

다이나믹 프로그래밍 3

최백준 choi@startlink.io

알약

<https://www.acmicpc.net/problem/4811>

- 약이 N 개 담겨있는 병이 있다
- 하루에 약을 하나씩 꺼내 먹는다
- 약은 한 조각 전체일 수도 있고, 쪼갠 반 조각일 수도 있다
- 한 조각이면 반으로 조각낸 다음, 반만 먹고 반을 병에 넣고
- 반 조각이면 먹는다
- 한 조각을 꺼낸 날은 W , 반 조각을 꺼낸 날은 H 를 적는다.
- 즉, $2N$ 일이 지나면, 길이가 $2N$ 인 문자열이 만들어진다.
- 이 때, 서로 다른 문자열의 개수를 구하는 문제

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우
- 반 조각을 먹는 경우

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F 개, 반 조각이 H 개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우: $D[F-1][H+1]$
- 반 조각을 먹는 경우: $D[F][H-1]$

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F개, 반 조각이 H개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우: $D[F-1][H+1]$
- 반 조각을 먹는 경우: $D[F][H-1]$
- $D[F][H] = D[F-1][H+1] + D[F][H-1]$

알약

<https://www.acmicpc.net/problem/4811>

- $D[F][H]$ = 약통에 약이 F 개, 반 조각이 H 개 있을 때, 약을 먹는 방법의 수
- 약을 먹는 경우: $D[F-1][H+1]$
- 반 조각을 먹는 경우: $D[F][H-1]$
- $D[F][H] = D[F-1][H+1] + D[F][H-1]$
- $D[F][0] = D[F-1][1]$
- $D[0][H] = 1$

알약

<https://www.acmicpc.net/problem/4811>

- 소스: <http://codeplus.codes/03cf361b3bd249bcbcb1de0d20b63a4a6>

욕심쟁이 판다

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

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

14	9	12	10
1	11	5	4
7	15	2	13
6	3	16	8

욕심쟁이 판다

10

<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

12

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

- Top-Down 소스: <http://codeplus.codes/96572f9207604fabb05ca349065b0a69>
- Bottom-Up 소스: <http://codeplus.codes/bbc11cdba8f749db9bcc9482f830240f>

내리막 길

<https://www.acmicpc.net/problem/1520>

- $N \times M$ 크기의 지도
- $(1,1)$ 에서 시작해서 (N,M) 로 가는 내리막 길의 개수

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

내리막 길

<https://www.acmicpc.net/problem/1520>

- $D[i][j]$ = (i,j)에서 시작해서 (N,M)로 가는 내리막 길의 개수
- $D[N][M] = 1$
- 이동하는 방향이 4방향이다.
- 이동하기와 다르게 문제의 크기가 줄어들지 않는다
- 하지만, 수가 감소하는 방향으로만 이동할 수 있기 때문에, 사이클은 생기지 않는다
- $D[i][j] += D[x][y]$
- $(i,j) \rightarrow (x,y)$ 로 이동할 수 있어야 함

내리막 길

<https://www.acmicpc.net/problem/1520>

```
int go(int x, int y) {
    if (x == n-1 && y == m-1) return 1;
    if (d[x][y]) return d[x][y];
    for (int k=0; k<4; k++) {
        int nx = x+dx[k];
        int ny = y+dy[k];
        if (0 <= nx && nx < n && 0 <= ny && ny < m) {
            if (a[x][y] > a[nx][ny]) d[x][y] += go(nx,ny);
        }
    }
    return d[x][y];
}
```

내리막 길

<https://www.acmicpc.net/problem/1520>

- Top-down 소스: <http://codeplus.codes/ceae64eabe414db4816e638ef046fe94>
- Bottom-up 소스: <http://codeplus.codes/f374c7875e0d4cf9b0e99938d3e71e94>

가장 큰 정사각형

17

<https://www.acmicpc.net/problem/1915>

- 1로 이루어진 가장 큰 정사각형을 찾는 문제

가장 큰 정사각형

18

<https://www.acmicpc.net/problem/1915>

- $D[i][j]$ = (i, j)를 오른쪽 끝으로 하는 가장 큰 정사각형의 변의 길이

가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>

- $D[i][j] = (i, j)$ 를 오른쪽 끝으로 하는 가장 큰 정사각형의 변의 길이

			1	1	1	1	
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1

가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>

- $D[i][j] = (i, j)$ 를 오른쪽 끝으로 하는 가장 큰 정사각형의 변의 길이

			1	1	1	1	
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1

가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>

- $D[i][j] = (i, j)$ 를 오른쪽 끝으로 하는 가장 큰 정사각형의 변의 길이

			1	1	1	1	
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1

가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>

- $D[i][j] = (i, j)$ 를 오른쪽 끝으로 하는 가장 큰 정사각형의 변의 길이

			1	1	1	1	
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1
			1	1	1	1	1

가장 큰 정사각형

23

<https://www.acmicpc.net/problem/1915>

- $D[i][j] = \min(D[i-1][j-1], D[i-1][j], D[i][j-1]) + 1;$

가장 큰 정사각형

24

<https://www.acmicpc.net/problem/1915>

- 소스: <http://codeplus.codes/1c7af9b150124c658222165b7b9245b3>

1, 2, 3 더하기 7

25

<https://www.acmicpc.net/problem/15992>

- 정수 n 을 1, 2, 3의 합으로 나타내는 방법의 수를 구하는 문제
- 단, 사용한 수의 개수는 m 개 이하야 한다.
- $n = 4, m = 2$
- $2+2$
- $1+3$
- $3+1$

1, 2, 3 더하기 7

26

<https://www.acmicpc.net/problem/15992>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개

1, 2, 3 더하기 7

<https://www.acmicpc.net/problem/15992>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$

1, 2, 3 더하기 7

<https://www.acmicpc.net/problem/15992>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][0] = 1$

1, 2, 3 더하기 7

<https://www.acmicpc.net/problem/15992>

- 소스: <http://codeplus.codes/298bc4f0e8f44289baf73b2b3249576f>

1, 2, 3 더하기 9

30

<https://www.acmicpc.net/problem/16195>

- 정수 n 을 1, 2, 3의 합으로 나타내는 방법의 수를 구하는 문제
- 단, 사용한 수의 개수는 m 개 **이하** 이어야 한다.
- $n = 4, m = 2$
- $2+2$
- $1+3$
- $3+1$

1, 2, 3 더하기 9

31

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][0] = 1$

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][0] = 1$
- m 개 이하로 사용해야한다.
- 정답: $D[n][1] + D[n][2] + \dots + D[n][m]$

1, 2, 3 더하기 9

35

<https://www.acmicpc.net/problem/16195>

- 소스: <http://codeplus.codes/74ce528a4bea4ce6974955b6df4eeca8>

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개 이하
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개 이하
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][j] = 1$ ($0 \leq j \leq m$)
- 정답: $D[n][m]$

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][0] = 1$
- m 개 이하로 사용해야한다.
- 정답: $D[n][1] + \dots + D[n][m-2] + D[n][m-1] + D[n][m]$

$i \setminus j$	0	1	...	$m-2$	$m-1$	m
0	1					
1						
...						
...						
$n-1$						
n						

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][0] = 1$
- m 개 이하로 사용해야한다.
- 정답: $D[n][1] + \dots + D[n][m-2] + D[n][m-1] + D[n][m]$

$i \setminus j$	$i \setminus j$	0	1	$m-2$	$m-1$	m
0	0	1				
1	1					
...	...					
...	...					
$n-1$	$n-1$					
n	n					

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][0] = 1$
- m 개 이하로 사용해야한다.
- 정답: $D[n][1] + \dots + D[n][m-2] + D[n][m-1] + D[n][m]$

$i \setminus j$	$i \setminus 0$	$i \setminus 1$	$i \setminus 2$	0	$m-2$	$m-1$	m
0	0	0	0	1			
1	1	1	1				
...				
...				
$n-1$	$n-1$	$n-1$					
n	n	n					

1, 2, 3 더하기 9

<https://www.acmicpc.net/problem/16195>

- $D[i][j]$ = i 를 1, 2, 3의 합으로 나타내는 방법의 수, 사용한 수의 개수는 j 개 이하
- $D[i][j] = D[i-1][j-1] + D[i-2][j-1] + D[i-3][j-1]$
- $D[0][j] = 1$ ($0 \leq j \leq m$)
- 정답: $D[n][m]$

1, 2, 3 더하기 9

42

<https://www.acmicpc.net/problem/16195>

- 소스: <http://codeplus.codes/8f58428ebf8644269958aa2ead02ba3c>

고층 빌딩

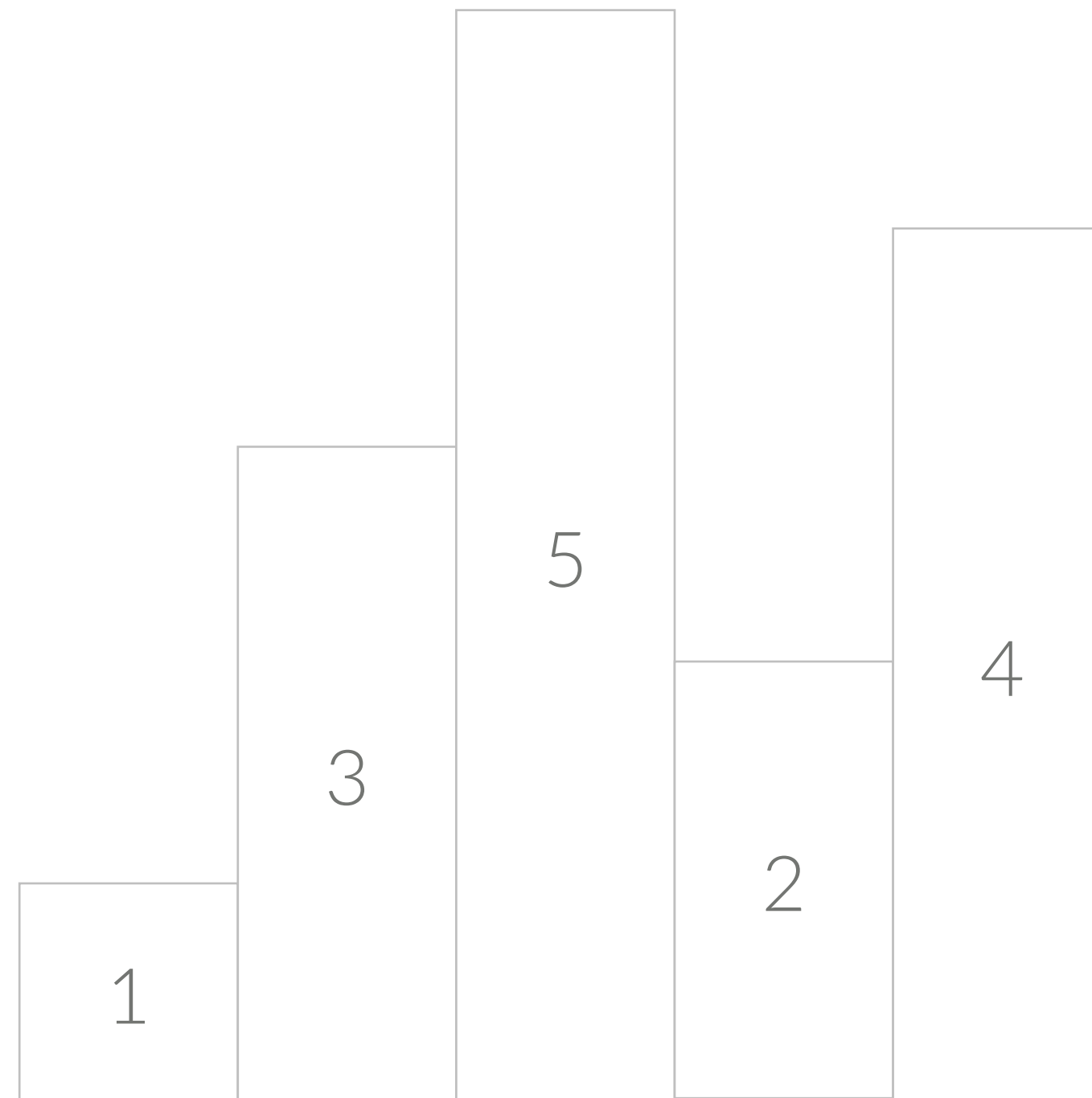
<https://www.acmicpc.net/problem/1328>

- 빌딩: N 개, 높이 $1 \sim N$
- 빌딩의 개수 N
- 가장 왼쪽에서 봤을 때 보이는 빌딩의 수 L
- 가장 오른쪽에서 봤을 때 보이는 빌딩의 수 R 이 주어졌을 때
- 가능한 빌딩 순서의 경우의 수

고층 빌딩

<https://www.acmicpc.net/problem/1328>

- $N = 5$, $L = 3$, $R = 2$ 인 경우 가능한 배치

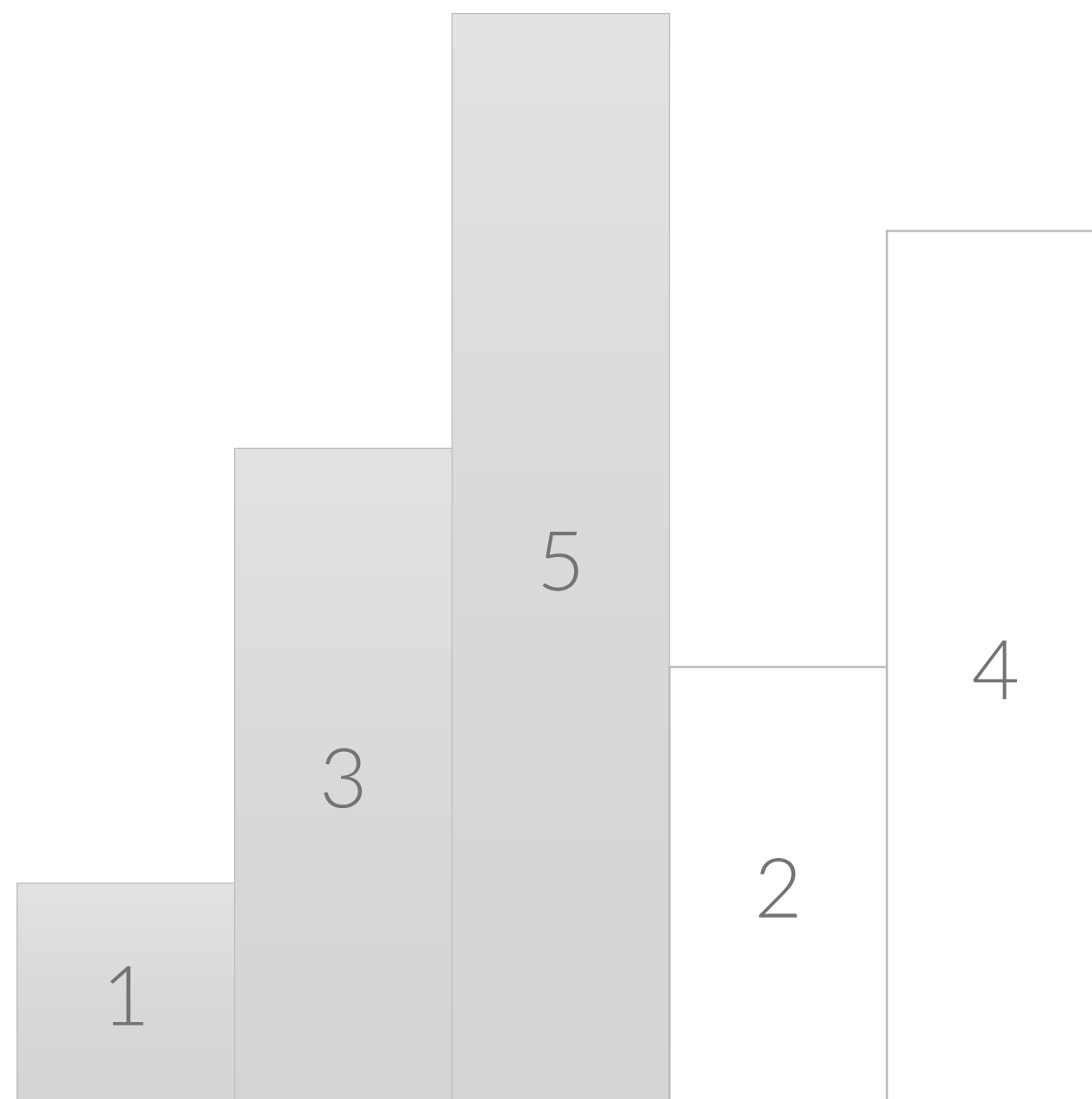


고층 빌딩

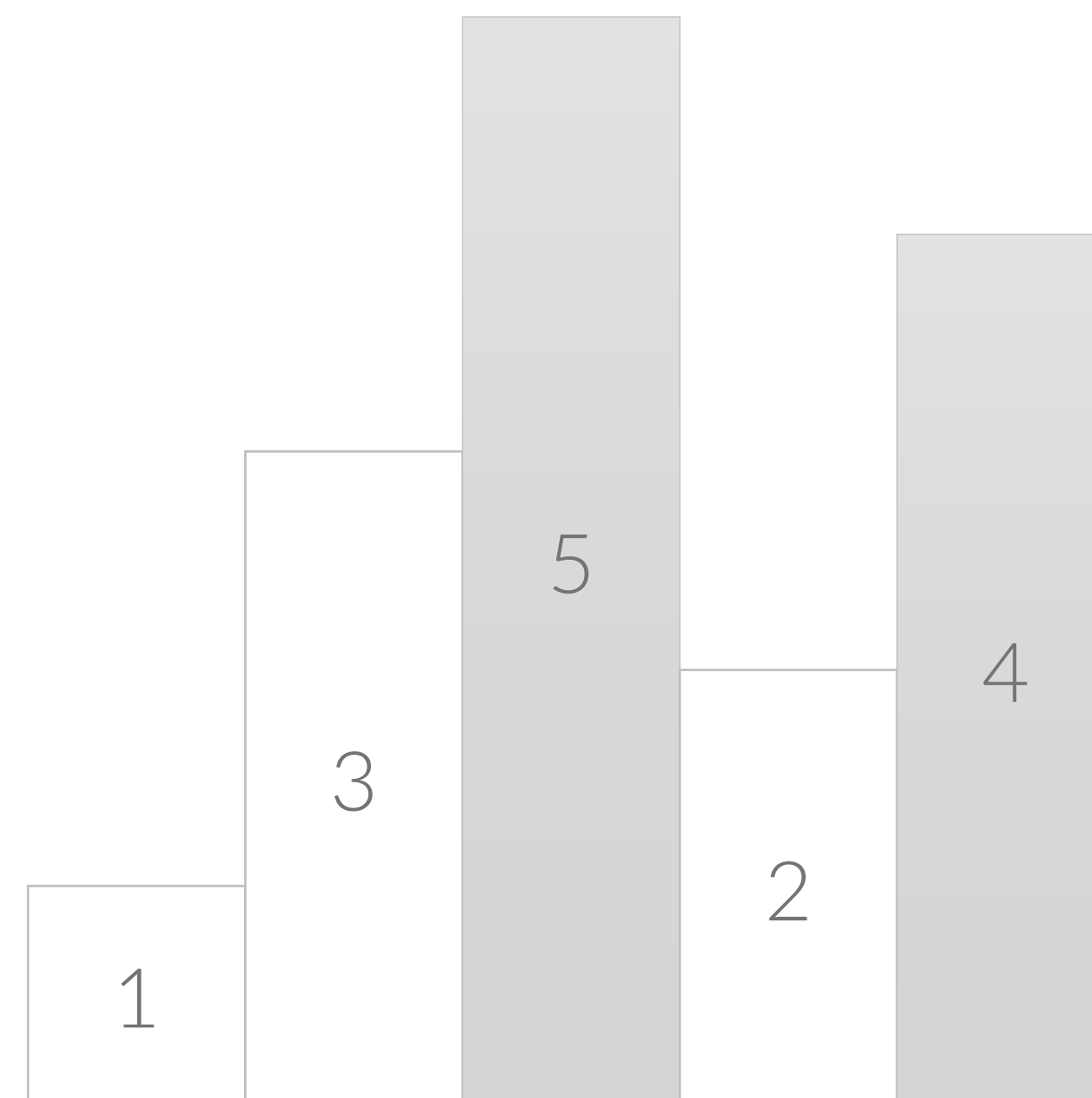
45

<https://www.acmicpc.net/problem/1328>

- $N = 5$, $L = 3$, $R = 2$ 인 경우 가능한 배치



왼쪽에서 봤을 때 3개



오른쪽에서 봤을 때 2개

고층 빌딩

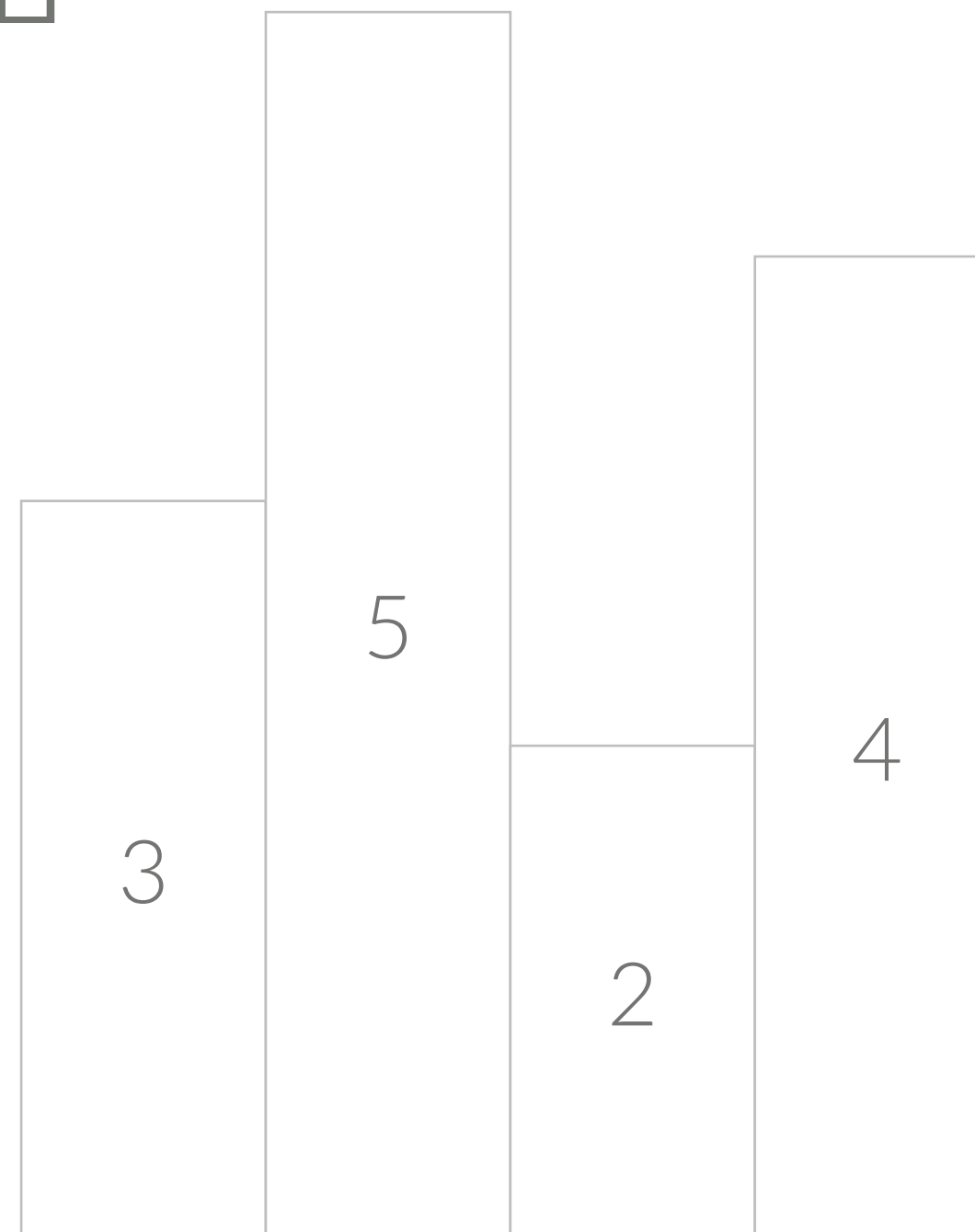
<https://www.acmicpc.net/problem/1328>

- $D[N][L][R]$ = 높이가 1~N인 빌딩 N개, 왼쪽에서 L개 보임, 오른쪽에서 R개 보이는 빌딩 배치의 개수
- 빌딩 2~N까지 이미 세워져있고, 여기에 높이가 1인 빌딩을 추가하는 방식으로 문제를 풀 수 있다.
- 빌딩 2~N까지 모두 세워져 있다.
- 여기에 높이가 1인 빌딩을 추가한다
- 2~N까지 모두 세워져 있을 때, 빌딩을 추가하는 방법의 수 N개

고층 빌딩

<https://www.acmicpc.net/problem/1328>

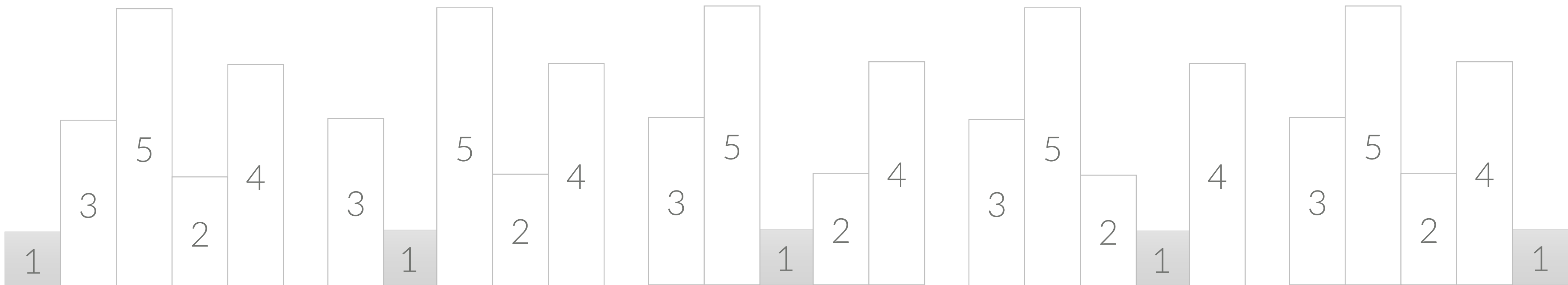
- 2~5까지 빌딩이 모두 있을 때, 높이가 1인 빌딩을 추가하는 방법
- 빌딩은 3, 5, 2, 4로 세워져 있다고 가정
- 왼쪽에서 2개, 오른쪽에서 2개 보임



고층 빌딩

<https://www.acmicpc.net/problem/1328>

- 2~5까지 빌딩이 모두 있을 때, 높이가 1인 빌딩을 추가하는 방법
- 빌딩은 3, 5, 2, 4로 세워져 있다고 가정
- 왼쪽에서 2개, 오른쪽에서 2개 보임

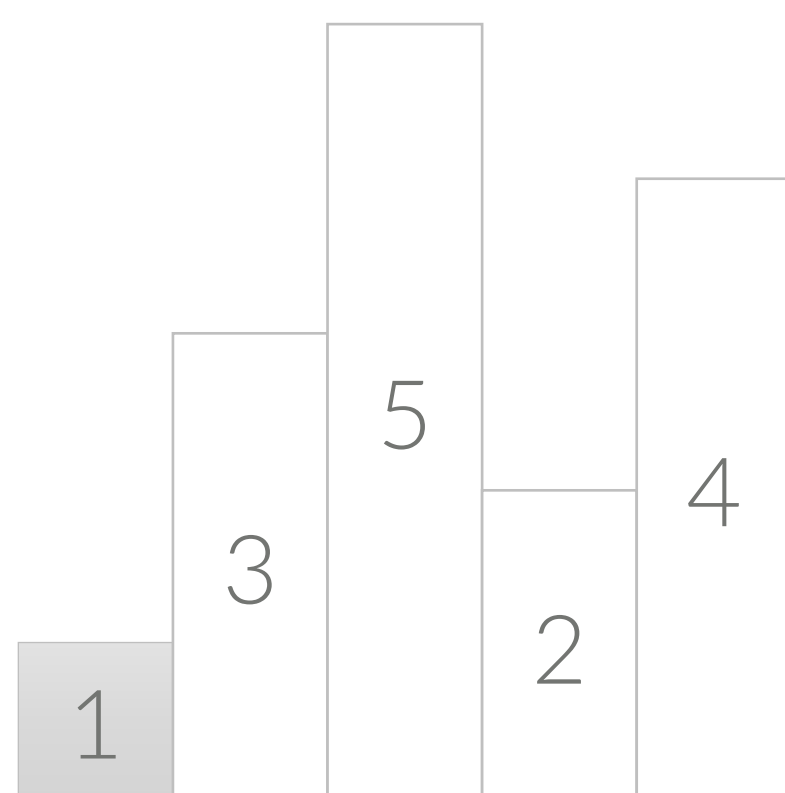


고층 빌딩

49

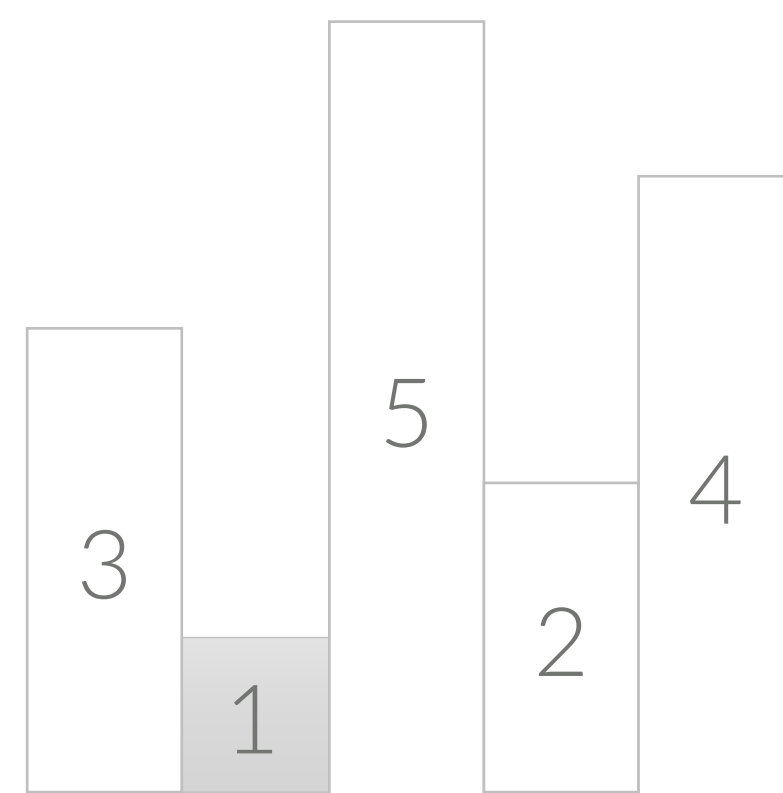
<https://www.acmicpc.net/problem/1328>

- 2~5까지 빌딩이 모두 있을 때, 높이가 1인 빌딩을 추가하는 방법
- 빌딩은 3, 5, 2, 4로 세워져 있다고 가정
- 왼쪽에서 2개, 오른쪽에서 2개 보임
- $N=4, L=2, R=2$



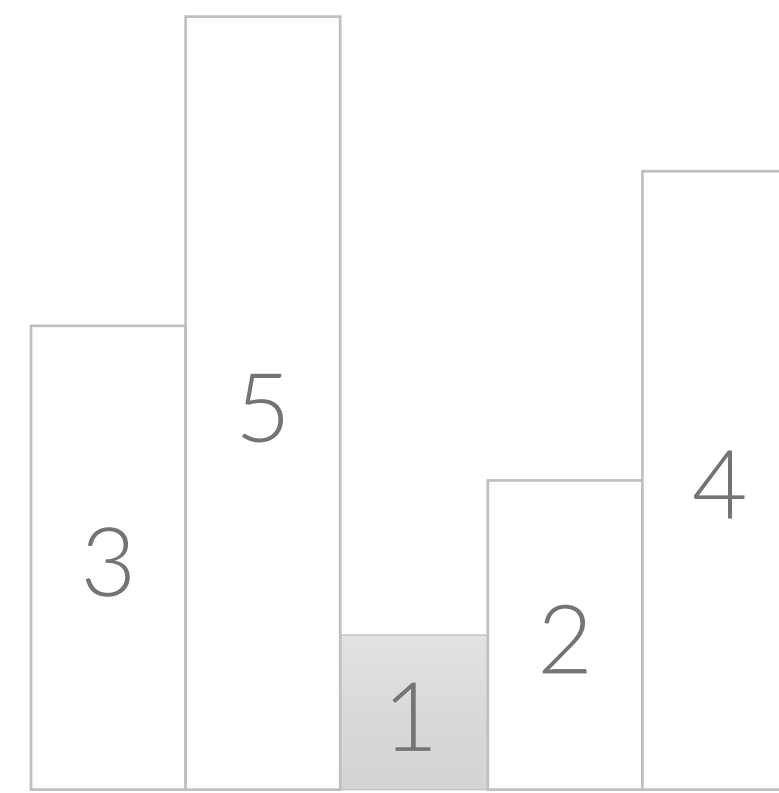
$N=5, L=3, R=2$

$N+=1, L+=1$



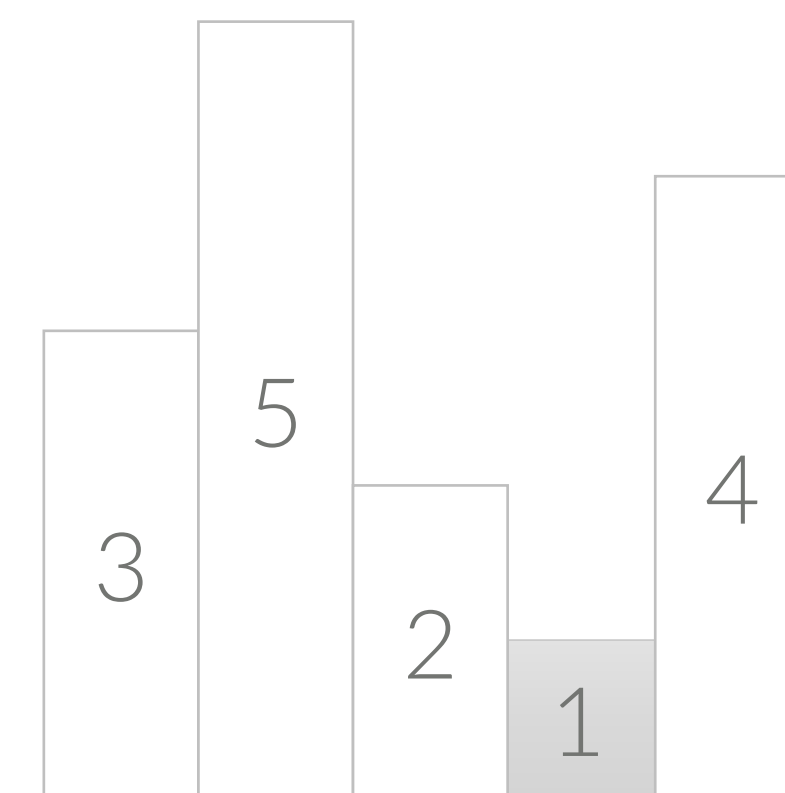
$N=5, L=2, R=2$

$N+=1$



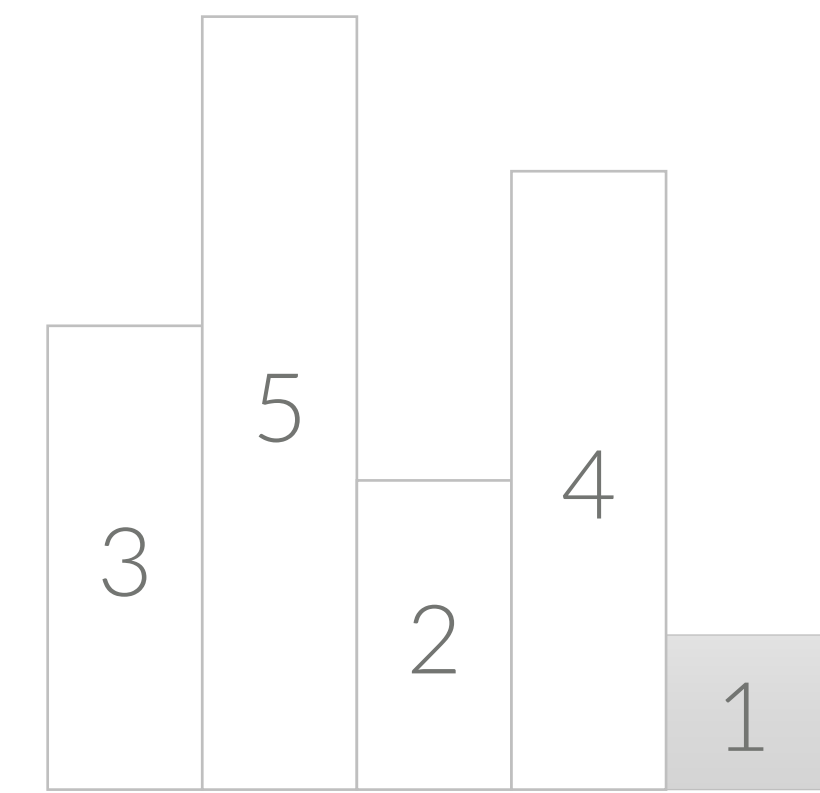
$N=5, L=2, R=2$

$N+=1$



$N=5, L=2, R=2$

$N+=1$



$N=5, L=2, R=3$

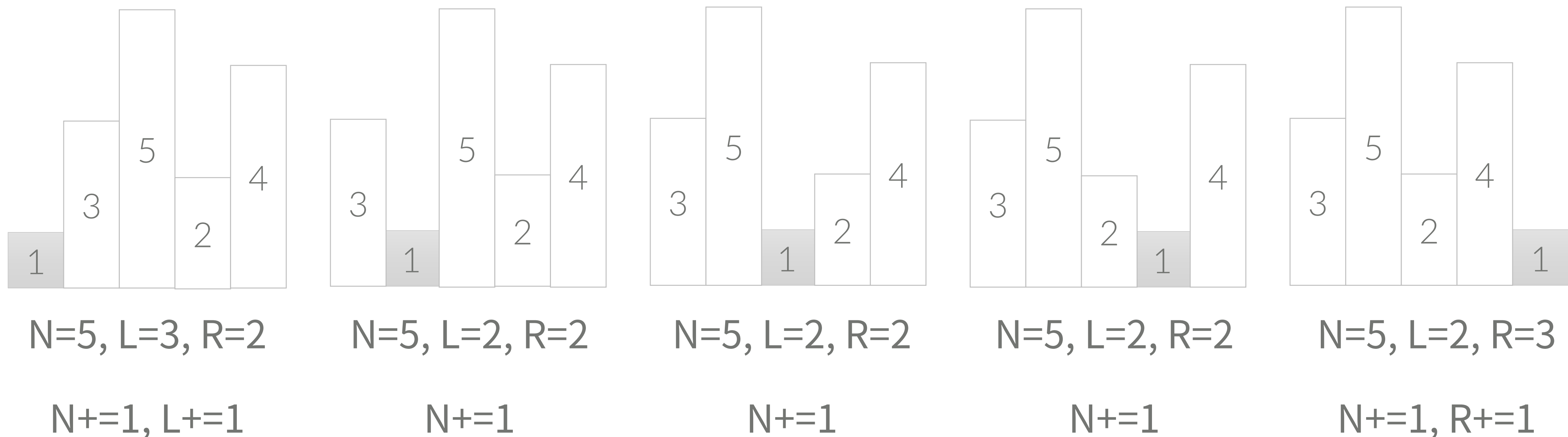
$N+=1, R+=1$

고층 빌딩

50

<https://www.acmicpc.net/problem/1328>

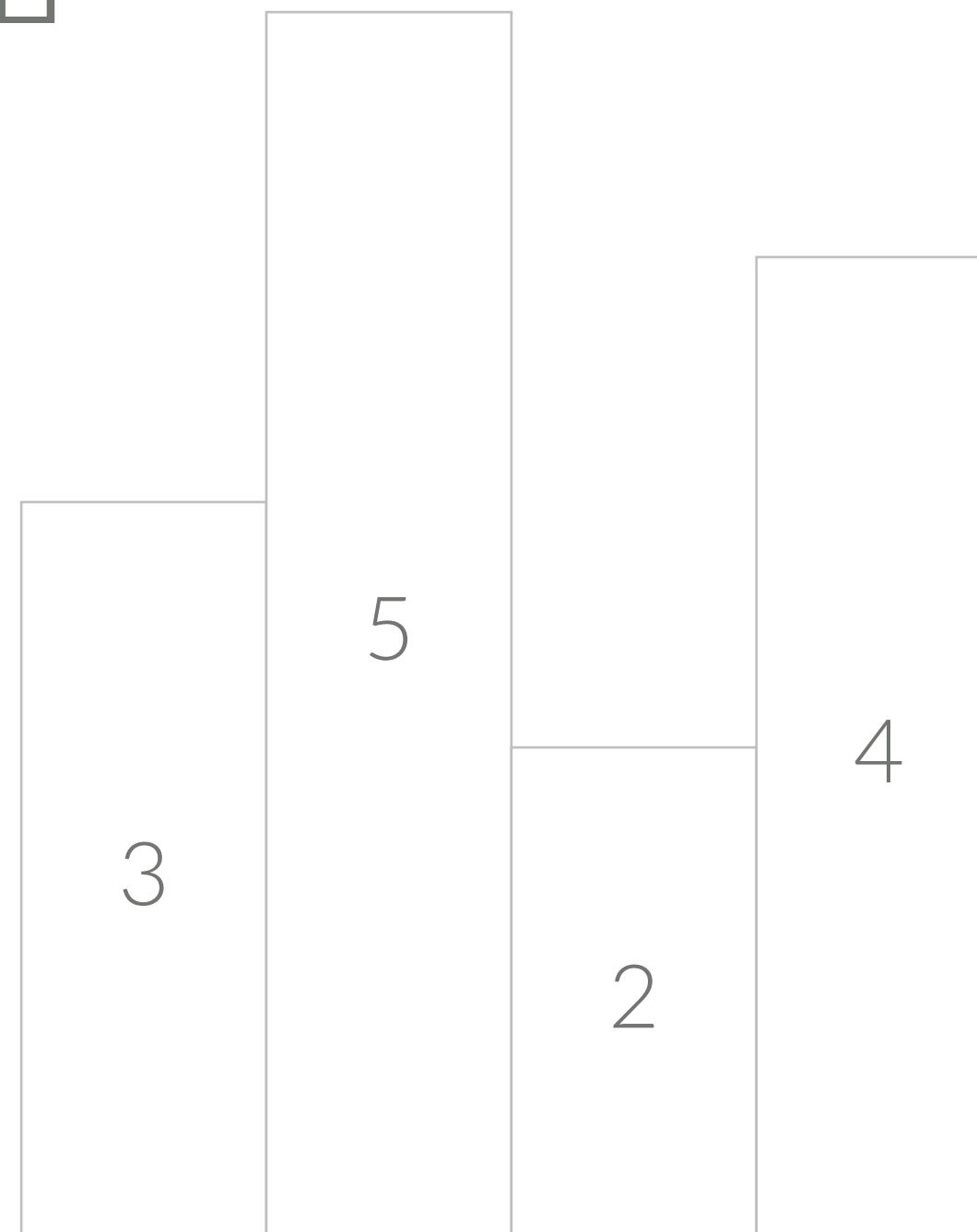
- 가운데 끼워넣는 경우는 L과 R이 변하지 않는다는 사실을 알 수 있다
- 가장 앞에 넣는 경우는 왼쪽에서 보이는 것이 하나 증가한다
- 가장 뒤는 오른쪽에서 보이는 것이 하나 증가한다



고층 빌딩

<https://www.acmicpc.net/problem/1328>

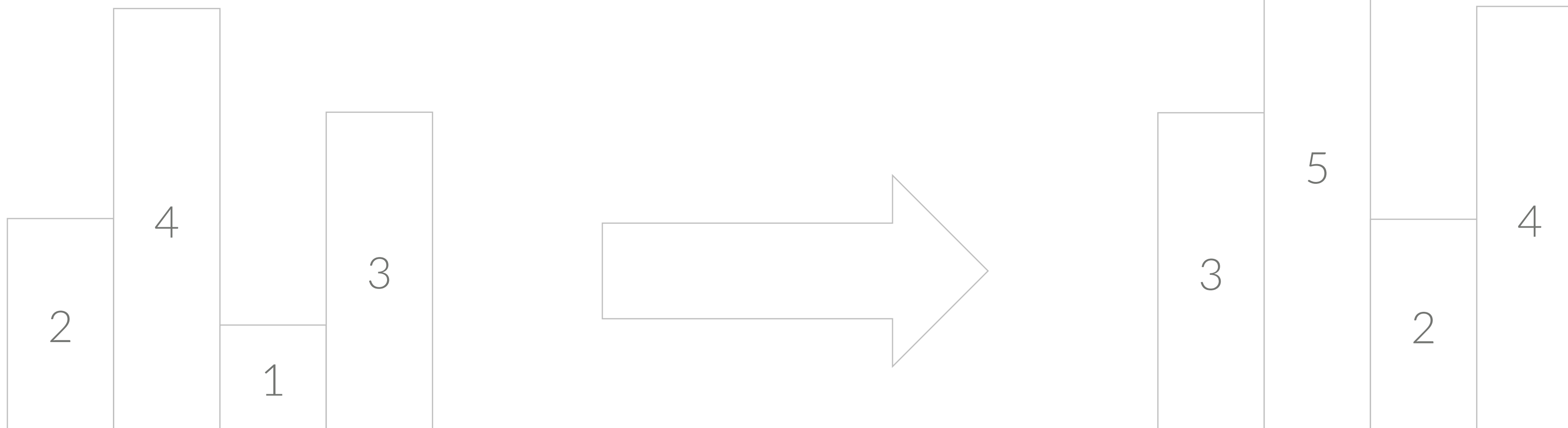
- 2~5까지 빌딩이 모두 있을 때, 높이가 1인 빌딩을 추가하는 방법
- 빌딩은 3, 5, 2, 4로 세워져 있다고 가정
- 왼쪽에서 2개, 오른쪽에서 2개 보임
- 이건 사실



고층 빌딩

<https://www.acmicpc.net/problem/1328>

- 2~5까지 빌딩이 모두 있을 때, 높이가 1인 빌딩을 추가하는 방법
- 빌딩은 3, 5, 2, 4로 세워져 있다고 가정
- 1~4까지 빌딩이 있고, 빌딩이 2, 4, 1, 3으로 세워져 있는 경우와 같다
- 이 경우에 모든 빌딩에 높이를 1씩 증가시키면, 같은 경우가 된다.



고층 빌딩

<https://www.acmicpc.net/problem/1328>

- $D[N][L][R]$ = 빌딩 N개, 왼쪽에서 L개 보임, 오른쪽에서 R개 보이는 빌딩 배치의 개수
- 가장 왼쪽에 빌딩 1을 추가하는 경우
 - L이 하나 증가해야 하기 때문에
 - $D[N+1][L+1][R] += D[N][L][R]$
- 가장 오른쪽에 빌딩 1을 추가하는 경우
 - R이 하나 증가해야 하기 때문에
 - $D[N+1][L][R+1] += D[N][L][R]$
- 사이에 빌딩 1을 추가하는 경우
 - $D[N+1][L][R] += D[N][L][R] * (N-1)$
 - 추가할 수 있는 경우가 N-1개 존재

고층 빌딩

<https://www.acmicpc.net/problem/1328>

- $D[N][L][R]$ = 빌딩 N개, 왼쪽에서 L개 보임, 오른쪽에서 R개 보이는 빌딩 배치의 개수
- $D[N+1][L+1][R] += D[N][L][R]$
- $D[N+1][L][R+1] += D[N][L][R]$
- $D[N+1][L][R] += D[N][L][R] * (N-1)$

고층 빌딩

55

<https://www.acmicpc.net/problem/1328>

- 소스: <http://codeplus.codes/1470c804808f42f7a4cee82a47d8dd53>

고층 빌딩

<https://www.acmicpc.net/problem/1328>

- $D[N][L][R]$ = 빌딩 N개, 왼쪽에서 L개 보임, 오른쪽에서 R개 보이는 빌딩 배치의 개수
- 가장 왼쪽에 빌딩 1이 있는 경우
 - L이 하나 증가해야 하기 때문에
 - $D[N-1][L-1][R]$
- 가장 오른쪽에 빌딩 1이 있는 경우
 - R이 하나 증가해야 하기 때문에
 - $D[N-1][L][R-1]$
- 사이에 빌딩 1이 있는 경우
 - $D[N-1][L][R] * (N-2)$
 - 추가할 수 있는 경우가 N-2개 존재

고층 빌딩

<https://www.acmicpc.net/problem/1328>

- $D[N][L][R]$ = 빌딩 N개, 왼쪽에서 L개 보임, 오른쪽에서 R개 보이는 빌딩 배치의 개수
- $D[N][L][R] = D[N-1][L-1][R] + D[N-1][L][R-1] + D[N-1][L][R] * (N-2)$

고층 빌딩

<https://www.acmicpc.net/problem/1328>

```
d[1][1][1] = 1LL;
for (int i=2; i<=n; i++) {
    for (int j=1; j<=l; j++) {
        for (int k=1; k<=r; k++) {
            d[i][j][k] = d[i-1][j-1][k] + d[i-1][j][k-1] + d[i-
1][j][k] * (i-2);
            d[i][j][k] %= mod;
        }
    }
}
```

고층 빌딩

<https://www.acmicpc.net/problem/1328>

- 소스: <http://codeplus.codes/bcd1fc11cbbf42748601eb9cb933b605>

홍준이의 친위대

60

<https://www.acmicpc.net/problem/3948>

- N명의 사람을 한 줄로 세우는데, 양 옆의 사람이 자신보다 크거나 작게 세우는 경우의 수
- 예를 들어, 4명이 있고, 키가 1, 2, 3, 4라면
- 1324, 2143, 3142, 2314, 3412, 4231, 4132, 2413, 3241, 1423
- 와 같은 10가지 배치가 가능하다.

홍준이의 친위대

<https://www.acmicpc.net/problem/3948>

- $Tall[n]$ = 1번 사람 > 2번 사람으로 줄을 서는 경우의 수
- $Short[n]$ = 1번 사람 < 2번 사람으로 줄을 서는 경우의 수

홍준이의 친위대

<https://www.acmicpc.net/problem/3948>

- $Tall[n]$ = 1번 사람 > 2번 사람으로 줄을 서는 경우의 수
- $Short[n]$ = 1번 사람 < 2번 사람으로 줄을 서는 경우의 수
- $Tall[n]$ 과 $Short[n]$ 과의 관계는?

홍준이의 친위대

<https://www.acmicpc.net/problem/3948>

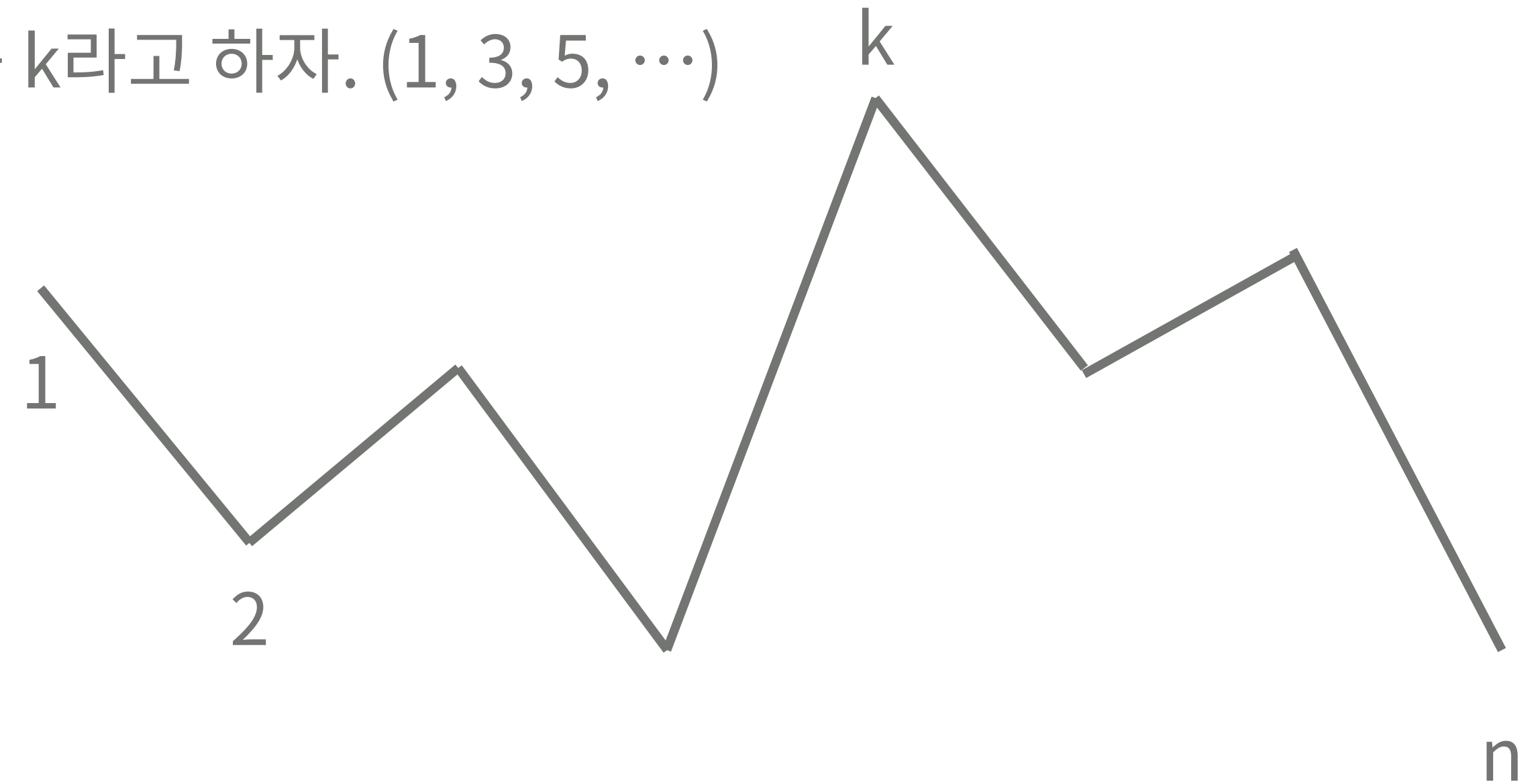
- $Tall[n]$ = 1번 사람 > 2번 사람으로 줄을 서는 경우의 수
- $Short[n]$ = 1번 사람 < 2번 사람으로 줄을 서는 경우의 수
- $Tall[n]$ 과 $Short[n]$ 과의 관계는?
- $Tall[n] = Short[n]$
- 모든 키가 k 인 사람을 $n-k+1$ 로 바꾸면 되기 때문
- 즉, $n = 1$ 인 경우를 제외하면, $Tall[n] + Short[n] = 2 \times Tall[n]$ s

홍준이의 친위대

64

<https://www.acmicpc.net/problem/3948>

- $Tall[n]$ = 1번 사람 > 2번 사람으로 줄을 서는 경우의 수
- $Tall[0] = 1$ 이면, $Tall[1] = Tall[2] = 1$
- $Tall[n]$ 을 구하기 위해, 가장 키가 큰 사람의 위치를 k 라고 하자. (1, 3, 5, ...)

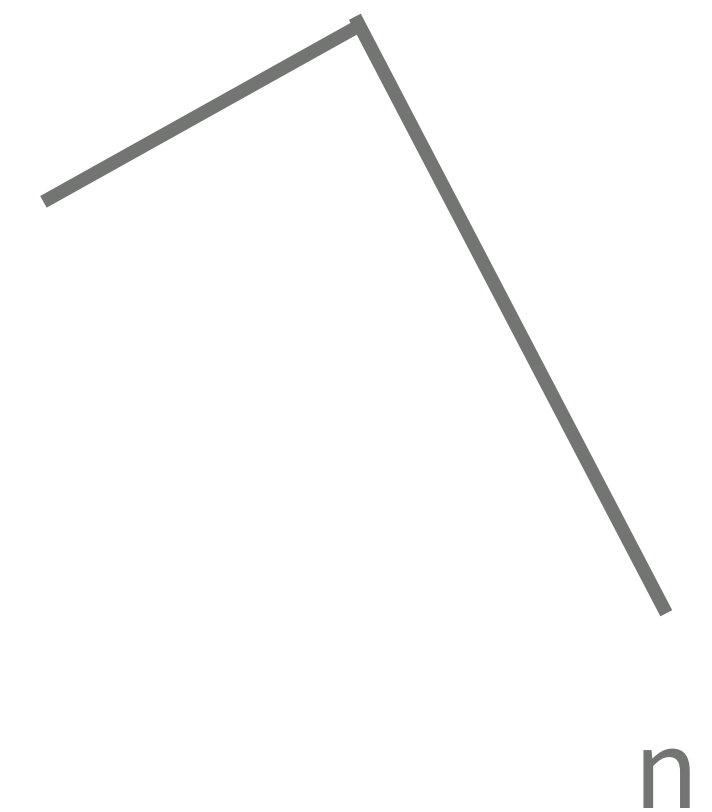
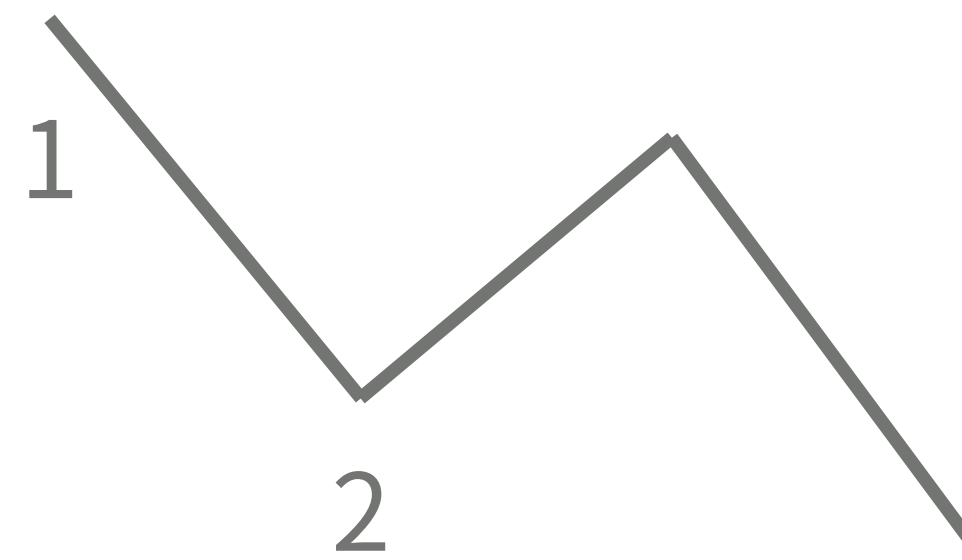


홍준이의 친위대

65

<https://www.acmicpc.net/problem/3948>

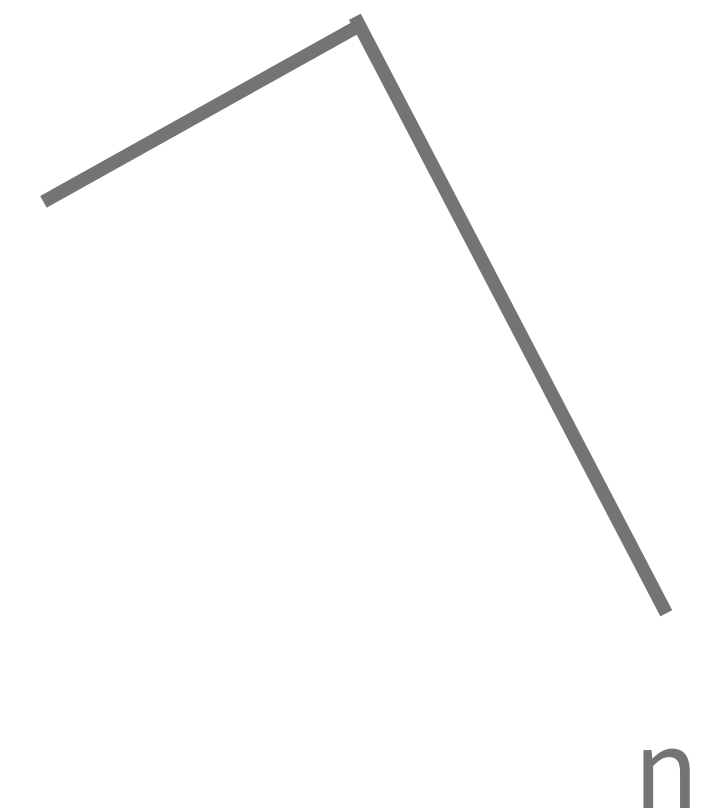
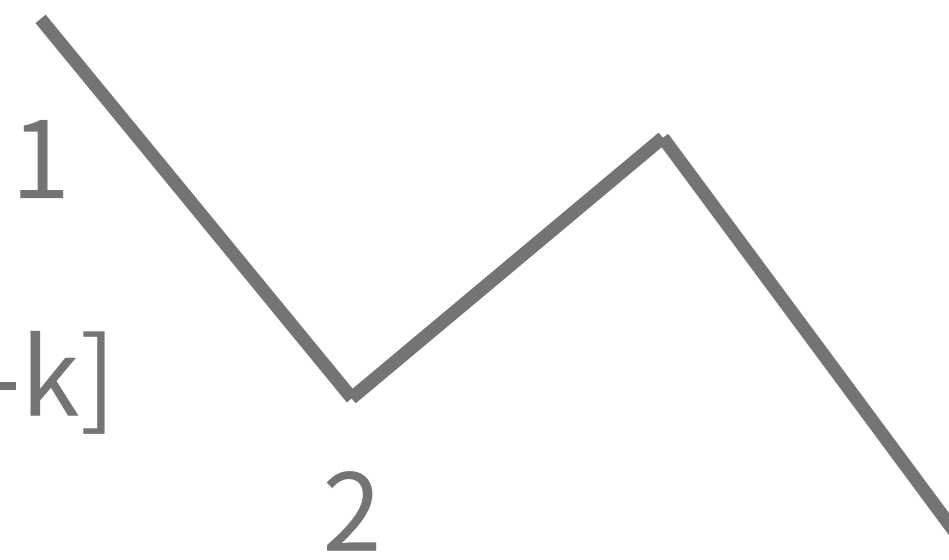
- $Tall[n]$ = 1번 사람 > 2번 사람으로 줄을 서는 경우의 수
- $Tall[0] = 1$ 이면, $Tall[1] = Tall[2] = 1$
- $Tall[n]$ 을 구하기 위해, 가장 키가 큰 사람의 위치를 k 라고 하자. (1, 3, 5, ...)
- k 를 기준으로 왼쪽과 오른쪽으로 나누어진다.



홍준이의 친위대

<https://www.acmicpc.net/problem/3948>

- $Tall[n] = 1$ 번 사람 > 2번 사람으로 줄을 서는 경우의 수
- $Tall[0] = 1$ 이면, $Tall[1] = Tall[2] = 1$
- $Tall[n]$ 을 구하기 위해, 가장 키가 큰 사람의 위치를 k 라고 하자. (1, 3, 5, ...)
- k 를 기준으로 왼쪽과 오른쪽으로 나누어진다.
- 왼쪽에 $k-1$ 명을 세워야 하는 경우의 수: $Tall[k-1]$
- 오른쪽에 $n-k$ 명을 세워야 하는 경우의 수: $Short[n-k]$
- $Comb(n-1, k-1) * Tall[k-1] * Short[n-k]$
- $= Comb(n-1, k-1) * Tall[k-1] * Tall[n-k]$



홍준이의 친위대

<https://www.acmicpc.net/problem/3948>

- 소스: <http://codeplus.codes/5d579051dc454fd9802ac8510c56f312>

좋아하는 배열

<https://www.acmicpc.net/problem/12911>

- 길이가 N이고
- 배열은 1부터 K까지의 수로 채울 수 있고
- 연속한 수가 A, B일 때, $A \leq B$ 또는 $A \% B \neq 0$ 을 만족해야 한다.

좋아하는 배열

<https://www.acmicpc.net/problem/12911>

- $D[N][A]$ = 길이가 N 이고, 첫 수가 A 인 좋아하는 배열의 개수
- 다음 수: B
- 1. $A \leq B$ 인 조건
 - $D[N][A] += D[N-1][B]$
- 2. $A > B$ 이면서 $A \% B \neq 0$ 인 조건
 - $D[N][A] += D[N-1][B]$ (B 는 약수가 아님)

좋아하는 배열

70

<https://www.acmicpc.net/problem/12911>

- 소스: <http://codeplus.codes/be297883bbb744f39db3449913f88534>
- $O(NK^2)$ 이라 시간 초과를 받게 된다

좋아하는 배열

<https://www.acmicpc.net/problem/12911>

- $D[N][A]$ = 길이가 N이고, 첫 수가 A인 좋아하는 배열의 개수
- 다음 수: B
- 1. $A \leq B$ 인 조건
 - $D[N][A] += D[N-1][B]$
- 2. $A > B$ 이면서 $A \% B \neq 0$ 인 조건
 - $D[N][A] += D[N-1][B]$ (B는 약수가 아님)
- $D[N][A]$ 에는 $D[N-1]$ 에 들어있는 전체 값에서 $D[N-1][B]$ ($A > B \ \&\& \ A \% B == 0$) 인 것을 빼면 된다

좋아하는 배열

72

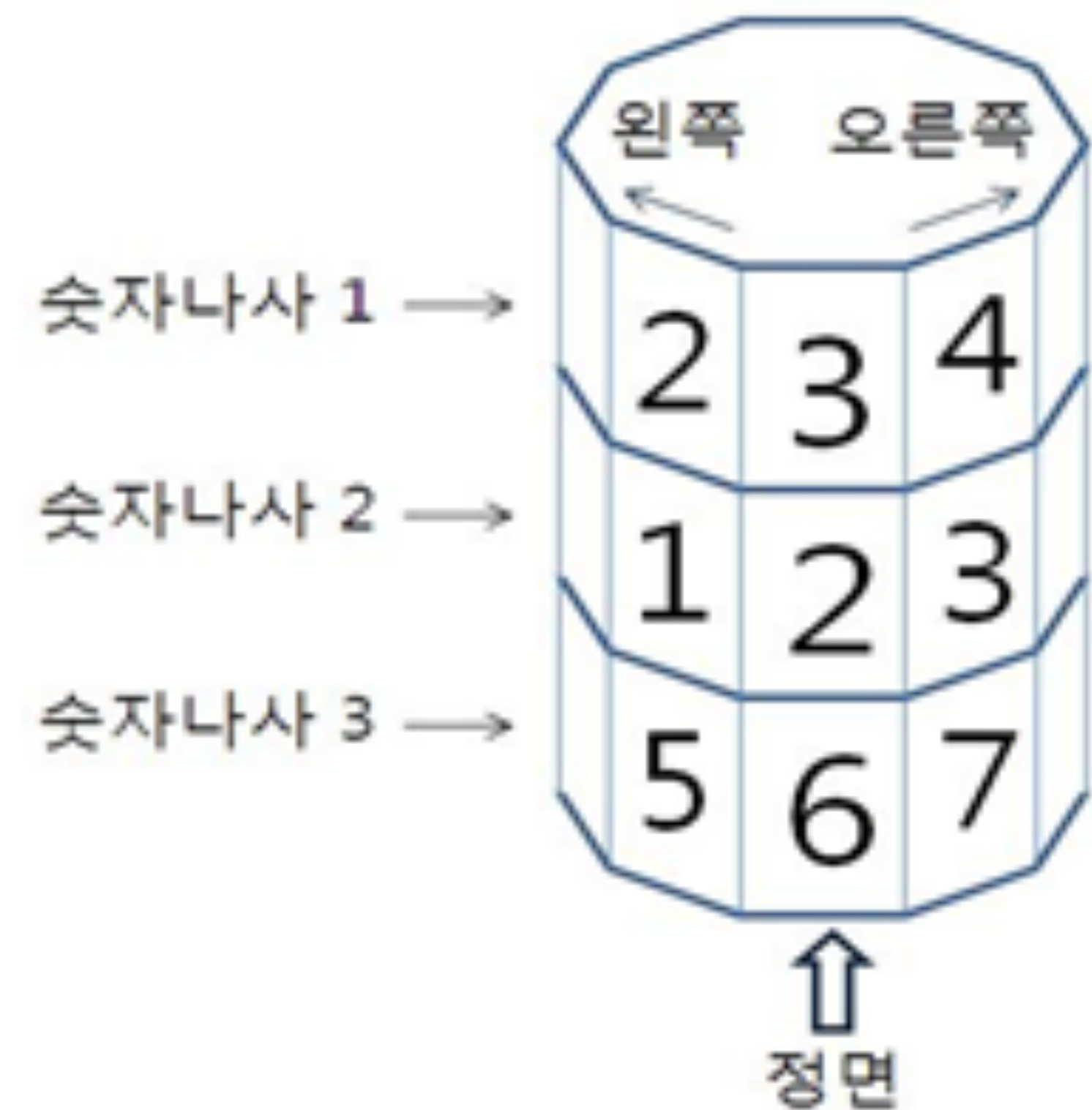
<https://www.acmicpc.net/problem/12911>

- 소스: <http://codeplus.codes/f522d20c9343423eb41bc6155d116969>

방법을 출력하지 않는 숫자 맞추기

<https://www.acmicpc.net/problem/13392>

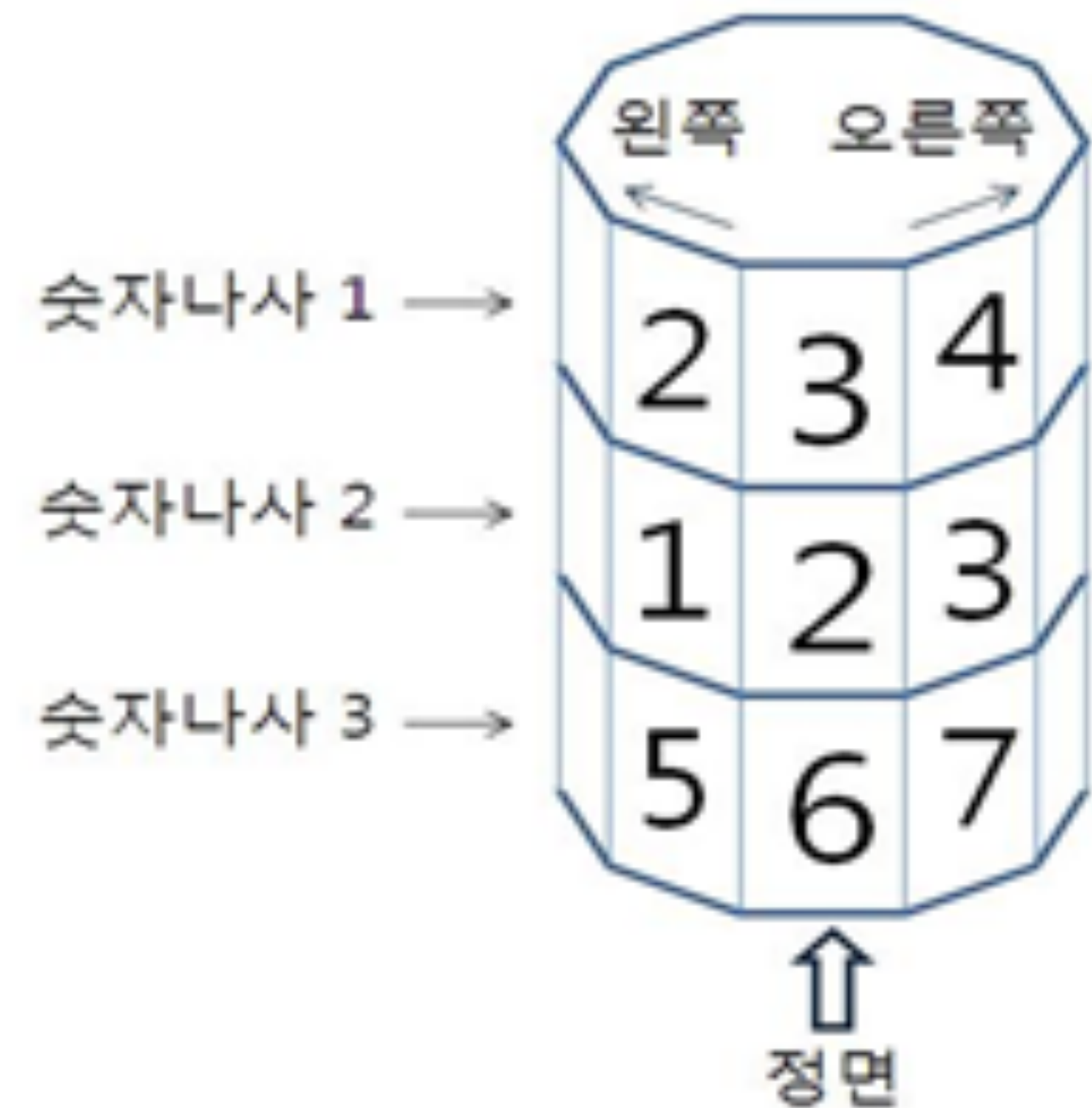
- 왼쪽으로 돌리면 아래에 있는 나사가 같이 돈다
- 오른쪽으로 돌리면 하나만 돌아간다



방법을 출력하지 않는 숫자 맞추기

<https://www.acmicpc.net/problem/13392>

- $D[i][j]$ = i 번 나사가 현재 왼쪽으로 j 번 돈 상황일 때 최소 횟수



방법을 출력하지 않는 숫자 맞추기

<https://www.acmicpc.net/problem/13392>

- 왼쪽으로 돌리는 경우

```
int cost_l = (b[index] + 10 - cur) % 10;  
int left = go(index+1, (turn + cost_l) % 10) + cost_l;
```

- 오른쪽으로 돌리는 경우

```
int cost_r = (cur + 10 - b[index]) % 10;  
int right = go(index+1, turn) + cost_r;
```

방법을 출력하지 않는 숫자 맞추기

76

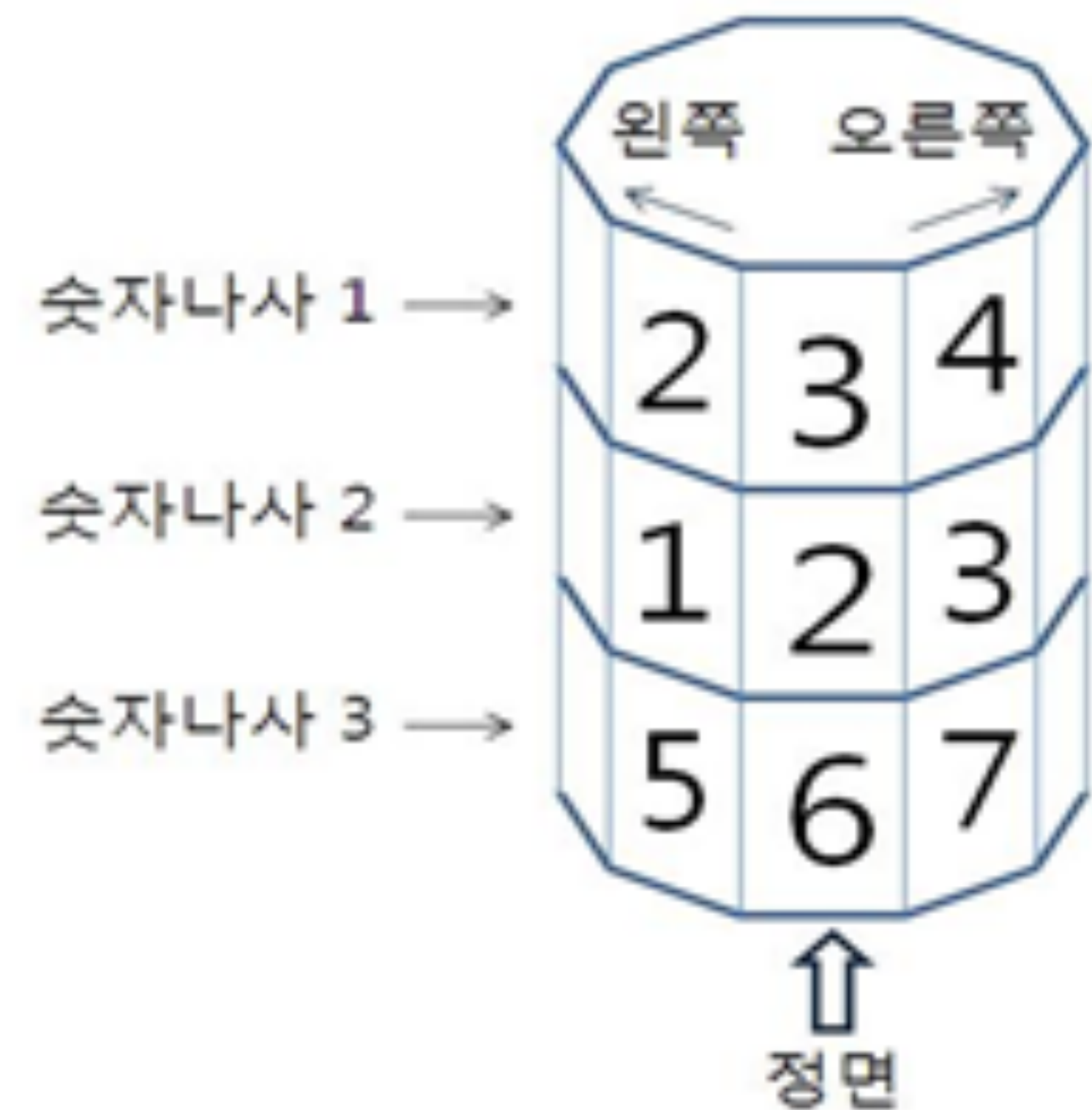
<https://www.acmicpc.net/problem/13392>

- 소스: <http://codeplus.codes/e74946799c6e4977be977224cff134b2>

숫자 맞추기

<https://www.acmicpc.net/problem/2494>

- 왼쪽으로 돌리면 아래에 있는 나사가 같이 돈다
- 오른쪽으로 돌리면 하나만 돌아간다



숫자 맞추기

78

<https://www.acmicpc.net/problem/2494>

- 소스: <http://codeplus.codes/3be616e7a2074c76afc5536e0f2c74ad>

자물쇠

<https://www.acmicpc.net/problem/1514>

- 동그란 디스크 N개로 이루어진 자물쇠가 있다. $N \leq 100$
- 현재 자물쇠의 상태가 주어졌을 때, 몇 번만에 자물쇠를 열 수 있는지 구하는 문제
- 한 번 자물쇠를 돌릴 때, 최대 세 칸을 시계 방향 또는 반시계 방향으로 돌릴 수 있다.
- 또, 최대 세 개의 인접한 디스크를 한 번에 돌릴 수 있다.

자물쇠

<https://www.acmicpc.net/problem/1514>

- 각각의 디스크마다 그 디스크를 가장 왼쪽으로 하면서 연속으로 한 칸, 두 칸, 세 칸을 몇 바퀴나 돌릴지 결정해야 한다.
- 연속 세 칸 돌리기에 의해서 영향을 받는 칸의 개수는 최대 2개이다.
- $D[i][j][k]$ = i 번째 디스크를 돌릴 것이고, 현재 칸은 j 만큼 돌아가 있고, 다음 칸은 k 만큼 돌아가 있다.

- N = 5인 경우 돌릴 수 있는 방법

[illegible]

자물쇠

<https://www.acmicpc.net/problem/1514>

- $N = 5$ 인 경우 돌릴 수 있는 방법

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

자물쇠

<https://www.acmicpc.net/problem/1514>

- $D[i][j][k]$ = i 번째 디스크를 돌릴 것이고, 현재 칸은 j 만큼 돌아가 있고, 다음 칸은 k 만큼 돌아가 있다.
- 현재 i 번째 디스크가 가리키고 있는 값: $(a[i] + j) \% 10$
- i 번째 디스크를 기준으로 연속 세 칸을 three만큼
- 연속 두 칸을 two만큼 돌린다.

자물쇠

<https://www.acmicpc.net/problem/1514>

- $D[i][j][k]$ = i 번째 디스크를 돌릴 것이고, 현재 칸은 j 만큼 돌아가 있고, 다음 칸은 k 만큼 돌아가 있다.
- 현재 i 번째 디스크가 가리키고 있는 값: $(a[i] + j) \% 10$
- i 번째 디스크를 기준으로 연속 세 칸을 three만큼
- 연속 두 칸을 two만큼 돌린다.
- 이제 i 번째 디스크는 $(a[i] + j + \text{two} + \text{three}) \% 10$ 을 가리키고 있다.
- 한 칸을 얼마나 돌려야하는지는 수식으로 구할 수 있다.
- 총 돌리는 비용: $\text{cost}[\text{one}] + \text{cost}[\text{two}] + \text{cost}[\text{three}]$

자물쇠

<https://www.acmicpc.net/problem/1514>

- $D[i][j][k]$ = i 번째 디스크를 돌릴 것이고, 현재 칸은 j 만큼 돌아가 있고, 다음 칸은 k 만큼 돌아가 있다.
- 현재 i 번째 디스크가 가리키고 있는 값: $(a[i] + j) \% 10$
- i 번째 디스크를 기준으로 연속 세 칸을 three만큼
- 연속 두 칸을 two만큼 돌린다.
- 이제 i 번째 디스크는 $(a[i] + j + two + three) \% 10$ 을 가리키고 있다.
- 한 칸을 얼마나 돌려야하는지는 수식으로 구할 수 있다.
- 총 돌리는 비용: $cost[one] + cost[two] + cost[three]$
- 다음 상태: $D[i+1][(k+two+three)\%10][three]$

자물쇠

<https://www.acmicpc.net/problem/1514>

- D[1]에서 고려해야 하는 돌리는 방법

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

자물쇠

<https://www.acmicpc.net/problem/1514>

- D[2]에서 고려해야 하는 돌리는 방법

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

자물쇠

<https://www.acmicpc.net/problem/1514>

- $D[i][j][k]$ 의 상태

자물쇠:

i	i+1	i+2	i+3
---	-----	-----	-----

돌아간 횟수:

j	k		
---	---	--	--

자물쇠

<https://www.acmicpc.net/problem/1514>

- $D[i][j][k]$ 의 상태
- two와 three는 2중 for문을 사용
- one은 수식으로 구할 수 있다.

자물쇠:

i	i+1	i+2	i+3
---	-----	-----	-----

돌아간 횟수:

j	k		
---	---	--	--

디스크 두 개 동시에 돌림:

two	two		
-----	-----	--	--

디스크 세 개 동시에 돌림:

three	three	three	
-------	-------	-------	--

자물쇠

90

<https://www.acmicpc.net/problem/1514>

- $D[i][j][k]$ 의 다음 상태

- $D[i+1][(k+two+three)\%10][three]$

자물쇠:

i	i+1	i+2	i+3
---	-----	-----	-----

돌아간 횟수:

j	k		
---	---	--	--

디스크 한 개 돌림:

one			
-----	--	--	--

디스크 두 개 동시에 돌림:

two	two		
-----	-----	--	--

디스크 세 개 동시에 돌림:

three	three	three	
-------	-------	-------	--

자물쇠

<https://www.acmicpc.net/problem/1514>

- 소스: <http://codeplus.codes/a754e9befaad47259bac0f42b86321ff>