

# 8/21 - 2

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

---

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-L 햄버거는 햄버거번, 레벨-(L-1) 버거, 패티, 레벨-(L-1) 버거, 햄버거번
- 레벨-1 버거는 BPPPB
- 레벨-2 버거는 BBPPBPBPPPB
- 레벨-N버거의 아래 X장을 먹었을 때, 먹은 패티의 개수는?

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-N 버거가 몇 장으로 이루어져 있는지 구할 수 있다.
- 레벨-0 햄버거는 패티
  - $D[0] = 1$
- 레벨-L 햄버거는 햄버거번, 레벨-(L-1) 버거, 패티, 레벨-(L-1) 버거, 햄버거번
  - $D[L] = 1 + D[L-1] + 1 + D[L-1] + 1 = 2D[L-1] + 3$

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-N 버거가 몇 장의 패티로 이루어져 있는지 구할 수 있다.
- 레벨-0 햄버거는 패티
  - $P[0] = 1$
- 레벨-L 햄버거는 햄버거번, 레벨-(L-1) 버거, 패티, 레벨-(L-1) 버거, 햄버거번
  - $P[L] = P[L-1] + 1 + P[L-1] = 2P[L-1] + 1$

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-N 햄버거는 햄버거번, 레벨-(N-1) 버거, 패티, 레벨-(N-1) 버거, 햄버거번
- $go(N, X)$ 를 레벨-N 햄버거의 아래 X장을 먹었을 때, 먹은 패티의 수 라고 정의하자.
- $N = 0$ 인 경우
  - $X = 0$ 이면 0, 그 외에는 1

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-N 햄버거는 햄버거번, 레벨-(N-1) 버거, 패티, 레벨-(N-1) 버거, 햄버거번
- $go(N, X)$ 를 레벨-N 햄버거의 아래 X장을 먹었을 때, 먹은 패티의 수 라고 정의하자.
- $X = 1$ 인 경우
  - 0

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-N 햄버거는 햄버거번, 레벨-(N-1) 버거, 패티, 레벨-(N-1) 버거, 햄버거번
- $go(N, X)$ 를 레벨-N 햄버거의 아래 X장을 먹었을 때, 먹은 패티의 수 라고 정의하자.
- $1 < X \leq 1 + D[N-1]$ 인 경우
  - $go(N-1, X-1)$

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-N 햄버거는 햄버거번, 레벨-(N-1) 버거, 패티, 레벨-(N-1) 버거, 햄버거번
- $go(N, X)$ 를 레벨-N 햄버거의 아래 X장을 먹었을 때, 먹은 패티의 수 라고 정의하자.
- $X = 1 + D[N-1] + 1$ 인 경우
  - $P[N-1] + 1$



# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-N 햄버거는 햄버거번, 레벨-(N-1) 버거, 패티, 레벨-(N-1) 버거, 햄버거번
- $go(N, X)$ 를 레벨-N 햄버거의 아래 X장을 먹었을 때, 먹은 패티의 수 라고 정의하자.
- $1 + D[N-1] + 1 < X \leq 1 + D[N-1] + 1 + D[N-1]$ 인 경우
  - $P[N-1] + 1 + go(N-1, X - 1 - D[N-1] - 1)$

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

- 레벨-0 햄버거는 패티
- 레벨-N 햄버거는 햄버거번, 레벨-(N-1) 버거, 패티, 레벨-(N-1) 버거, 햄버거번
- $go(N, X)$ 를 레벨-N 햄버거의 아래 X장을 먹었을 때, 먹은 패티의 수 라고 정의하자.
- $X = 1 + D[N-1] + 1 + D[N-1] + 1$ 인 경우
  - $P[N-1] + 1 + P[N-1]$

# 레벨 햄버거

<https://www.acmicpc.net/problem/16974>

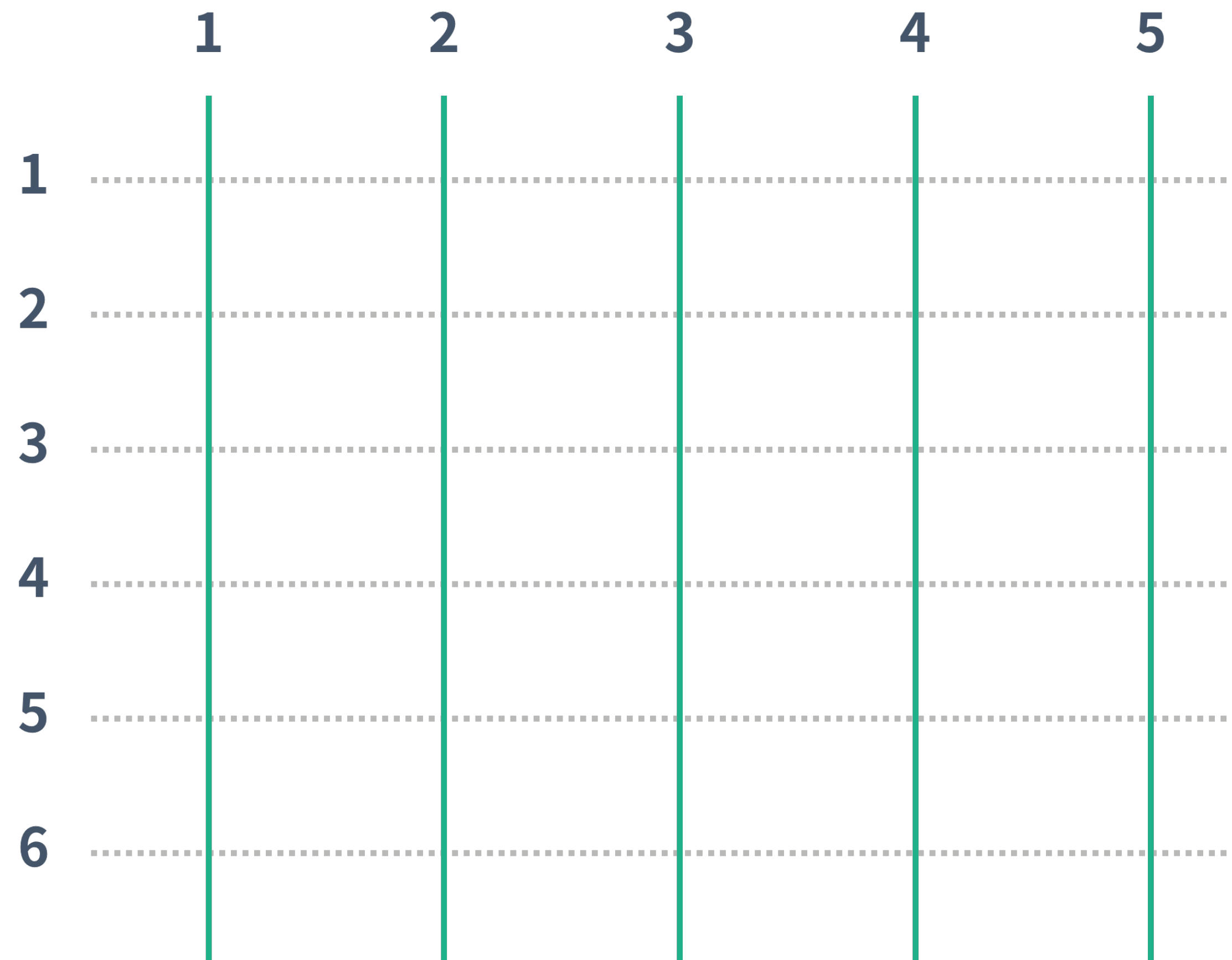
- 소스: <http://codeplus.codes/ca95d82a459d42969004f0b9e442bf69>

# 사다리 조작

12

<https://www.acmicpc.net/problem/15684>

- N개의 세로선, M개의 가로선으로 이루어진 사다리 게임이 있다

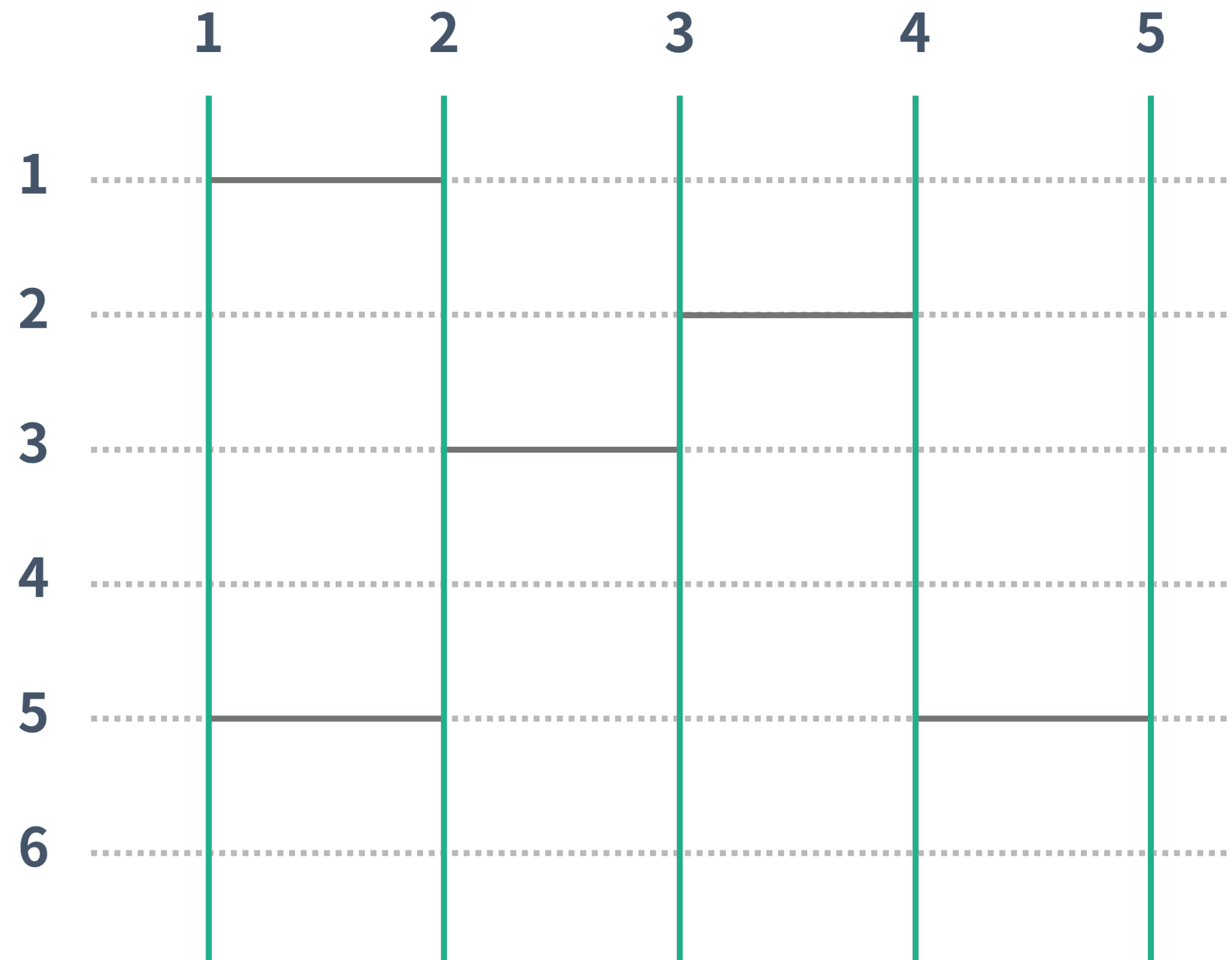


# 사다리 조작

13

<https://www.acmicpc.net/problem/15684>

- 가로선을 최소로 추가해서  $i$ 번 세로선의 결과를  $i$ 로 만드는 문제



# 사다리 조작

<https://www.acmicpc.net/problem/15684>

- 세로선의 개수  $2 \leq N \leq 10$
- 세로선마다 가로선을 놓을 수 있는 위치의 개수  $1 \leq H \leq 30$
- 가로선의 개수  $0 \leq M \leq (N-1) \times H$
- 정답이 3보다 크면 -1, 불가능해도 -1

# 사다리 조작

15

<https://www.acmicpc.net/problem/15684>

- 전체 경우의 수는  $(N-1) \times H$ 개 중에서 3개를 고르는 경우의 수와 같다
- $9 \times 30$  중에 3개를 고르는 것은  $270^3$ 과 같다.

# 사다리 조작

<https://www.acmicpc.net/problem/15684>

- 소스: <http://codeplus.codes/3ae8c3b29285412e86fd6f9a29ee602c>



# 벽 부수고 이동하기 4

17

<https://www.acmicpc.net/problem/16946>

- $N \times M$  크기의 지도가 있다.
- 0은 이동할 수 있는 곳(빈 칸), 1은 이동할 수 없는 곳 (벽)
- 두 칸이 변을 공유할 때, 인접하다고 한다.
- 각각의 벽을 빈 칸으로 바꾸고, 그 위치에서 이동할 수 있는 칸의 개수 % 10을 구해보자.

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

4	6	0	0	3
0	0	7	3	2
0	6	0	4	0
5	0	4	0	3

# 벽 부수고 이동하기 4

18

<https://www.acmicpc.net/problem/16946>

- 이동할 수 있는 빈 칸을 모두 그룹 짓고
- 몇 개의 칸을 이루어져 있는지 계산해보자
- 그룹 1의 크기: 7
- 그룹 2의 크기: 1
- 그룹 3의 크기: 7

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

		1	1	
1	1	1		
1		1		2
	3		3	
3	3	3	3	3

그룹

# 벽 부수고 이동하기 4

<https://www.acmicpc.net/problem/16946>

- 이동할 수 있는 빈 칸을 모두 그룹 짓고
- 몇 개의 칸을 이루어져 있는지 계산해보자
- 그룹 1의 크기: 7
- 그룹 2의 크기: 1
- 그룹 3의 크기: 7
- 근처에 있는 빈 칸의 그룹은 1 이다. 따라서 정답 = 1 + 7 = 8

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

		1	1	
1	1	1		
1		1		2
	3		3	
3	3	3	3	3

그룹

# 벽 부수고 이동하기 4

20

<https://www.acmicpc.net/problem/16946>

- 이동할 수 있는 빈 칸을 모두 그룹 짓고
- 몇 개의 칸을 이루어져 있는지 계산해보자
- 그룹 1의 크기: 7
- 그룹 2의 크기: 1
- 그룹 3의 크기: 7
- 근처에 있는 빈 칸의 그룹은 1, 2, 3 이다. 따라서 정답 =  $1 + 7 + 1 + 7 = 16$

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

		1	1	
1	1	1		
1		1		2
	3		3	
3	3	3	3	3

그룹

# 벽 부수고 이동하기 4

<https://www.acmicpc.net/problem/16946>

- 이동할 수 있는 빈 칸을 모두 그룹 짓고
- 몇 개의 칸을 이루어져 있는지 계산해보자
- 그룹 1의 크기: 7
- 그룹 2의 크기: 1
- 그룹 3의 크기: 7
- 근처에 있는 빈 칸의 그룹은 1, 3 이다. 따라서 정답 = 1 + 7 + 7= 15

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

		1	1	
1	1	1		
1		1		2
	3		3	
3	3	3	3	3

그룹

# 벽 부수고 이동하기 4

22

<https://www.acmicpc.net/problem/16946>

- 소스: <http://codeplus.codes/2a5f61289f6f4ffaa14488e259cd853a>