

# 8/21

최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

# 욕심쟁이 판다

<https://www.acmicpc.net/problem/1937>

- $N \times N$  크기의 이차원 배열에서 최대 부분 증가 수열을 구하는 문제

# 욕심쟁이 판다

<https://www.acmicpc.net/problem/1937>

- $D[i][j]$  = 판다가  $(i, j)$ 에서 이동을 시작했을 때, 최대한 살 수 있는 일수

# 욕심쟁이 판다

<https://www.acmicpc.net/problem/1937>

- $D[i][j] = \max(D[x][y]) + 1$
- $(i, j) \rightarrow (x, y)$  이동 가능
- $A[i][j] < A[x][y]$

# 욕심쟁이 판다

<https://www.acmicpc.net/problem/1937>

- C/C++: <https://gist.github.com/Baekjoon/db16601416d364f0e2a6db9e0f072007>
- Java: <https://gist.github.com/Baekjoon/adacd7cd92787b4b5837e597c2584ddb>

# 부등식

<https://www.acmicpc.net/problem/3110>

- A1, A2, B2, C2, D2, E1, E2가 주어졌을 때
- $A1/A2 < B1/B2 < C1/C2 < D1/D2 < E1/E2$
- 를 만족하는 (B1, C1, D1)의 개수를 구하는 문제
- 모든 수는 1,000보다 작거나 같은 자연수

# 부등식

<https://www.acmicpc.net/problem/3110>

- A1, A2, B2, C2, D2, E1, E2가 주어졌을 때
- $A1/A2 < B1/B2 < C1/C2 < D1/D2 < E1/E2$
- 를 만족하는 (B1, C1, D1)의 개수를 구하는 문제
- 모든 수는 1,000보다 작거나 같은 자연수

# 부등식

<https://www.acmicpc.net/problem/3110>

- M을 부등식에서 나올 수 있는 가장 큰 분자라고 했을 때
- M은  $1,000 * 1,000 = 1,000,000$  정도 될 수 있다.
- 예를 들어, A2가 크고, E2가 작은 경우에 저정도 될 수 있다



# 부등식

<https://www.acmicpc.net/problem/3110>

- 모든 B1, C1, D1을 순회해보는 방식은  $O(M^3)$  이라서 시간이 너무 오래 걸린다.

# 부등식

<https://www.acmicpc.net/problem/3110>

- 모든 B1, C1 순회해보면서, D1의 개수를 세어보는 방식을 생각해보자.
- $C1/C2 < D1/D2 < E1/E2$
- 가능한 D1의 개수는 수식으로 구할 수 있다.
- 하지만, B1, C1을 순회해야 하기 때문에, 복잡도는  $O(M^2)$ 이다.

# 부등식

<https://www.acmicpc.net/problem/3110>

- 앞 페이지의 방법을 확장해서 C1만 순회하면서
- B1, D1의 개수를 각각 구해서 곱해주면 된다.

# 부등식

<https://www.acmicpc.net/problem/3110>

- C++: <https://gist.github.com/Baekjoon/f2e79ac7eb1766086c951f557227886b>
- Java: <https://gist.github.com/Baekjoon/a6ba56549e2660b9fe46b3e8dbfca0b1>

# 기타 레슨

<https://www.acmicpc.net/problem/2343>

- 블루레이에는 총  $N$ 개의 레슨이 들어가는데, 블루레이를 녹화할 때, 레슨의 순서가 바뀌면 안된다
- 즉,  $i$ 번 레슨과  $j$ 번 레슨을 같은 블루레이에 녹화하려면  $i$ 와  $j$  사이의 모든 레슨도 같은 블루레이에 녹화해야 한다
- $M$ 개의 블루레이에 모든 기타 레슨 동영상을 녹화하기로 했다
- 블루레이의 크기(녹화 가능한 길이)를 최소로 하려고 한다
- 단,  $M$ 개의 블루레이는 모두 같은 크기이어야 한다

# 기타 레슨

<https://www.acmicpc.net/problem/2343>

- 블루레이에는 총  $N$ 개의 레슨이 들어가는데, 블루레이를 녹화할 때, 레슨의 순서가 바뀌면 안된다
- 즉,  $i$ 번 레슨과  $j$ 번 레슨을 같은 블루레이에 녹화하려면  $i$ 와  $j$  사이의 모든 레슨도 같은 블루레이에 녹화해야 한다
- $M$ 개의 블루레이에 모든 기타 레슨 동영상을 녹화하기로 했다
- **블루레이의 크기(녹화 가능한 길이)**를 최소로 하려고 한다
- 단,  $M$ 개의 블루레이는 모두 같은 크기이어야 한다

# 기타 레슨

15

<https://www.acmicpc.net/problem/2343>

- 블루레이의 크기를 결정!
- 이 때, 필요한 블루레이의 개수가  $M$ 보다 작거나 같은지 확인
- 작거나 같으면 크기를 크게
- 크면 작게

# 기타 레슨

<https://www.acmicpc.net/problem/2343>

- C++: <https://gist.github.com/Baekjoon/88166624472f014b8945>
- Java: <https://gist.github.com/Baekjoon/00226e2230abe5d0bf7d>



# 행렬

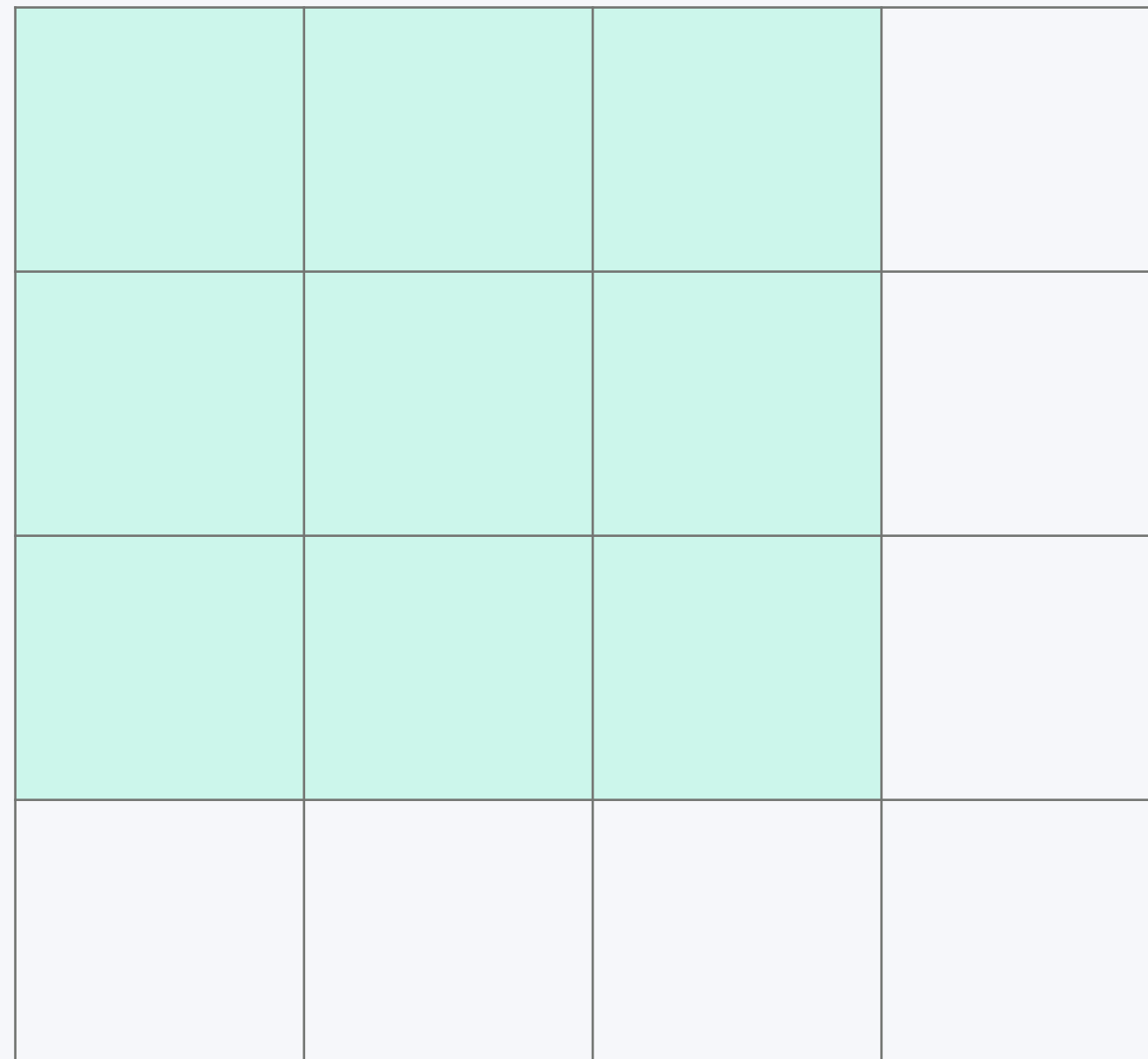
<https://www.acmicpc.net/problem/1080>

- 0과 1로만 이루어진 행렬 A와 행렬 B가 있다. 이 때, 행렬 A를 행렬 B로 바꾸는데 필요한 연산의 횟수의 최소값을 구하는 문제
- 행렬을 변환하는 연산은 어떤 3\*3크기의 부분 행렬에 있는 모든 원소를 뒤집는 것이다. (0 -> 1, 1 -> 0)

# 행렬

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

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



<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.


# 행렬

<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.

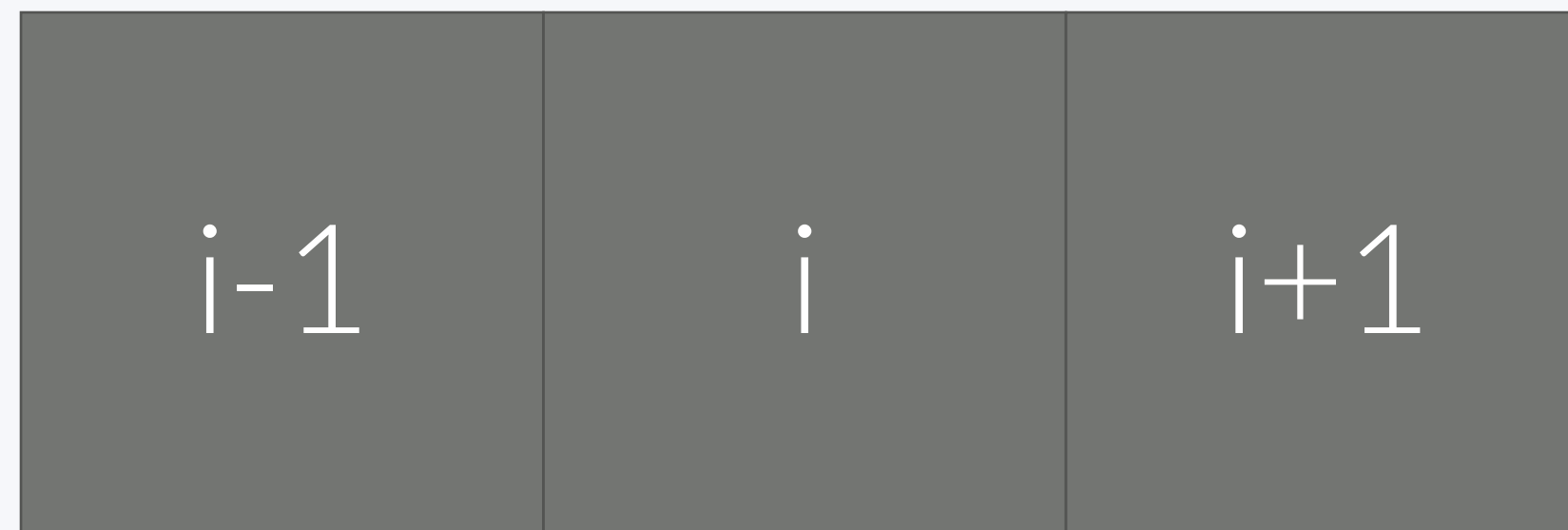
<https://www.acmicpc.net/problem/1080>

- C/C++: <https://gist.github.com/Baekjoon/f88dc83db3507180c61e12b7a7339118>
- Java: <https://gist.github.com/Baekjoon/e70158facb68e0968870c155e7a54dad>

# 전구와 스위치

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

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

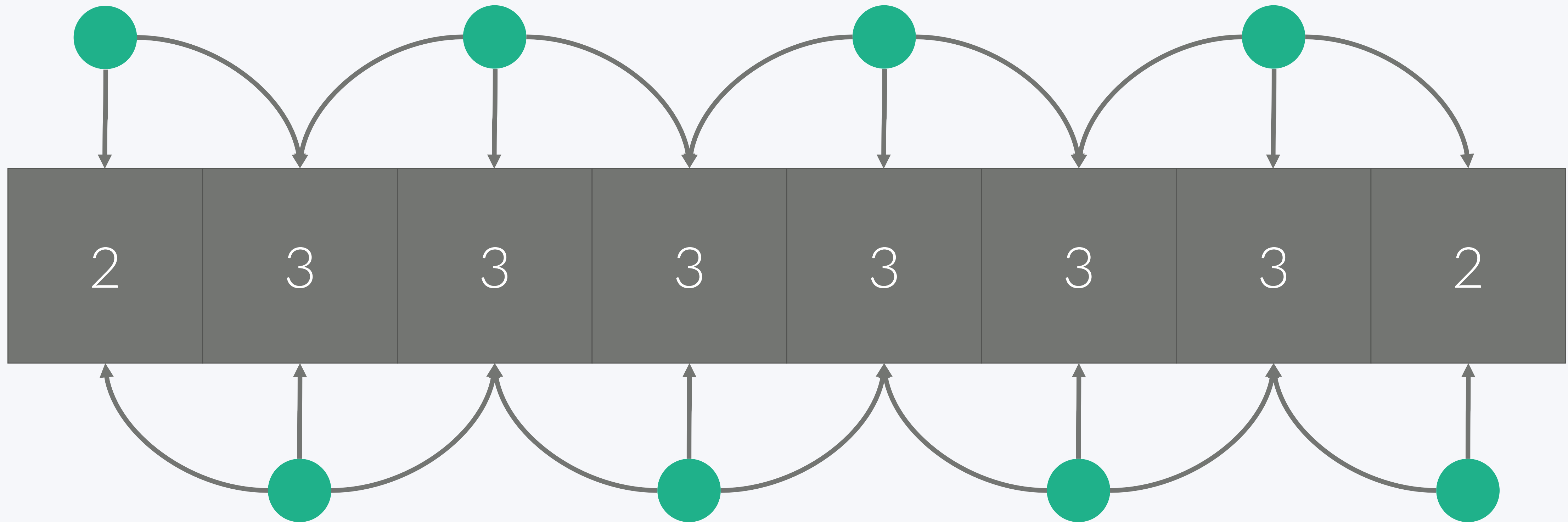


# 전구와 스위치

23

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

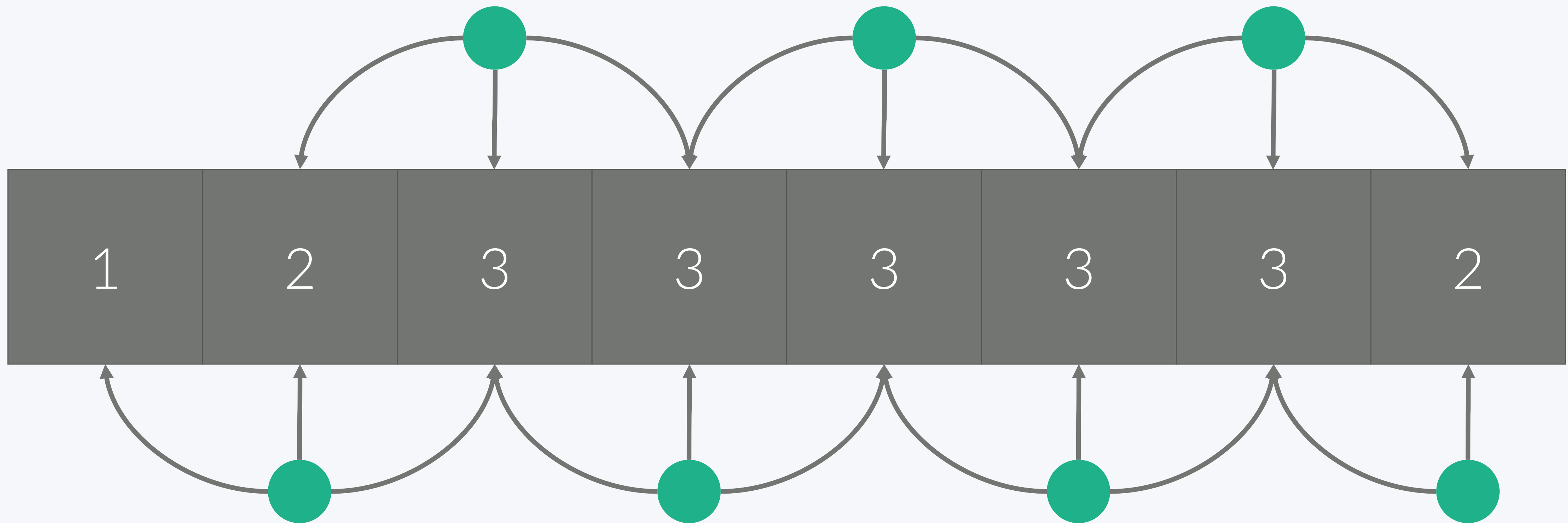
- $i$ 번 스위치를 누르면,  $i-1, i, i+1$ 번 전구의 상태가 바뀐다



# 전구와 스위치

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

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



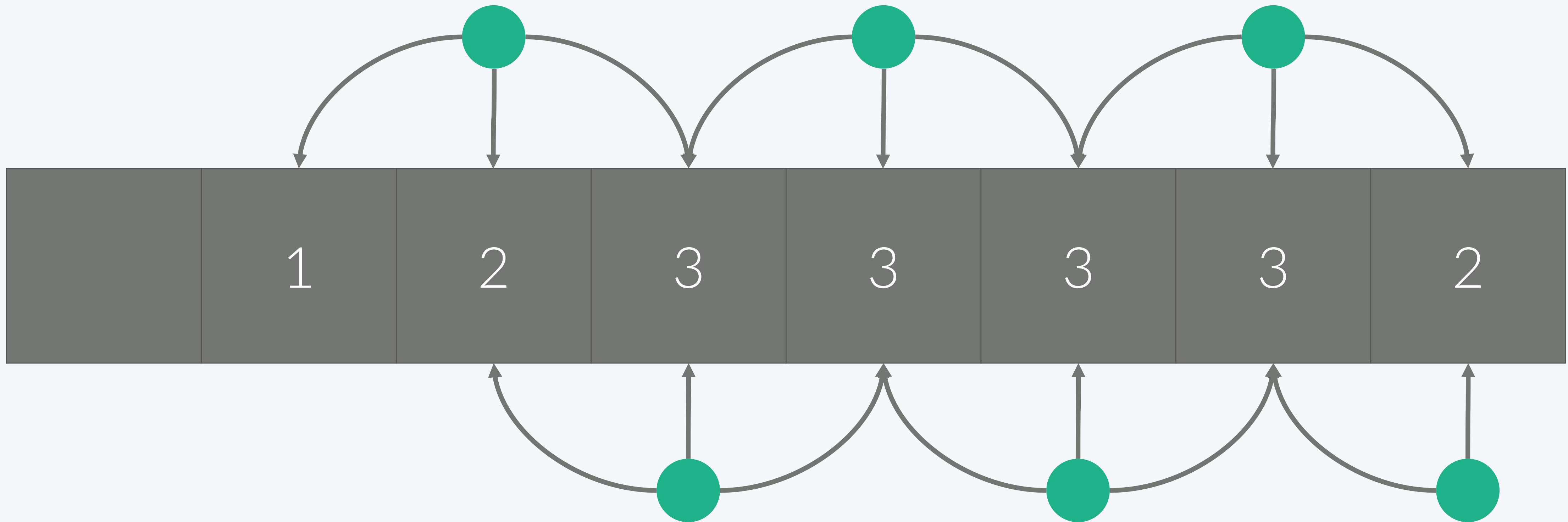


# 전구와 스위치

25

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

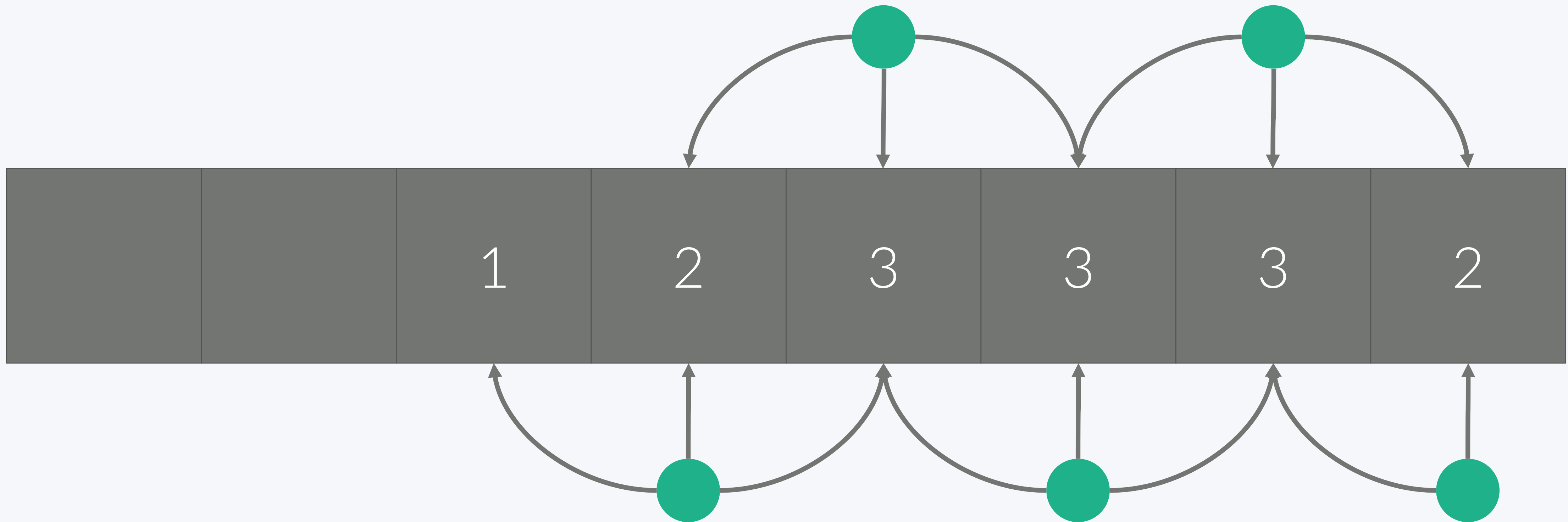
- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다



# 전구와 스위치

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

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

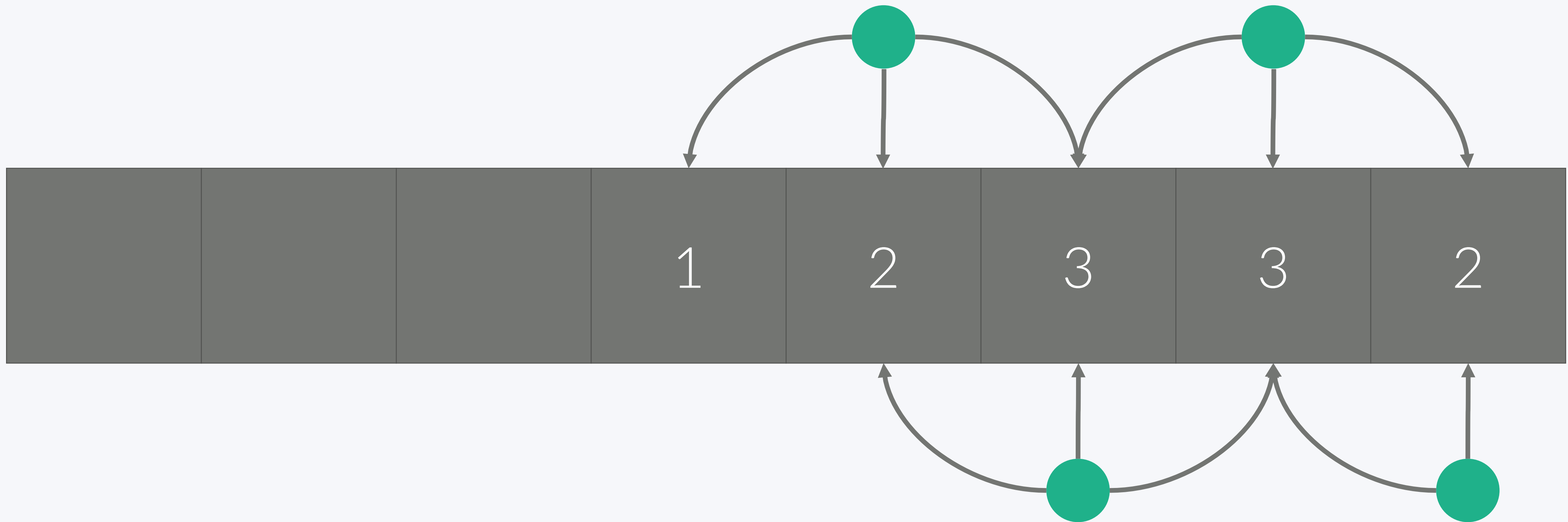


# 전구와 스위치

27

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

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

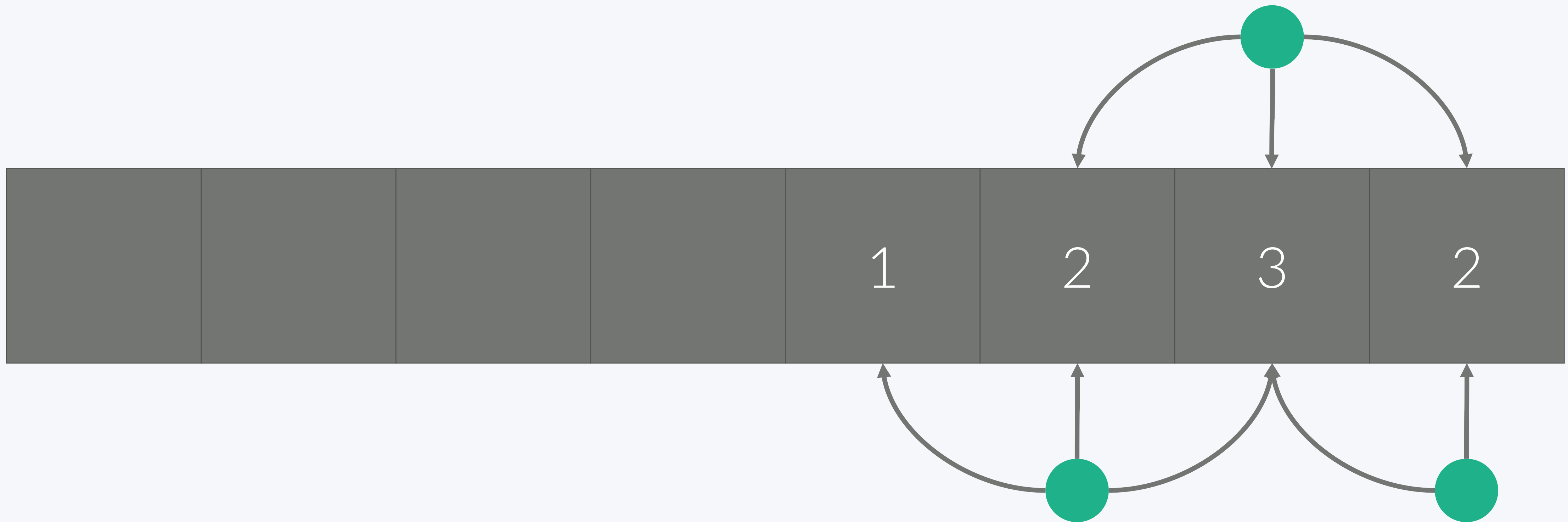


# 전구와 스위치

28

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

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



# 전구와 스위치

29

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

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

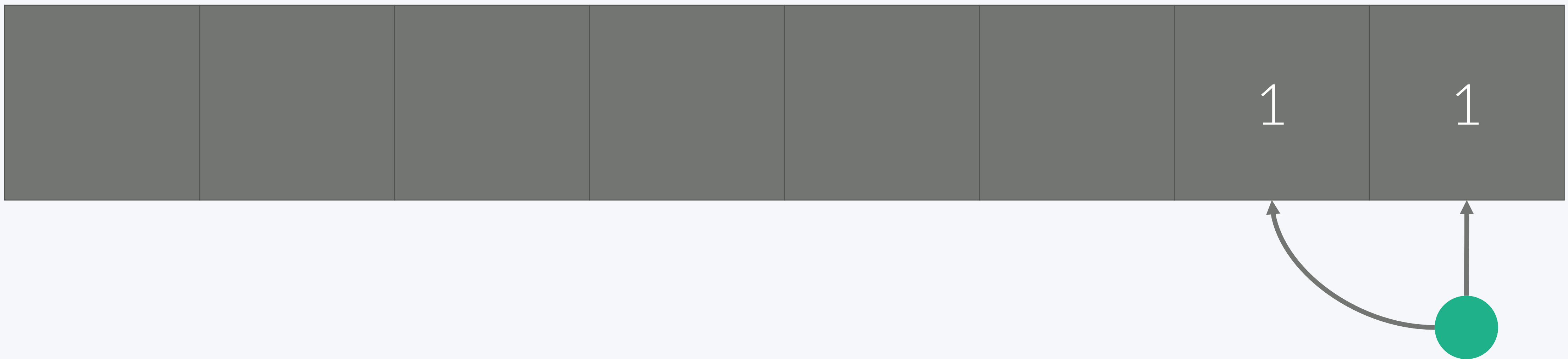


# 전구와 스위치

30

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

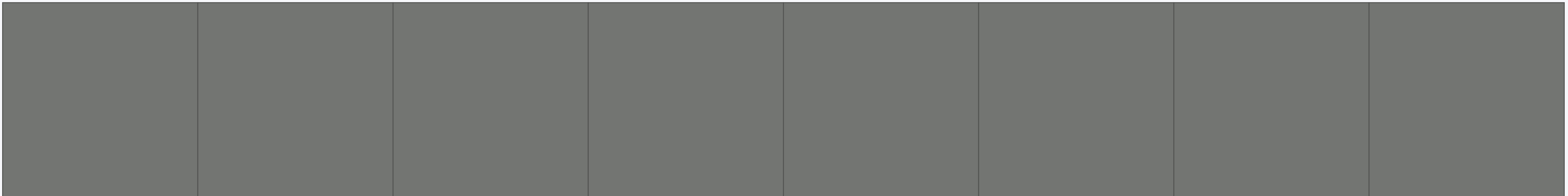
- 1번 스위치를 어떻게 누를지 결정하면, 각각의 칸을 바꿀 수 있는 방법이 한 개가 된다



# 전구와 스위치

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

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



# 전구와 스위치

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

- C++: <https://gist.github.com/Baekjoon/b2aec87657d026eee4999f539211e824>
- Java: <https://gist.github.com/Baekjoon/a6ec0a929ceb5086e722d26615864349>



# 동전 뒤집기

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

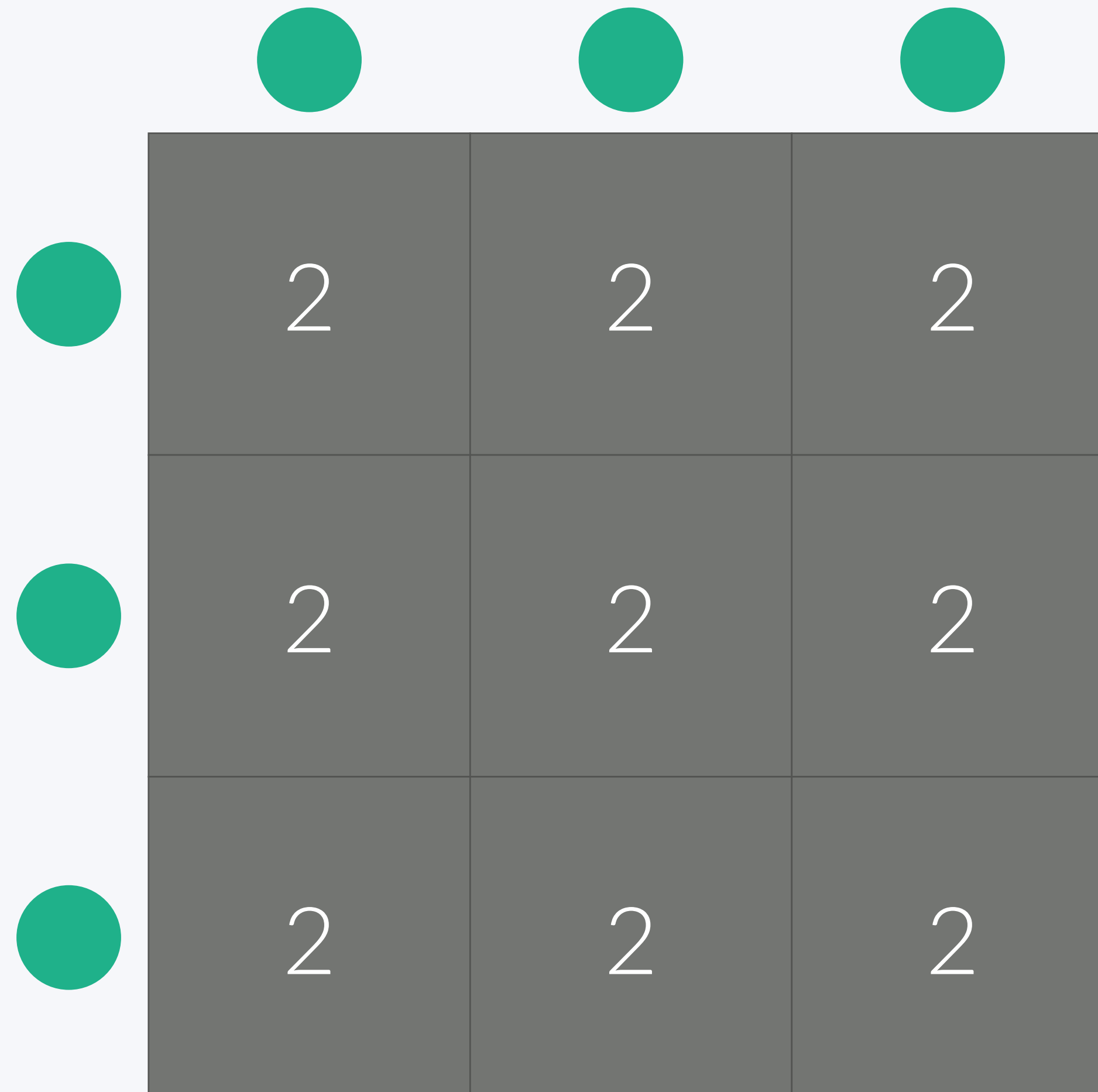
- $N^2$ 개의 동전이  $N$ 행  $N$ 열로 놓여져 있다.
- 임의의 한 행, 한 열에 놓인  $N$ 개의 동전을 뒤집는 작업을 수행할 수 있다
- 동전을 적절히 뒤집어서  $T$ 의 최소 개수를 구하는 문제

# 동전 뒤집기

34

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

- 각각의 칸을 바꿀 수 있는 방법은 모두 두 가지이다

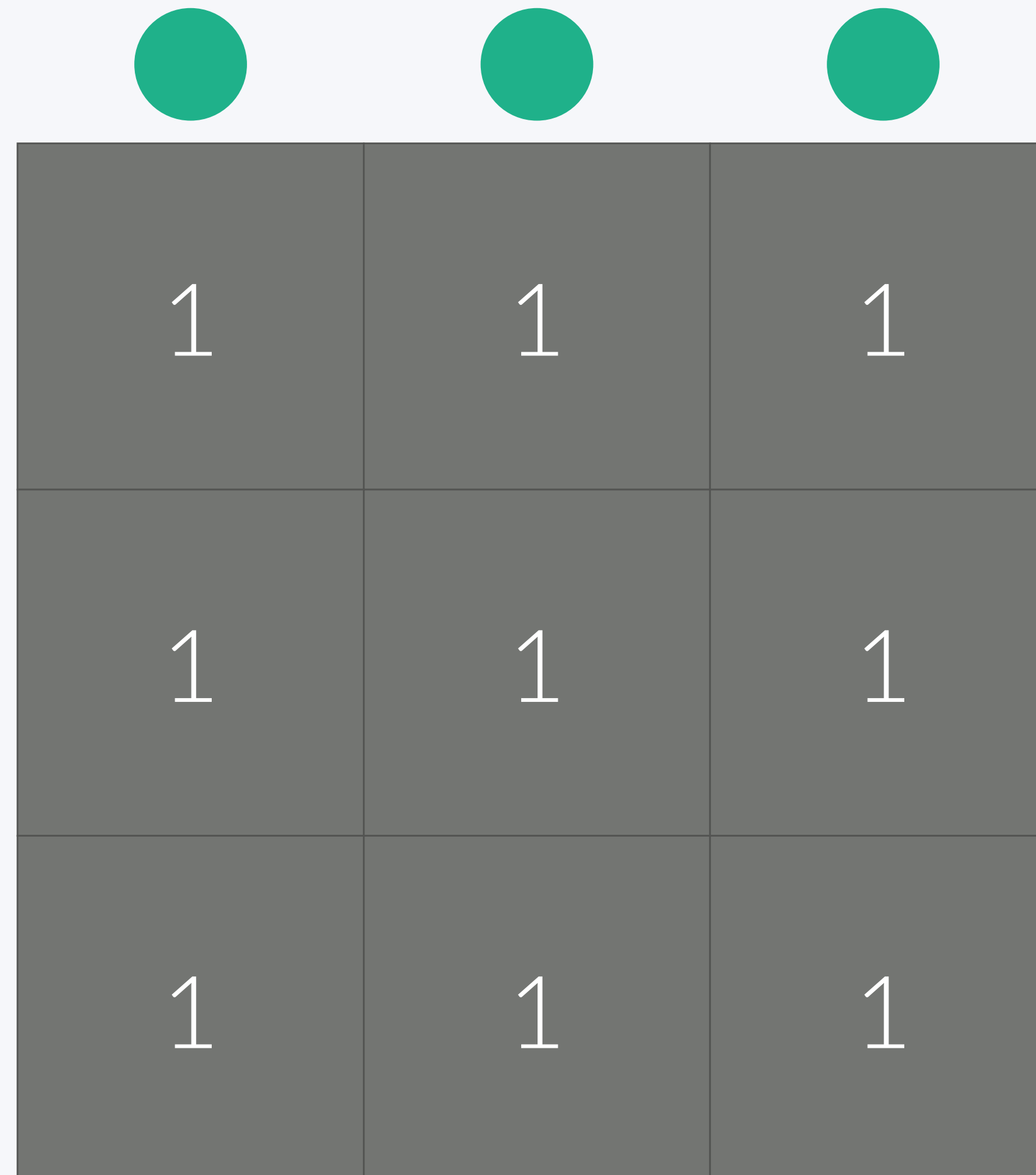


# 동전 뒤집기

35

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

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



1	1	1
1	1	1
1	1	1

# 동전 뒤집기

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

- C++: <https://gist.github.com/Baekjoon/a34b4c0a7079bd99885849acae28714e>
- Java: <https://gist.github.com/Baekjoon/c06eff95ca4692ca0ff25729efd1e654>

# 가르침

<https://www.acmicpc.net/problem/1062>

- N개의 단어가 주어졌을 때
- K개의 글자로만 이루어진 단어의 개수를 고르는 문제
- 모든 단어는 anta로 시작하고
- 모든 단어는 tica로 끝난다
- $N \leq 50$ , 단어의 길이  $\leq 15$

# 가르침

<https://www.acmicpc.net/problem/1062>

- N개의 단어가 주어졌을 때
- K개의 글자로만 이루어진 단어의 개수를 고르는 문제
- 모든 단어는 anta로 시작하고
- 모든 단어는 tica로 끝난다
- 먼저, a, n, t, i, c는 가르쳐야 한다
- 즉, 26-5개의 글자 중에서 K-5개를 고르는 문제

# 가르침

<https://www.acmicpc.net/problem/1062>

- 경우의 수가 몇 개 되지 않기 때문에, 모든 경우를 다 시뮬레이션 해본다

# 가르침

<https://www.acmicpc.net/problem/1062>

```
int go(int index, int k, int mask, vector<int> &word) {
    if (k < 0) return 0;
    if (index == 26) {
        int cnt = 0;
        for (int i=0; i<word.size(); i++) {
            if ((word[i] & ((1<<26)-1-mask)) == 0) cnt += 1;
        }
        return cnt;
    }
    int ans = 0;
    int t1 = go(index+1, k-1, mask | (1 << index), word);
    if (ans < t1) ans = t1;
```



# 가르침

41

<https://www.acmicpc.net/problem/1062>

```
    if (index != 'a'-'a' && index != 'n'-'a' && index != 't'-'a'
        && index != 'i'-'a' && index != 'c'-'a') {
        t1 = go(index+1, k, mask, word);
        if (ans < t1) ans = t1;
    }
    return ans;
}
```

# 가르침

<https://www.acmicpc.net/problem/1062>

- C++: <https://gist.github.com/Baekjoon/f4025edddb6f9b5c71002110df23d420>
- Java: <https://gist.github.com/Baekjoon/d115a4eece06aa734f27fba256cec013>

# 동전

<https://www.acmicpc.net/problem/9084>

- 동전의 종류가 주어졌을 때, 주어진 금액을 만드는 모든 방법을 세는 문제

# 동전

<https://www.acmicpc.net/problem/9084>

- 동전이 1, 2, 3원이 있을 때, 4원을 만드는 방법은 4가지이다.
- $4 = 1+1+1+1 = 1+1+2 = 1+3 = 2+2$
- $1+1+2$ 와  $1+2+1$ 은 같은 방법이다.

# 동전

<https://www.acmicpc.net/problem/9084>

- 동전이 1, 2, 3원이 있을 때, 4원을 만드는 방법은 4가지이다.
- $4 = 1+1+1+1 = 1+1+2 = 1+3 = 2+2$
- $1+1+2$ 와  $1+2+1$ 은 같은 방법이다.
- 즉,  $4 = 1*a + 2*b + 3*c$ 를 만족하는  $(a, b, c)$  쌍의 개수를 세는 문제이다.

# 동전

<https://www.acmicpc.net/problem/9084>

- $D[i]$  = i원을 만드는 방법의 개수
- $D[i] = \sum D[i - \text{coin}[j]]$
- 이 때, 동전의 개수를 먼저 고려하지 않으면, 중복해서 경우의 수를 세어버리게 된다.

# 동전

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1

# 동전

48

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1, 2+1, 1+2, 3



# 동전

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1, 2+1, 1+2, 3
- 4원: 1+1+1+1, 2+1+1, 1+2+1, 3+1, 1+1+2, 2+2, 1+3

# 동전

50

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1

# 동전

51

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1

# 동전

52

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1
- 3원: 1+1+1

# 동전

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1
- 3원: 1+1+1
- 4원: 1+1+1+1

# 동전

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1
- 4원: 1+1+1+1

# 동전

55

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1, 1+2
- 4원: 1+1+1+1

# 동전

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1, 1+2
- 4원: 1+1+1+1, 1+1+2, 2+2



# 동전

57

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1, 1+2, 3
- 4원: 1+1+1+1, 1+1+2, 2+2

# 동전

58

<https://www.acmicpc.net/problem/9084>

- 0원:
- 1원: 1
- 2원: 1+1, 2
- 3원: 1+1+1, 1+2, 3
- 4원: 1+1+1+1, 1+1+2, 2+2, 1+3

# 동전

<https://www.acmicpc.net/problem/9084>

- C++: <https://gist.github.com/Baekjoon/ef886c2e6f799f2382f77ebc095a38a8>
- Java: <https://gist.github.com/Baekjoon/7b4e611f08d03b0479e1773dc5c6f4a6>

# 2048 (Easy)

<https://www.acmicpc.net/problem/12100>

60

- 2048 게임에서 최대 5번 이동시켜서 얻을 수 있는 가장 큰 블록을 출력하는 문제

# 2048 (Easy)

<https://www.acmicpc.net/problem/12100>

- 이동 횟수가 5번이기 때문에,  $4^5 = 1024$ 번 이동을 시켜보면 된다.

# 2048 (Easy)

62

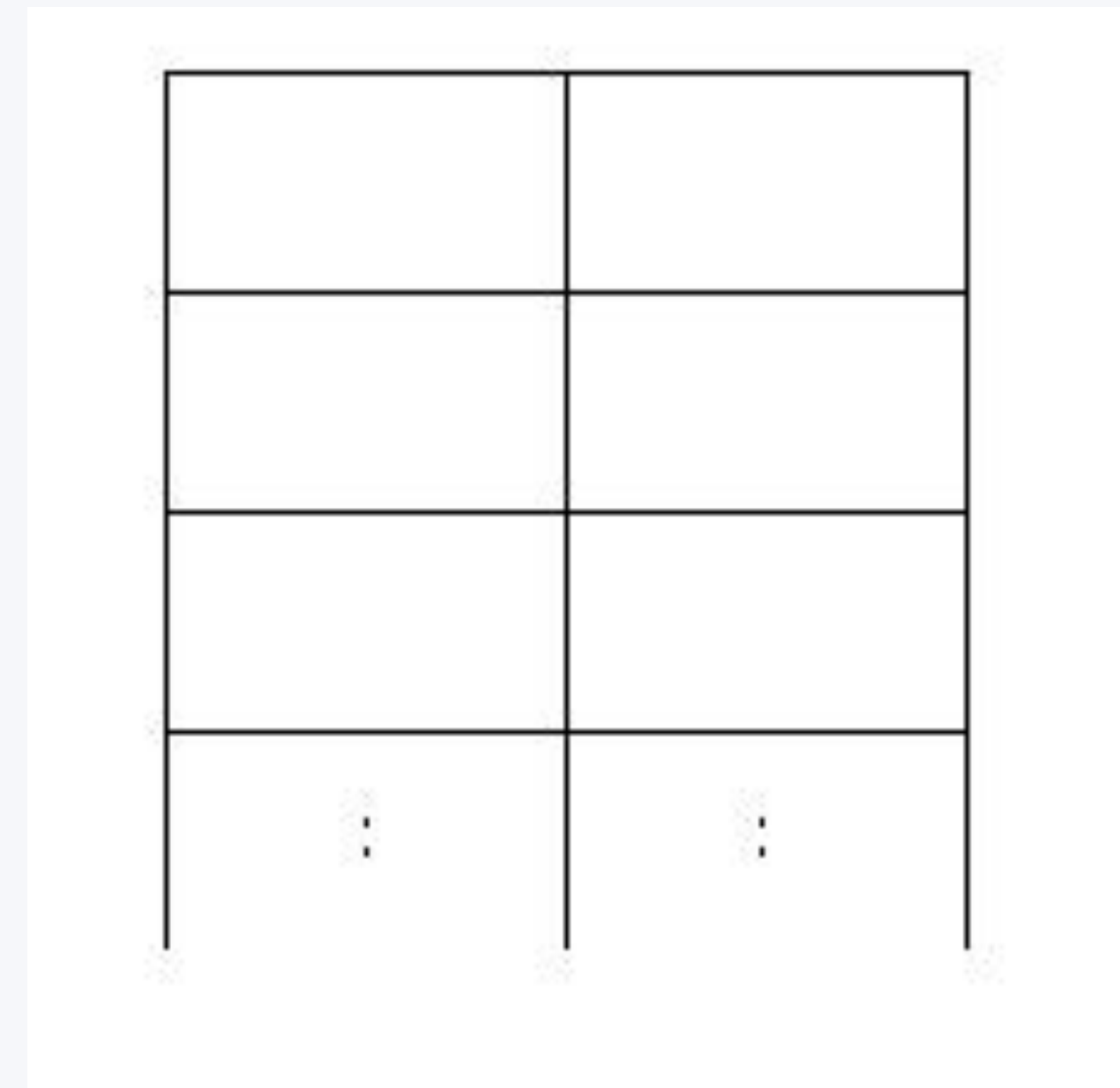
<https://www.acmicpc.net/problem/12100>

- C++: <https://gist.github.com/Baekjoon/522d138cf5ff54abc22e49374a7aeedc>
- Java: <https://gist.github.com/Baekjoon/4a8748c9675eb1562d744b7f87a6b1da>

# 동물원

<https://www.acmicpc.net/problem/1309>

- 가로로 두 칸, 세로로  $N$  칸인 동물원이 있다
- 가로, 세로로 붙어 있게 배치하면 안된다
- 가능한 배치의 수

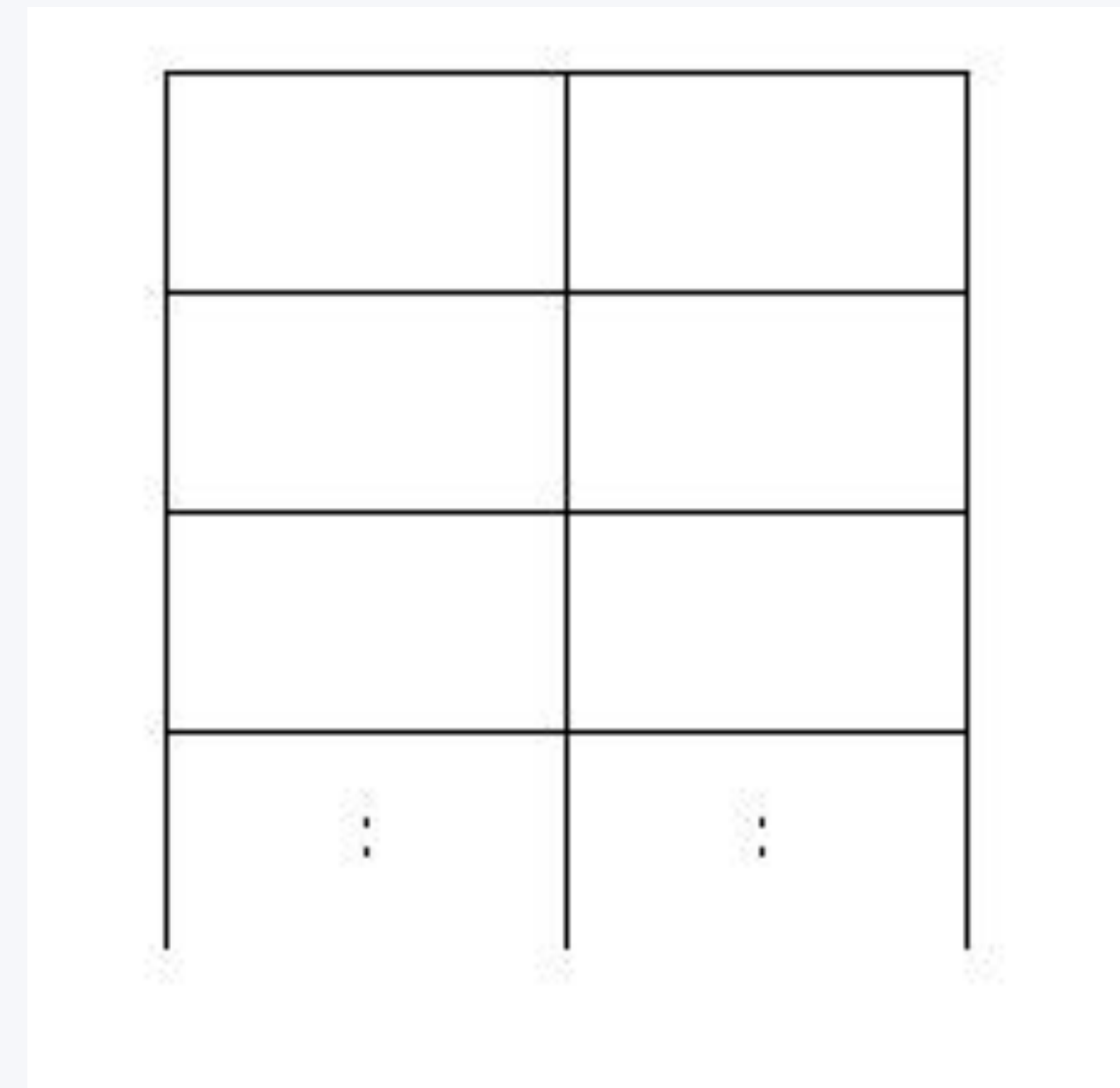


# 동물원

64

<https://www.acmicpc.net/problem/1309>

- $D[N][0]$  = N번 줄에 배치하지 않음
- $D[N][1]$  = N번 줄의 왼쪽에만 배치함
- $D[N][2]$  = N번 줄의 오른쪽에만 배치함



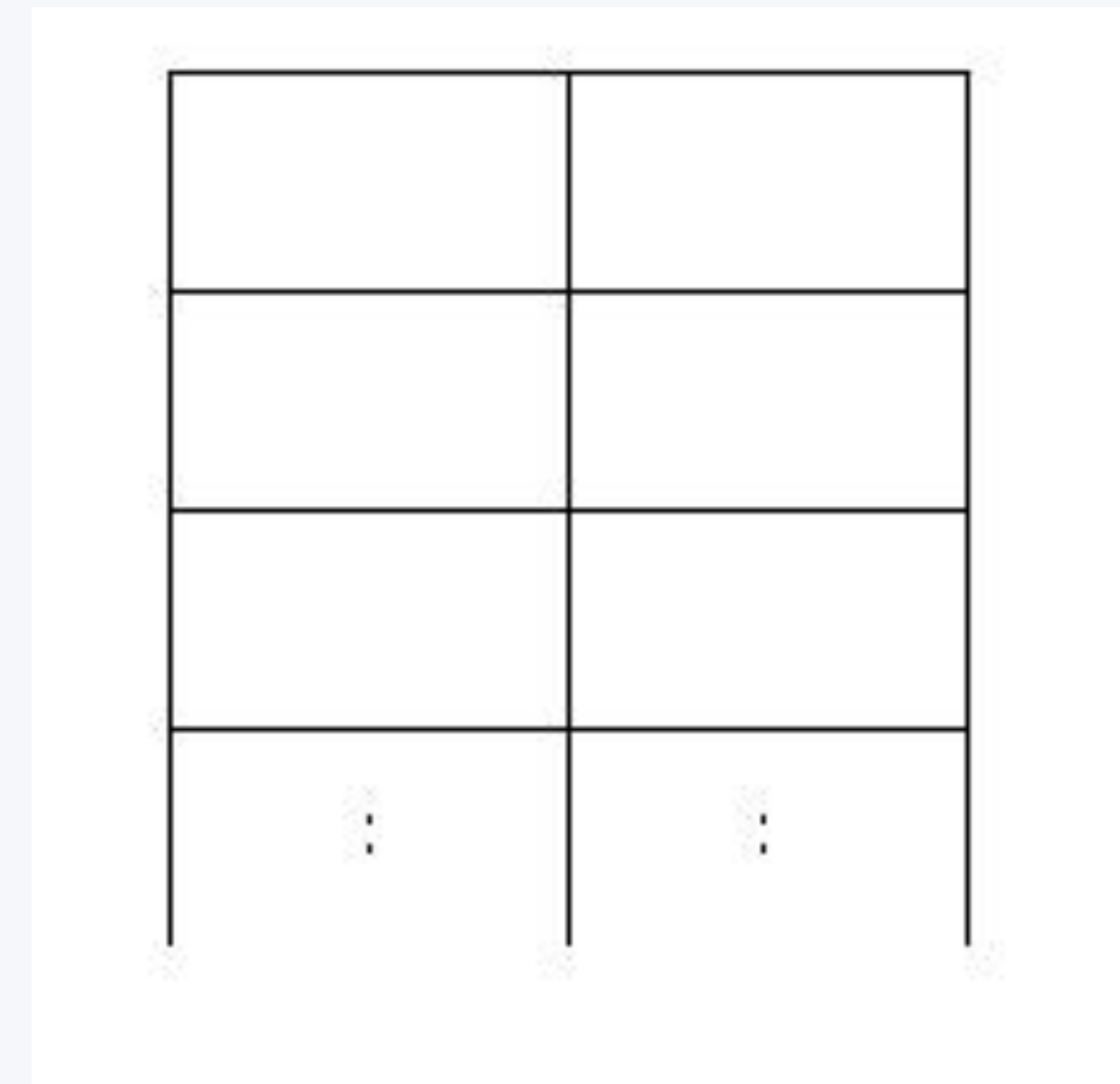


# 동물원

65

<https://www.acmicpc.net/problem/1309>

- $D[N][0] = N$ 번 줄에 배치하지 않음
- $D[N][1] = N$ 번 줄의 왼쪽에만 배치함
- $D[N][2] = N$ 번 줄의 오른쪽에만 배치함
- $D[N][0] = D[N-1][0] + D[N-1][1] + D[N-1][2]$
- $D[N][1] = D[N-1][0] + D[N-1][2]$
- $D[N][2] = D[N-1][0] + D[N-1][1]$



# 동물원

<https://www.acmicpc.net/problem/1309>

- C/C++: <https://gist.github.com/Baekjoon/a15a88eee552b047b80d31ba9cf7a789>
- Java: <https://gist.github.com/Baekjoon/0bd270c392c8bb4f3dc3dc108b18c99f>

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- 바이너리 파워 비숍이  $(x, y)$ 에 있다면
- $(x + 2^k, y + 2^k)$ ,  $(x + 2^k, y - 2^k)$ ,  $(x - 2^k, y + 2^k)$ ,  $(x - 2^k, y - 2^k)$  로 이동할 수 있으며
- $k$ 는 음이 아닌 정수이다
- 한 번 사용한  $k$ 는 다시 사용할 수 없다
- $(0, 0)$ 에서  $(x, y)$ 로 이동하려고 할 때, 방문하는 칸의 개수를 최소로 하는 문제

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- $go(x, y, p)$ 를 다음과 같이 정의하자
- $(0, 0)$ 에서  $(x, y)$ 까지 가는 최소 칸을 구하는 함수
- 이 때, 사용할 수 있는  $k$ 는  $0 \leq k \leq p$

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- $go(x, y, p)$ :  $(0, 0)$ 에서  $(x, y)$ 까지 가는 최소 칸을 구하는 함수, 사용할 수 있는  $k$ 는  $0 \leq k \leq p$
- $x = 0$  and  $y = 0$ 인 경우: 0
- $x = 0$  or  $y = 0$ 인 경우: 방법이 없다
- $x \neq 0$  and  $y \neq 0$ 인 경우

# 바이너리 파워 비숍

70

<https://www.acmicpc.net/problem/1601>

- $go(x, y, p)$ :  $(0, 0)$ 에서  $(x, y)$ 까지 가는 최소 칸을 구하는 함수, 사용할 수 있는  $k$ 는  $0 \leq k \leq p$
- $x \neq 0$  and  $y \neq 0$ 인 경우
- $2^p$ 를 사용하지 않는 경우:  $go(x, y, p-1)$
- $2^p$ 를 사용하는 경우

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- $go(x, y, p)$ :  $(0, 0)$ 에서  $(x, y)$ 까지 가는 최소 칸을 구하는 함수, 사용할 수 있는  $k$ 는  $0 \leq k \leq p$
- $x \neq 0$  and  $y \neq 0$ 인 경우
- $2^p$ 를 사용하는 경우
  - $x > 0$  and  $y > 0$ :  $go(x-2^p, y-2^p, p-1)$
  - $x > 0$  and  $y < 0$ :  $go(x-2^p, y+2^p, p-1)$
  - $x < 0$  and  $y > 0$ :  $go(x+2^p, y-2^p, p-1)$
  - $x < 0$  and  $y < 0$ :  $go(x+2^p, y+2^p, p-1)$

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- $1 \leq x, y \leq 100,000,000 < 2^{27}$
- 즉,  $go(x, y, 27)$ 을 호출해야 한다
- 시간 복잡도:



# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- $1 \leq x, y \leq 100,000,000 < 2^{27}$
- 즉,  $go(x, y, 27)$ 을 호출해야 한다
- 시간 복잡도:  $O(2^{28})$

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- $|x| > 2^{(p+1)}$  또는  $|y| > 2^{(p+1)}$  이면 정답을 구할 수 없다
- $2^0 + 2^1 + \dots + 2^p = 2^{(p+1)} - 1$  이기 때문

# 바이너리 파워 비숍

<https://www.acmicpc.net/problem/1601>

- C++: <https://gist.github.com/Baekjoon/05298ac3f96ca75655a773dddf455fdaa>
- Java: <https://gist.github.com/Baekjoon/3f2ea11a725efc38fcaa32ffa4b0a00c>

# 단위 분수로 분할

<https://www.acmicpc.net/problem/2404>

- 네 개의 양의정수  $p, q, a, n$ 이 주어졌을 때 다음 두 조건을 만족하는  $p/q$ 의 분할의 개수를 구하는 문제
1.  $n$ 개 이하의 단위분수의 합으로 나타내야 한다.
  2. 분할을 이루는 단위분수의 분모의 크기의 곱은  $a$ 보다 작거나 같아야 한다.
- $(p, q, a, n)$ 이  $(2, 3, 120, 3)$ 일 때 답은 4
  - $2/3 = 1/3 + 1/3 = 1/2 + 1/6 = 1/4 + 1/4 + 1/6 = 1/3 + 1/6 + 1/6$

# 단위 분수로 분할

<https://www.acmicpc.net/problem/2404>

- `go(p, q, a, n, mul, last)`
- 네 개의 양의정수  $p, q, a, n$ 이 주어졌을 때  $p/q$ 의 분할의 개수
- 이 때, 분모의 곱: `mul`
- 마지막으로 사용한 분모: `last`

# 단위 분수로 분할

<https://www.acmicpc.net/problem/2404>

- `go(p, q, a, n, mul, last)`
- 네 개의 양의정수  $p, q, a, n$ 이 주어졌을 때  $p/q$ 의 분할의 개수
- 이 때, 분모의 곱: `mul`
- 마지막으로 사용한 분모: `last`
- 정답을 찾은 경우: `p == 0`일 때

# 단위 분수로 분할

<https://www.acmicpc.net/problem/2404>

- `go(p, q, a, n, mul, last)`
- 네 개의 양의정수  $p, q, a, n$ 이 주어졌을 때  $p/q$ 의 분할의 개수
- 이 때, 분모의 곱: `mul`
- 마지막으로 사용한 분모: `last`
- 불가능한 경우
- $n == 0$  일 때 (더 이상 사용할 분수가 없음)
- $p/q > n/last$  일 때 (사용할 수 있는 가장 큰 분수를 아무리 많이 사용해도 0을 만들 수 없음)
- $mul * last > a$  일 때 (문제의  $a$  조건을 넘어가버림)

# 단위 분수로 분할

<https://www.acmicpc.net/problem/2404>

```
long long go(long long p, long long q, long long a, int n, long
long mul, long long last) {
    if (p == 0) {
        if (mul <= a) return 1;
        else return 0;
    }
    if (n == 0 || p*last > n*q || mul*last > a) return 0;
    long long ans = 0;
    for (long long i=last; i<=a; i++) {
        if (mul*i > a) break;
        long long bj = p*i-q, bm = q*i;
        if (bj >= 0) ans += go(bj, bm, a, n-1, mul*i, i);
    }
    return ans;
}
```



# 단위 분수로 분할

<https://www.acmicpc.net/problem/2404>

- C++: <https://gist.github.com/Baekjoon/8f324aaead396e92fdffdbd880a7717e>
- Java: <https://gist.github.com/Baekjoon/fd797e08c2c27b1a9630f85fc51e1612>

# 소형기관차

<https://www.acmicpc.net/problem/2616>

- $N$  ( $\leq 50,000$ )개의 객차가 있고, 각 객차에 타고 있는 사람의 수  $A[i]$  명
- 3개의 소형 기관차가 최대로 끌 수 있는 객차의 수는 같다
- 소형 기관차 3대를 이용하여 최대한 많은 손님을 목적지까지 운송하도록 한다
- 각 소형 기관차는 번호가 연속적으로 이어진 객차를 끌게 한다.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
35	40	50	10	30	45	60

# 소형기관차

<https://www.acmicpc.net/problem/2616>

- $D[i][j]$  =  $i$ 번 사람까지 있을 때,  $j$ 개의 소형 기관차로 운송할 수 있는 최대 손님 수

# 소형기관차

<https://www.acmicpc.net/problem/2616>

- $D[i][j] = \max(D[i-1][j], D[i-m][j-1] + (A[i-m+1] \sim A[i] \text{ 까지 합}))$

# 소형기관차

<https://www.acmicpc.net/problem/2616>

- C++: <https://gist.github.com/Baekjoon/5e8ddb17610f4694afa8deafbdabd86a>
- Java: <https://gist.github.com/Baekjoon/2d6a467424250c03bb485f33fac1a8a7>

# 정상 회담 2

86

<https://www.acmicpc.net/problem/1670>

- 팔이 교차되면 안되는 악수의 수를 구하는 문제

# 정상 회담 2

<https://www.acmicpc.net/problem/1670>

- 잘 생각해보면, 올바른 괄호 문자열의 개수를 세는 문제와 같다
- $D[N] = N$ 명이 있었을 때, 가능한 악수 방법의 수

# 정상 회담 2

<https://www.acmicpc.net/problem/1670>

- 잘 생각해보면, 올바른 괄호 문자열의 개수를 세는 문제와 같다
- $D[N] = N$ 명이 있었을 때, 가능한 악수 방법의 수
- $D[N] = \sum D[i] * D[n-i-1] \ (0 < i \leq n)$



# 정상 회담 2

<https://www.acmicpc.net/problem/1670>

- 잘 생각해보면, 올바른 괄호 문자열의 개수를 세는 문제와 같다
- $D[N] = N$ 명이 있었을 때, 가능한 악수 방법의 수
- $D[N] = \sum D[i] * D[n-i-1] \ (0 < i \leq n)$

# 정상 회담 2

<https://www.acmicpc.net/problem/1670>

- C++: <https://gist.github.com/Baekjoon/637b19f5b0c78897d00d67349583ab43>
- Java: <https://gist.github.com/Baekjoon/aaaaacd4120eb664dfd3afa1057660f0>

# 돌다리 건너기

<https://www.acmicpc.net/problem/2602>

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

# 돌다리 건너기

<https://www.acmicpc.net/problem/2602>

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

# 돌다리 건너기

<https://www.acmicpc.net/problem/2602>

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

# 돌다리 건너기

<https://www.acmicpc.net/problem/2602>

- $D[k][i][j]$  = k번째 돌다리의 i번째 문자와 j번째 문자를 밟는 경우의 수

# 돌다리 건너기

<https://www.acmicpc.net/problem/2602>

```
d[0][0][0] = d[1][0][0] = 1;
for (int j=1; j<=m; j++) {
    for (int i=1; i<=n; i++) {
        for (int k=0; k<2; k++) {
            if (a[k][i] != s[j]) continue;
            for (int l=i-1; l>=0; l--) {
                if (a[1-k][l] == s[j-1]) {
                    d[k][i][j] += d[1-k][l][j-1];
                }
            }
        }
    }
}
```

# 돌다리 건너기

<https://www.acmicpc.net/problem/2602>

- C++: <https://gist.github.com/Baekjoon/c6ed60dfc5bc8e43698da6bd44602a3b>
- Java: <https://gist.github.com/Baekjoon/7de92cbd4fc8fb45cb19269311f3d07c>



# 미네랄

<https://www.acmicpc.net/problem/2933>

- 동굴의 모양이 주어진다
- 두 사람은 던질 막대의 높이를 결정하고 막대를 던진다.
- 이 때, 미네랄을 만나면 미네랄은 모두 파괴되고, 막대는 그 위치에서 이동을 멈춘다.
- 새로운 클러스터가 공중에 떠있으면 중력에 의해서 아래로 내려온다.

# 미네랄

<https://www.acmicpc.net/problem/2933>

- 입력으로 주어진 대로 시뮬레이션을 해본다.
- 어려운 부분은 클러스터가 떨어지는 것을 구현하는 부분

# 미네랄

<https://www.acmicpc.net/problem/2933>

- 동굴의 모양이 주어진다
- 두 사람은 던질 막대의 높이를 결정하고 막대를 던진다.
- 이 때, 미네랄을 만나면 미네랄은 모두 파괴되고, 막대는 그 위치에서 이동을 멈춘다.
- 새로운 클러스터가 공중에 떠있으면 중력에 의해서 아래로 내려온다.

# 미네랄

100

<https://www.acmicpc.net/problem/2933>

- C++: <https://gist.github.com/Baekjoon/3b5b4905a2c2a23ccfadc9f3a1cc7006>
- Java: <https://gist.github.com/Baekjoon/3d53790b365c897dbabb84bd7be730cc>

# 벽 부수고 이동하기 2

101

<https://www.acmicpc.net/problem/14442>

- $N \times M$ 의 행렬로 나타내는 지도에서 (1, 1)에서 (N,M)으로 최단 거리로 이동하는 문제
- 0은 빈 칸, 1은 벽
- 단, 벽은 한 번 부수고 지나갈 수 있다

# 벽 부수고 이동하기 2

<https://www.acmicpc.net/problem/14442>

- 벽을 부순다는 조건이 없으면 일반적인 미로 탐색 문제이다
- 어떤 칸에 방문했을 때, 벽을 부순 적이 있는 경우와 아직 부순 적이 없는 경우는 다른 경우이기 때문에
- 상태  $(i, j)$  대신에  $(i, j, k)$  ( $k$ 는 벽을 부순 횟수) 으로 BFS 탐색을 진행한다.

# 벽 부수고 이동하기 2

103

<https://www.acmicpc.net/problem/14442>

- C++: <https://gist.github.com/Baekjoon/848ef609f899b93bf7b9221f7c598462>
- Java: <https://gist.github.com/Baekjoon/b32c17491c283bd522d4d5bb0d44f75a>

# 짜합 수열

<https://www.acmicpc.net/problem/12103>

- 정수 수열  $a_1, a_2, \dots, a_n$ 이 있을 때
- $1 \leq i \leq n - k + 1$  인 모든 정수  $i$ 에 대해서,  $a_i + a_{i+1} + \dots + a_{i+k-1}$  이 짝수라면, 이 수열을  $k$ -짜합 수열이라고 정의한다.
- 당신은 수열에 있는 몇 개의 원소를 원하는 정수로 바꿀 수 있다.
- 최소 몇 개의 원소를 바꿔야지 수열을  $k$ -짜합 수열로 만들 수 있는가?



# 짜합 수열

<https://www.acmicpc.net/problem/12103>

- 정수 수열  $a_1, a_2, \dots, a_n$ 이 있을 때
- $1 \leq i \leq n - k + 1$  인 모든 정수  $i$ 에 대해서,  $a_i + a_{i+1} + \dots + a_{i+k-1}$  이 짝수라면, 이 수열을  $k$ -짜합 수열이라고 정의한다.
- 당신은 수열에 있는 몇 개의 원소를 원하는 정수로 바꿀 수 있다.
- 최소 몇 개의 원소를 바꿔야지 수열을  $k$ -짜합 수열로 만들 수 있는가?

# 짜합 수열

106

<https://www.acmicpc.net/problem/12103>

- 모든  $1 \leq i \leq n - k$  에 대해서
- $a_i + a_{i+1} + \dots + a_{i+k-1} \equiv a_{i+1} + \dots + a_{i+k-1} + a_{i+k} \pmod{2}$
- 이다.

# 짜합 수열

107

<https://www.acmicpc.net/problem/12103>

- 양변을 정리하면
- $1 \leq i \leq n - k$
- 에 대해서
- $a_i \equiv a_{i+k} \pmod{2}$  이다.

# 짜합 수열

<https://www.acmicpc.net/problem/12103>

- $a_1, \dots, a_k$ 의 홀짝을 정해주면, 나머지의 홀짝은 자동으로 결정된다.

# 짝합 수열

109

<https://www.acmicpc.net/problem/12103>

- $a_1 + a_2 + \dots + a_k \equiv 0 \pmod{2}$
- 이다.
- $a_i$ 가 짝수이면,  $a_i, a_{i+k}, a_{i+2k}, \dots$  을 짝수로 만들어야 한다.

# 짜합 수열

110

<https://www.acmicpc.net/problem/12103>

- $D[i][j] = i$ 번째 원소까지 합이  $j \pmod{2}$  인 경우의 수

# 짜합 수열

<https://www.acmicpc.net/problem/12103>

- C++: <https://gist.github.com/Baekjoon/af2be0a8f2fa5797c0c203e1c8686237>
- Java: <https://gist.github.com/Baekjoon/66ca0e1840ba402380d0f91a2a3bb8e5>