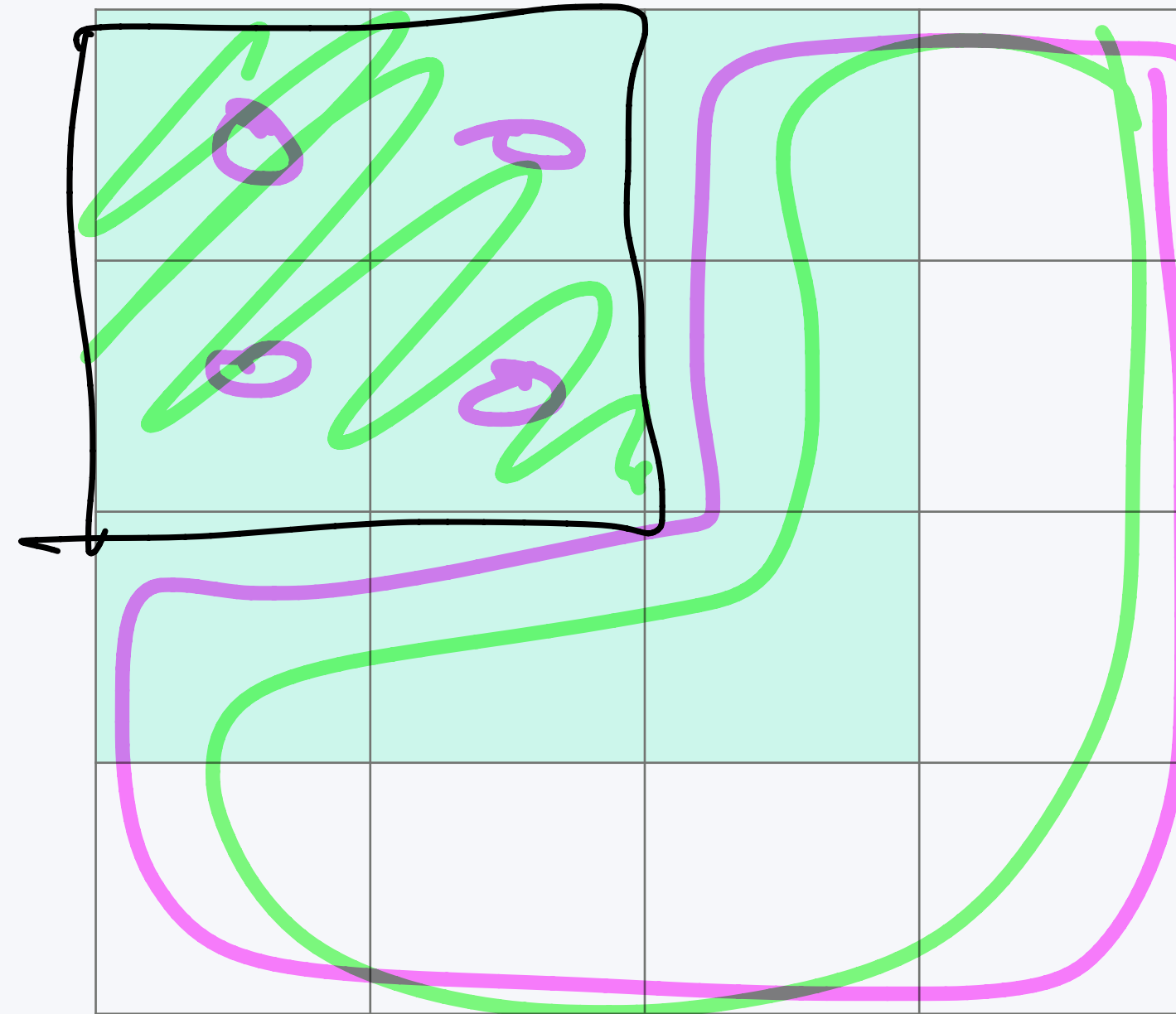
행렬

$O(NM)$

59

<https://www.acmicpc.net/problem/1080>

- $(0, 0)$ 을 바꿀 수 있는 칸은 $(0, 0) \sim (2, 2)$ 를 뒤집는 것 밖에 없다.



행렬

<https://www.acmicpc.net/problem/1080>

- $(0, 1)$ 을 바꿀 수 있는 칸은 $(0, 0) \sim (2, 2)$ 와 $(0, 1) \sim (2, 3)$ 을 뒤집는 것 밖에 없다.

<https://www.acmicpc.net/problem/1080>

- 모든 (i, j) ($0 \leq i < N-2$, $0 \leq j < M-2$) 에 대해서, row-major 순서로 순회하면서, 다른 경우에 $(i, j) \sim (i+2, j+2)$ 를 뒤집어 주면 된다.

행렬

<https://www.acmicpc.net/problem/1080>

- 소스: <http://boj.kr/dede7c911c684fd092af5b7804d53af3>

전구와 스위치

N 개 전구

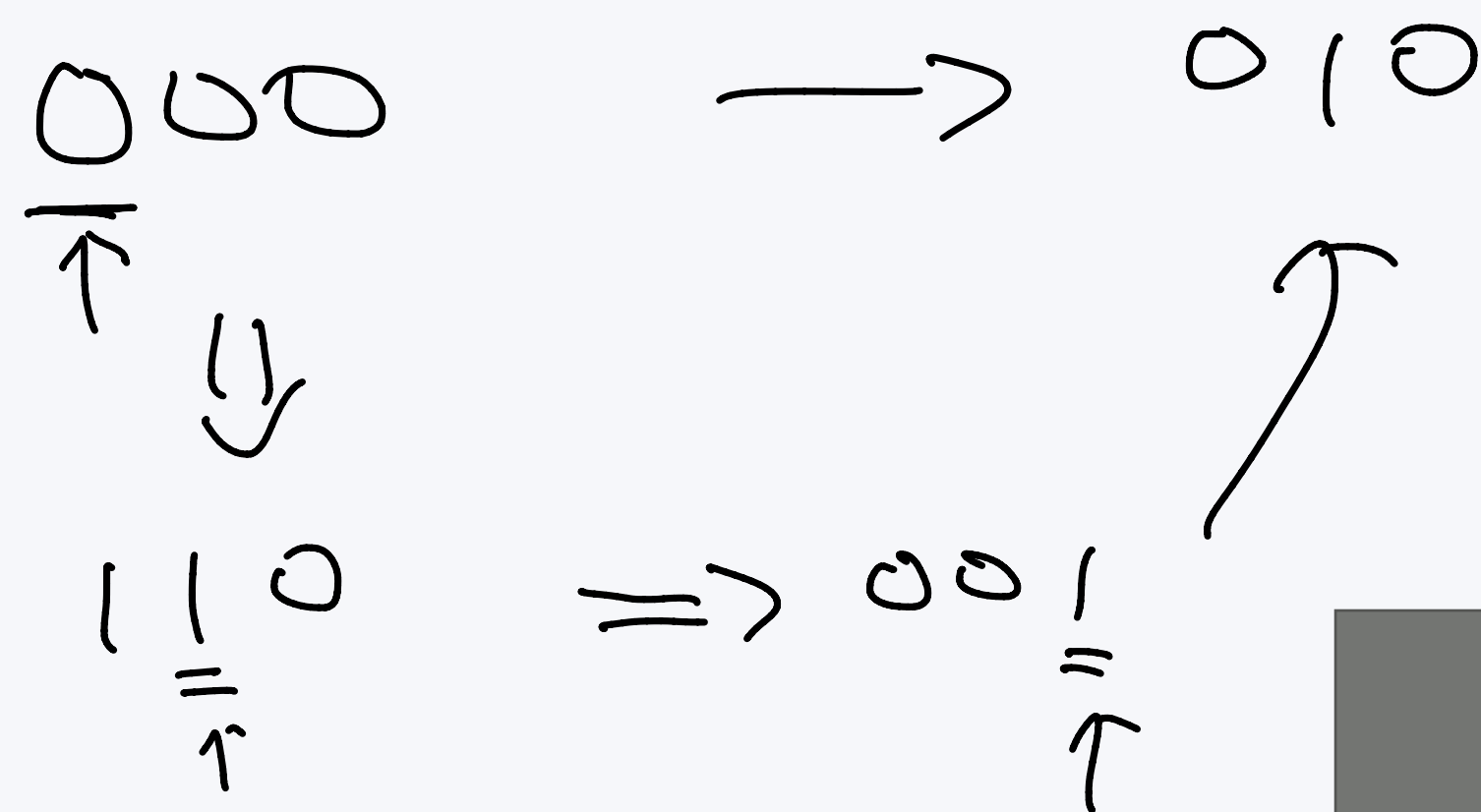
\longleftrightarrow
 $1 \ 2 \ \dots \ N-1 \ N$

63

<https://www.acmicpc.net/problem/2138>

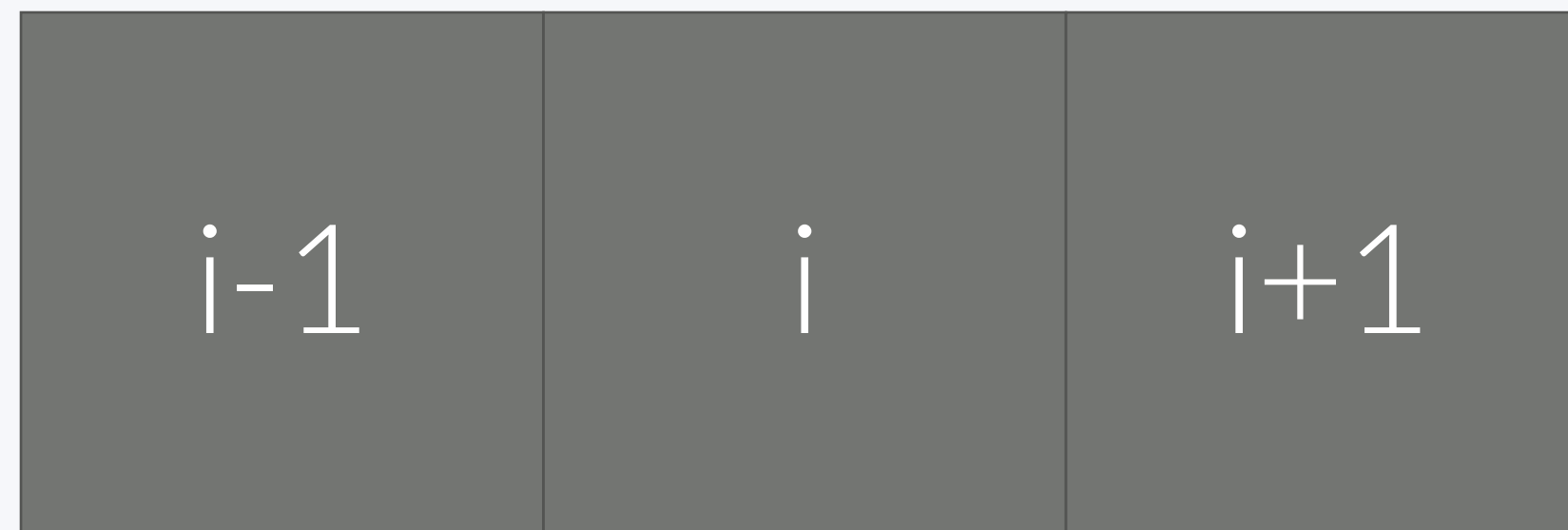
(i 번 \rightarrow ~~$1, 2$~~)

- i 번 스위치를 누르면, $i-1, i, i+1$ 번 전구의 상태가 바뀐다



i 번 스위치

$i-1$ i $i+1$



A B

최소

$O(N)$

$N \leq 10^4$ 만

전구와 스위치

<https://www.acmicpc.net/problem/2138>

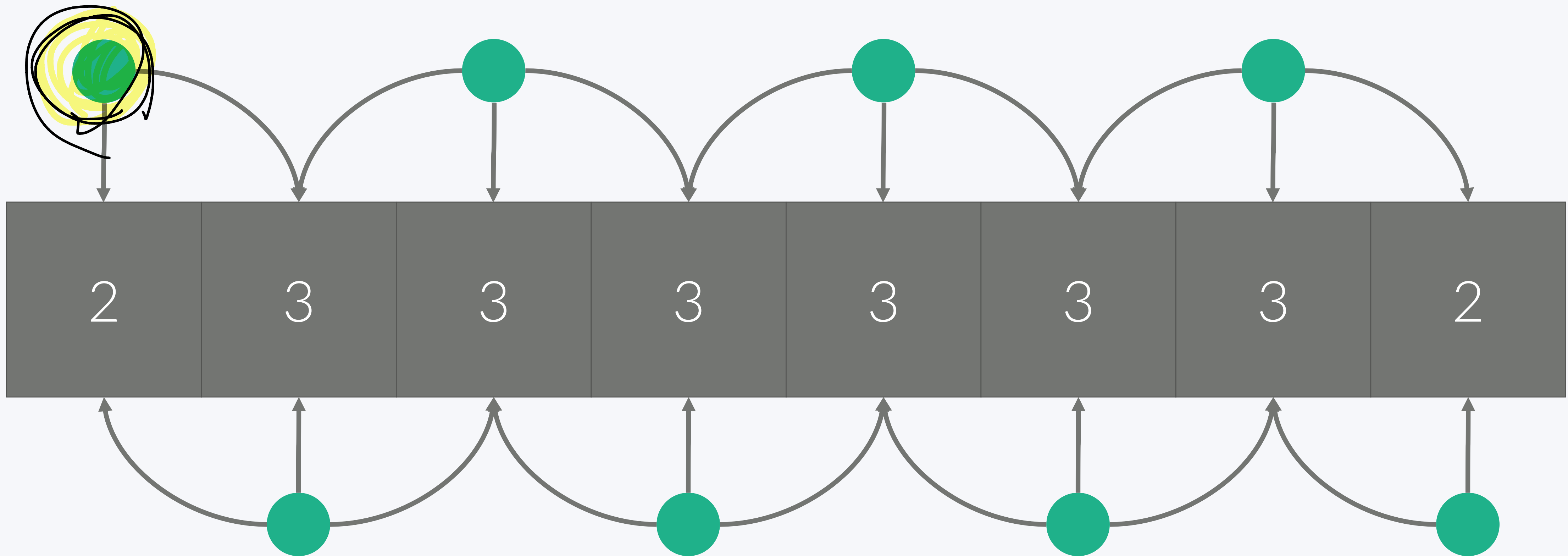
64

- i 번 스위치를 누르면, $i-1, i, i+1$ 번 전구의 상태가 바뀐다

0, 1

0 : 스위치
□ : 전구
0, 1

□ : 전구
스 : 전구의 상태를
바꿀 수 있는 스위치



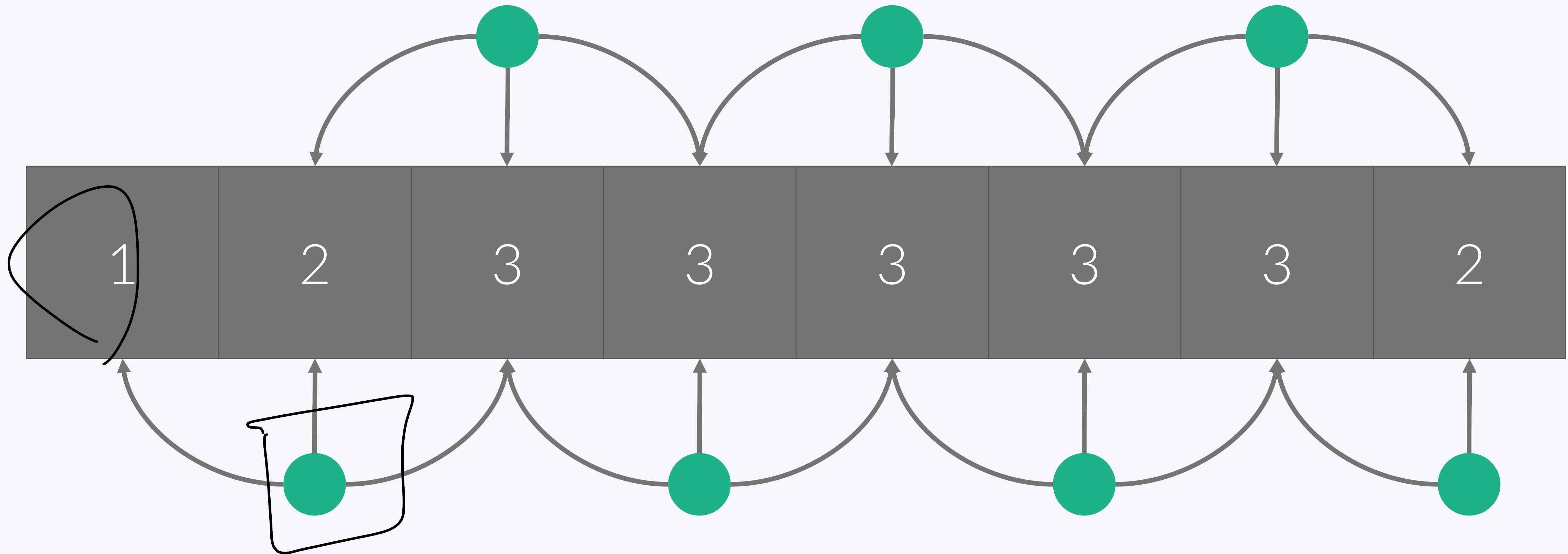
전구와 스위치

<https://www.acmicpc.net/problem/2138>

A \Rightarrow 13

65

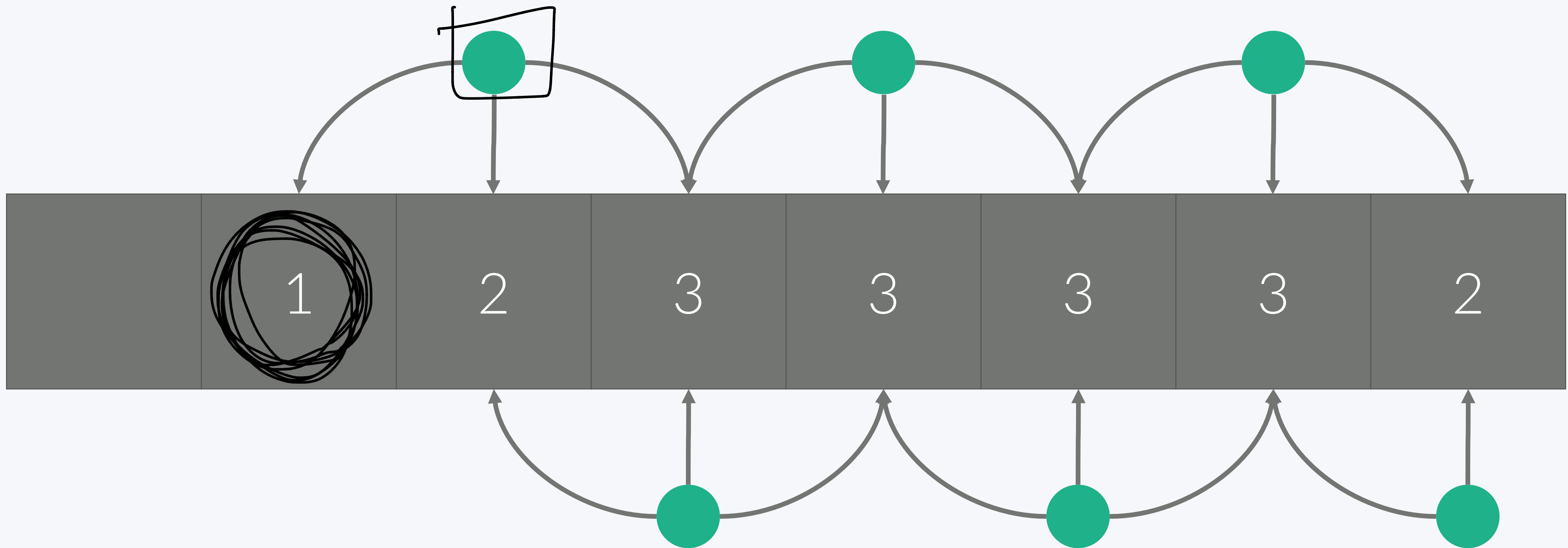
- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다



전구와 스위치

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다

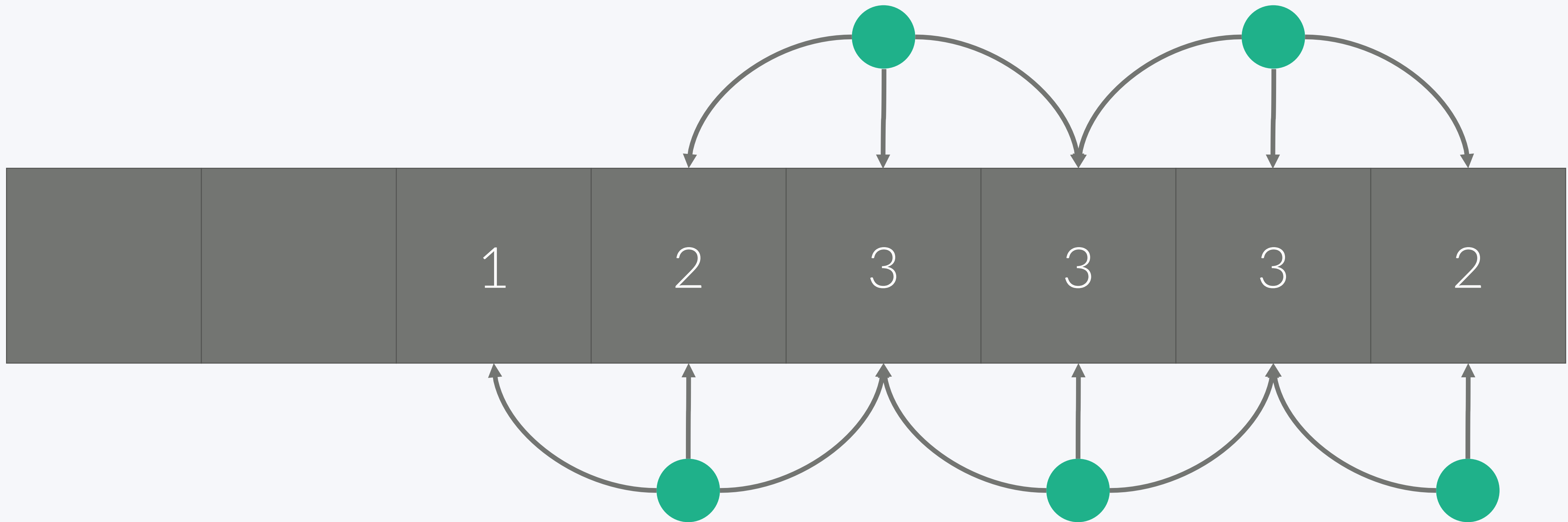


전구와 스위치

67

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다

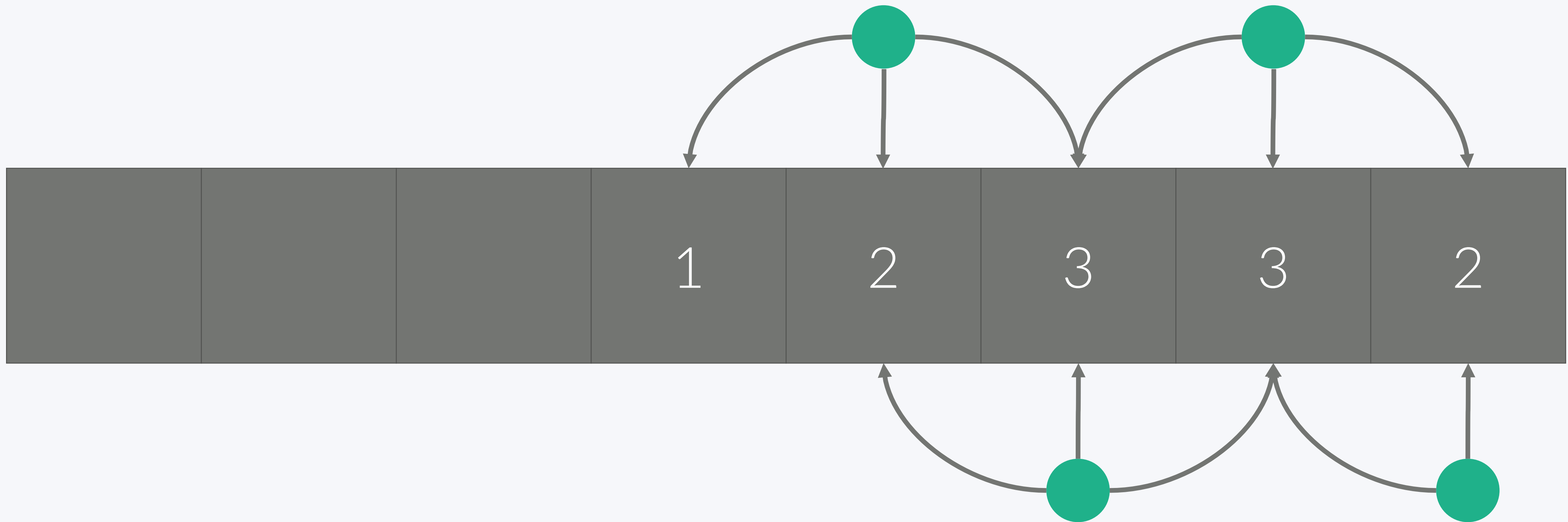


전구와 스위치

68

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다

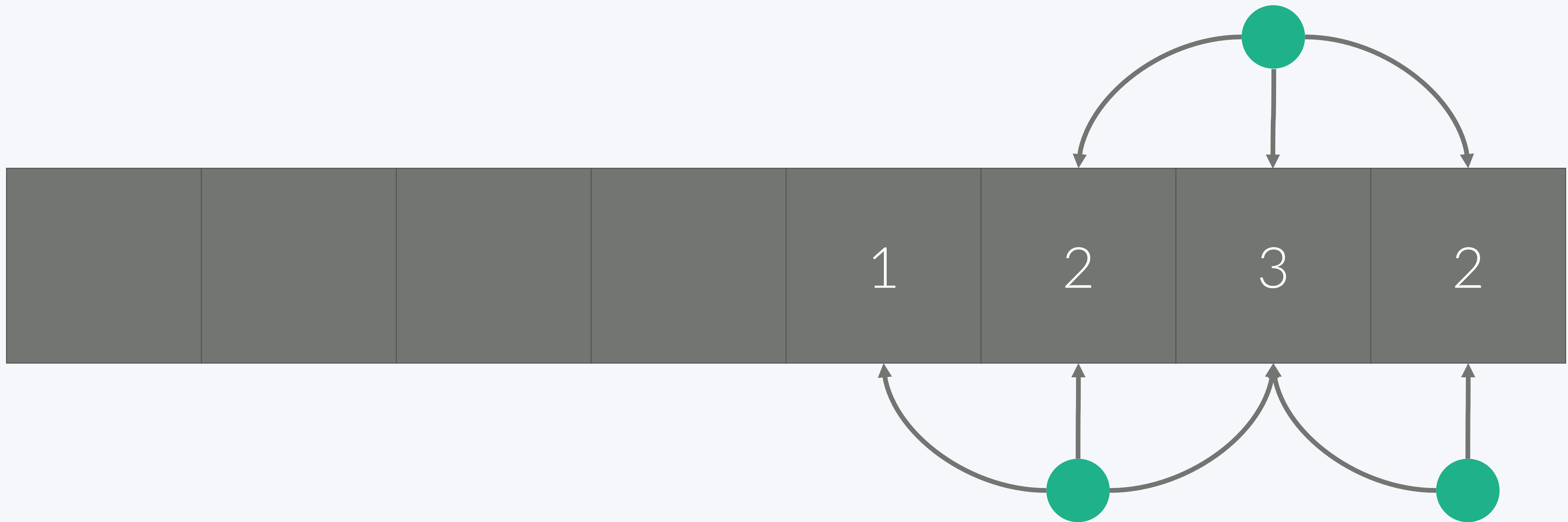


전구와 스위치

69

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다



전구와 스위치

70

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다

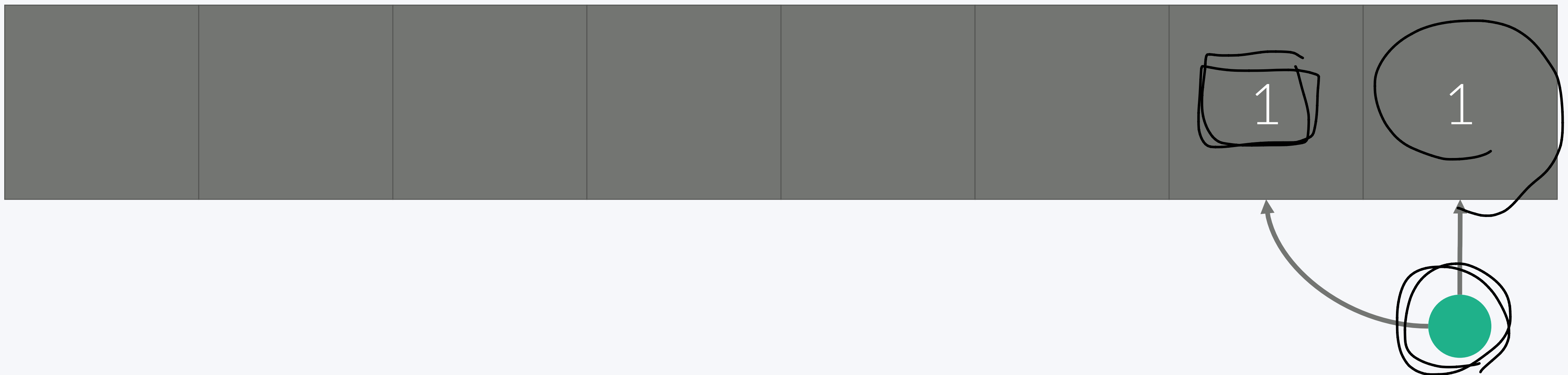


전구와 스위치

71

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다

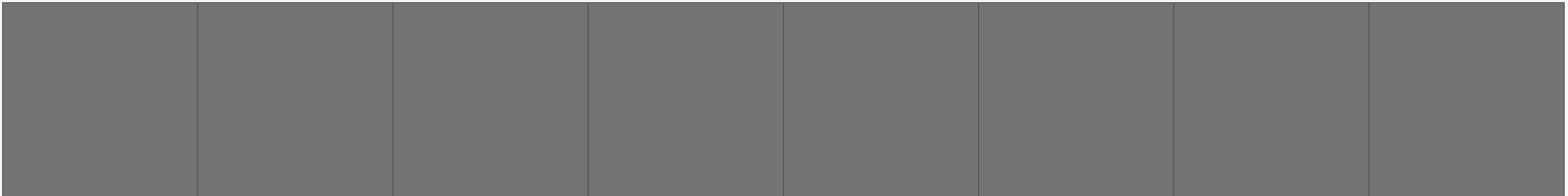


전구와 스위치

72

<https://www.acmicpc.net/problem/2138>

- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다

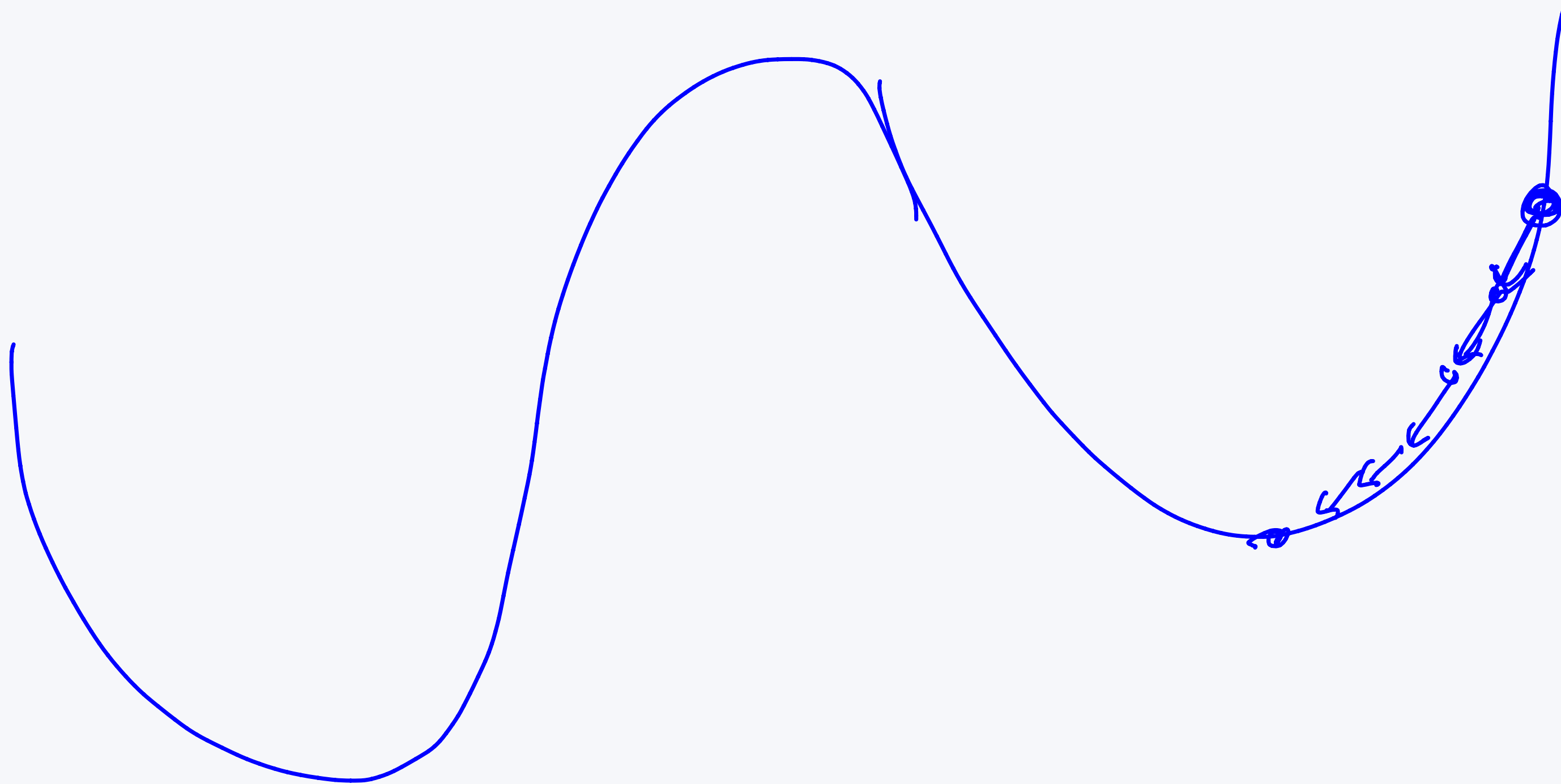


전구와 스위치

73

<https://www.acmicpc.net/problem/2138>

- 소스: <http://boj.kr/b19ee6782f3e4e24ae70806f51c4513e>

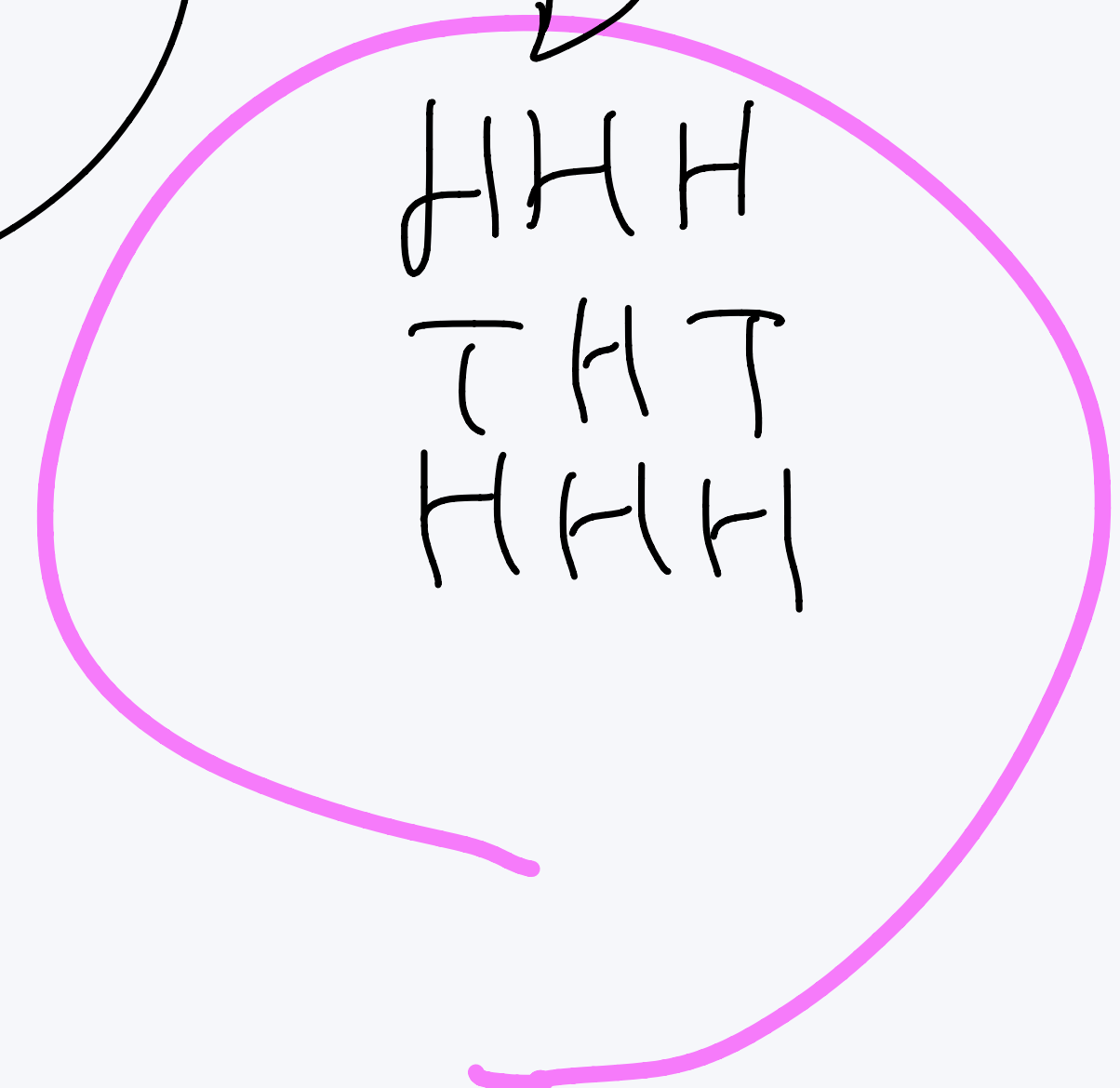


동전 뒤집기

<https://www.acmicpc.net/problem/1285>

- N^2 개의 동전이 N 행 N 열로 놓여져 있다.
- 임의의 한 행, 한 열에 놓인 N 개의 동전을 뒤집는 작업을 수행할 수 있다
- 동전을 적절히 뒤집어서 T의 최소 개수를 구하는 문제

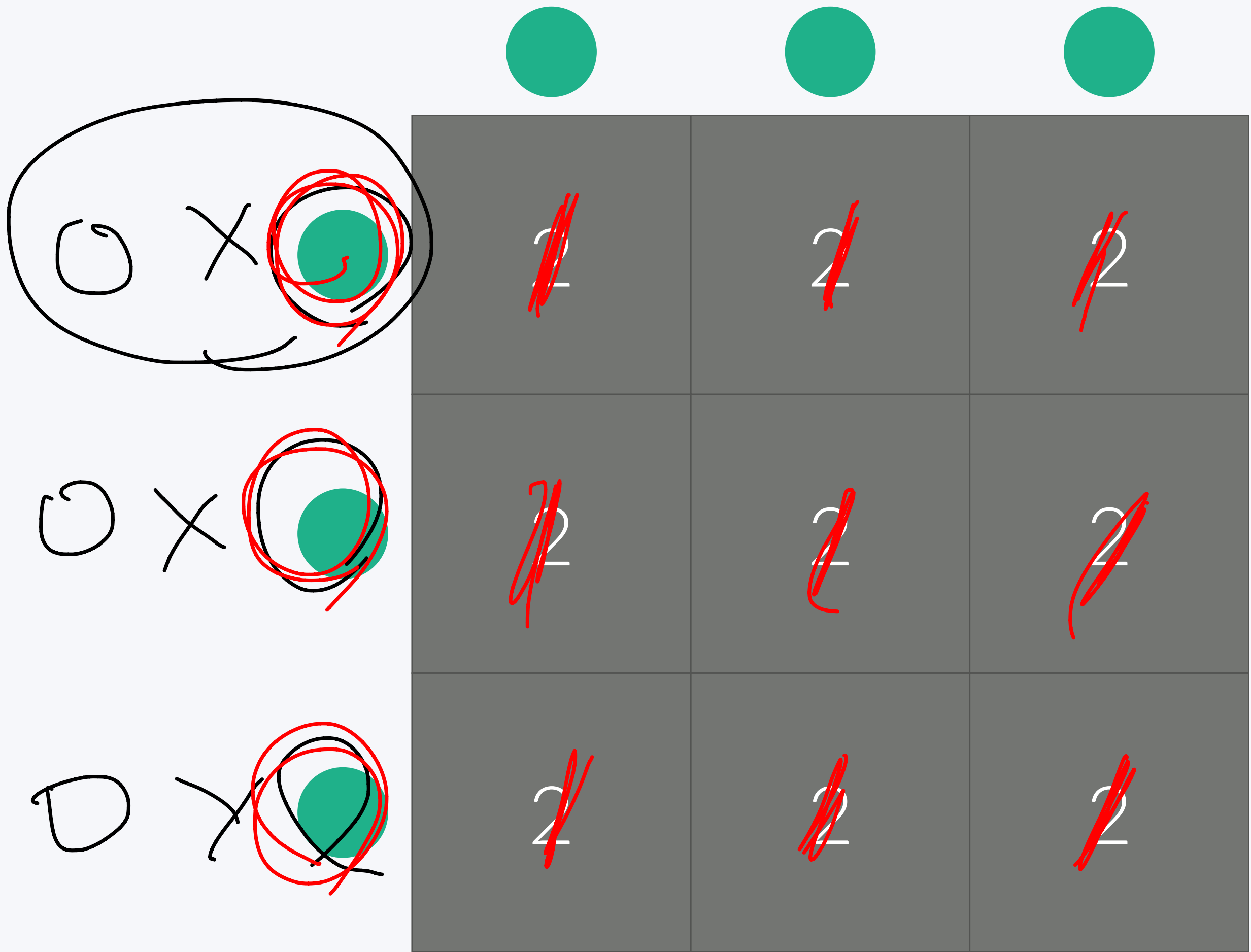
H, T
 $N \leq 20$



동전 뒤집기

<https://www.acmicpc.net/problem/1285>

- 각각의 칸을 바꿀 수 있는 방법은 모두 두 가지이다



$N \times N$

75

행

2^N

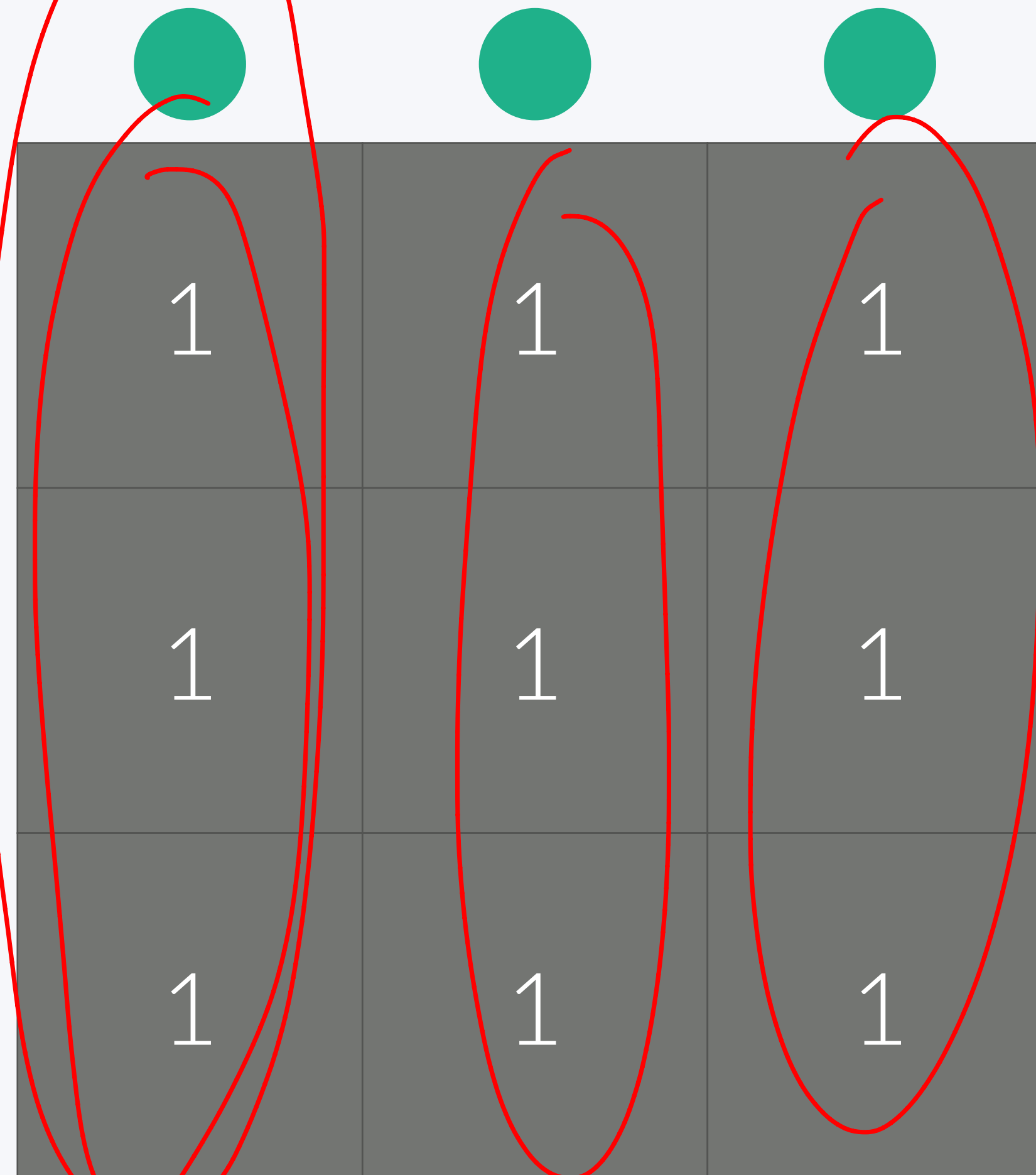
동전 뒤집기

<https://www.acmicpc.net/problem/1285>

76

- 한 행에 대해서 어떻게 돌릴지 결정을 해놓았다면, 열에 대해서는 T가 적은 쪽이 결정되어 버린다

$$N \leq \cancel{20}$$
$$3 \quad 2$$



1	1	1
1	1	1
1	1	1

$$\frac{2^N \times N^2}{(048576 \times 400)}$$
$$= 403$$

동전 뒤집기

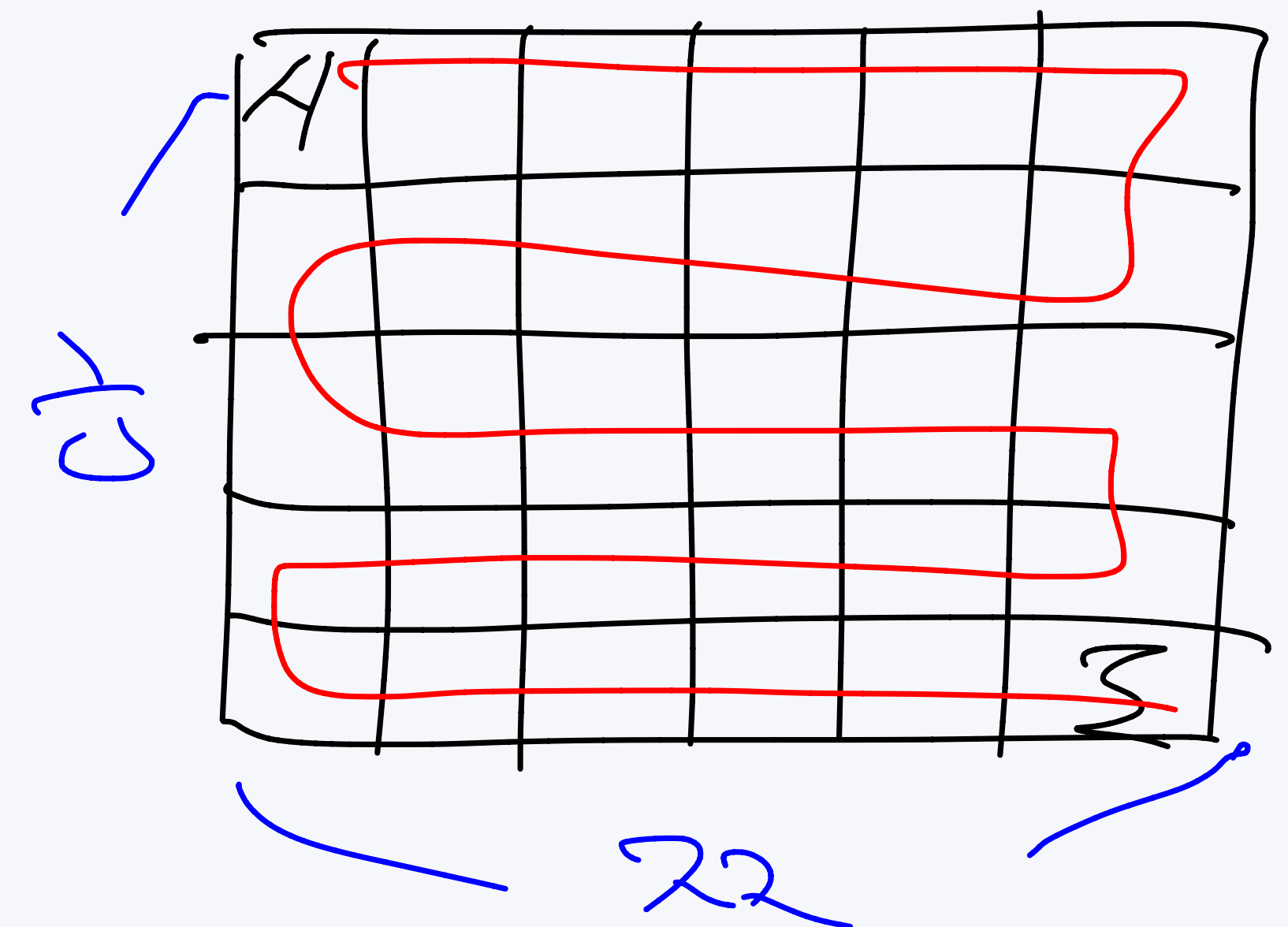
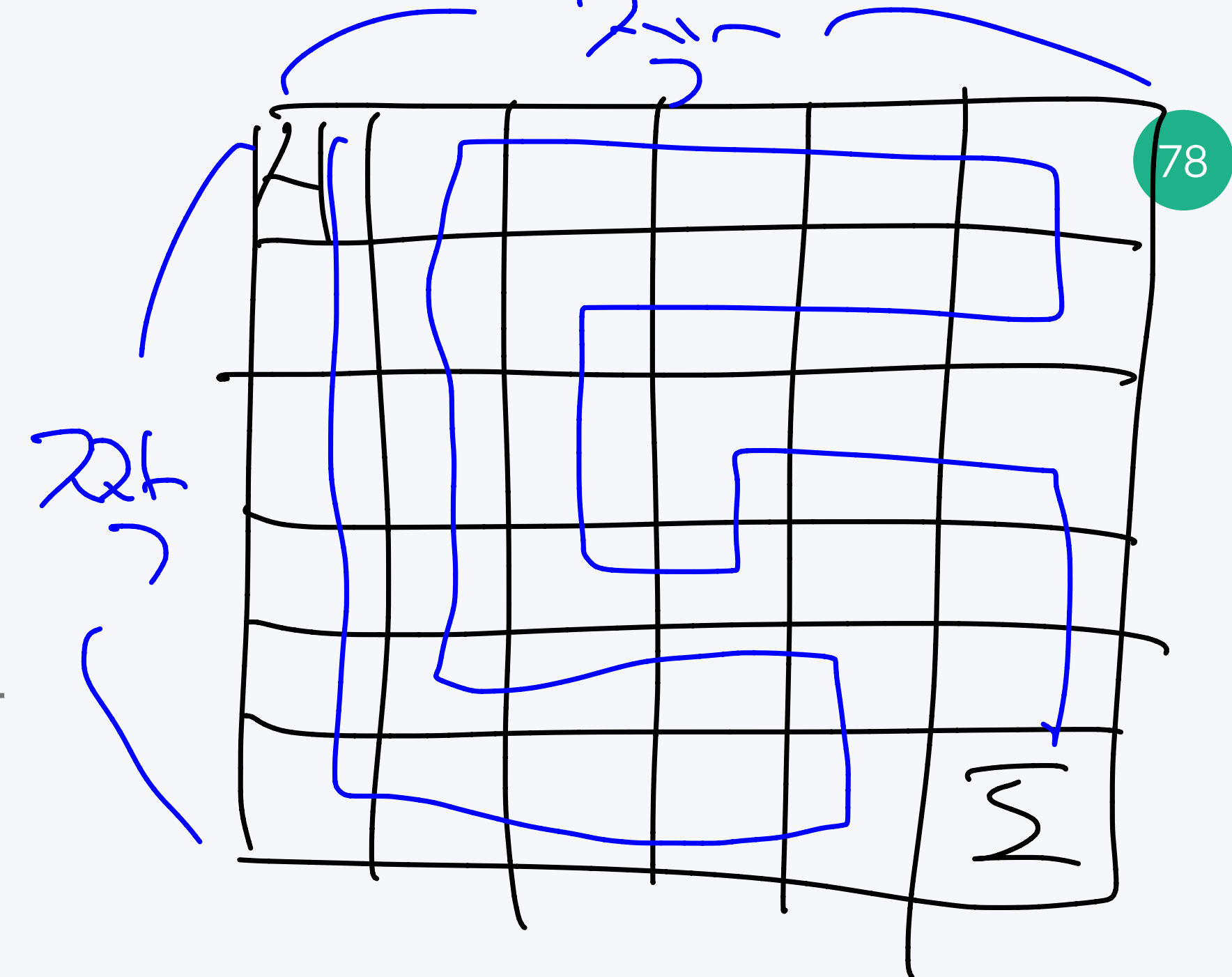
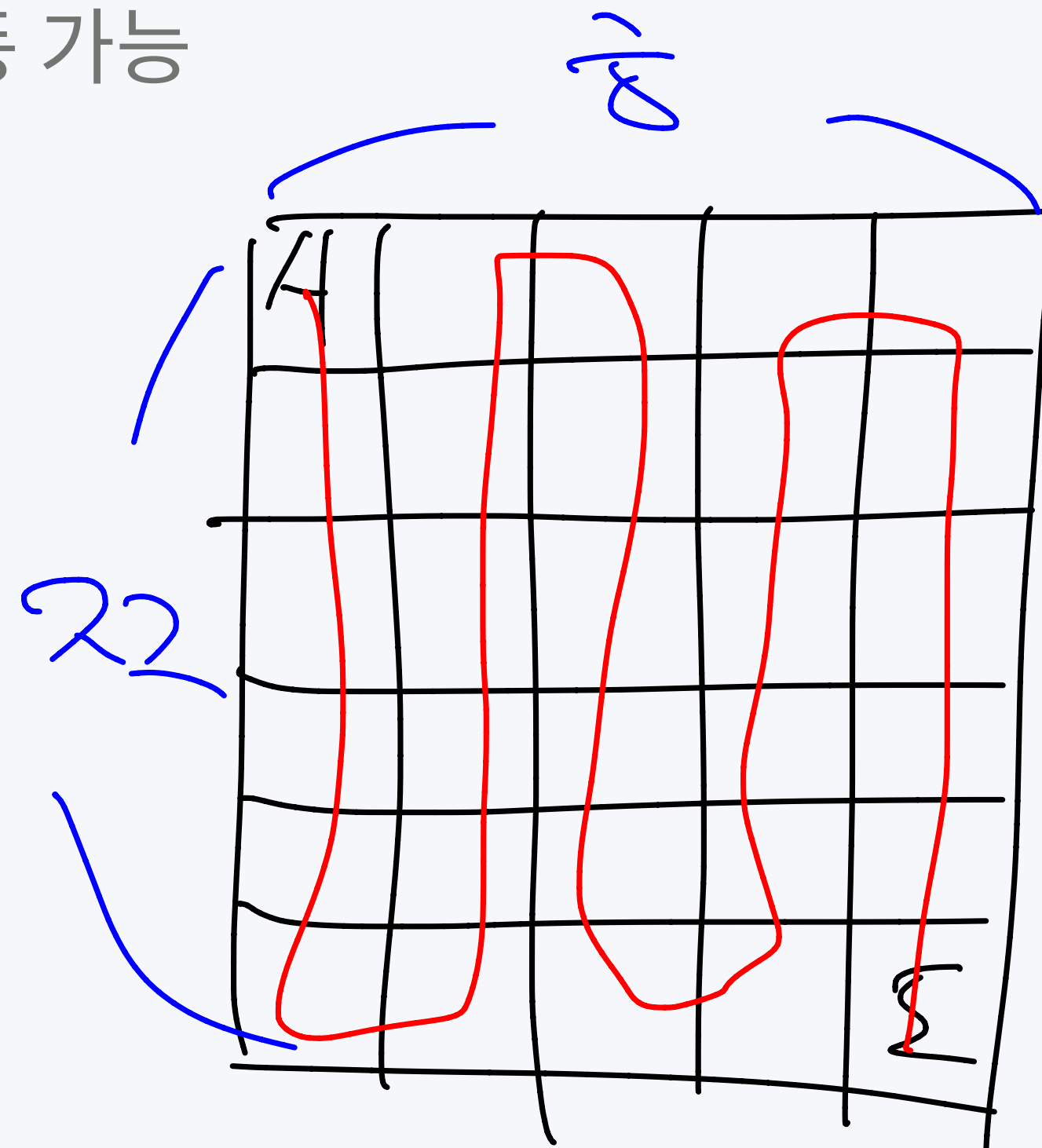
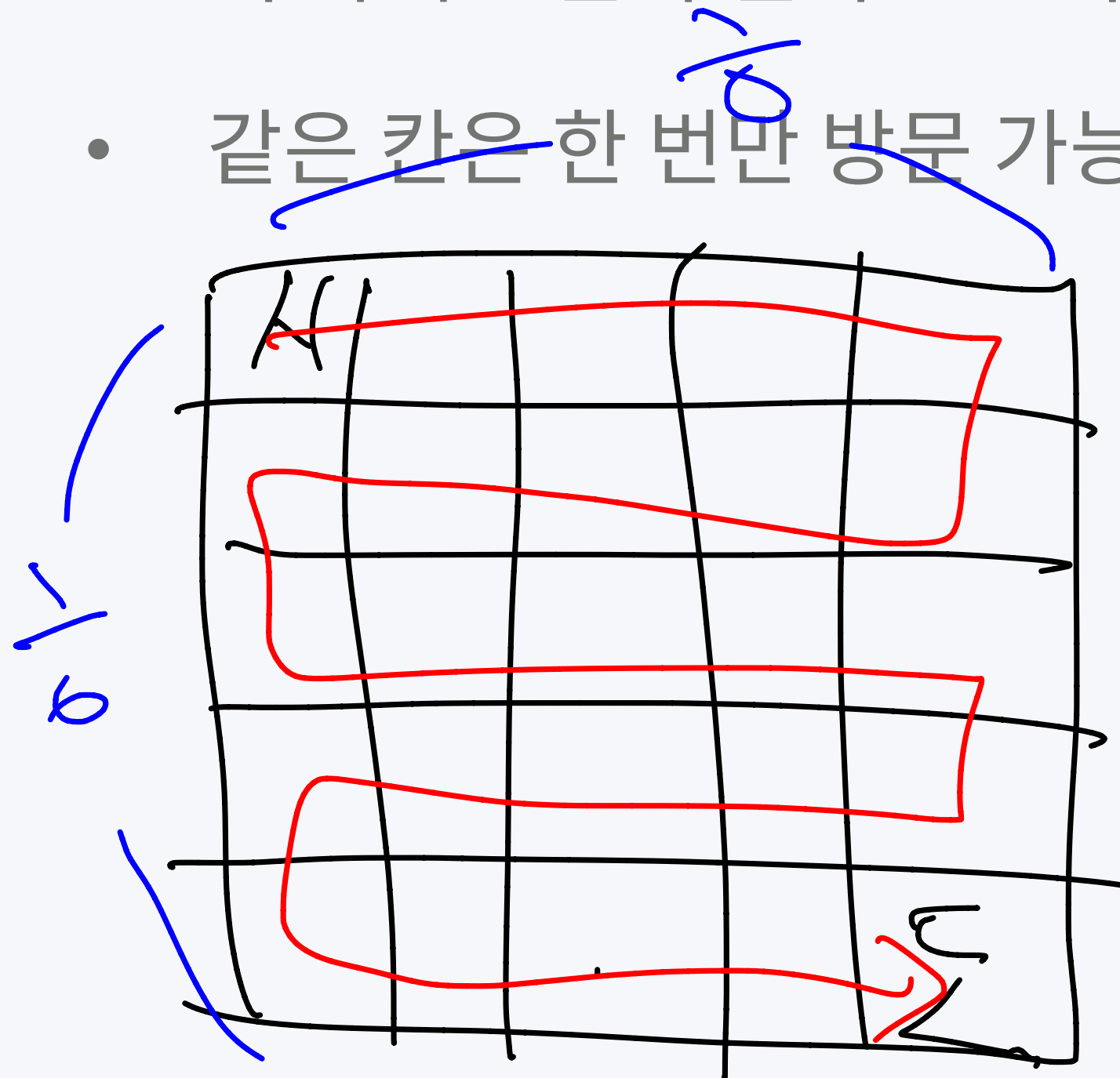
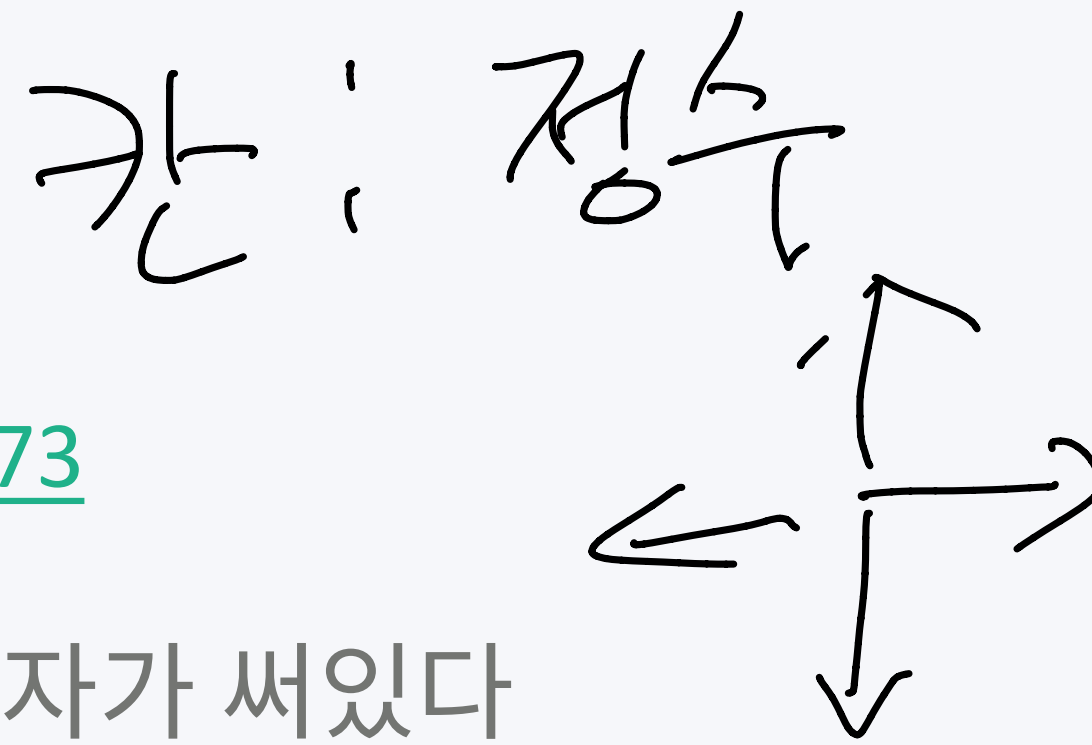
<https://www.acmicpc.net/problem/1285>

- 소스: <http://boj.kr/b1b351ef73e24d4dbea0421efdf6cfd1>

롤러코스터

<https://www.acmicpc.net/problem/2873>

- $R * C$ 크기의 보드의 각 칸에 숫자가 써있다
- $(1, 1) \rightarrow (R, C)$ 로 이동하는데 방문한 칸의 숫자의 합을 최대로
- 위 아래 오른쪽 왼쪽으로 이동 가능
- 같은 칸은 한 번만 방문 가능



롤러코스터

<https://www.acmicpc.net/problem/2873>

- R 또는 C가 홀수면 모든 칸을 방문할 수 있음

롤러코스터

80

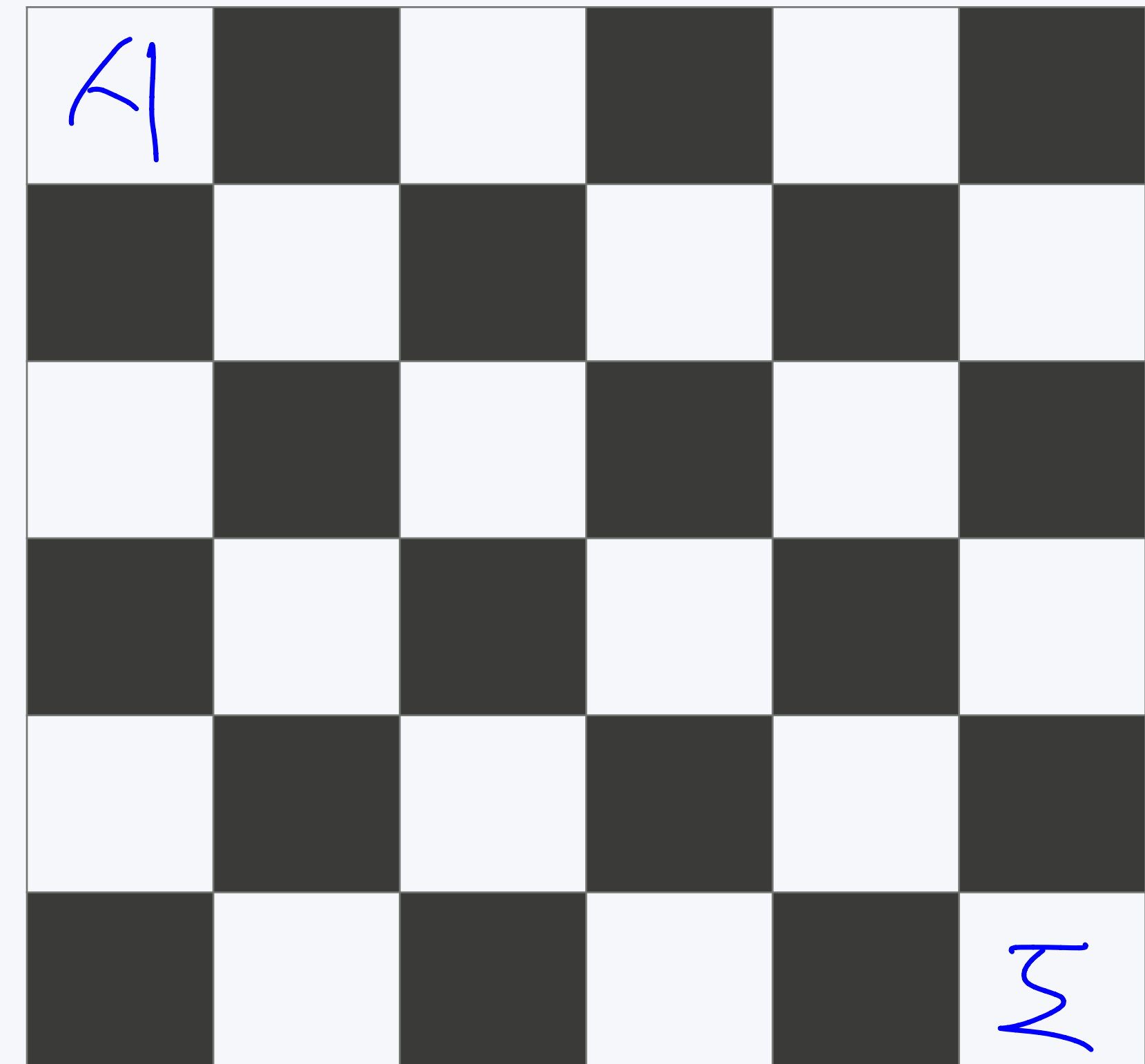
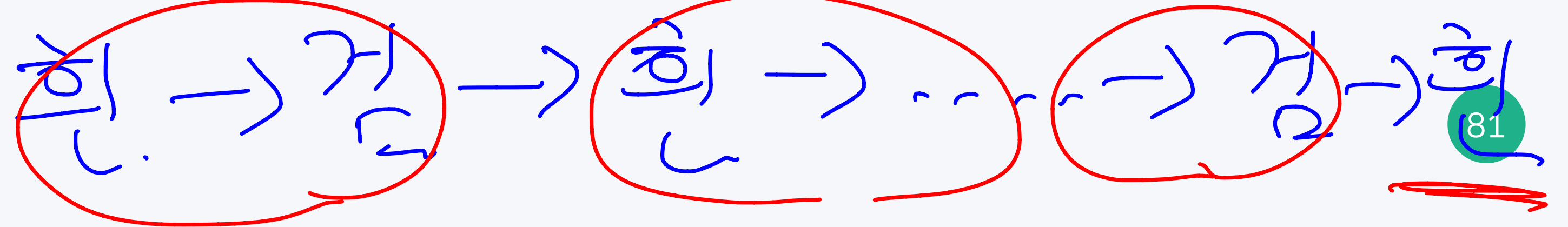
<https://www.acmicpc.net/problem/2873>

- R과 C가 모두 짝수면 모든 칸을 방문하는 것은 불가능

롤러코스터

<https://www.acmicpc.net/problem/2873>

- 모든 칸을 체스판 처럼 검정과 흰색으로 칠했다고 하자
- (1, 1)과 (R, C)의 색은 흰색이다
- (1, 1)과 (R, C)로 가는 모든 경로는
흰->검->흰->...->흰->검->흰
- 방문한 칸은 흰>검이다.
- 방문하지 않은 칸 흰<검
- 따라서, 모든 칸을 방문하는 것이 불가능



R, C 짝수

흰 = 검

롤러코스터

<https://www.acmicpc.net/problem/2873>

- 흰 칸 한 칸을 방문하지 않는다면, 나머지 칸은 모두 방문 불가
- 검정 칸 한 칸을 방문하지 않으면, 나머지 칸을 모두 방문 가능
- 따라서, 방문하지 않을 검정 칸 하나를 선택해야 함
- 방문한 칸의 합의 최대를 구하는 문제이기 때문에, 가장 작은 값을 가지는 검정 칸을 선택!

롤러코스터

<https://www.acmicpc.net/problem/2873>

- 문제를 변형해서 풀기
- 두 사람이 $(1, 1)$, (R, C) 에 있고, 서로 만날때까지 이동하는 문제
- $(1, 1)$ 에 있는 사람을 A, (R, C) 에 있는 사람을 B

롤러코스터

<https://www.acmicpc.net/problem/2873>

- 선택한 칸이 첫 두 행에 없다면
 - A는 첫 행의 오른쪽으로 갔다가 아래로 한 칸 내려오고 두 번째 칸의 왼쪽으로 이동한 다음, 한 칸 아래로 내려온다
 - 이렇게 되면, 위의 두 행을 무시하고 다시 문제를 풀 수 있다.
-
- 선택한 칸이 마지막 두 행에 없다면
 - 위와 같은 식으로 B를 이동시켜
 - 마지막 두 행을 무시하고 문제를 다시 풀 수 있다

롤러코스터

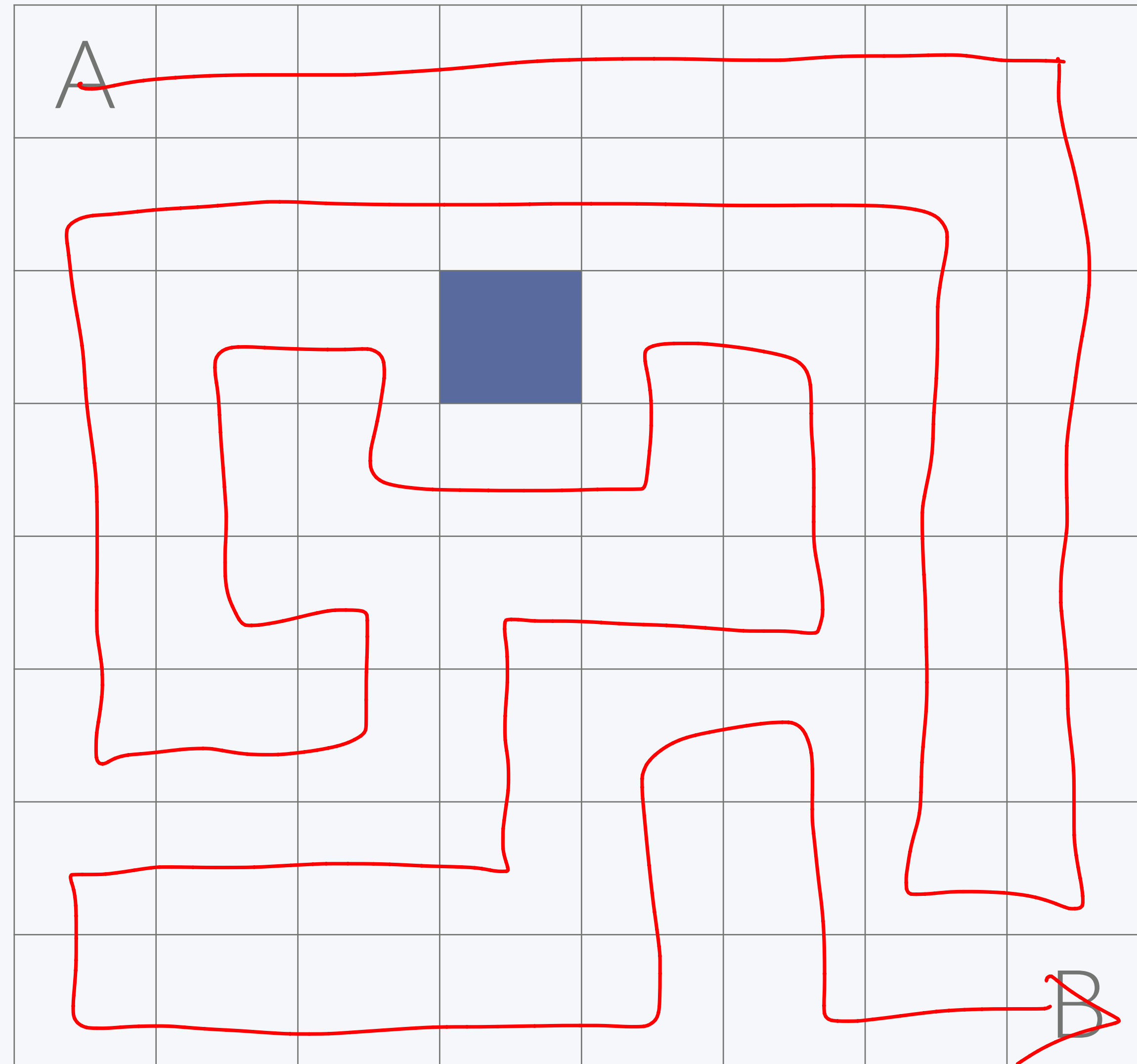
<https://www.acmicpc.net/problem/2873>

- 이런식으로 하면, 행은 2개만 남게 되고
- 여기서부터는 열을 행과 같은 식으로 처리하면
- 결국 2×2 크기의 칸만 남게 된다

Meet in the Middle
MITM

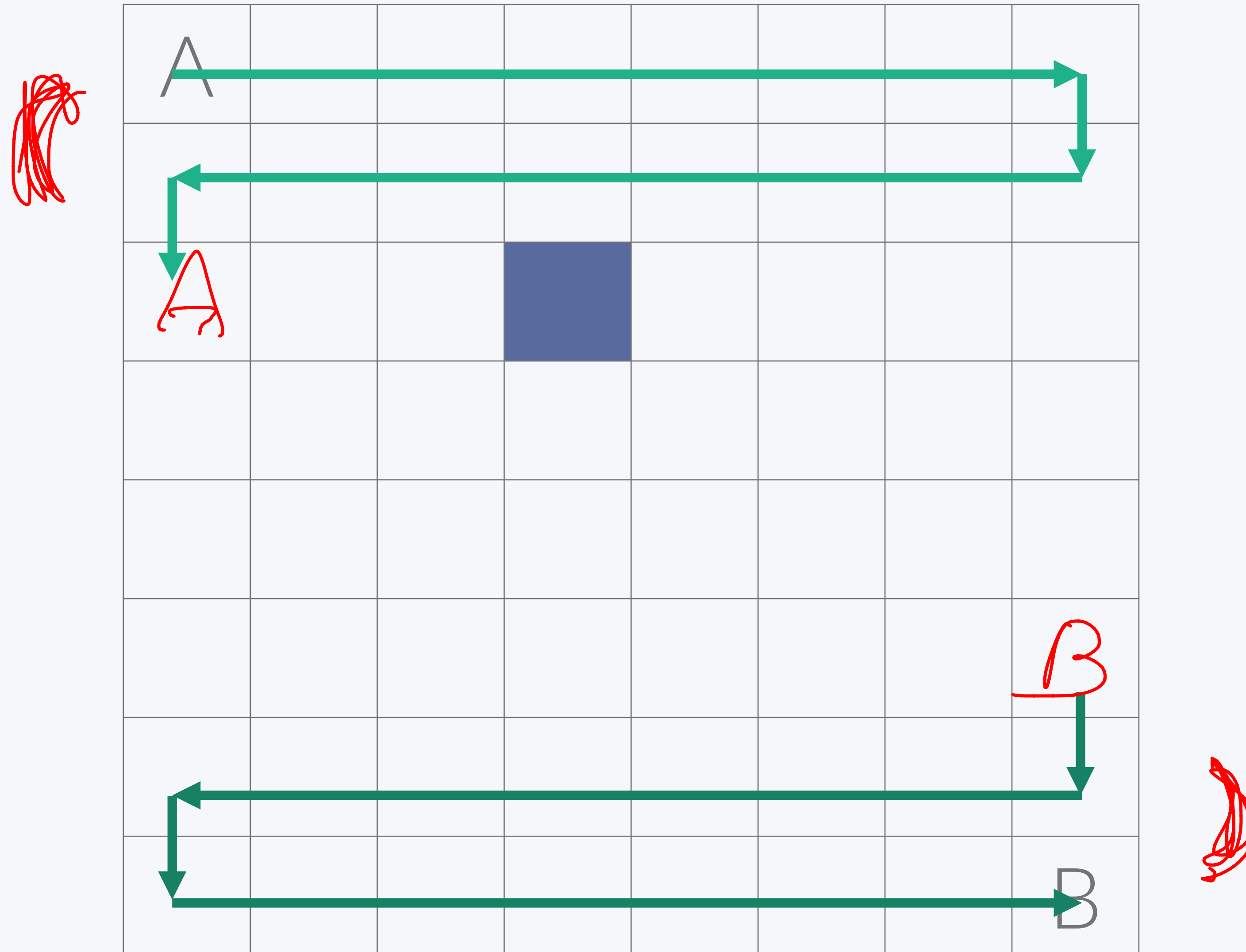
롤러코스터

<https://www.acmicpc.net/problem/2873>



롤러코스터

<https://www.acmicpc.net/problem/2873>



롤러코스터

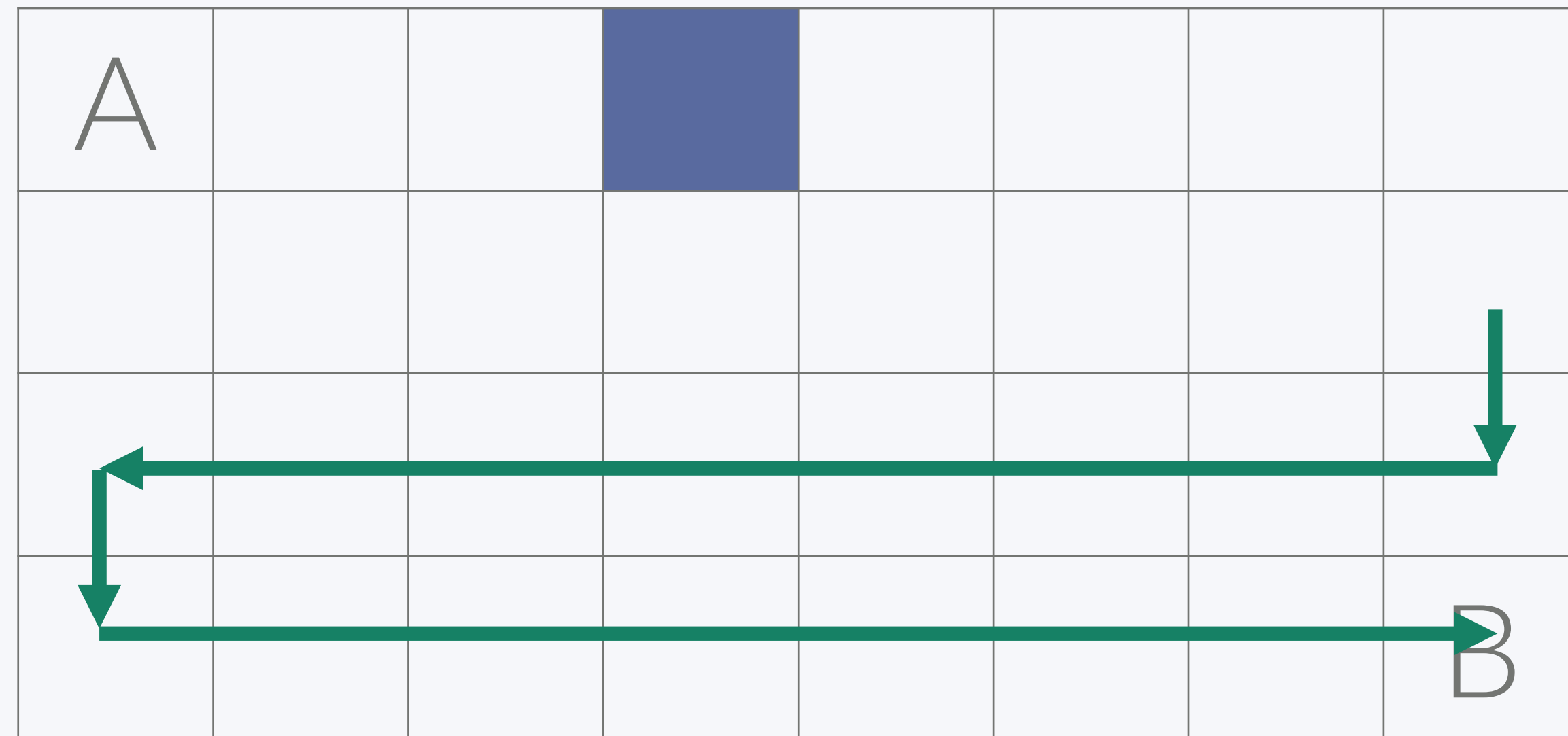
<https://www.acmicpc.net/problem/2873>

② (

A							
							B

롤러코스터

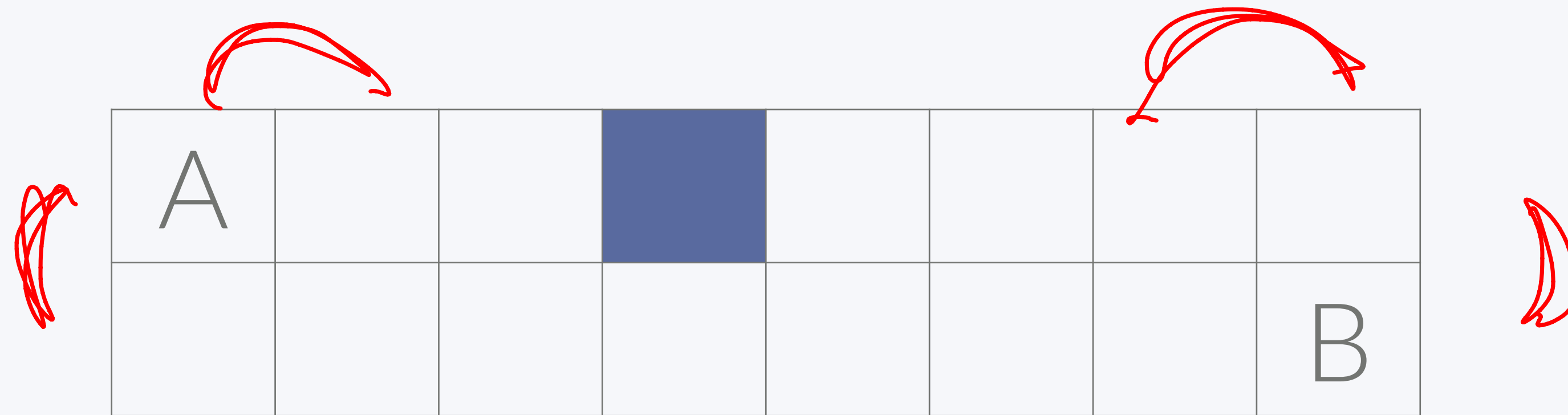
<https://www.acmicpc.net/problem/2873>



롤러코스터

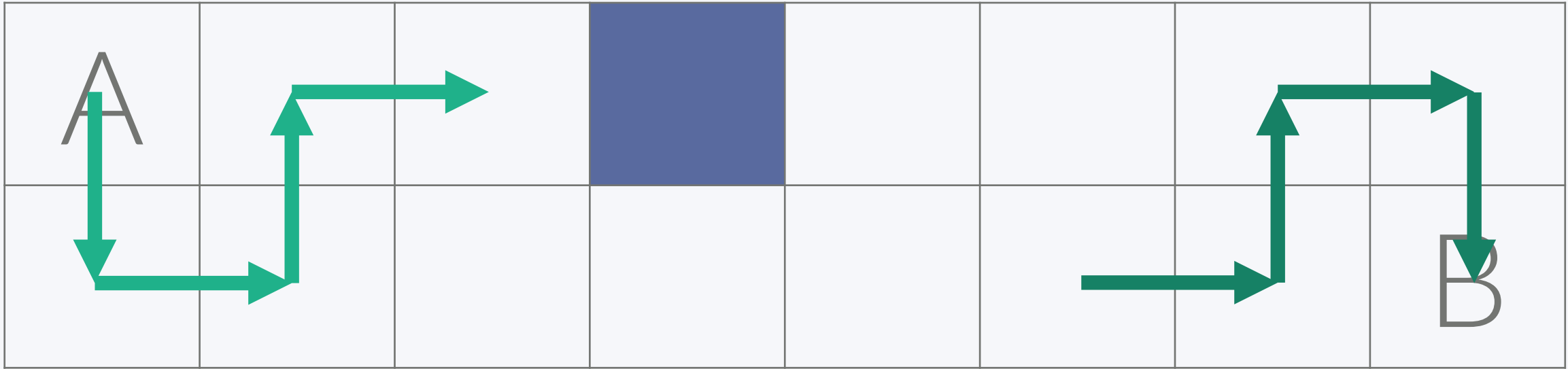
<https://www.acmicpc.net/problem/2873>

90



롤러코스터

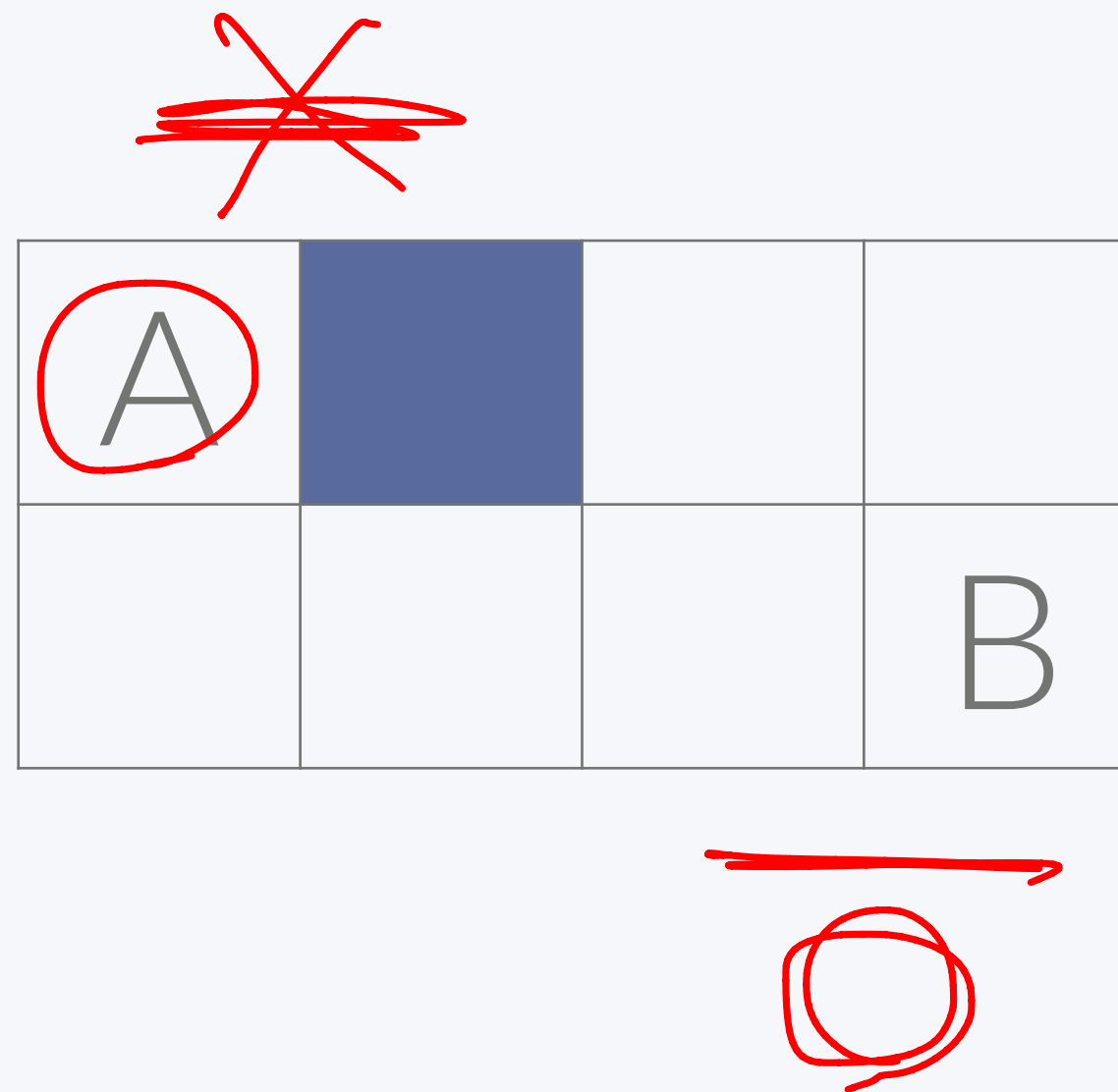
<https://www.acmicpc.net/problem/2873>



롤러코스터

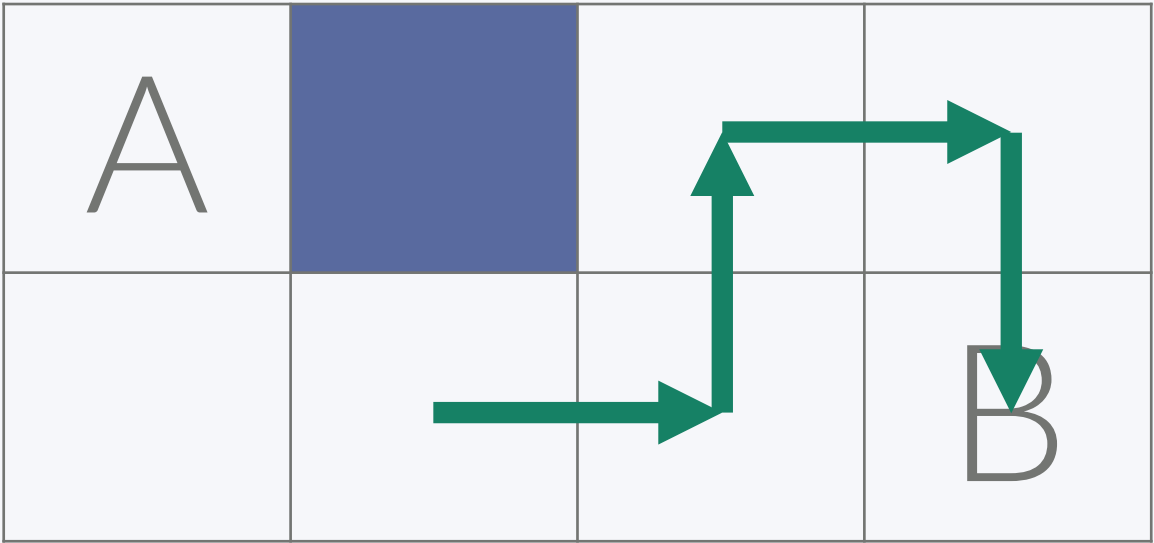
<https://www.acmicpc.net/problem/2873>

92



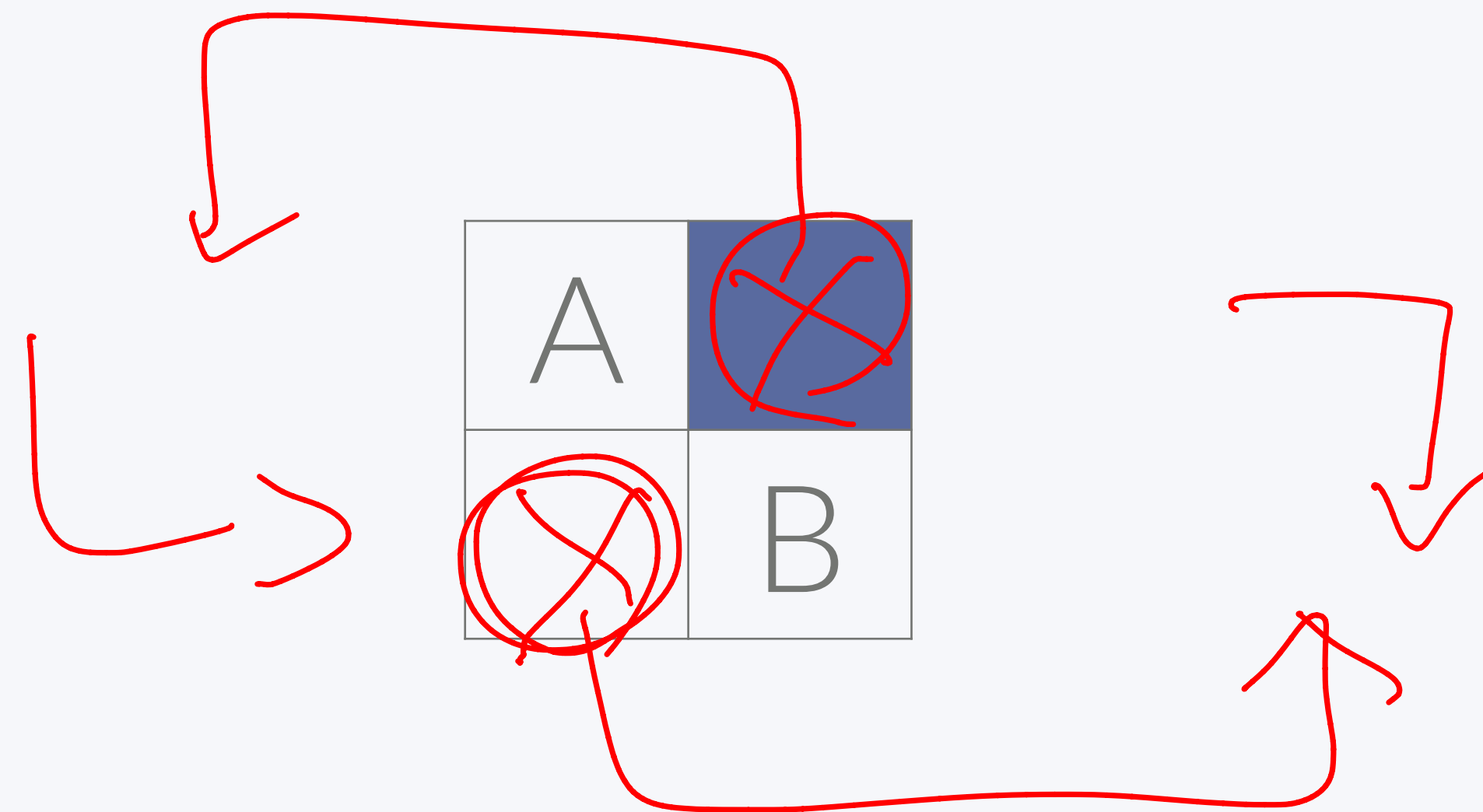
롤러코스터

<https://www.acmicpc.net/problem/2873>



롤러코스터

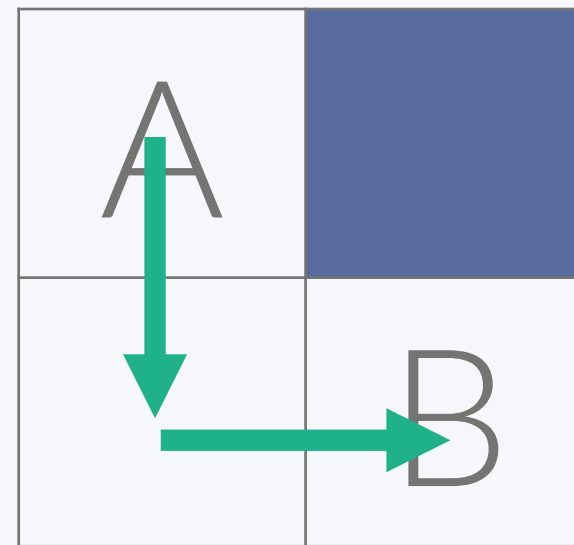
<https://www.acmicpc.net/problem/2873>



롤러코스터

<https://www.acmicpc.net/problem/2873>

95



롤러코스터

<https://www.acmicpc.net/problem/2873>

- 소스: <http://boj.kr/92ecaa7e867241ce83015d1bf0720a8a>

보석 도둑

<https://www.acmicpc.net/problem/1202>

- 보석이 N 개
- 각 보석은 무게 $M[i]$ 와 가격 $V[i]$ 를 가지고 있음
- 가방은 K 개
- 가방에 담을 수 있는 최대 무게 $C[i]$
- 가방에는 보석 1개만 넣을 수 있음
- 가방에 담을 수 있는 보석의 최대 가격 구하는 문제

보석 도둑

<https://www.acmicpc.net/problem/1202>

- 가격이 높은 보석부터 차례대로
- 각 보석을 담을 수 있는 가방 중 $C[i]$ 가 가장 작은 가방에 넣는다

보석 도둑

<https://www.acmicpc.net/problem/1202>

1. 슯자를 넣고
2. 어떤 수 x 보다 큰 슯자 중에 가장 작은 수를 찾고
3. 슯자를 지운다

보석 도둑

100

<https://www.acmicpc.net/problem/1202>

- multiset을 사용한다
- 소스: <http://boj.kr/a88709932eec47a8a75f4a3344062351>

보석 도둑

101

<https://www.acmicpc.net/problem/1202>

- Max-heap을 이용해서 푸는 방법

보석 도둑

<https://www.acmicpc.net/problem/1202>

- 보석과 가방을 배열 하나에 넣고, 무게를 기준으로 오름차순 정렬한다
 - 보석은 $M[i]$, 가방은 $C[i]$
- 보석인 경우 힙에 $V[i]$ 를 넣는다
- 가방인 경우 힙에서 가장 큰 $V[i]$ 를 뺀다

보석 도둑

103

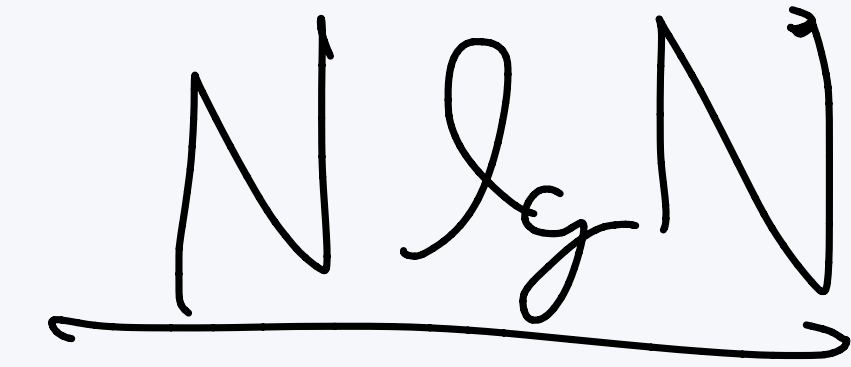
<https://www.acmicpc.net/problem/1202>

- 소스: <http://boj.kr/da02ffafbe994cbd99109e6c8c5c8af0>

가장 긴 증가하는 부분 수열 3

<https://www.acmicpc.net/problem/12738>

104



- 수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열을 구하는 프로그램을 작성하시오
- 예를 들어, 수열 $A = \{10, 20, 10, 30, 20, 50\}$ 인 경우에 가장 긴 증가하는 부분 수열은 $A = \{10, 20, 10, 30, 20, 50\}$ 이고, 길이는 4이다

가장 긴 증가하는 부분 수열 2

105

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 리스트가 없기 때문에, 새로 추가한다

Handwritten notes illustrating the process of finding the Longest Increasing Subsequence (LIS) for the sequence {1, 3, 1, 2, 4, 3, 4, 2}:

• 1 2, 3 4 5

1, 3

~~3 (4 5 ...)~~

The diagram shows a sequence of numbers 1, 2, 3, 4, 5. A bracket connects the 3 in the sequence to the 3 in the set {1, 3}. Below this, the sequence 3 (4 5 ...) is shown, which is crossed out with a large X, indicating it is not a valid LIS.

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
 - 가능한 정답의 리스트를 만들어가면서 답을 구해보자
 - 1을 복사해서 새로 만들고, 뒤에 3을 붙이는 것이 좋다.
-
- 1
 - 1, 3

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 새로 만든다. 1은 중복되니까 지워버린다.

• 1

• 1

• 1, 3

~~2~~

1, 2

~~1, 3, 2~~

3

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 1을 복사한 다음, 뒤에 2를 붙이는 것이 좋다. 1, 3은 절대 정답이 될 수 없기 때문에 지운다

- 1
- 1, 2
- ~~1, 3~~

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 1, 2를 복사한 다음, 뒤에 4를 붙이는 것이 좋다.

5
3

- 1
- 1, 2
- ~~1, 2, 4~~ 1, 2, 3
1, 2, ~~4~~ 5
3

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, **3**, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 1, 2를 복사한 다음, 뒤에 3를 붙이는 것이 좋다. 1, 2, 4는 정답이 될 수 없다.

- 1

- 1, 2

- 1, 2, 3

- ~~1, 2, 4~~

1, 3, 4

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 1, 2, 3를 복사한 다음, 뒤에 4를 붙이는 것이 좋다.
- 1
- 1, 2
- 1, 2, 3
- 1, 2, 3, 4

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 1을 복사한 다음, 뒤에 2를 붙이는 것이 좋다. 1, 2는 2개이니까 하나를 지운다.

길이: 4

N

- 1
- 1, 2
- ~~1, 2~~
- 1, 2, 3
- 1, 2, 3, 4

가장 긴 증가하는 부분 수열 2

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, **2**} 인 경우
- 가능한 정답의 리스트를 만들어가면서 답을 구해보자
- 1
- 1, 2
- ~~1, 2~~
- 1, 2, 3
- 1, 2, 3, 4 (정답)

- lower bound : 크거나 같은 최소값
upper bound : 크거나 같은 최대값

Handwritten sequence: 1 2 2 3 3 3 3 5 5 5 7 8 8 8 8 8 8

Annotations below the sequence:

- An arrow points from "50 lb" under the first "5".
- An arrow points from "50 lb" under the "7".

가장 긴 증가하는 부분 수열 2

116

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]					

가장 긴 증가하는 부분 수열 2

117

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1				



lower bound

가장 긴 증가하는 부분 수열 2

118

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	3			



lower bound

가장 긴 증가하는 부분 수열 2

119

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	2			



lower bound

가장 긴 증가하는 부분 수열 2

120

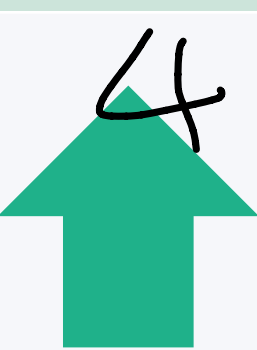
<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	2			

4 4 4



lower bound

가장 긴 증가하는 부분 수열 2

121

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	2	4		

1
3
4
2
3

lower bound

가장 긴 증가하는 부분 수열 2

122

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	2	3		



lower bound

가장 긴 증가하는 부분 수열 2

123

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	2	3	4	



lower bound

가장 긴 증가하는 부분 수열 2

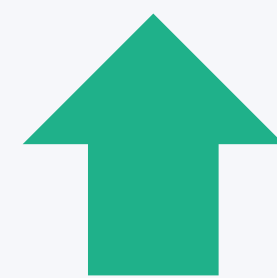
124

<https://www.acmicpc.net/problem/12015>

- 수열 = {1, 3, 1, 2, 4, 3, 4, 2} 인 경우

i	1	2	3	4	5	6	7	8
A[i]	1	3	1	2	4	3	4	2

i	0	1	2	3	4
D[i]	1	2	3	4	



lower bound

가장 긴 증가하는 부분 수열 2

125

<https://www.acmicpc.net/problem/12015>

- 소스: <http://boj.kr/e01be38b0667462d9c4fe35db690ccd1>

A = [3, 4, 5, 1]
앞
2 3 4
/ 4 5