

다이나믹 프로그래밍 4 (도전)

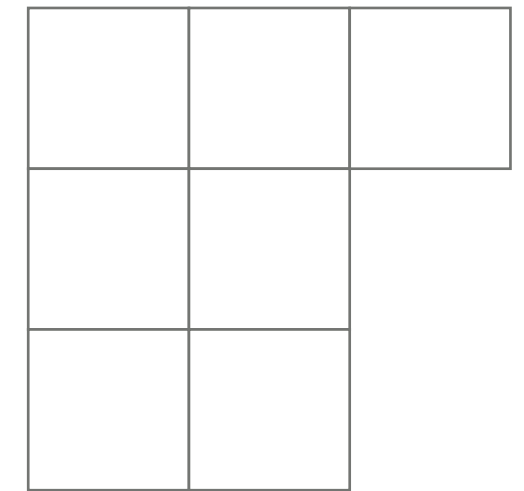
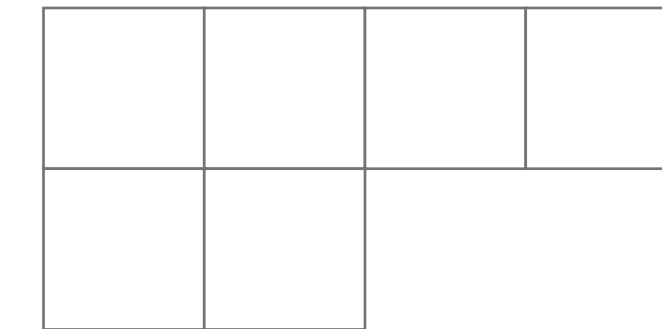
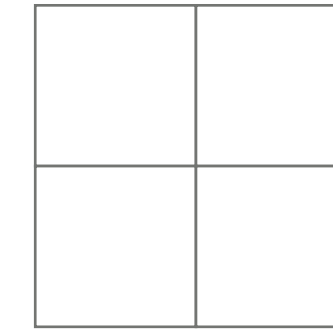
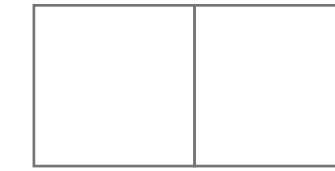
최백준 choi@startlink.io

상태 다이나믹

다이어그램과 태블로

<https://www.acmicpc.net/problem/9646>

3



- 영 다이어그램

1. 박스는 각각의 행과 열에 대해서 연속적이어야 한다.
2. 모든 행은 모두 가장 왼쪽을 기준으로 정렬되어 있어야 한다.
3. 각각의 행은 바로 위에 있는 행보다 길 수 없다.

다이어그램과 태블로

<https://www.acmicpc.net/problem/9646>

- 영 태블로

1. 각각의 박스에는 1과 N을 포함하는 그 사이의 정수가 채워져 있다.
2. 각 박스에 적혀있는 정수는 왼쪽에 있는 정수보다 크거나 같아야 한다.
3. 각 박스에 적혀있는 정수는 위에 있는 정수보다 커야 한다.

1	1
2	

1	1
3	

1	2
2	

1	2
3	

1	3
2	

1	3
3	

2	2
3	

2	3
3	

다이어그램과 태블로

<https://www.acmicpc.net/problem/9646>

- 영 다이어그램이 주어졌을 때, 영 태블로를 만드는 방법의 수를 구하는 문제
- $1 \leq \text{행의 개수 } k \leq 7$
- 각 행에 있는 박스의 개수 l_1, l_2, \dots, l_k
- $7 \geq l_1 \geq l_2 \geq \dots \geq l_k \geq 1$
- $k \leq N \leq 7$

다이어그램과 태블로

<https://www.acmicpc.net/problem/9646>

6

- 영 태블로

1. 각각의 박스에는 1과 N을 포함하는 그 사이의 정수가 채워져 있다.
2. 각 박스에 적혀있는 정수는 왼쪽에 있는 정수보다 크거나 같아야 한다.
3. 각 박스에 적혀있는 정수는 위에 있는 정수보다 커야 한다.

1	1
2	

1	1
3	

1	2
2	

1	2
3	

1	3
2	

1	3
3	

2	2
3	

2	3
3	

다이어그램과 태블로

<https://www.acmicpc.net/problem/9646>

- 각각의 열에 대해서 같은 수는 올 수 없다.
- 또, 각 열에 대해서, 수는 아래로 갈수록 증가해야 한다.
- 따라서, 각 열에 등장한 수를 상태로 나타낼 수 있다.

다이어그램과 테이블로

<https://www.acmicpc.net/problem/9646>

- $D[c][s]$ = c번 열에 수의 상태가 s일 때, 경우의 수
- $D[c][s] = \sum D[c-1][ps]$

다이어그램과 태블로

<https://www.acmicpc.net/problem/9646>

- m = 열의 개수
- $b[c]$ = c 번 열에 있는 박스의 개수
- $\text{bit_count}(s)$ = s 에 있는 비트 1의 개수
- $\text{convert}(s)$ = 비트 마스크 s 를 배열로 바꿈

다이어그램과 테이블로

<https://www.acmicpc.net/problem/9646>

```
for (int s=0; s<(1<<n); s++) {  
    int cnt = bit_count(s);  
    if (cnt != b[0]) continue;  
    d[0][s] = 1;  
}
```

다이어그램과 테이블로

<https://www.acmicpc.net/problem/9646>

```
for (int j=1; j<m; j++) {
    for (int s=0; s<(1<<n); s++) {
        int cnt = bit_count(s); if (cnt != b[j]) continue;
        for (int ps=0; ps<(1<<n); ps++) {
            int pcnt = bit_count(ps); if (pcnt != b[j-1]) continue;
            vector<int> v1 = convert(ps), v2 = convert(s);
            bool ok = true;
            for (int i=0; i<v2.size(); i++) {
                if (v1[i] > v2[i]) ok = false;
            }
            if (ok == false) continue;
            d[j][s] += d[j-1][ps];
        }
    }
}
```

다이어그램과 테이블로

<https://www.acmicpc.net/problem/9646>

```
for (int s=0; s<(1<<n); s++) {  
    int cnt = bit_count(s);  
    if (cnt != b[0]) continue;  
    d[0][s] = 1;  
}
```

다이어그램과 태블로

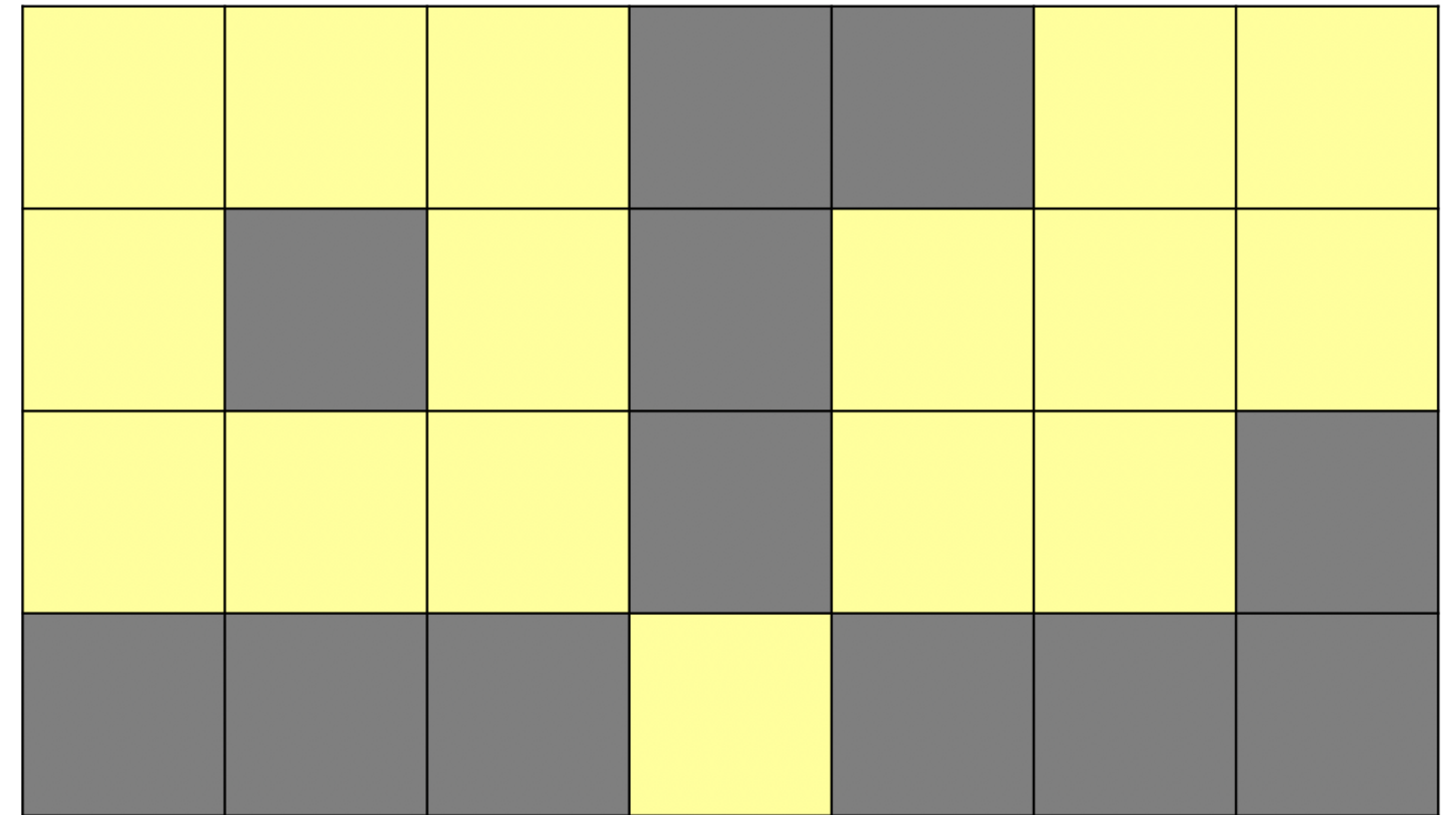
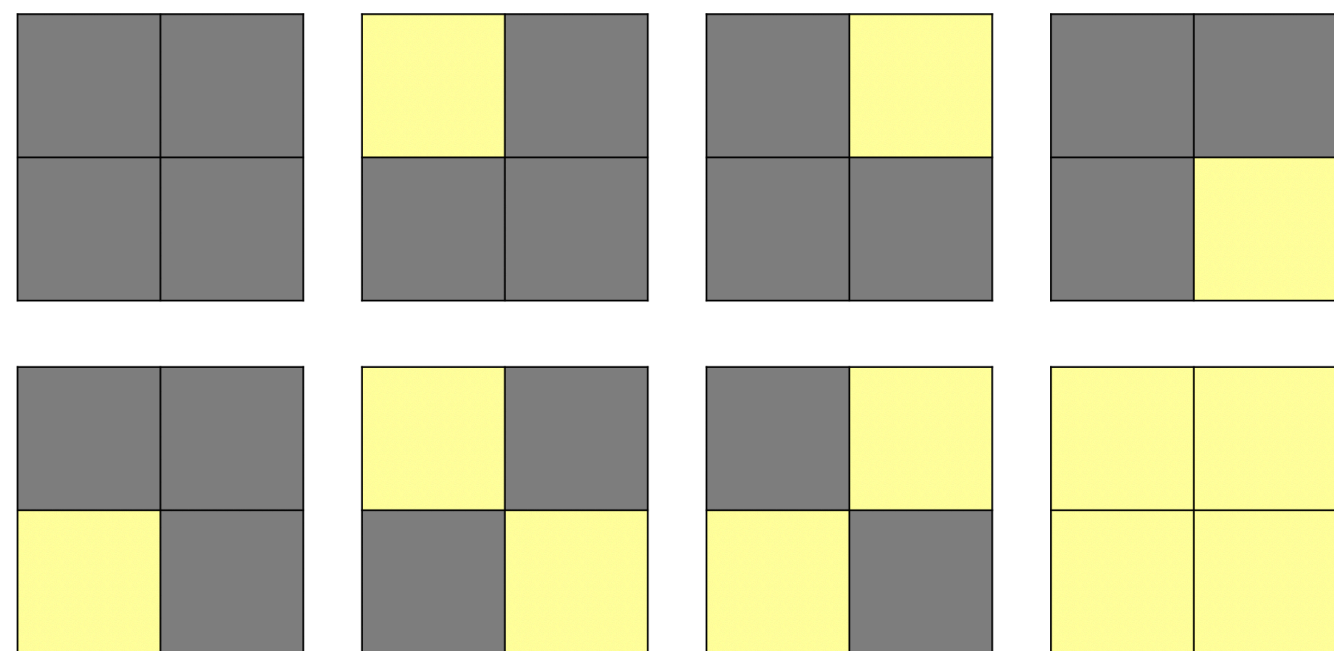
<https://www.acmicpc.net/problem/9646>

- 소스: <http://codeplus.codes/e6717b8c15724b8e80bbc0ada40819e8>

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

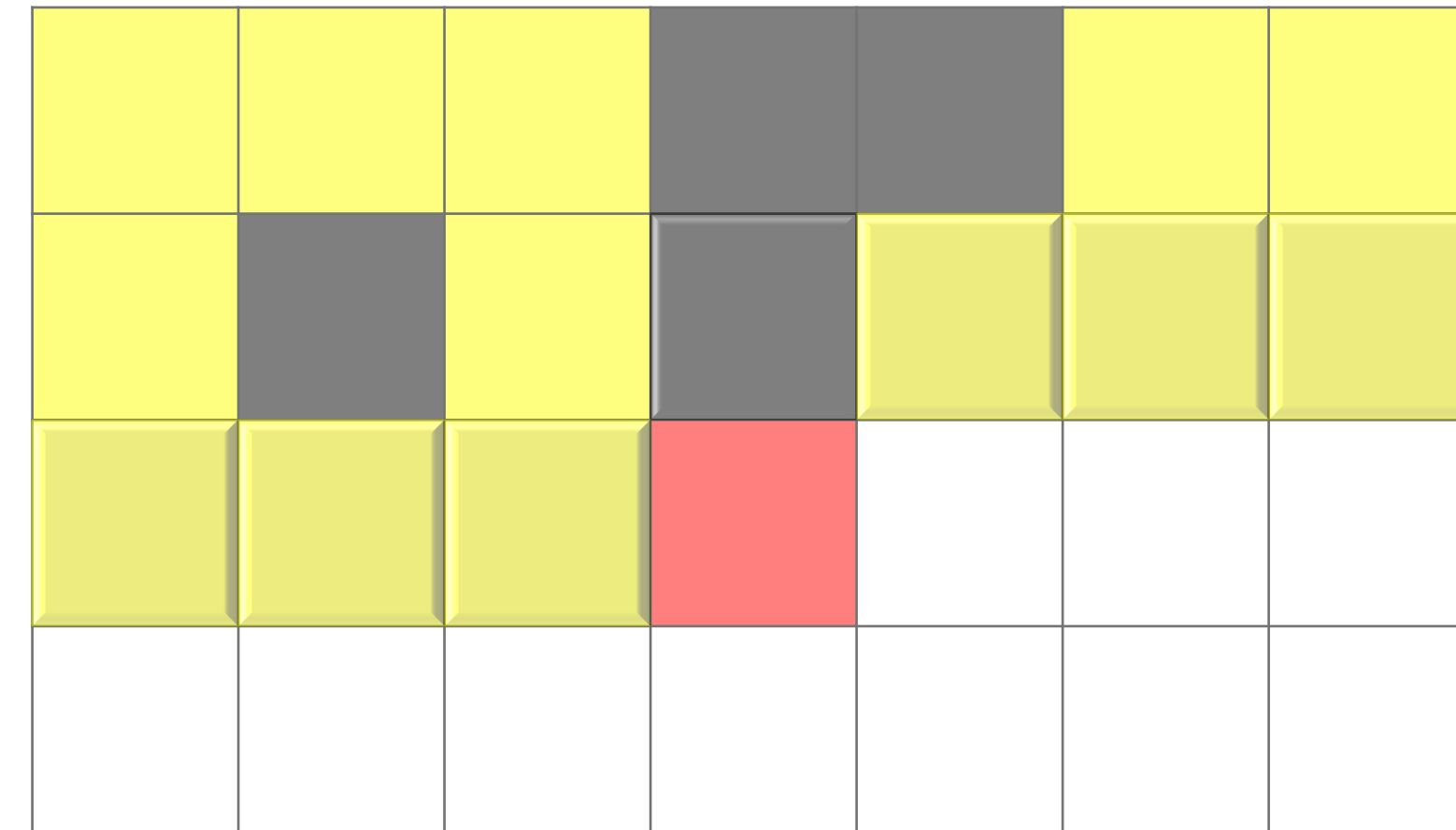
- 단위 정사각형으로 나누어져 있는 $N \times M$ 크기의 직사각형이 있을 때 $1 \leq N \leq 100, 1 \leq M \leq 8$
- 아래 조건을 만족하게 색칠하는 방법의 수를 구하는 문제
- 1. 모든 칸은 색칠되어 있거나 비어있다.
- 2. 색칠된 칸과 인접한 색칠된 칸의 개수는 짝수개
- $N = 2, M = 20$ 이면 8가지 경우가 가능하다



직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 격자판 채우기와 비슷한 문제이다.
- 어떤 칸을 색칠하거나 색칠하지 않으면
- M개의 칸이 영향을 받는다.
- 각 칸은 총 몇 개의 상태가 있을까?



직사각형 색칠

<https://www.acmicpc.net/problem/14275>

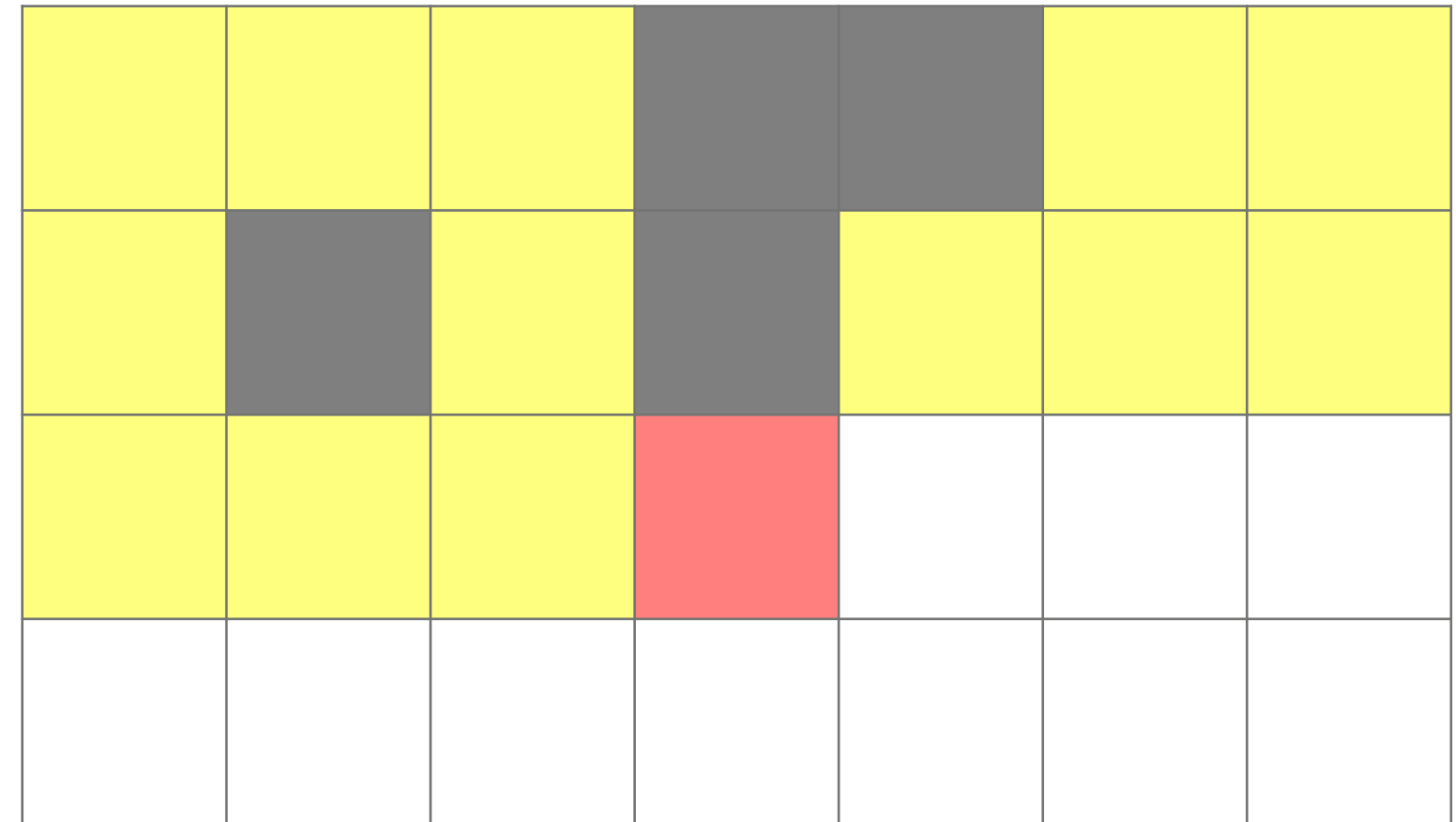
- 격자판 채우기와 비슷한 문제이다.
- 어떤 칸을 색칠하거나 색칠하지 않으면
- M개의 칸이 영향을 받는다.
- 각 칸은 총 몇 개의 상태가 있을까?
- 3개의 상태가 있다.
- 0: 색칠되어 있고, 인접한 색칠된 칸의 개수는 짝수
- 1: 색칠되어 있고, 인접한 색칠된 칸의 개수는 홀수
- 2: 색칠되어 있지 않다 (비어있는 칸)

			2	1	1	0
0	0	0				

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

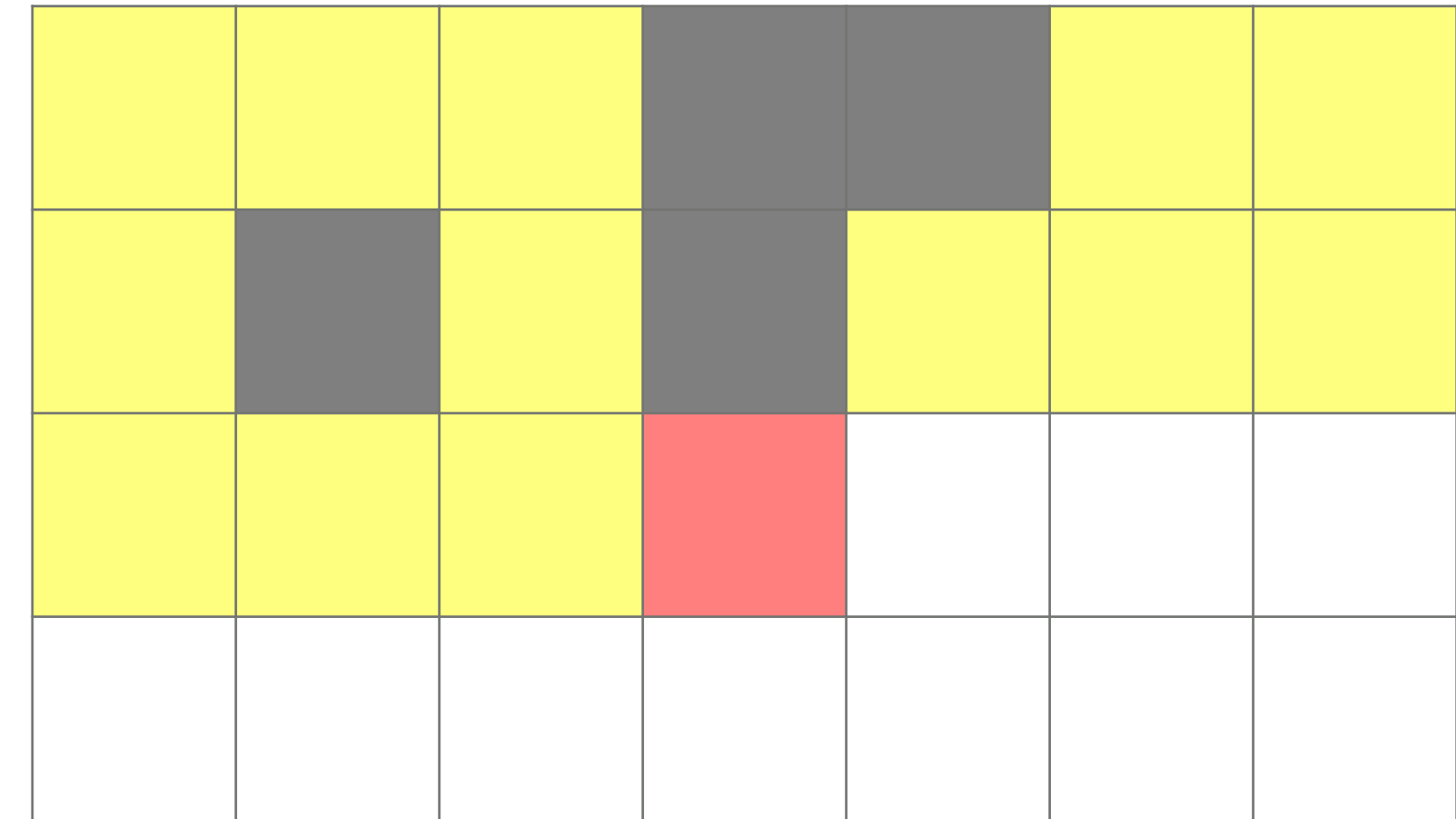
- 빨간 칸을 색칠하거나 색칠하지 않을 때
- 영향을 받는 칸은 2개가 있다. (위, 왼쪽)



직사각형 색칠

<https://www.acmicpc.net/problem/14275>

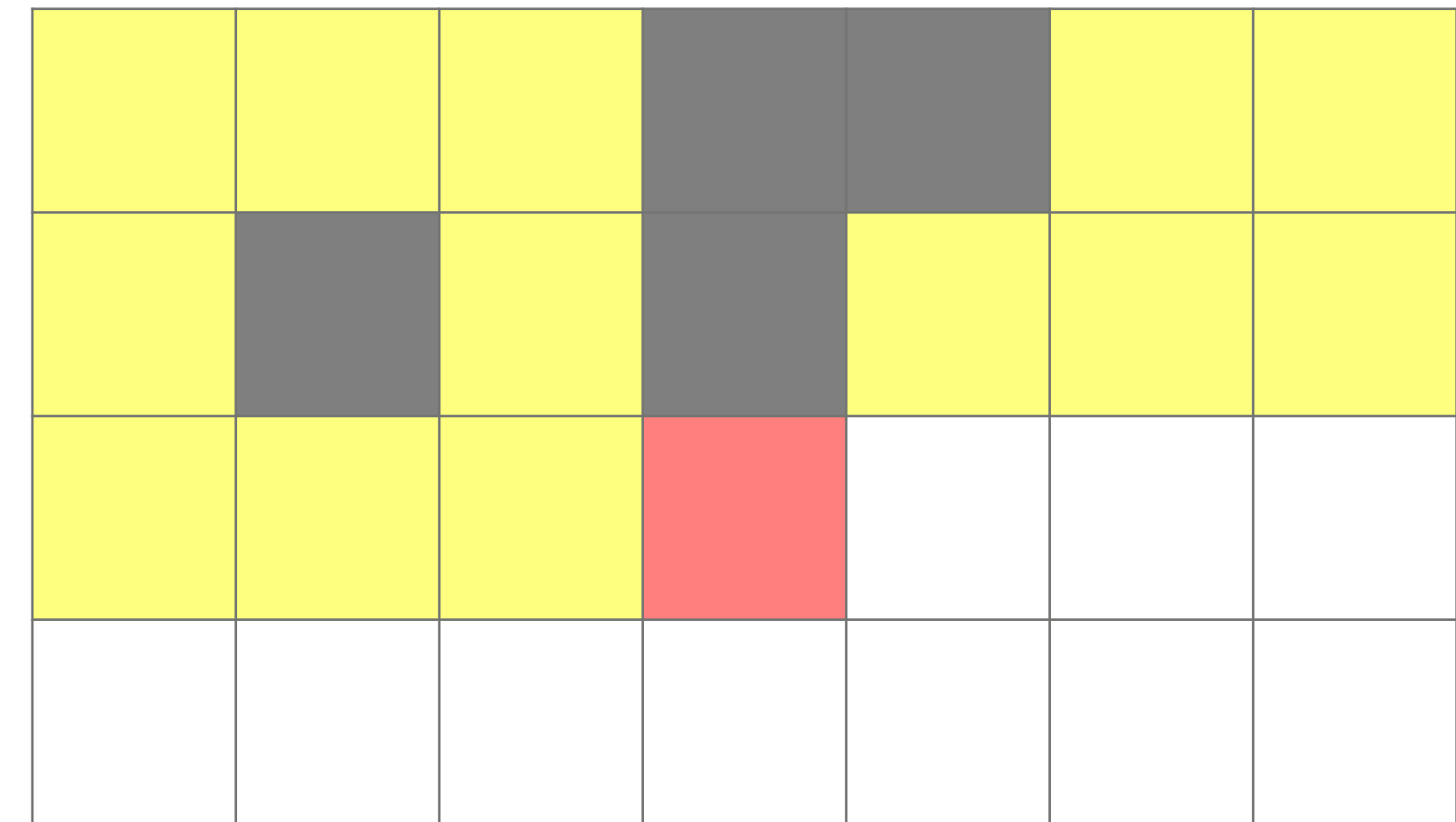
- 위 칸이 비어있는 경우에는 현재 칸(빨간색)을 색칠하는 것과
- 색칠하지 않는 것 모두 가능하다.
- 현재 칸을 색칠하면 왼쪽 칸의 상태가 변경되어야 한다.
- 현재 칸을 색칠하지 않으면 왼쪽 칸의 상태는 그대로 유지하면 된다.



직사각형 색칠

<https://www.acmicpc.net/problem/14275>

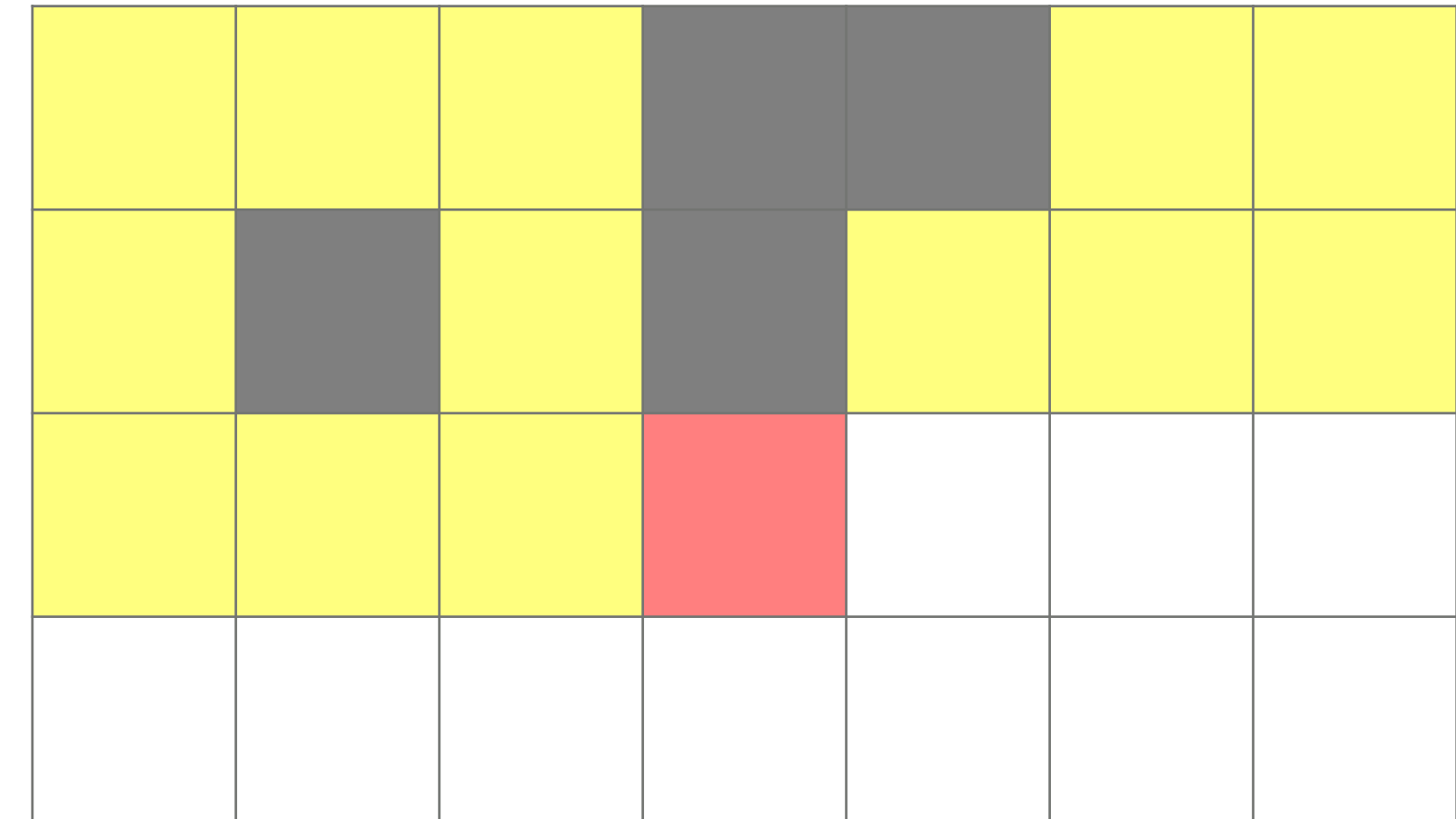
- 위 칸이 색칠되어 있는 경우에는
- 현재 칸(빨간색)을 색칠하는 것과 색칠하지 않는 것 중에서
- 하나만 할 수 있다.
- 이 선택은 위 칸의 상태에 의해서 결정된다.
- 위 칸과 인접한 색칠된 칸이 홀수 개면 현재 칸은 색칠해야 하고
- 색칠된 칸이 짝수 개면 현재 칸을 색칠하면 안된다.



직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- $go(num, state) = \text{num번 칸을 채우고, 이전 } M \text{개 칸의}$
- $\text{상태가 } state \text{일 때 채우는 방법의 수}$
- 위 칸의 상태 up 은 $(state / \text{pow3}[m-1]) \% 3$ 로 구할 수 있다.
- 왼쪽 칸의 상태 $left$ 는 $(state / \text{pow3}[0]) \% 3$ 로 구할 수 있다.



직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 현재 칸을 색칠했을 때 왼쪽 칸의 변경된 상태 cleft
- $\text{left} == 2$ 라면 (왼쪽 칸이 비어있으면)
 - $\text{cleft} = 2$
- $\text{left} == 0$ 이라면 (왼쪽 칸이 색칠되어 있고, 짝수 개의 색칠된 칸과 인접해 있다면)
 - $\text{cleft} = 1$ (현재 칸이 색칠되어서 왼쪽 칸은 홀수 개의 색칠된 칸과 인접해 있다)
- $\text{left} == 1$ 이라면 (왼쪽 칸이 색칠되어 있고, 홀수 개의 색칠된 칸과 인접해 있다면)
 - $\text{cleft} = 0$ (현재 칸이 색칠되어서 왼쪽 칸은 짝수 개의 색칠된 칸과 인접해 있다)

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 현재 칸을 색칠했을 때 현재 칸의 변경된 상태 cur (위 칸은 비어있다고 가정)
- $left == 2$ 라면 (왼쪽 칸이 비어있으면)
 - $cur = 0$
- $left != 2$ 이라면 (왼쪽 칸이 색칠되어 있다면)
 - $cur = 1$ (현재 칸이 홀수 개의 색칠된 칸과 인접해 있다))
- 위 칸이 색칠되어 있으면 cur 을 반대로 ($0 \rightarrow 1, 1 \rightarrow 0$) 만들면 된다.

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 현재 칸이 가장 왼쪽 칸이면 왼쪽 칸이 없기 때문에
 - `cleft = left`
 - `cur = 0`
- 이라고 볼 수 있다.

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 위 칸의 정보는 다음 상태를 구할 때 의미가 없어진다.
- 다음 상태를 구하기 위해서 위 칸의 정보와 왼쪽 칸의 정보를 제거한다.
- $nstate = (state - up * pow3[m-1] - left * pow3[0]) * 3;$

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 위 칸이 비어있으면, 현재 칸은 색칠할 수도 있고, 하지 않을 수도 있다.

```
if (up == 2) { // up: empty
    // cur: empty
    ans += go(num+1, nstate + left*pow3[1] + 2);
    // cur: colored
    ans += go(num+1, nstate + cleft*pow3[1] + cur);
}
```

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 위 칸이 비어있지 않으면, 위 칸의 상태에 따라서 색칠할지 말지 결정된다.

```
else { // up: colored
    cur = 1-cur;
    if (up == 0) { // up: even
        // cur: empty
        ans += go(num+1, nstate + left*pow3[1] + 2);
    } else { // up: odd
        // cur: colored
        ans += go(num+1, nstate + cleft*pow3[1] + cur);
    }
}
```

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 시간 복잡도: $O(NM3^M)$
- 가장 첫 칸의 위는 모두 비어있다고 가정하면 된다.

직사각형 색칠

<https://www.acmicpc.net/problem/14275>

- 소스: <http://codeplus.codes/f65cb1ac92a64ab2961ec013a27aeadb>

도로 건설

<https://www.acmicpc.net/problem/14276>

- N개의 도시가 있고, 1번부터 N번까지 번호가 매겨져 있다. 도로는 없다.
- 아래 조건을 지키면서 총 M개의 양방향 도로를 만들어야 한다. 양방향 도로는 두 집을 연결한다.
- 1. 서로 다른 두 집 A와 B가 있을 때, $0 < |A-B| \leq K$ 를 만족해야 도로를 연결할 수 있다. 같은 집의 쌍에 대해서 여러 개의 도로를 만들 수 있다.
- 2. 모든 집은 짝수개의 도로와 인접해야 한다. (0도 짝수)
- N, M, K가 주어졌을 때, 도로를 만드는 방법의 수
- $1 \leq N \leq 30, 0 \leq M \leq 30, 1 \leq K \leq 8$

도로 건설

<https://www.acmicpc.net/problem/14276>

- 번호가 높은 집에서 낮은 집으로 도로를 연결한다고 해도 된다.

도로 건설

<https://www.acmicpc.net/problem/14276>

- $d[i][j][state]$
 - i 번 집의 도로를 연결하려고 한다. ($i-1$ 번 집까지는 도로를 모두 연결한 상태)
 - 현재 j 개의 도로를 연결
 - i 번 집부터 K 개의 집이 연결된 도로의 상태는 $state$ (짝수면 0, 홀수면 1)

도로 건설

<https://www.acmicpc.net/problem/14276>

- $d[i][j][state]$
 - i 번 집의 도로를 연결하려고 한다. ($i-1$ 번 집까지는 도로를 모두 연결한 상태)
 - 현재 j 개의 도로를 연결
 - i 번 집부터 K 개의 집이 연결된 도로의 상태는 $state$ (짝수면 0, 홀수면 1)
- i 번 집에서 도로를 연결할 수 있는 집의 번호는 $i-1, i-2, \dots, i-K$ 이다.
- 이 집 중에서 어떤 집에 도로를 각각 몇 개 연결해야 하는지 구해야 한다.

도로 건설

<https://www.acmicpc.net/problem/14276>

- 예를 들어, $K = 2$ 이고, 3번 집에 도로를 연결해야 하는 경우에
- 3번-2번, 3번-1번을 순서대로 연결하는 경우와 3번-1번, 3번-2번을 순서대로 연결하는 경우를 다른 경우라고 해야 한다.
- 앞의 점화식에서는 이것을 처리하기 매우 어렵기 때문에
- 변수 하나를 더 추가해야 한다.

도로 건설

<https://www.acmicpc.net/problem/14276>

- $d[i][j][cur][state]$
 - i 번 집의 도로를 연결하려고 한다. ($i-1$ 번 집까지는 도로를 모두 연결한 상태)
 - 현재 j 개의 도로를 연결
 - i 번 집과 $i-(cur+1)$ 번 집에 도로를 연결하는 것을 고려하는 중
 - i 번 집부터 K 개의 집이 연결된 도로의 상태는 $state$ (짝수면 0, 홀수면 1)
- i 번 집에서 도로를 연결할 수 있는 집의 번호는 $i-1, i-2, \dots, i-K$ 이다.
- 이 집 중에서 어떤 집에 도로를 각각 몇 개 연결해야 하는지 구해야 한다.

도로 건설

<https://www.acmicpc.net/problem/14276>

- $d[i][j][cur][state]$
 - i 번 집의 도로를 연결하려고 한다. ($i-1$ 번 집까지는 도로를 모두 연결한 상태)
 - 현재 j 개의 도로를 연결
 - i 번 집과 $i-(cur+1)$ 번 집에 도로를 연결하는 것을 고려하는 중
 - i 번 집부터 K 개의 집이 연결된 도로의 상태는 $state$ (짝수면 0, 홀수면 1)
- $cur = K$ 인 경우: i 번 집에서 연결할 수 있는 모든 도로의 상태를 계산한 것
 - 경우의 수: $d[i+1][j][0][state \ll 1]$
 - 이때 $i-K$ 번은 더 이상 도로와 연결할 수 없기 때문에, $i-K$ 번 집의 도로 수가 짝수인 것을 확인해야 한다.

도로 건설

<https://www.acmicpc.net/problem/14276>

- $d[i][j][cur][state]$
 - i 번 집의 도로를 연결하려고 한다. ($i-1$ 번 집까지는 도로를 모두 연결한 상태)
 - 현재 j 개의 도로를 연결
 - i 번 집과 $i-(cur+1)$ 번 집에 도로를 연결하는 것을 고려하는 중
 - i 번 집부터 K 개의 집이 연결된 도로의 상태는 $state$ (짝수면 0, 홀수면 1)
- $cur < K$ 인 경우: i 번 집과 $i-(cur+1)$ 집의 도로를 연결해야 함
 - 도로를 건설하는 경우: $d[i][j+1][cur][state \oplus (1 \ll 0) \oplus (1 \ll (cur+1))]$
 - 도로를 건설하지 않는 경우: $d[i][j][cur+1][state]$

도로 건설

<https://www.acmicpc.net/problem/14276>

- 소스: <http://codeplus.codes/f1775b58db9b46e7a484e651ddfbe425>