

다이나믹 프로그래밍 3

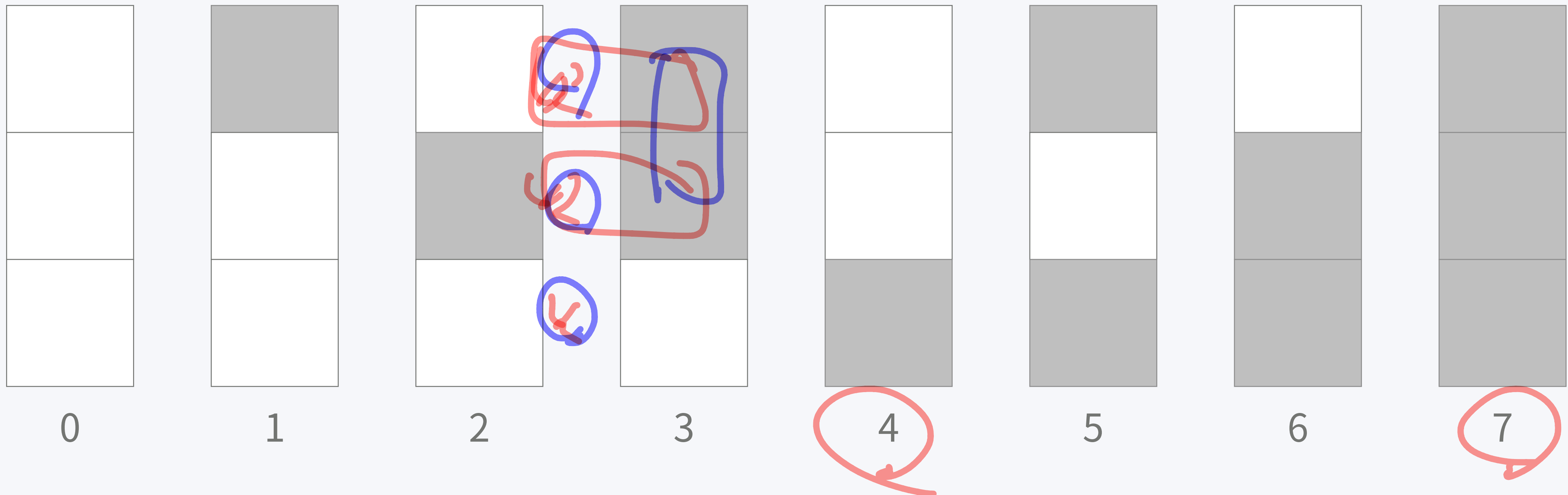
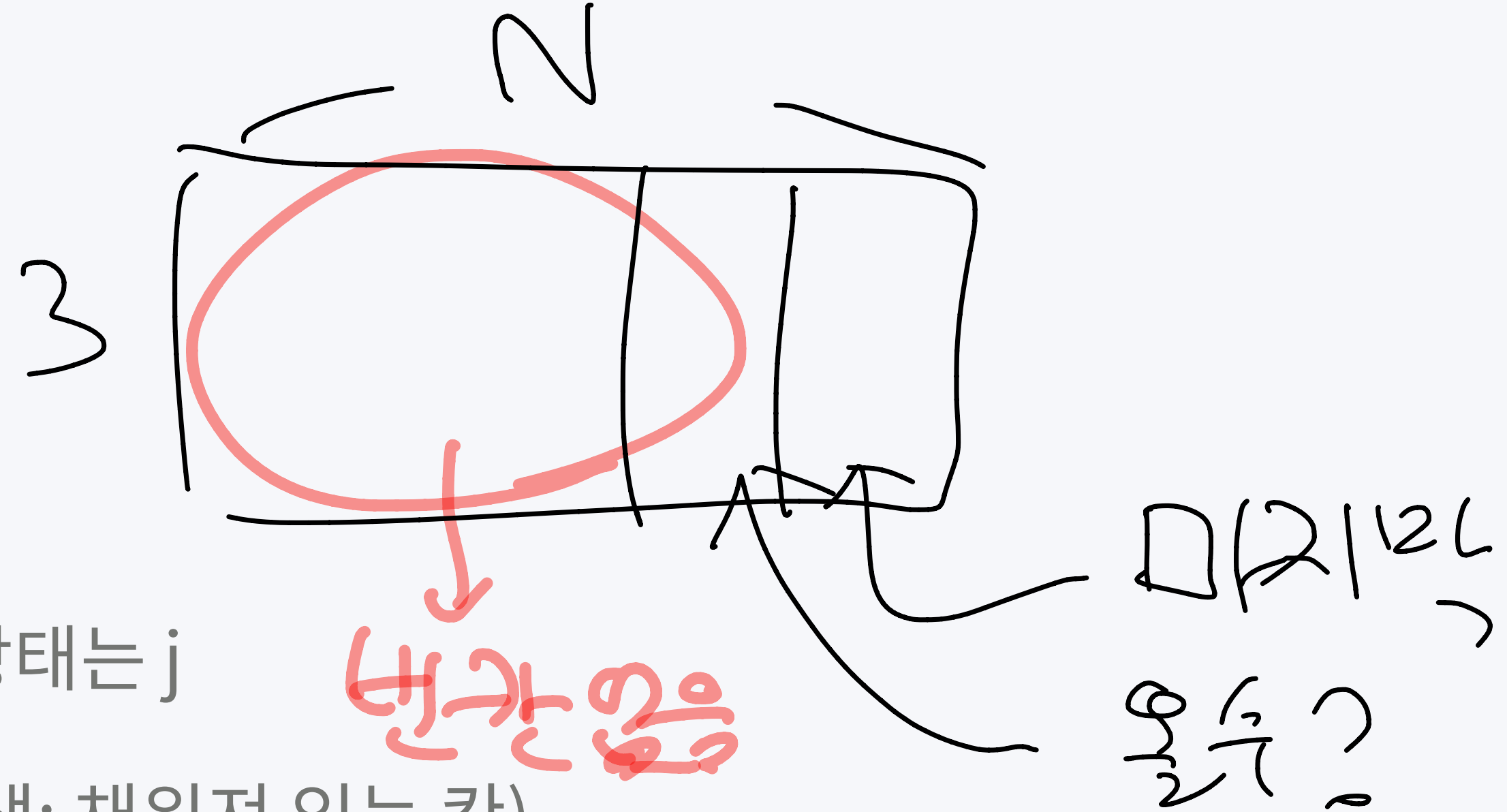
최백준 choi@startlink.io

상태 다이나믹

타일 채우기

<https://www.acmicpc.net/problem/2133>

- $3 \times N$ 을 1×2 , 2×1 로 채우는 방법의 수
- $D[i][j] = 3 \times i$ 를 채우는 방법의 수, i 열의 상태는 j
- 마지막에 올 수 있는 가능한 경우의 수 (회색: 채워져 있는 칸)

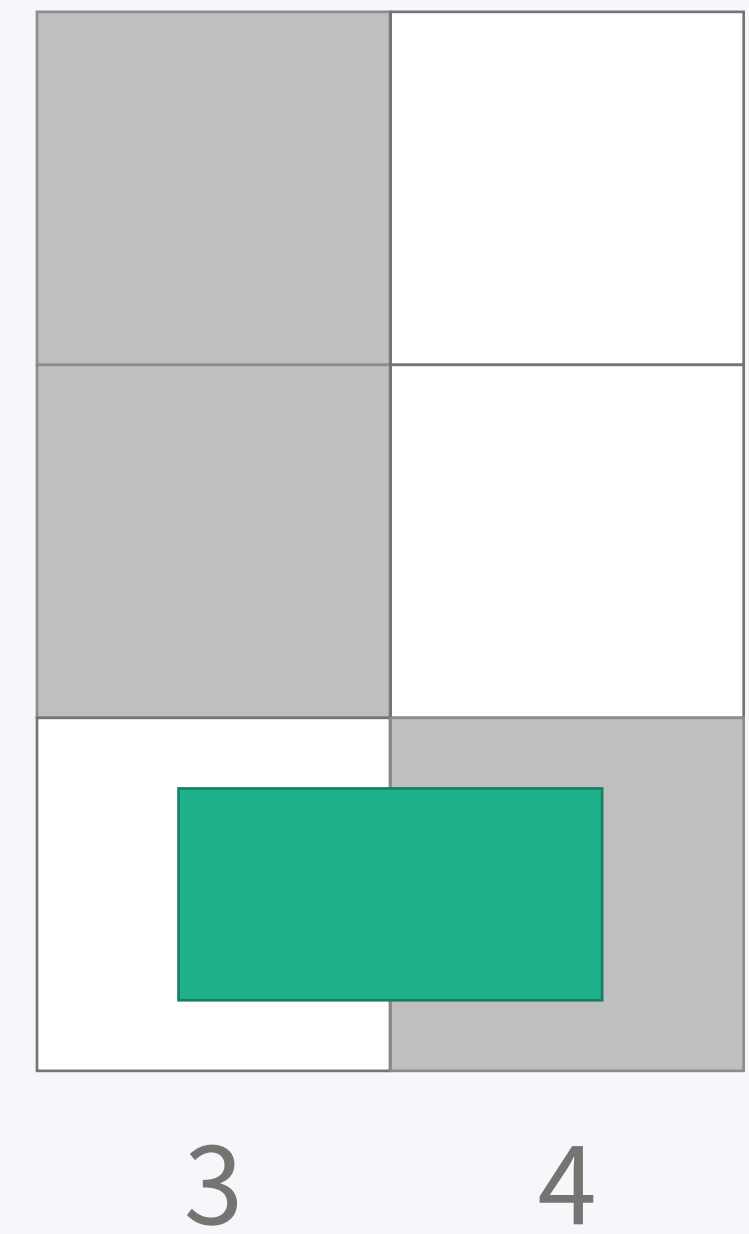
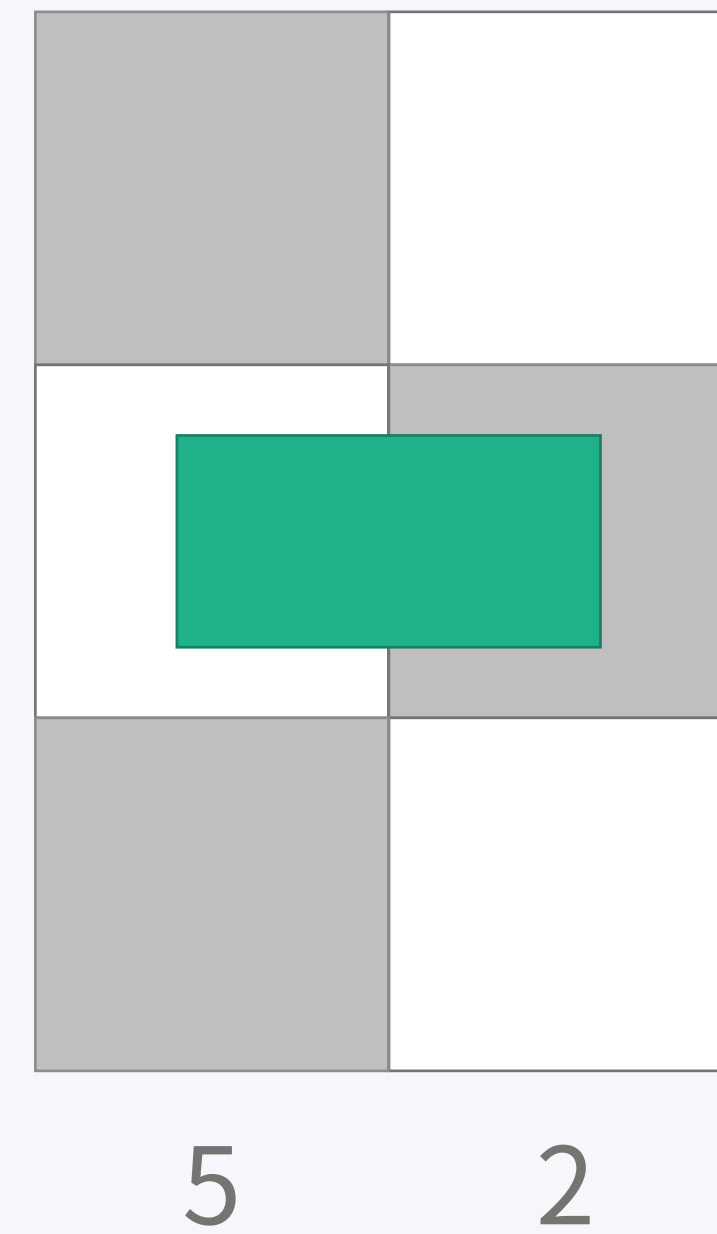
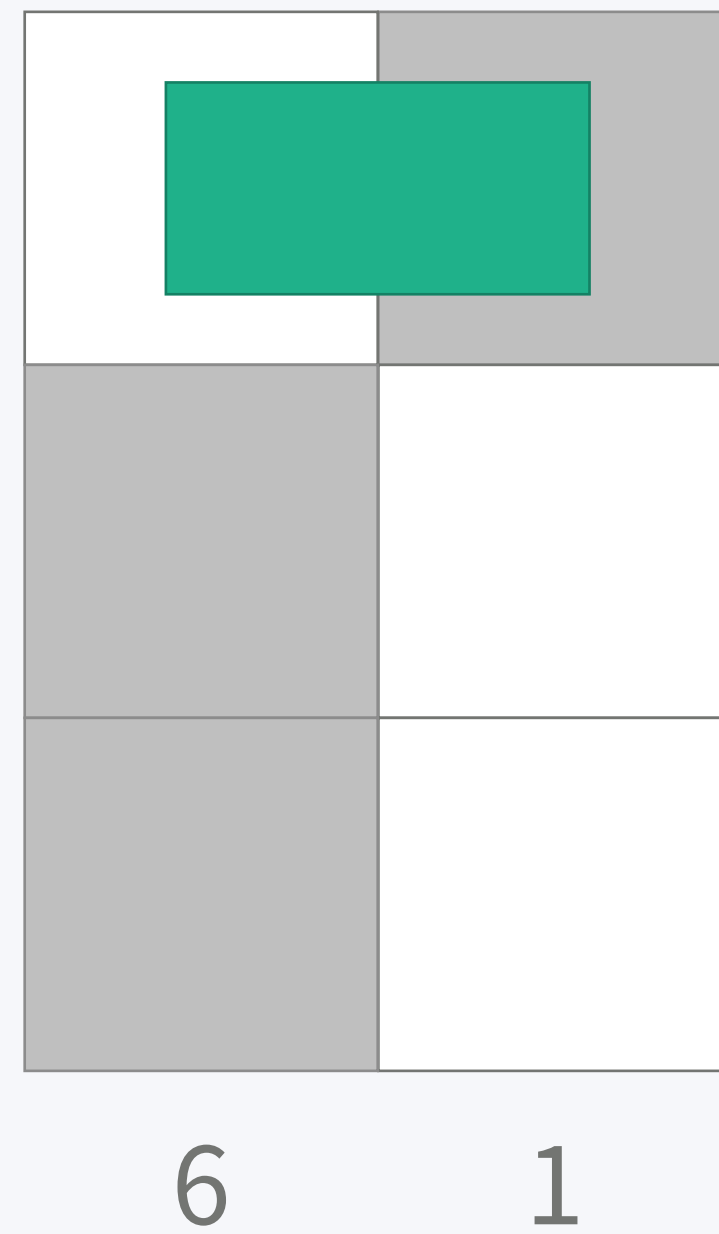
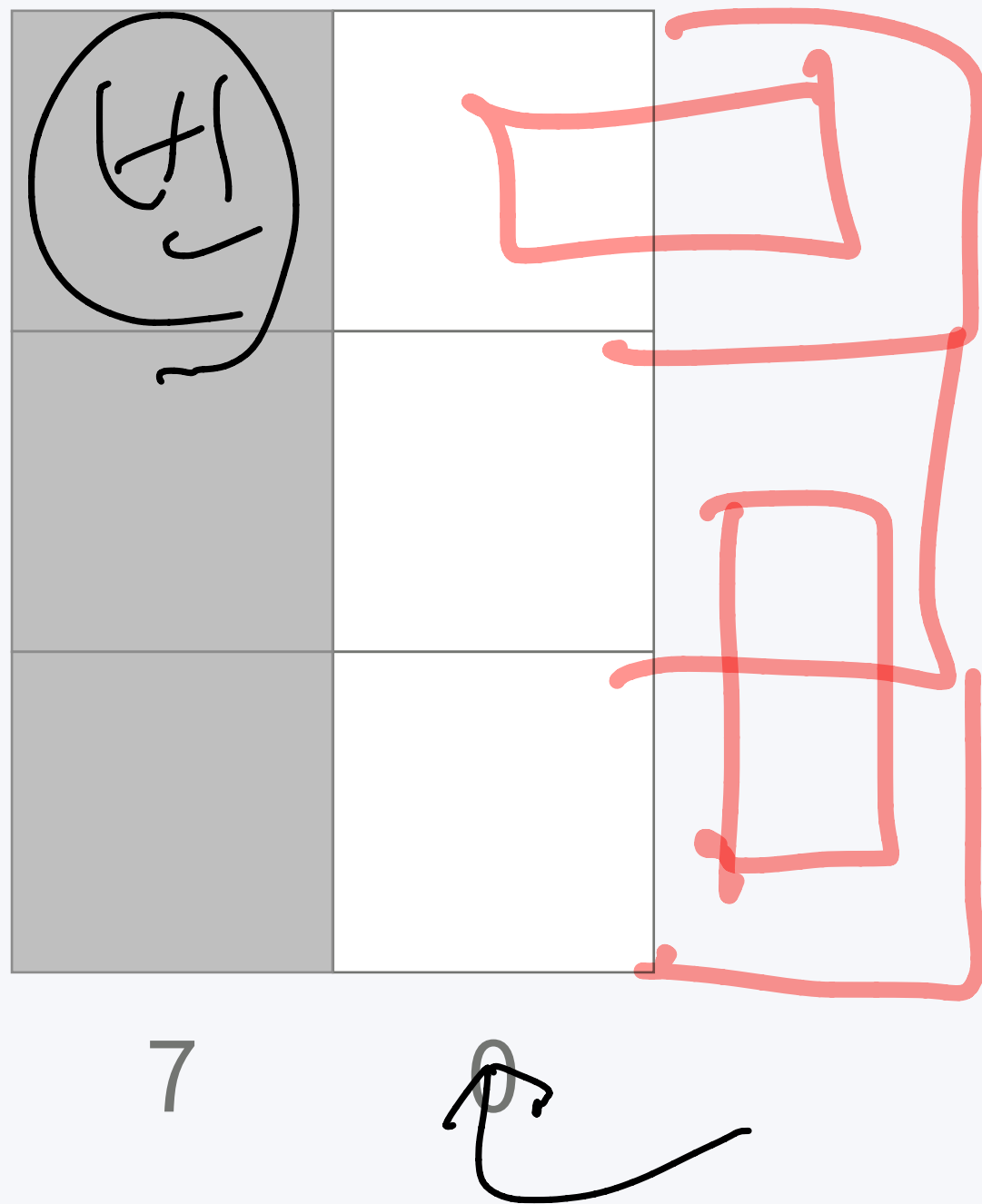


타일 채우기

<https://www.acmicpc.net/problem/2133>

- i 열을 채울 때, $i-1$ 에 빈 칸이 있으면 안된다

$N-1$ N

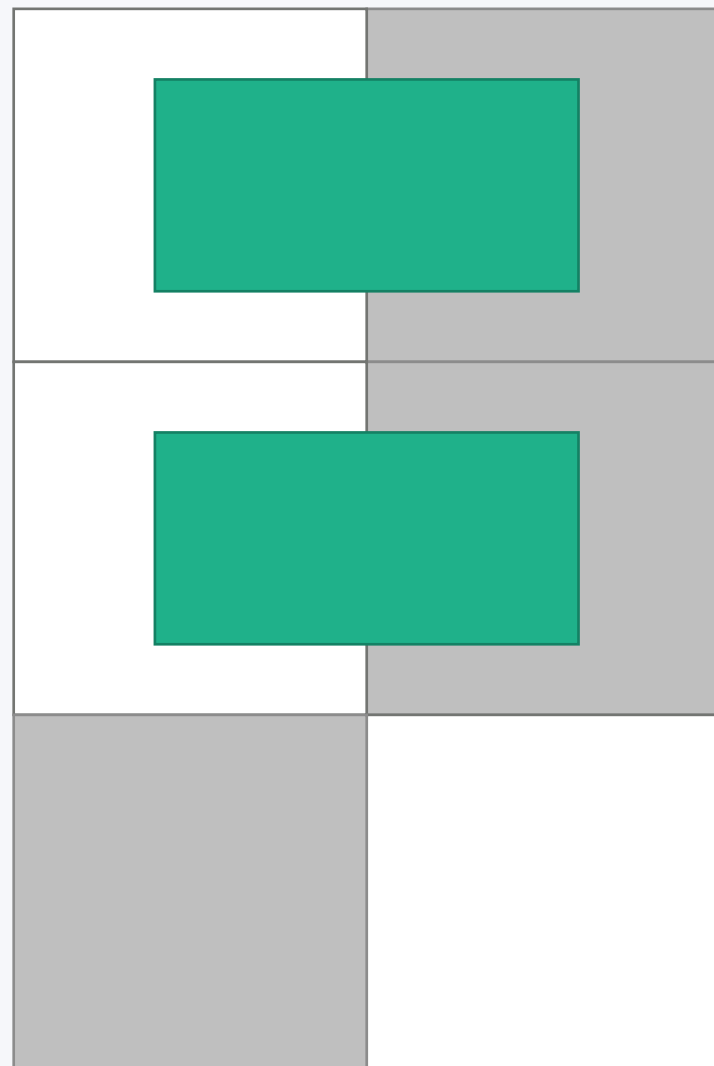


타일 채우기

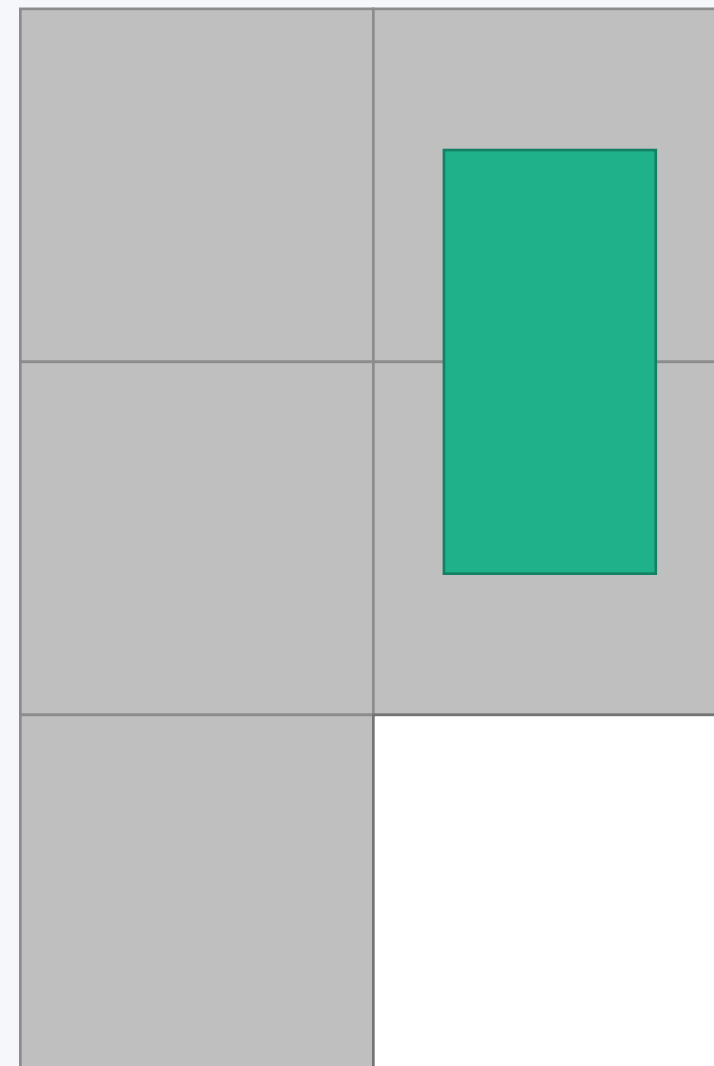
5

<https://www.acmicpc.net/problem/2133>

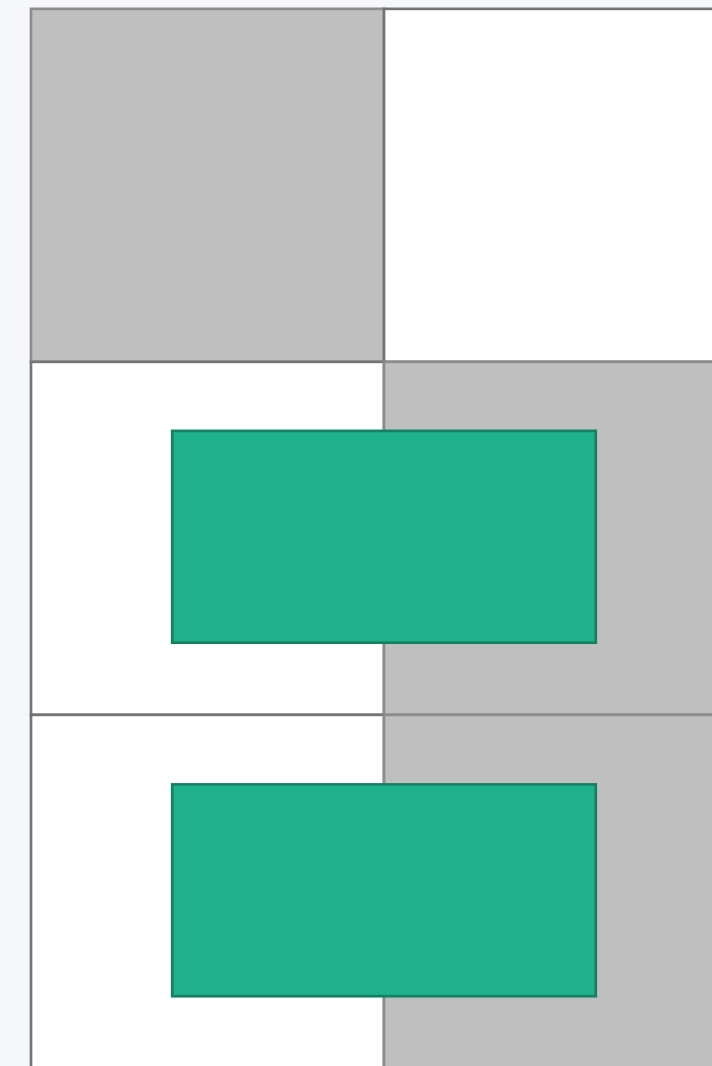
- i 열을 채울 때, $i-1$ 에 빈 칸이 있으면 안된다



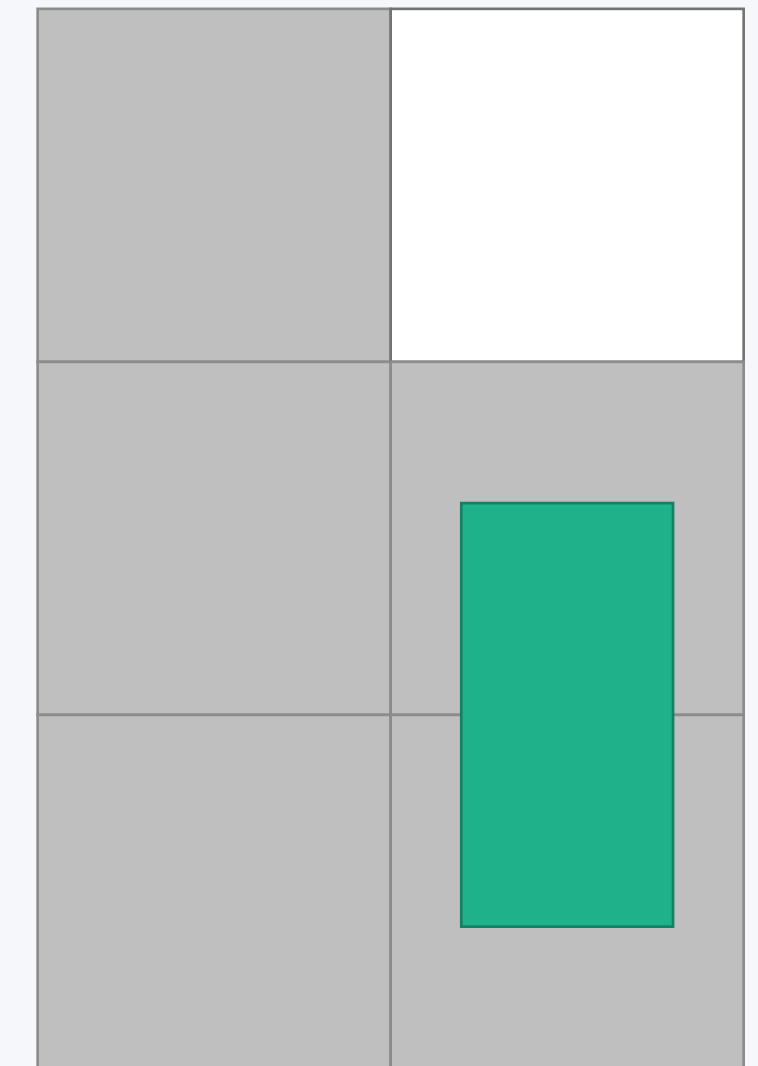
4 3



7 3



1 6



7 6

타일 채우기

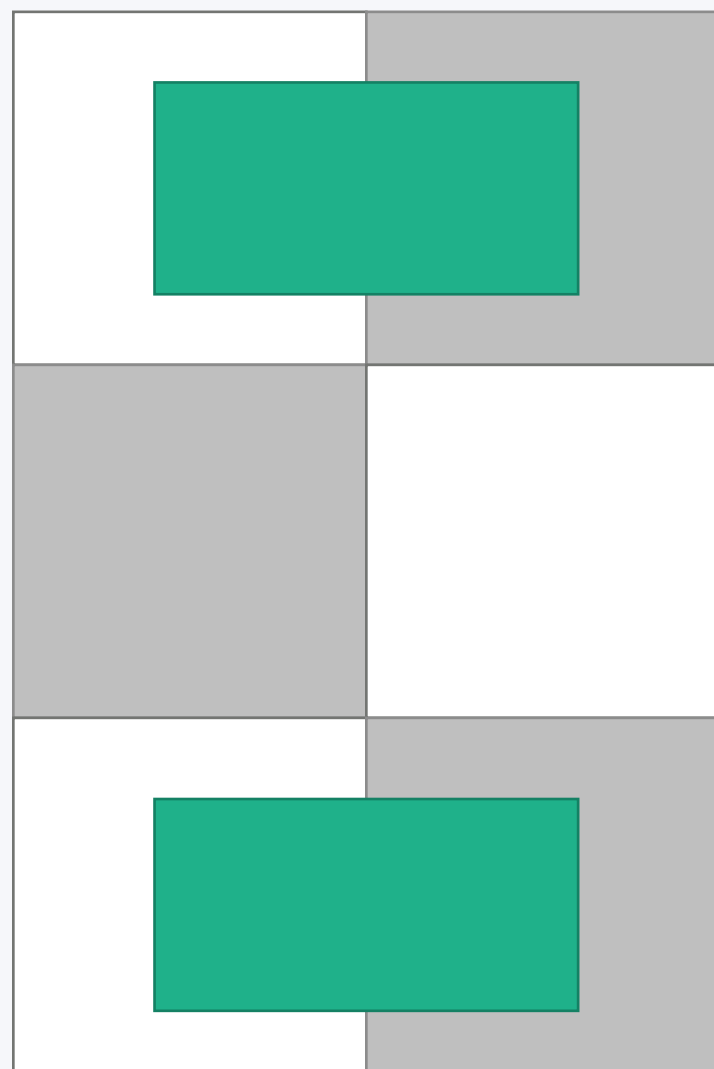
<https://www.acmicpc.net/problem/2133>

$$D[N][S] = 3 \times N$$

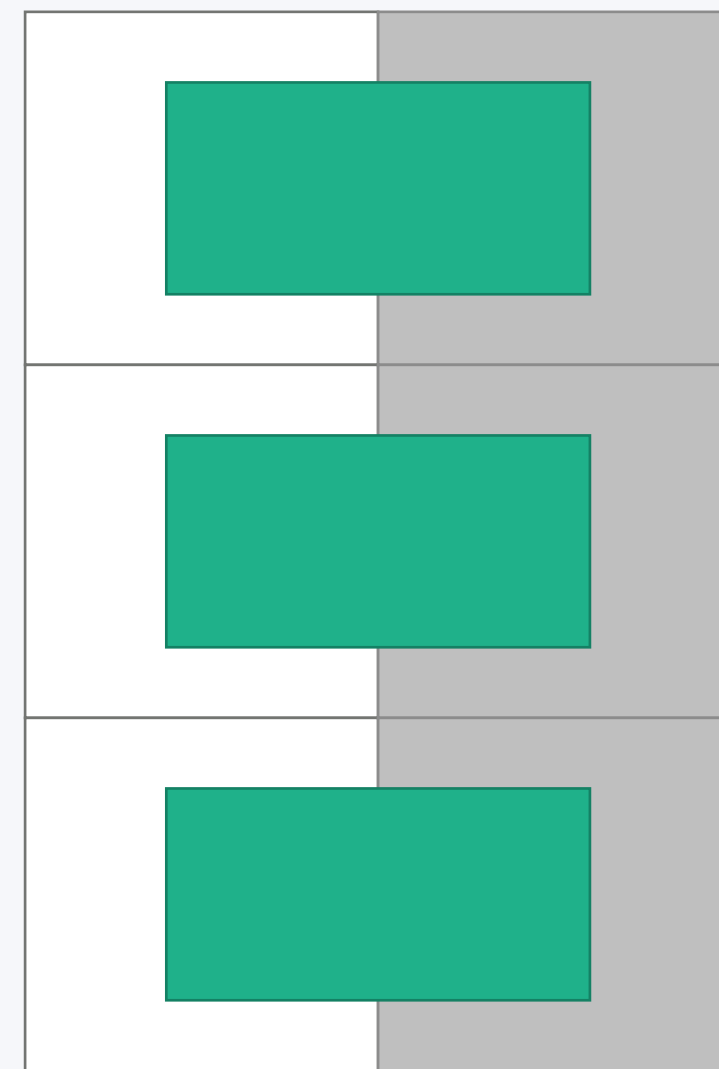
N개의 상태 S

6

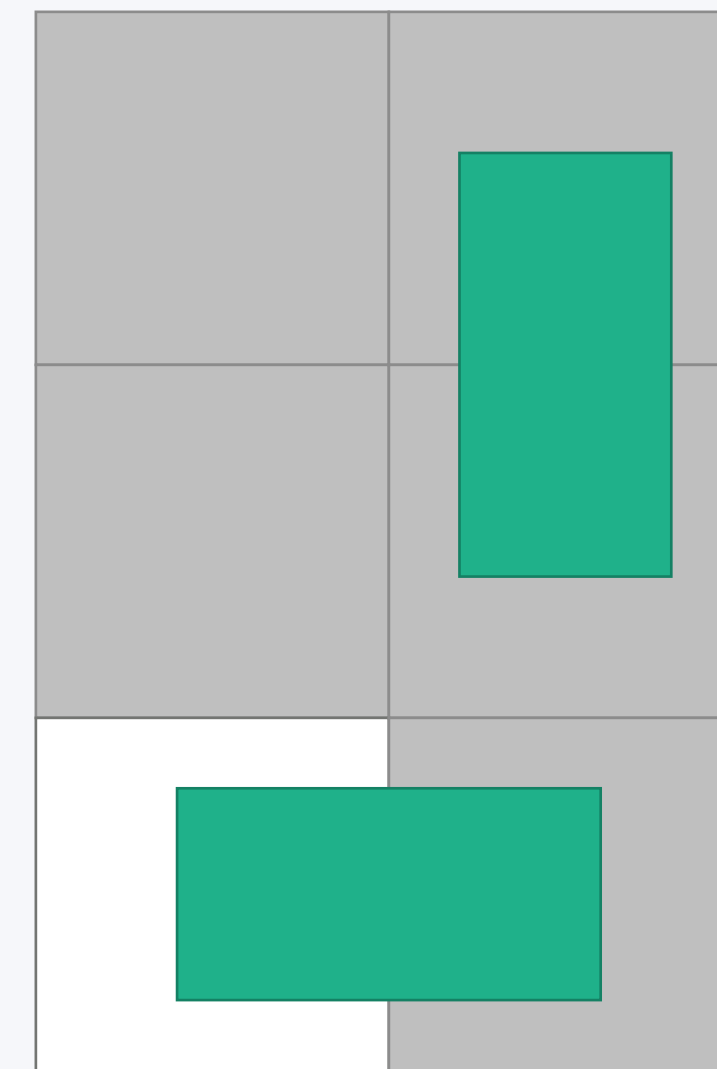
- i열을 채울 때, i-1에 빈 칸이 있으면 안된다



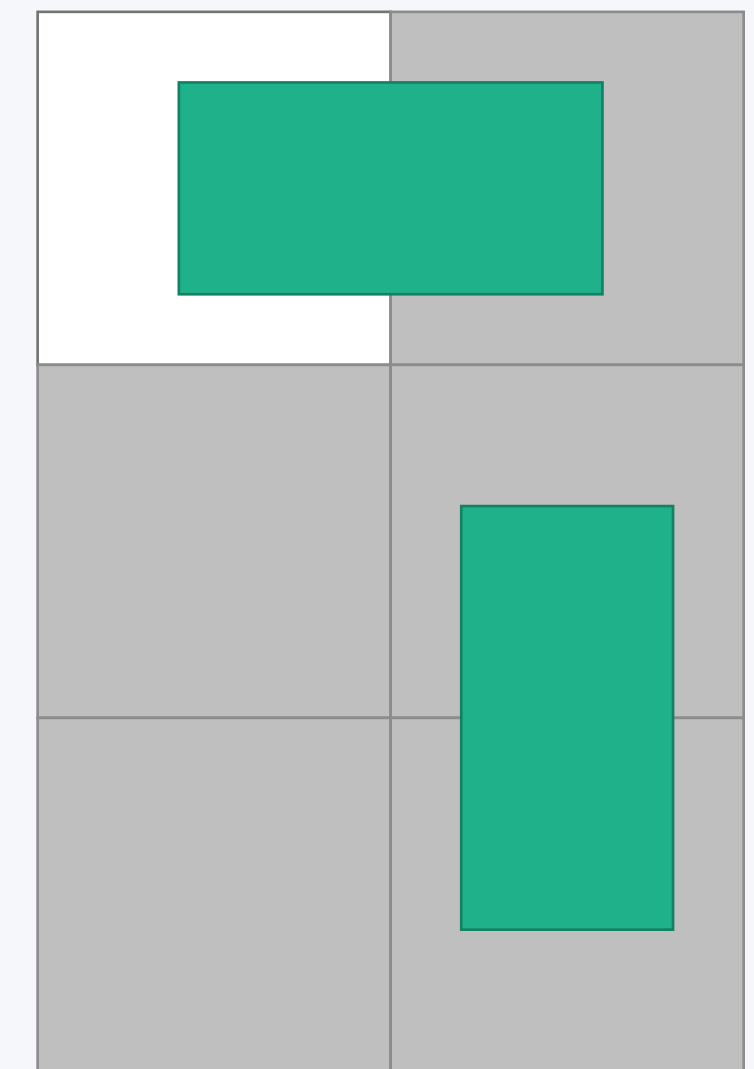
2 5



0 7



3 7



6 7

타일 채우기

<https://www.acmicpc.net/problem/2133>

- $D[i][0] = D[i-1][7]$
- $D[i][1] = D[i-1][6]$
- $D[i][2] = D[i-1][5]$
- $D[i][4] = D[i-1][3]$
- $D[i][3] = D[i-1][4] + D[i-1][7]$
- $D[i][6] = D[i-1][1] + D[i-1][7]$
- $D[i][5] = D[i-1][2]$
- $D[i][7] = D[i-1][0] + D[i-1][3] + D[i-1][6]$

타일 채우기

<https://www.acmicpc.net/problem/2133>

8

$D[N][7]$

```
D[0][7] = 1;
for (int i=1; i<=n; i++) {
    D[i][0] = D[i-1][7];
    D[i][1] = D[i-1][6];
    D[i][2] = D[i-1][5];
    D[i][4] = D[i-1][3];
    D[i][3] = D[i-1][4] + D[i-1][7];
    D[i][6] = D[i-1][1] + D[i-1][7];
    D[i][5] = D[i-1][2];
    D[i][7] = D[i-1][0] + D[i-1][3] + D[i-1][6];
}
```


타일 채우기

<https://www.acmicpc.net/problem/2133>

- 소스: <http://boj.kr/856e4a7c8ab74d18a3891c75d2774fbf>

$$D[5][2] \\ (0, 2)$$

$$D[27][3] \\ (0, 1, 3, 4) \quad 3$$

$$1 \rightarrow 4$$

$$0 \rightarrow ? \rightarrow 3 \\ 4 \rightarrow 1$$

16
8
2
1

외판원 순회

<https://www.acmicpc.net/problem/2098>

- 도시 1번부터 N번까지 있을 때

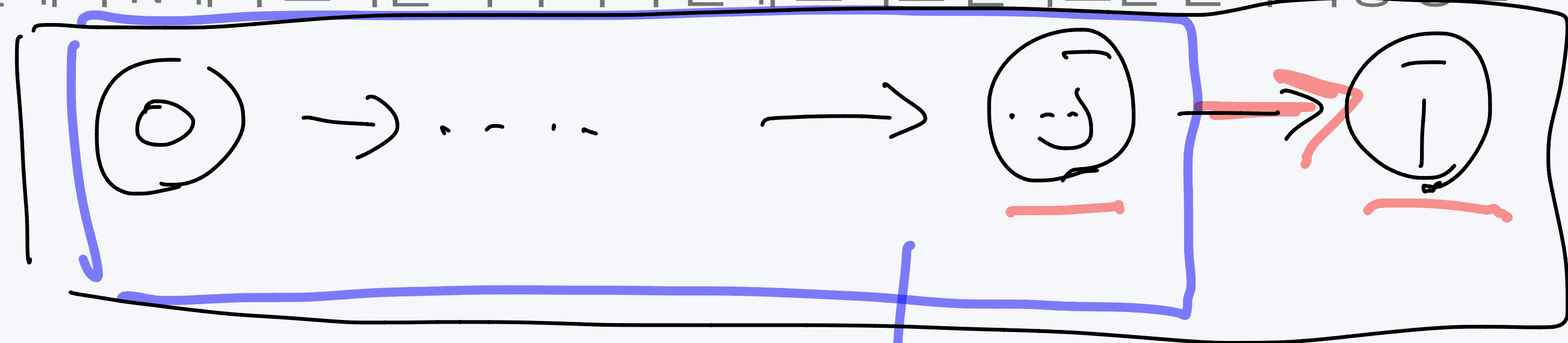
- 어느 한 도시에서 출발해서 N개의 도시를 거쳐 다시 원래 도시로 돌아오는 순회 여행 경로

- 비용의 최소값

- $N \leq 16$

$D[S][i] =$ 방문한 도시의 집합 S
마지막 도시 i

$$(2^N \times N) \times N = 2^N \cdot N^2$$



$$① A[S][i] \neq 0$$

$$② S \& (1 < i) \neq 0$$

$$③ S \& (1 < i) \neq 0$$

$$\rightarrow S^{\wedge} (1 < i)$$

$$D[S^{\wedge} (1 < i)][i] + A[i][1]$$

외판원 순회

<https://www.acmicpc.net/problem/2098>

- $D[S][i]$ = 도시를 방문한 상태가 S 이고, 현재 있는 위치가 i 일때 최소값
- 시작 도시는 1로 고정한다.
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$
- $2 \rightarrow 3 \rightarrow 1 \rightarrow 2$
- 모두 같기 때문

외판원 순회

<https://www.acmicpc.net/problem/2098>

- 상태를 사용할 수 있는 이유
- $1 \rightarrow 4$ 까지 정답이
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 라고 하자.
- 그러면, $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ 가 더 크다면, 방문한 도시의 집합은 같기 때문에 필요없는 값이 된다.
- $1 \rightarrow 5$ 로 갈 때 4에서 가는 경우라면
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ 가
- $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5$ 보다 무조건 최소값이다

외판원 순회

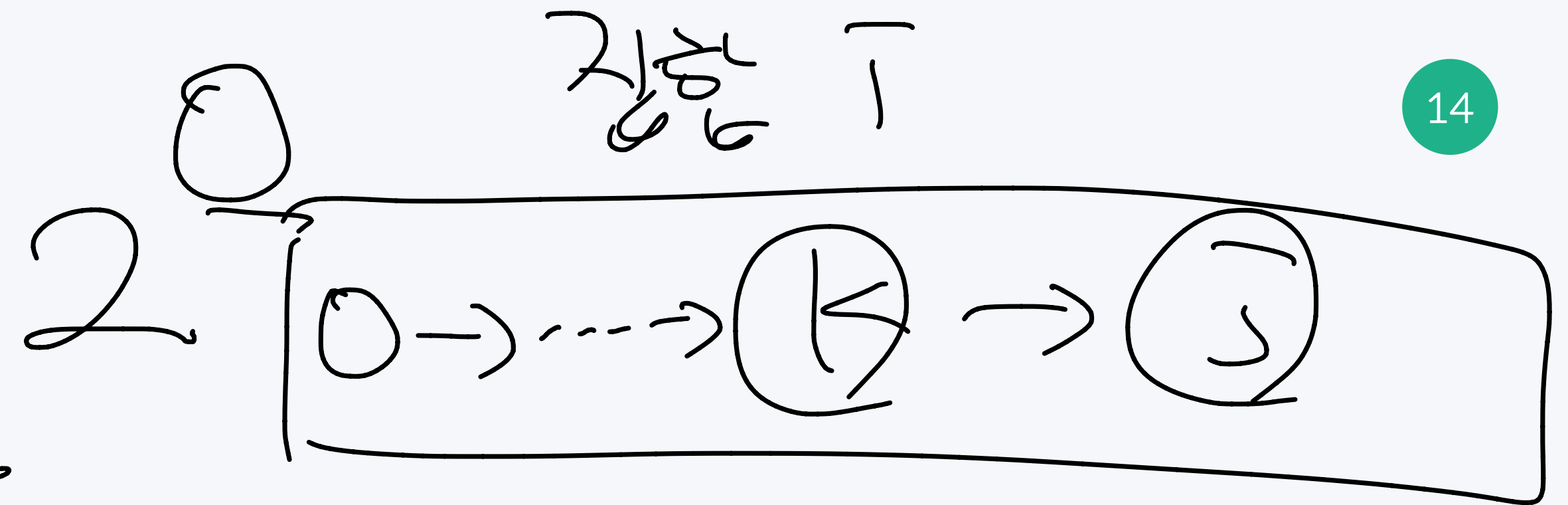
<https://www.acmicpc.net/problem/2098>

- $D[S][i] = D[S2][j] + A[j][i]$
- $S2 = S$ 에서 i 를 뺀 값
- i 는 S 에는 포함되어 있어야 하고, $S2$ 에는 포함되어 있지 않아야 한다
- j 는 S 에 포함되어 있어야 한다
- 초기값
 - $D[1][0] = 0$
 - 나머지 = MAX
- 정답
 - $\text{Min}(D[(1 \ll N) - 1][i] + A[i][0])$

외판원 순회

<https://www.acmicpc.net/problem/2098>

14



```
d[1][0] = 0;
```

```
for (int i=0; i<(1<<n); i++) {
```

```
    for (int j=1; j<n; j++) {
```

```
        if (i&(1<<j)) {
```

```
            for (int k=0; k<n; k++) {
```

```
                if (k!=j && (i&(1<<k)) && a[k][j]) {
```

```
                    d[i][j] = min(d[i][j], d[i-(1<<j)][k] + a[k][j]);
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

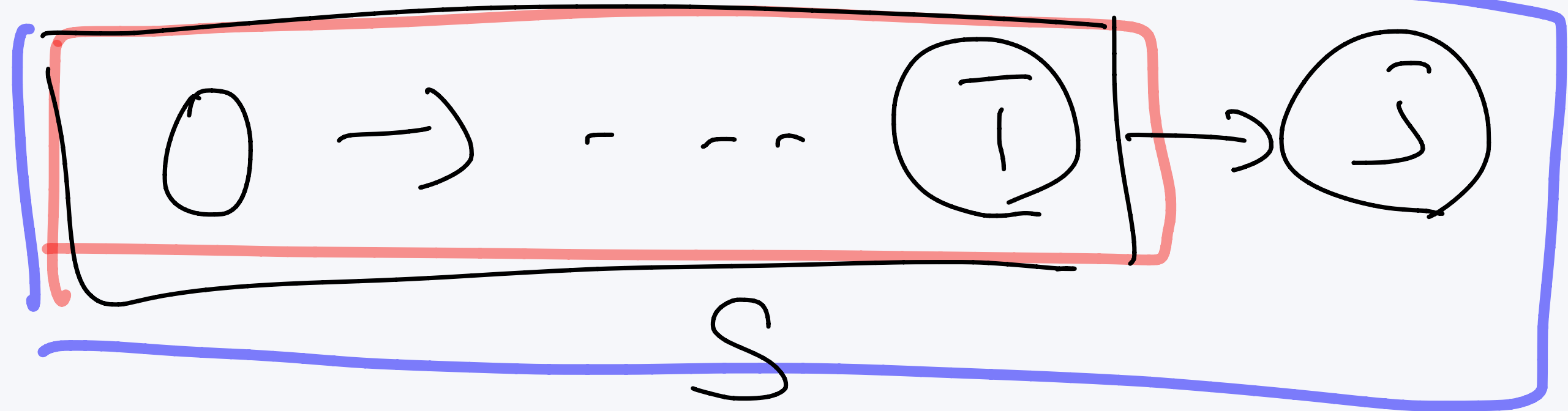
$$\frac{T \wedge (1 \ll j)}{T \wedge (1 \ll j)}$$

외판원 순회

<https://www.acmicpc.net/problem/2098>

- 소스: <http://boj.kr/3f544d19839c42de990874e4c724eb7b>

15



$DCS][C]$

$$DCS[(K<S)][C] = \min(DCS][C] + A[C][C])$$

$$① A[C][C] \neq 0$$

$$② S \& (K<?) \neq 0$$

$$③ S \& (K<?) == 0$$

발전소

<https://www.acmicpc.net/problem/1102>

- 발전소를 고치는 방법은 간단하다.
- 고장나지 않은 발전소를 이용해서 고장난 발전소를 재시작하면 된다.
- 하지만, 이 때 비용이 발생한다.
- 이 비용은 어떤 발전소에서 어떤 발전소를 재시작하느냐에 따라 다르다.
- 적어도 P개의 발전소가 고장나 있지 않도록, 발전소를 고치는 비용의 최솟값을 구하는 프로그램을 작성하시오.
- $N \leq 16$

발전소

<https://www.acmicpc.net/problem/1102>

- $D[i]$ = 발전소의 상태를 i 로 만들기 위해 필요한 최소 비용
- i = 발전소를 이진수로 나타낸 상태 (1: 켜있음)
- $D[i \mid (1 \ll k)] = D[i] + A[j][k]$
- j 는 i 에서 켜져있는 발전소
- k 는 i 에서 꺼져있는 발전소

발전소

<https://www.acmicpc.net/problem/1102>

- 소스: <http://boj.kr/3314a6640d3d4b11a9ab010c94ef9de0>

계단 수

19

<https://www.acmicpc.net/problem/1562>

- N이 주어질 때, 길이가 N이면서 0에서 9가 모두 등장하는 계단 수가 총 몇 개 있는지 구하기

쉬운 계단 수

20

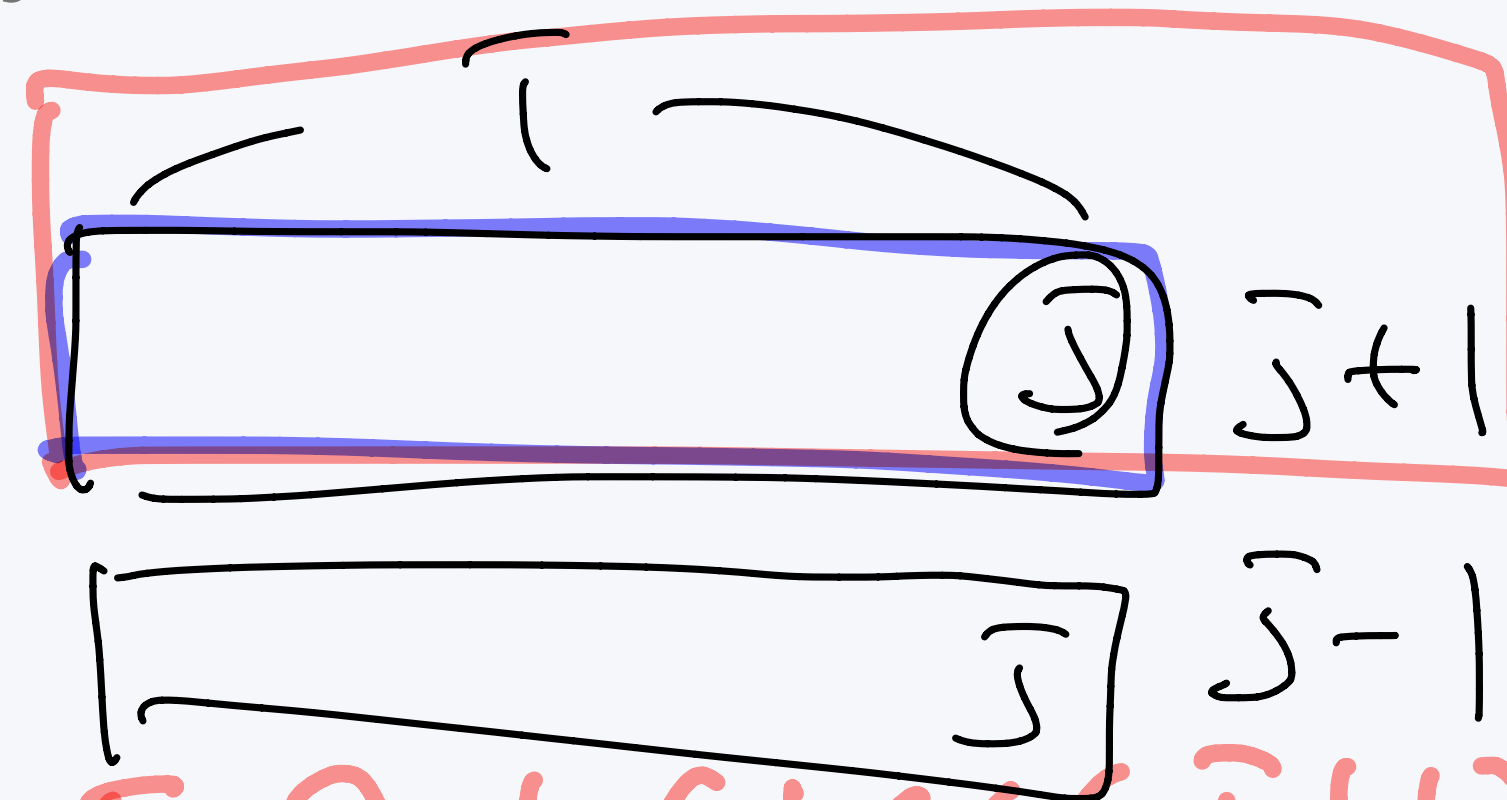
<https://www.acmicpc.net/problem/10844>

- 인접한 자리의 차이가 1이 나는 수를 계단 수라고 한다
- 예: 45656
- 길이가 N인 계단 수의 개수를 구하는 문제

DT [W] [S]

$$= \frac{1}{2} \tau, \text{ 마크로 } j, \text{ 미시 } S$$

- ~~$D[i][j] = D[i-1][j-1] + D[i-1][j+1]$~~




$D[i+1][j+1][s | (1 \ll (j+1))]$

$D [[[[[2] [2] [2] \dots [2]]]]]$
 $\quad \quad \quad \uparrow \quad \quad \uparrow \quad \quad \uparrow \quad \quad \uparrow$
 $\quad \quad \quad \textcircled{5} \quad \quad 1 \quad \quad 2 \quad \quad 9$

$P[\text{true}] \text{ } [S] \text{ } [S] \text{ } [f]$

$\frac{1}{x}$
 $\frac{1}{x}$

- 2

- 

$$2 \rightarrow 3 \rightarrow 4$$

계단 수

23

<https://www.acmicpc.net/problem/1562>

- 소스: <http://boj.kr/78be8926f65441b5bea965474b59c8ba>

계단 수

<https://www.acmicpc.net/problem/1562>

- 이 문제는 계단수 이기 때문에, 상태 S 를 범위로 나타낼 수 있다.
- $D[N][M][L][R]$ = 길이가 N 이고, M 으로 끝나는 계단수, 지금까지 나온 수: $L \sim R$

계단 수

25

<https://www.acmicpc.net/problem/1562>

- 소스: <http://boj.kr/d5f4014fa7644d77ae283899fa8047e1>

박성원

<https://www.acmicpc.net/problem/1086>

- 서로 다른 정수로 이루어진 집합이 있다
- 이 집합의 순열을 합치면 큰 정수 하나를 만들 수 있다
- 예를 들어, {5221,40,1,58,9}로 5221401589를 만들 수 있다
- 합친수가 정수 K로 나누어 떨어지는 순열을 구하는 프로그램을 작성하시오
- 그냥 랜덤하게 순열 하나를 정답이라고 출력하려고 한다
- 이 문제에는 정답이 여러 개 있을 수도 있고, 우연히 문제의 정답을 맞출 수도 있다.
- 우연히 정답을 맞출 확률을 분수로 출력하는 프로그램을 작성하시오

박성원

27

<https://www.acmicpc.net/problem/1086>

- 서로 다른 정수로 이루어진 집합이 있다
- 이 집합의 순열을 합치면 큰 정수 하나를 만들 수 있다
- 예를 들어, {5221,40,1,58,9}로 5221401589를 만들 수 있다
- 합친수가 정수 K로 나누어 떨어지는 순열을 구하는 프로그램을 작성하시오
- 그냥 랜덤하게 순열 하나를 정답이라고 출력하려고 한다
- 이 문제에는 정답이 여러 개 있을 수도 있고, 우연히 문제의 정답을 맞출 수도 있다.
- 우연히 정답을 맞출 확률을 분수로 출력하는 프로그램을 작성하시오

<https://www.acmicpc.net/problem/1086>

- $D[S][M]$ = 사용한 순열에 포함된 수의 집합이 S 이고, 나머지가 M 인 것의 개수
- S 에 포함되어 있지 않은 수를 L 이라고 했을 때
- L 번째 수 : $A[L]$
- L 번째 수의 길이: $Len[L]$
- L 번째 수가 포함된 경우의 나머지 = $(M * 10^{Len[L]} + A[L]) \% M$
- $D[S \mid (1 \ll L)][next] += D[S][M]$

박성원

29

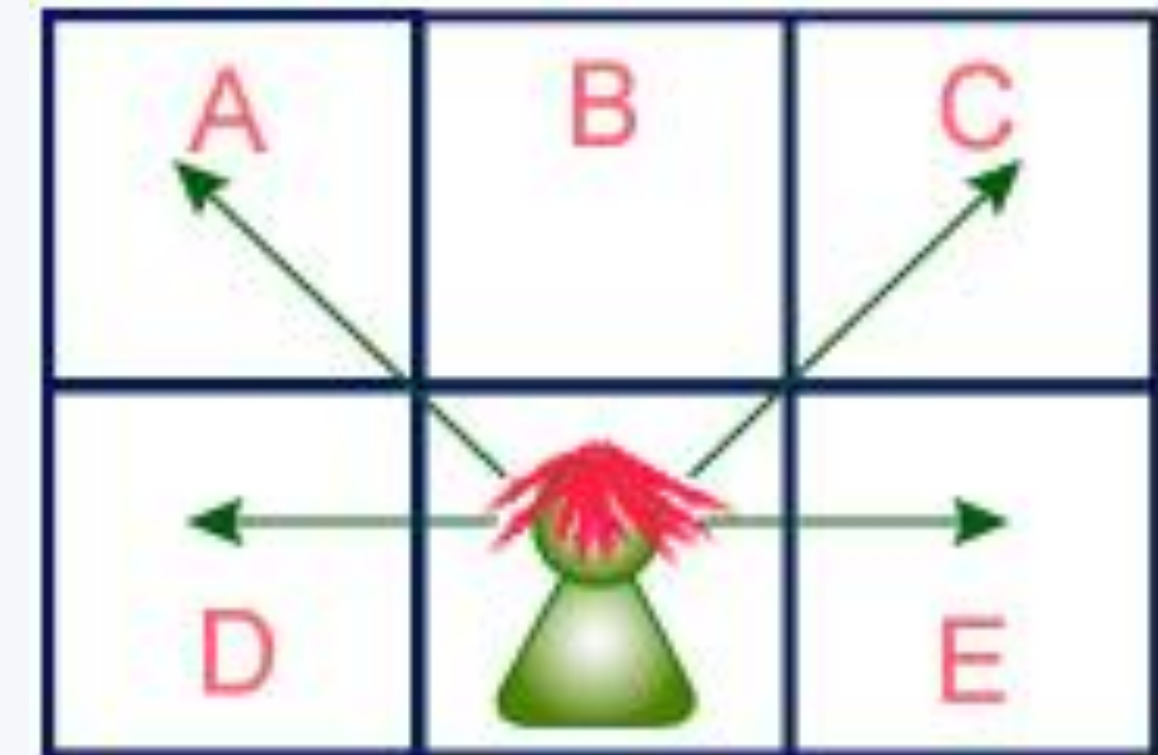
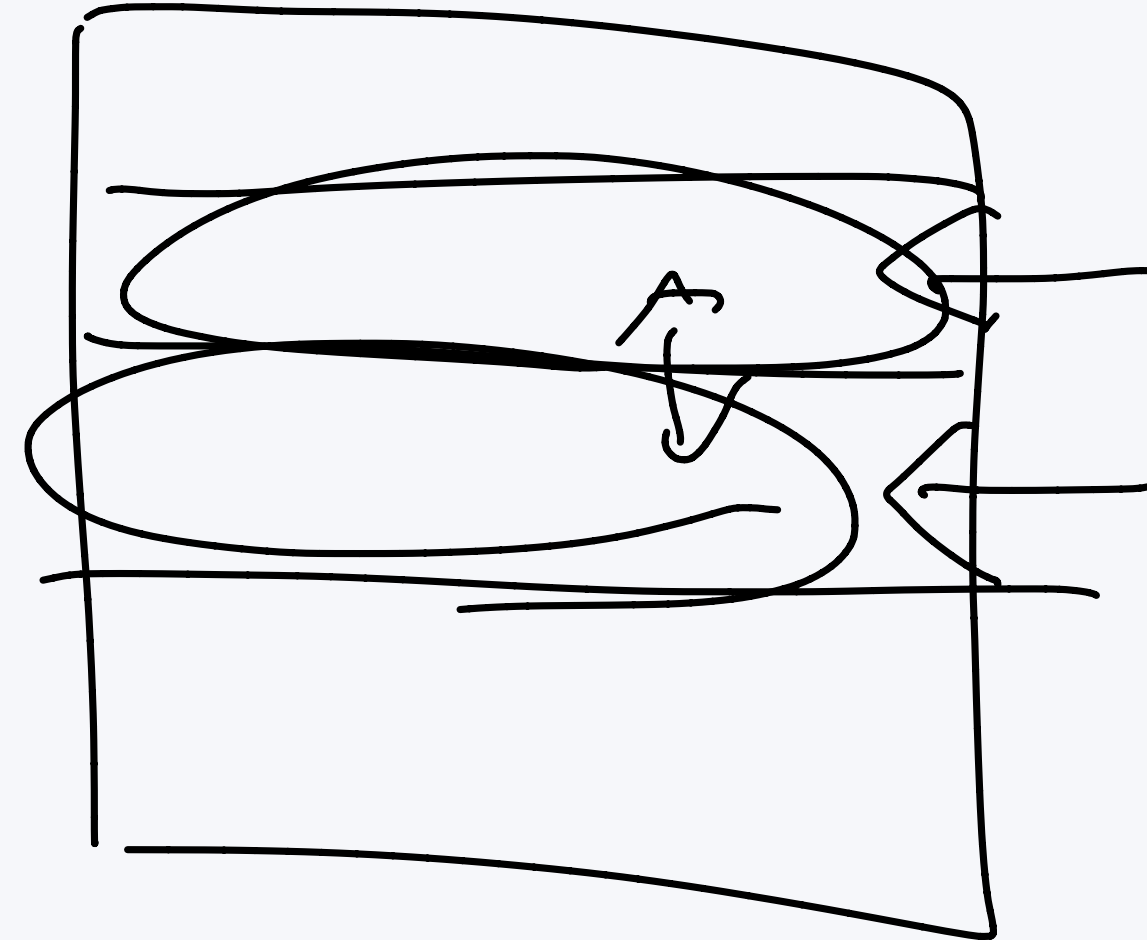
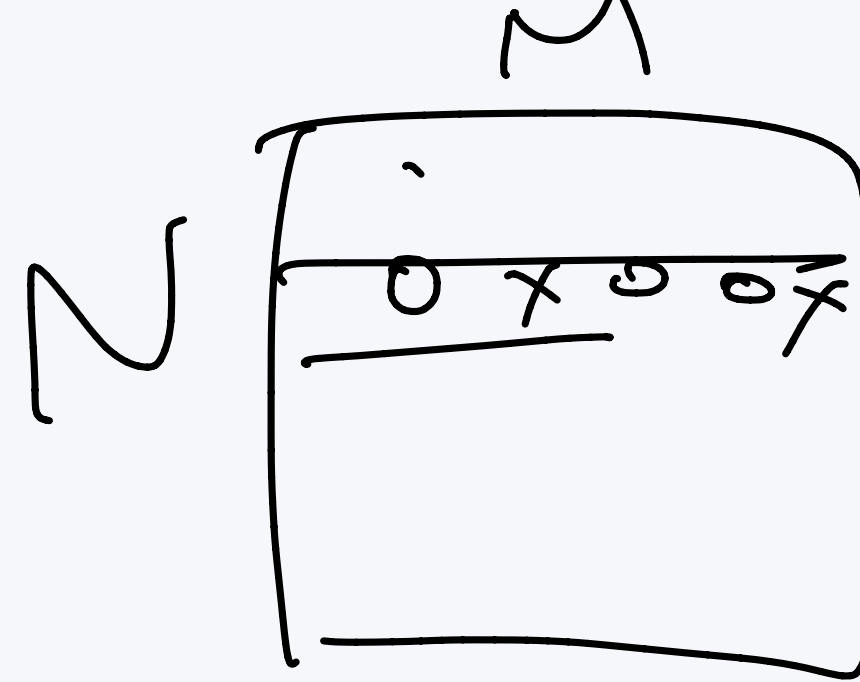
<https://www.acmicpc.net/problem/1086>

- 소스: <http://boj.kr/3f00f7e37e0d435b994a7087204d72ca>

컨닝

<https://www.acmicpc.net/problem/1014>

- N행 M열 직사각형 교실에서 시험을 보려고 한다.
- 최대 몇 명의 학생이 시험을 볼 수 있는가?
- $1 \leq N, M \leq 10$

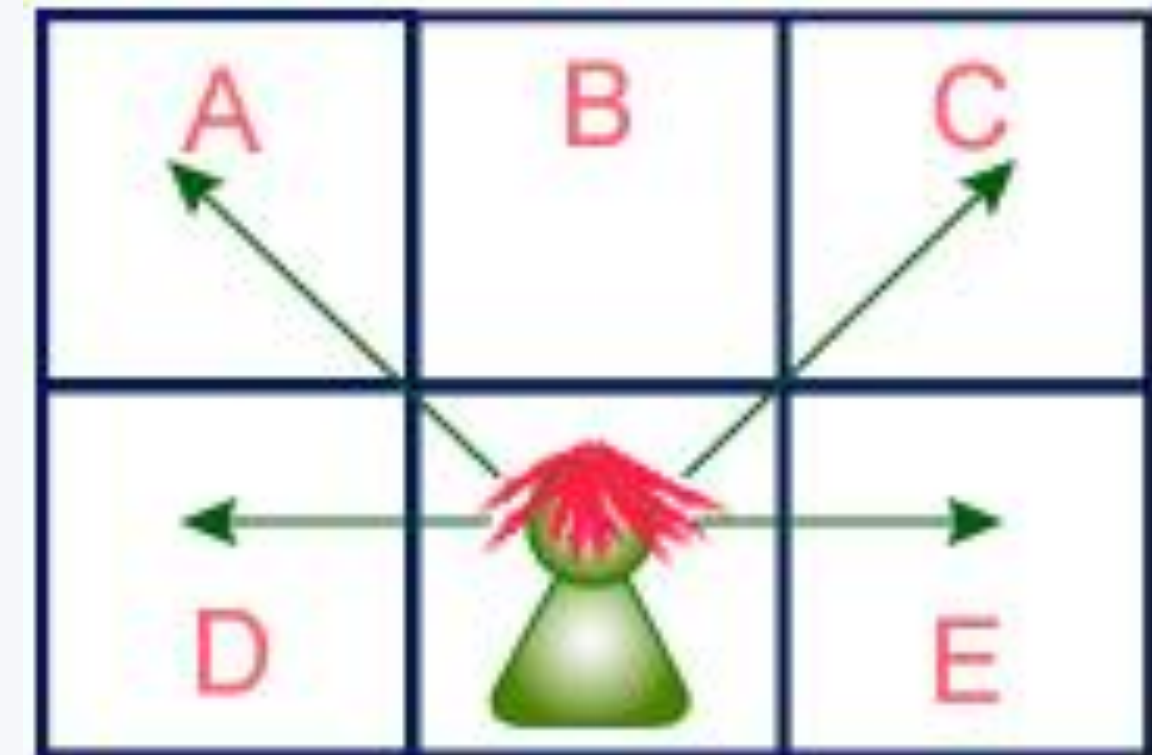


컨닝

31

<https://www.acmicpc.net/problem/1014>

- $D[i][j]$ = i 번 행의 상태가 j 일 때, 앞을 수 있는 학생의 수



컨닝

$$N \times 2^M \times 2^M = N \times 2^{2M}$$

32

<https://www.acmicpc.net/problem/1014>

• $D[i][j]$ = i 번 행의 상태가 j 일 때, 앉을 수 있는 학생의 수

• $D[i][j] = \max_k (D[i-1][k] + (j \text{ 상태에 있는 학생의 수}))$

• k 와 j 는 서로 ~~안~~ 충돌할 수 있는 상태이어야 함

안충돌

(14)

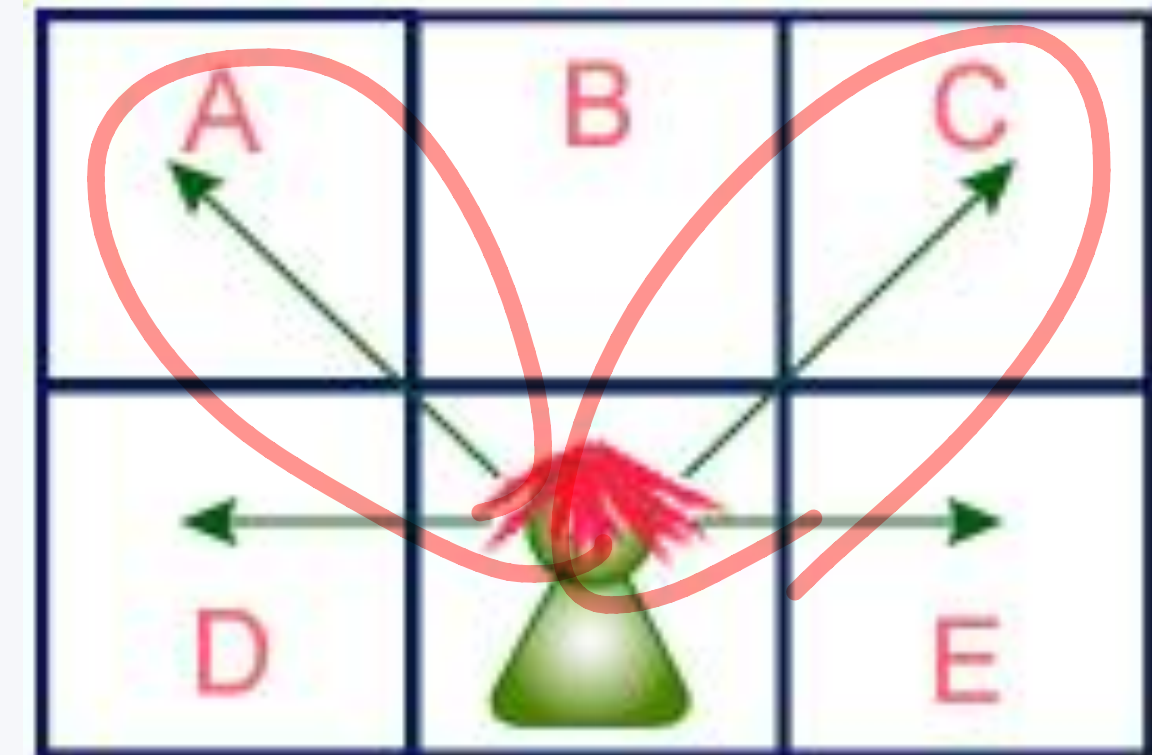
(14)

(14)

상태 K

상태 J

N



컨닝

<https://www.acmicpc.net/problem/1014>

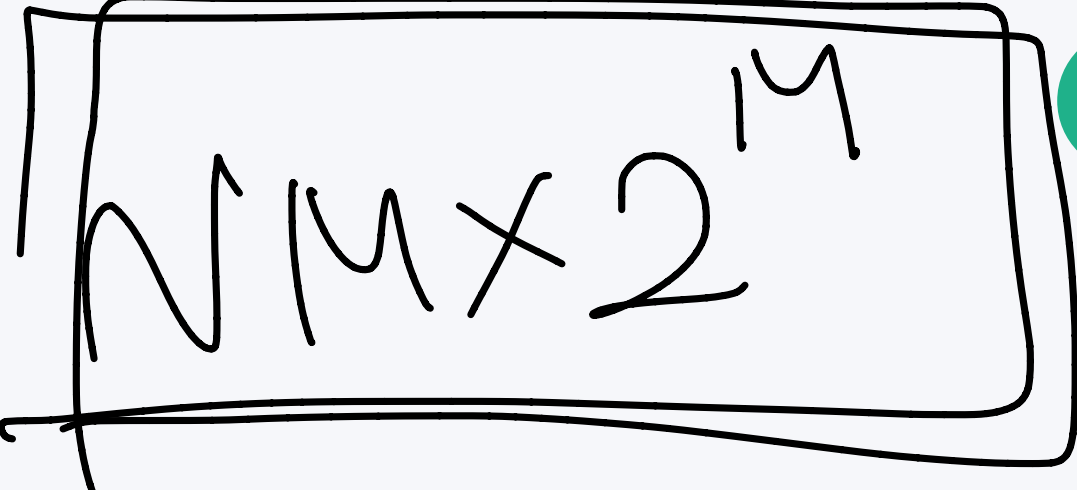
- 소스: <http://boj.kr/f8b97be0667c4fcc818f9157f78a1a72>

격자판 채우기

<https://www.acmicpc.net/problem/1648>

- $N \times M$ 격자판을 2×1 크기의 도미노로 채우는 방법의 수
- $1 \leq N, M \leq 14$

$$N \times 2^{2^M}$$



A hand-drawn rectangular box containing the formula $N \times M \times 2^M$. To the right of the box is a green circle containing the number 34.

격자판 채우기

$DP[i][S] =$ i번째 칸까지 채울 것임

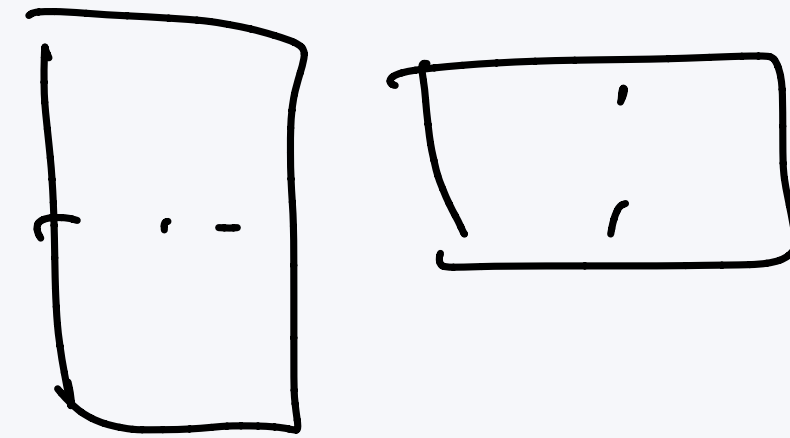
35

<https://www.acmicpc.net/problem/1648>

• $go(i, S) =$ i번째 칸을 채울 것이고, i번째 칸부터 칸 M개의 상태가 S일 때, 경우의 수

• i번째 칸?

• Row-major order 순서



0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

- $go(i, S)$ = i 번째 칸을 채울 것이고, i 번째 칸부터 칸 M 개의 상태가 S 일 때, 경우의 수
- $i = 7$ 인 경우 상태 S 가 나타내는 범위 7~12번 칸 (M 개 칸)

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

$$S = 4 = \underline{\underline{100}}_2$$

37

- $go(i, S)$ = i 번째 칸을 채울 것이고, i 번째 칸부터 칸 M 개의 상태가 S 일 때, 경우의 수
- $i = 7$ 인 경우 상태 S 가 나타내는 범위 7~12번 칸 (M 개 칸)
- $S = 4$ 인 경우: 4는 2진수로 000100(2) 이다. 따라서, 9번 칸만 이미 채워져 있는 상태

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

$$S = 5 = \underline{101}_2$$

38

<https://www.acmicpc.net/problem/1648>

- $go(i, S)$ = i 번째 칸을 채울 것이고, i 번째 칸부터 칸 M 개의 상태가 S 일 때, 경우의 수
- $i = 7$ 인 경우 상태 S 가 나타내는 범위 7~12번 칸 (M 개 칸)
- $S = 5$ 인 경우: 5는 2진수로 000101(2) 이다. 따라서, 7, 9번 칸이 이미 채워져 있는 상태

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

$$S=13$$

39

<https://www.acmicpc.net/problem/1648>

- $go(i, S)$ = i 번째 칸을 채울 것이고, i 번째 칸부터 칸 M 개의 상태가 S 일 때, 경우의 수
- $i = 7$ 인 경우 상태 S 가 나타내는 범위 7~12번 칸 (M 개 칸)
- $S = 13$ 인 경우: 13은 2진수로 001101(2) 이다. 따라서, 7, 9, 10번 칸이 이미 채워져 있는 상태

0	1	2	3	4	5
6	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	11
<u>12</u>	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

- $go(i, S)$ = i 번째 칸을 채울 것이고, i 번째 칸부터 칸 M 개의 상태가 S 일 때, 경우의 수
- $i = 7$ 인 경우 상태 S 가 나타내는 범위 7~12번 칸 (M 개 칸)
- $S = 0$ 인 경우: 0은 2진수로 000000(2) 이다. 따라서, 모두 비어있는 상태

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

- 왜 M개를 저장할까?
- i번째 칸에 블록을 놓는 경우에 항상 i번째 칸이 왼쪽 또는 위가 되게 놓기 때문

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

- 상태 S에서 이미 1로 되어있는 곳은 아래 그림과 같이 위에서 채웠다는 의미

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

43

① $\text{if}((S \& 1) == 1)$

$\text{go}(i+1, S \gg 1)$

- 블럭을 놓을 수 없는 경우
- 7이 이미 채워져 있는 경우
- 상태 S를 이진수로 나타냈을 때, 마지막 비트가 1인 경우 $((S \& 1) == 1)$
- 이런 경우에는 채울 수 없기 때문에, 다음 칸을 채워야 함
- S를 오른쪽으로 한 비트 shift 해야함 $\text{go}(i+1, (S \gg 1))$

$\text{go}(i, S)$

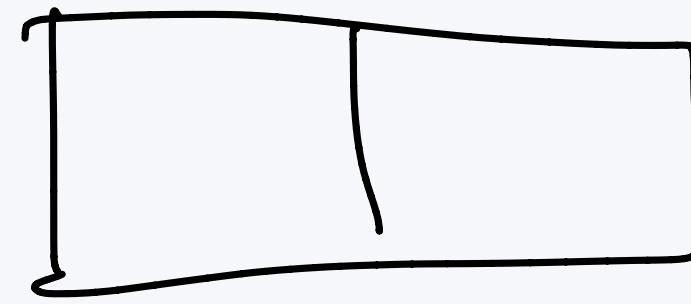
0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

44



- 블록을 놓을 수 있는 경우 (1x2를 놓는 경우)
- $S \& 1$ 과 $S \& 2$ 모두 0이어야 함
- 이 때, 가장 오른쪽 칸인지 아닌지 확인하는 절차도 필요
- $go(N+2, (S \gg 2))$

$S \& (1 \ll 0)$
 $S \& (1 \ll 1)$

$go(N+2, S \gg 2)$

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

$N \times M \times 2^M$

45

- 블록을 놓을 수 있는 경우 (2x1를 놓는 경우)

- 현재 칸이 비어있으면 항상 가능함

- $go(N+1, (S \gg 1) | (1 \ll (M-1)))$

$go(T+1, (S \gg 1) | (1 \ll (M-1)))$

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

격자판 채우기

<https://www.acmicpc.net/problem/1648>

$$T = N \times M$$
$$S = 0$$

46

- 올바르게 채웠는지는 어떻게 확인하나?
- $N \times M$ 번째 칸이 존재한다고 가정
- $N \times M$ 번째 칸에서 상태가 0이면 올바르게 채운 것!

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18					

격자판 채우기

47

<https://www.acmicpc.net/problem/1648>

- 소스: <http://boj.kr/2035af9b05ee4e688a223e173a24c634>

두부장수 장홍준

48

<https://www.acmicpc.net/problem/1657>

- 세로크기 N, 가로크기 M인 두부판을 가지고 2x1짜리 두부로 잘라서 판다.
- 두부판의 위치마다 등급이 다르다.
- $1 \leq N, M \leq 14$

	A	B	C	D	F
A	10	8	7	5	1
B	8	6	4	3	1
C	7	4	3	2	1
D	5	3	2	2	1
F	1	1	1	1	0

두부장수 장홍준

<https://www.acmicpc.net/problem/1657>

- 격자판 채우기와 비슷하게 해결한다

두부장수 장홍준

50

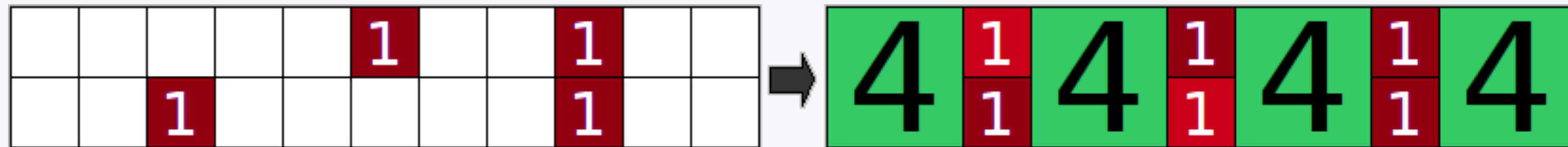
<https://www.acmicpc.net/problem/1657>

- 소스: <http://boj.kr/7a57fec62222491aaabf9a20090e0c24>

4블럭

<https://www.acmicpc.net/problem/14389>

- 1블럭의 크기는 1×1 , 점수는 1점
- 4블럭의 점수는 2×2 , 점수는 4점
- 1블럭이 일부 채워진 $N \times M$ 크기의 보드가 주어졌을 때, 점수의 최대값을 구하는 문제



4블럭

<https://www.acmicpc.net/problem/14389>

```
if (a[i][j] == '1') {
    return ans = go(index+1, state >> 1) + 1;
}
ans = go(index+1, state>>1);
if ((state & 1) == 0) {
    ans = max(ans, go(index+1, state>>1) + 1);
}
if (i != n-1 && j != m-1 && (state & 1) == 0 && (state & 2) == 0) {
    if (a[i][j+1] == '.' && a[i+1][j] == '.' && a[i+1][j+1] == '.') {
        ans = max(ans, go(index+2, (state>>2) | (1<<(m-1)) | (1<<(m-2)))) + 16);
    }
}
```

4블럭

<https://www.acmicpc.net/problem/14389>

- 소스: <http://boj.kr/1c90ee16cd664eda950d5825be5805ab>

체스판

<https://www.acmicpc.net/problem/12960>

- 윤호는 L-모양으로 생긴 타일을 매우 많이 가지고 있다.
- 윤호는 동혁이의 체스판에 다음 조건을 만족시키면서 타일을 체스판 위에 올려 놓으려고 한다.
 - 모든 타일은 회전 시킬 수 있다. (90, 180, 270도)
 - 모든 타일은 체스판 위의 세 칸을 덮어야 한다.
 - 타일은 겹치면 안된다.
 - 말이 이미 올려져있는 칸은 타일이 덮을 수 없다.
 - 타일의 꼭지점 칸(두 정사각형과 붙어있는 칸)은 체스판의 검정 칸을 덮어야 한다.
- 윤호가 놓을 수 있는 타일의 최대 개수를 구하는 문제

체스판

<https://www.acmicpc.net/problem/12960>

```
// BW
// W.
if (j+1 < m && i+1 < n) {
    if ((state&2) == 0 && a[i][j+1] == '.' && a[i+1][j] == '.') {
        ans = max(ans, go(index+2, (state>>2) | (1<<(m-2))) + 1);
    }
}
// WB
// .W
if (j+1 < m && i+1 < n) {
    if ((state&2) == 0 && a[i][j+1] == '.' && a[i+1][j+1] == '.') {
        ans = max(ans, go(index+2, (state>>2) | (1<<(m-1))) + 1);
    }
}
```

체스판

56

<https://www.acmicpc.net/problem/12960>

```
// .W
// WB
if (j-1 >= 0 && i+1 < n) {
    if ((state&(1<<(m-1))) == 0 && a[i+1][j-1] == '.' && a[i+1][j] == '.') {
        ans = max(ans, go(index+1, (state>>1) | (1<<(m-1)) | (1<<(m-2)))) + 1);
    }
}

// W.
// BW
if (i+1 < n && j+1 < m) {
    if (a[i+1][j] == '.' && a[i+1][j+1] == '.') {
        ans = max(ans, go(index+2, (state>>2) | (1<<(m-1)) | (1<<(m-2)))) + 1);
    }
}
```


체스판

57

<https://www.acmicpc.net/problem/12960>

- 소스: <http://boj.kr/5c2f75606b4e4a40aa241a9fcfc571eb>

타일 놓기

58

<https://www.acmicpc.net/problem/14390>

- 직사각형을 타일로 가득 채우는 최소 타일 개수를 구하는 문제

타일 놓기

<https://www.acmicpc.net/problem/14390>

- $D[i][state]$ = i 행의 상태가 $state$ 일 때, 타일의 최소 개수
- $D[i][state] = \sum D[i][prev_state]$

타일 놓기

60

<https://www.acmicpc.net/problem/14390>

```
int cnt = 0; int last = -100;
// 0 : -, 1: |
for (int k=0; k<m; k++) {
    if (a[i][k] == '#') continue;
    if (cur & (1<<k)) {
        if (i-1 == 0) cnt++;
        else if ((prev & (1<<k)) == 0) cnt++;
        else if (a[i-1][k] == '#') cnt++;
    } else {
        if (last+1 != k) cnt++;
        last = k;
    }
}
```

타일 놓기

61

<https://www.acmicpc.net/problem/14390>

- 소스: <http://boj.kr/21a2d8f41b304bda93cc36de9e617d1e>

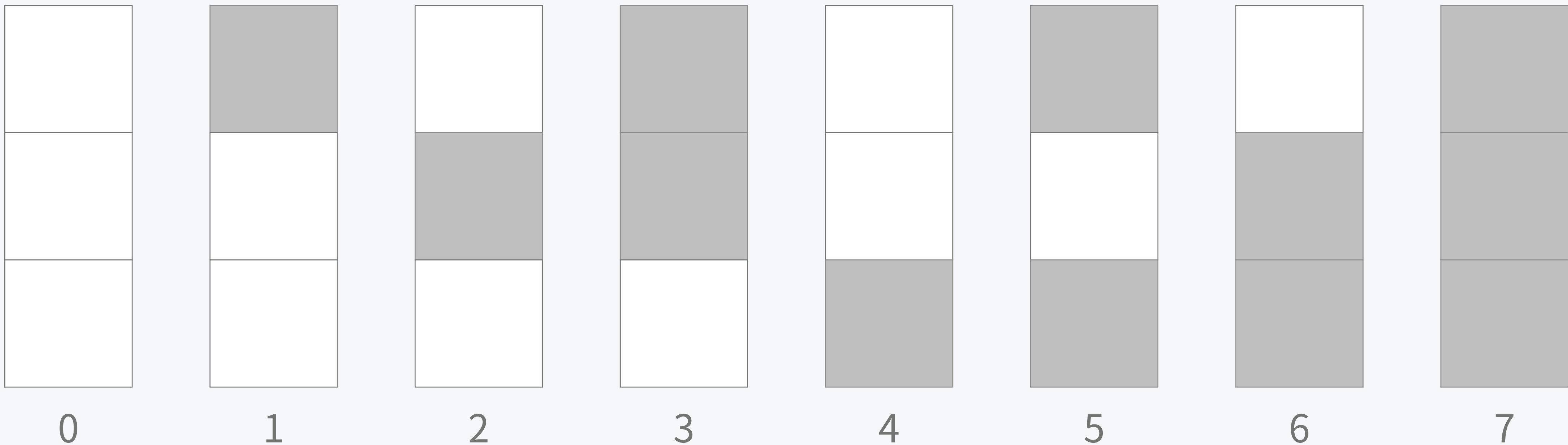
행렬과 다이나믹

타일 채우기 2

63

<https://www.acmicpc.net/problem/13976>

- $3 \times N$ 을 1×2 , 2×1 로 채우는 방법의 수
- $D[i][j] = 3 \times i$ 를 채우는 방법의 수, i 열의 상태는 j
- 마지막에 올 수 있는 가능한 경우의 수 (회색: 채워져 있는 칸)



타일 채우기 2

<https://www.acmicpc.net/problem/13976>

- $D[i][0] = D[i-1][7]$
- $D[i][1] = D[i-1][6]$
- $D[i][2] = D[i-1][5]$
- $D[i][4] = D[i-1][3]$
- $D[i][3] = D[i-1][4] + D[i-1][7]$
- $D[i][6] = D[i-1][1] + D[i-1][7]$
- $D[i][5] = D[i-1][2]$
- $D[i][7] = D[i-1][0] + D[i-1][3] + D[i-1][6]$

타일 채우기 2

65

<https://www.acmicpc.net/problem/13976>

$$\begin{bmatrix} D[i][0] \\ D[i][1] \\ D[i][2] \\ D[i][3] \\ D[i][4] \\ D[i][5] \\ D[i][6] \\ D[i][7] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} D[i-1][0] \\ D[i-1][1] \\ D[i-1][2] \\ D[i-1][3] \\ D[i-1][4] \\ D[i-1][5] \\ D[i-1][6] \\ D[i-1][7] \end{bmatrix}$$

타일 채우기 2

<https://www.acmicpc.net/problem/13976>

- 소스: <http://boj.kr/cc209021212a4c4c914ce5e9715c2a88>

동민 수열

<https://www.acmicpc.net/problem/1529>

- 4와 7로 이루어진 수를 금민수라고 한다
- 수 N개가 주어졌을 때, 만들 수 있는 동민 수열의 개수를 구하는 문제
- 다음 조건을 만족하는 수열 A를 동민 수열이라고 한다
 1. 수열을 구성하는 수가 금민수이다
 2. 수열을 구성하는 수가 입력으로 주어지는 수에 있어야 한다
 3. 모든 i 에 대해서, $A[i]$ 의 마지막 자리는 $A[i+1]$ 의 첫 번째 자리와 같아야 한다

동민 수열

<https://www.acmicpc.net/problem/1529>

- 4와 7로 이루어진 수를 금민수라고 한다
- 수 N 개가 주어졌을 때, 만들 수 있는 동민 수열의 개수를 구하는 문제
- 다음 조건을 만족하는 수열 A 를 동민 수열이라고 한다
 1. 수열을 구성하는 수가 금민수이다
 2. 수열을 구성하는 수가 입력으로 주어지는 수에 있어야 한다
 3. 모든 i 에 대해서, $A[i]$ 의 마지막 자리는 $A[i+1]$ 의 첫 번째 자리와 같아야 한다

동민 수열

<https://www.acmicpc.net/problem/1529>

- 4와 7로 이루어진 수를 금민수라고 한다
- 수 N 개가 주어졌을 때, 만들 수 있는 동민 수열의 개수를 구하는 문제
- 다음 조건을 만족하는 수열 A 를 동민 수열이라고 한다
 1. 수열을 구성하는 수가 금민수이다
 2. 수열을 구성하는 수가 입력으로 주어지는 수에 있어야 한다
 3. 모든 i 에 대해서, $A[i]$ 의 마지막 자리는 $A[i+1]$ 의 첫 번째 자리와 같아야 한다
- 실제 수가 무엇인지 중요한 것이 아니고, 첫 숫자와 마지막 숫자가 중요하다

동민 수열

<https://www.acmicpc.net/problem/1529>

- 4로 시작하고 4로 끝나는 수의 개수를 c44
- 4로 시작하고 7로 끝나는 수의 개수를 c47
- 7로 시작하고 4로 끝나는 수의 개수를 c74
- 7로 시작하고 7로 끝나는 수의 개수를 c77

동민 수열

<https://www.acmicpc.net/problem/1529>

- 4로 시작하고 4로 끝나는 길이가 n 인 동민 수열의 개수를 $d44[n]$
- 4로 시작하고 7로 끝나는 길이가 n 인 동민 수열의 개수를 $d47[n]$
- 7로 시작하고 4로 끝나는 길이가 n 인 동민 수열의 개수를 $d74[n]$
- 7로 시작하고 7로 끝나는 길이가 n 인 동민 수열의 개수를 $d77[n]$

동민 수열

<https://www.acmicpc.net/problem/1529>

- 4로 시작하고 4로 끝나는 길이가 n 인 동민 수열의 개수를 $d44[n]$
 - 4로 시작하고 7로 끝나는 길이가 n 인 동민 수열의 개수를 $d47[n]$
 - 7로 시작하고 4로 끝나는 길이가 n 인 동민 수열의 개수를 $d74[n]$
 - 7로 시작하고 7로 끝나는 길이가 n 인 동민 수열의 개수를 $d77[n]$
-
- $d47[n] = d44[n-1] * c47 + d47[n] * c77$

동민 수열

<https://www.acmicpc.net/problem/1529>

- 그런데, 수열의 길이가 너무 길다.

동민 수열

<https://www.acmicpc.net/problem/1529>

- 그런데, 수열의 길이가 너무 길다.
- 다음과 같은 행렬을 만들어서 수열의 개수를 구할 수 있다.
- $A = \begin{pmatrix} c_{44} & c_{47} \\ c_{74} & c_{77} \end{pmatrix}$
- 정답은?

동민 수열

75

<https://www.acmicpc.net/problem/1529>

- 그런데, 수열의 길이가 너무 길다.
- 다음과 같은 행렬을 만들어서 수열의 개수를 구할 수 있다.
- $A = \begin{pmatrix} c_{44} & c_{47} \\ c_{74} & c_{77} \end{pmatrix}$
- 정답은?
- A^N 의 모든 값을 더한다

동민 수열

76

<https://www.acmicpc.net/problem/1529>

- 소스: <http://boj.kr/30e6281866f0495e807bd60361121780>