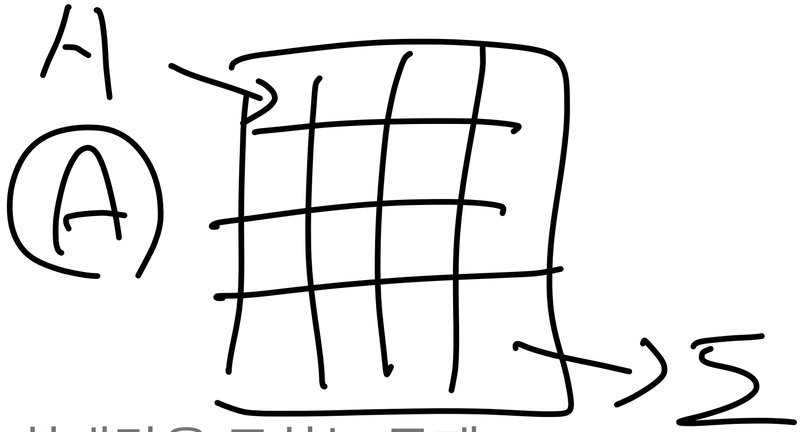
다이나믹 프로그래밍 3 (연습)

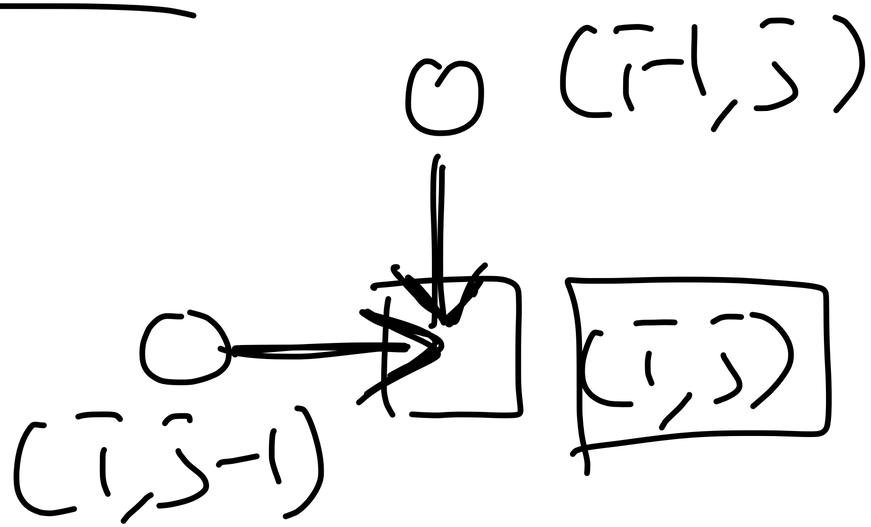
최백준 choi@startlink.io

로봇 조종하기

https://www.acmicpc.net/problem/2169

- (1, 1)에서 (N, M)으로 이동하려고 한다
- 왼쪽, 오른쪽, 아래로만 이동할 수 있다. 합의 최대값을 구하는 문제





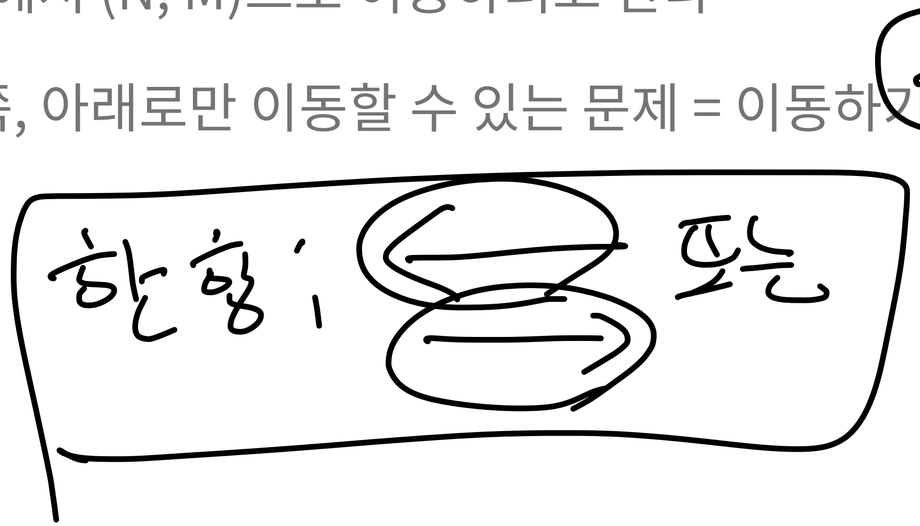
$DP[i][j]$ = $(1,1) \rightarrow (i,j)$ 최대

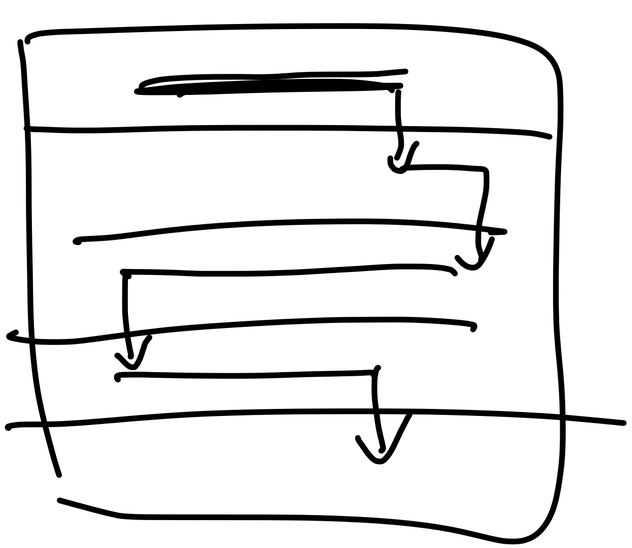
$\max(DP[i-1][j], DP[i][j-1]) + A[i][j]$

로봇 조종하기

https://www.acmicpc.net/problem/2169

- (1, 1)에서 (N, M)으로 이동하려고 한다
- 오른쪽, 아래로만 이동할 수 있는 문제 = 이동하기 2





로봇 조종하기

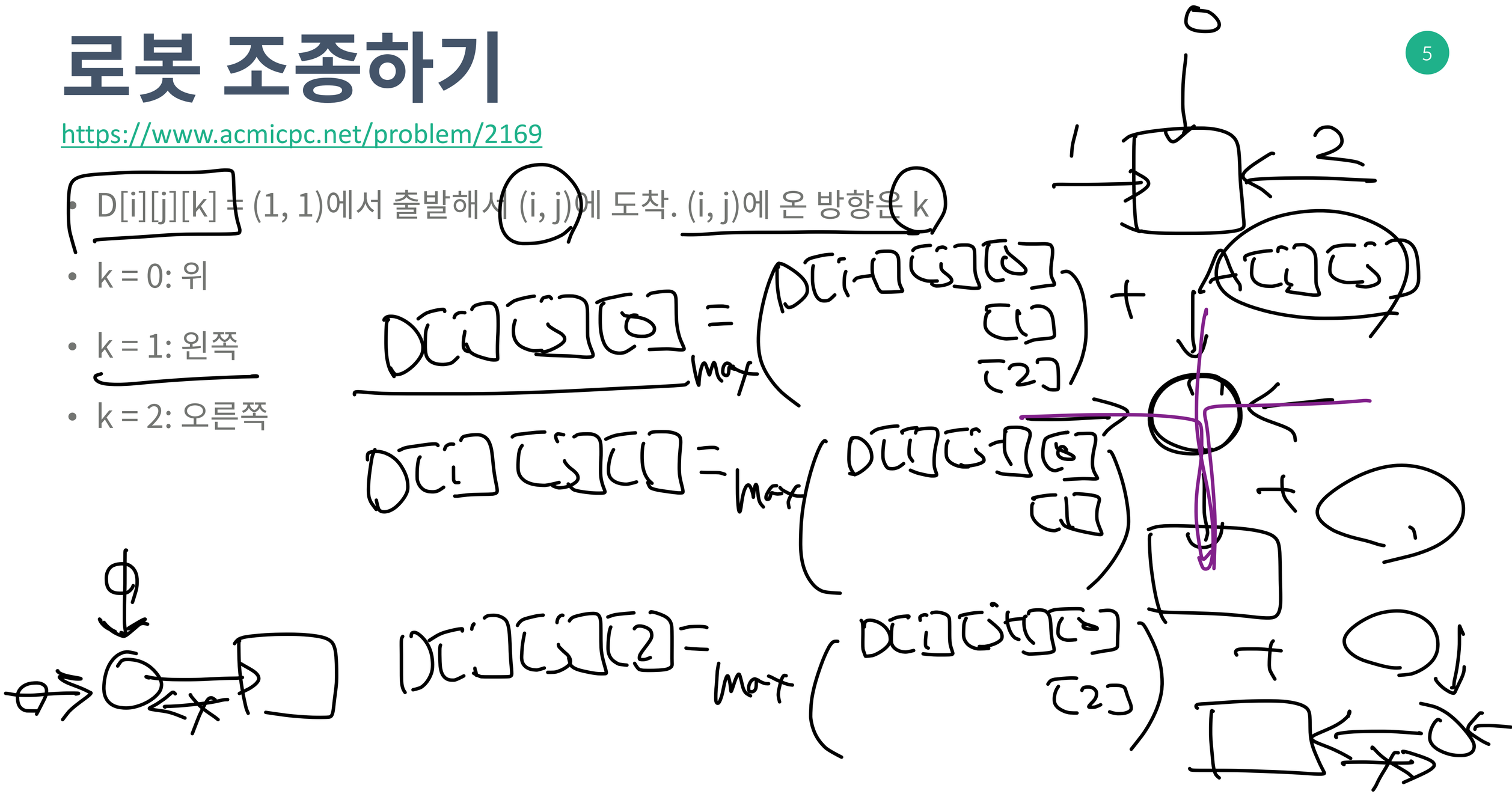
https://www.acmicpc.net/problem/2169

- 가장 윗 행부터 차례대로 처리를 하면 된다
- 위, 왼쪽, 오른쪽 순서대로 처리를 해야 한다

로봇 조종하기

https://www.acmicpc.net/problem/2169

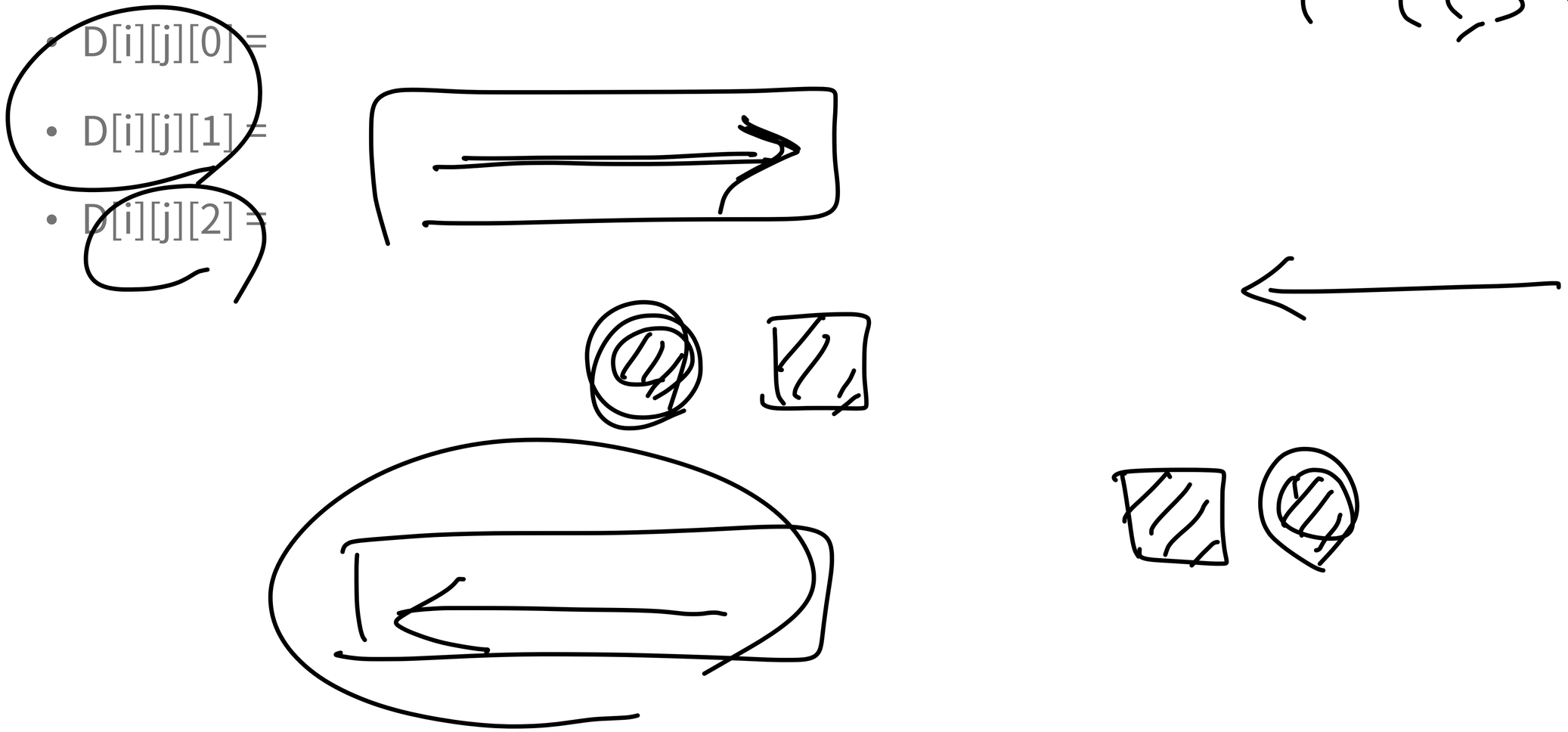
- D[i][j][k] = (1, 1)에서 출발해서 (i, j)에 도착. (i, j)에 온 방향은 k
- k = 0: 위
- k = 1: 왼쪽
- k = 2: 오른쪽



로봇 조종하기

https://www.acmicpc.net/problem/2169

- D[i][j][0] =
- D[i][j][1] =
- D[i][j][2] =



로봇 조종하기

https://www.acmicpc.net/problem/2169

- D[i][j][0] = max(D[i-1][j][0], D[i-1][j][1], D[i-1][j][2]) + A[i][j];
- D[i][j][1] = max(D[i][j-1][0], D[i][j-1][1]) + A[i][j];
- D[i][j][2] = max(D[i][j+1][0], D[i][j+1][2]) + A[i][j];

로봇 조종하기

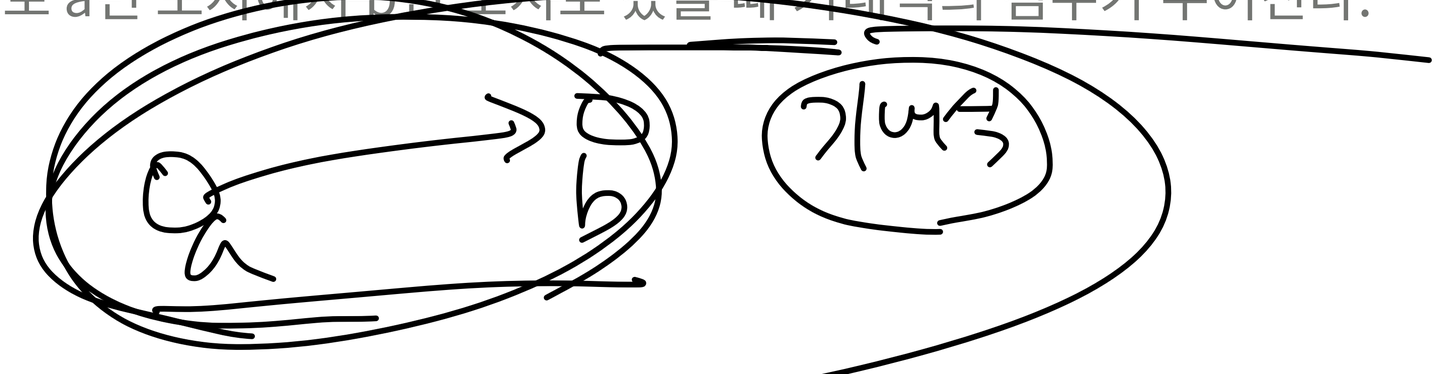
https://www.acmicpc.net/problem/2169

- 소스: http://codeplus.codes/01b30e6c28ba45f3af8acdc3e151deff

여행

https://www.acmicpc.net/problem/2157

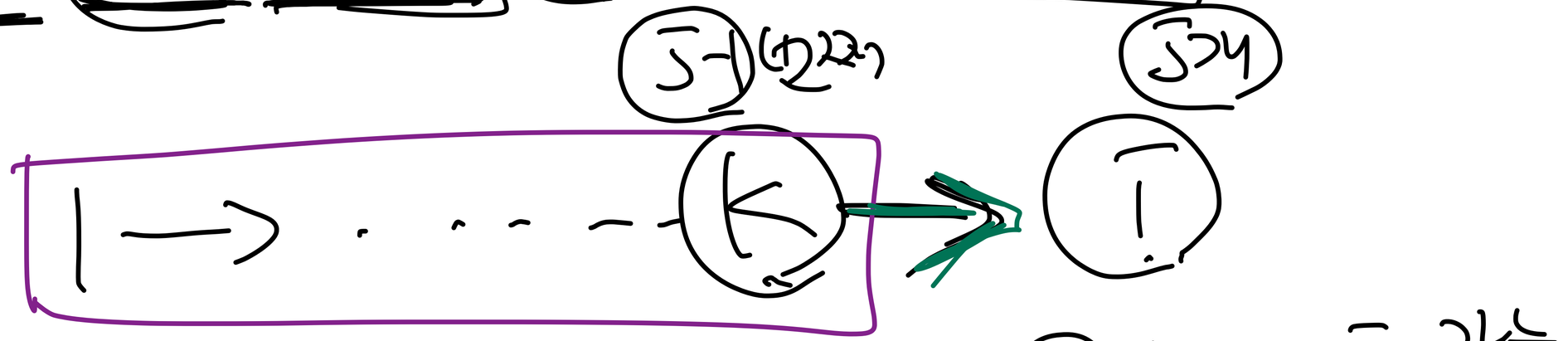
- N개의 도시가 동쪽에서 서쪽으로 순서대로 위치해 있다 $1 \leq N \leq 300$
- 제일 동쪽에 있는 도시는 1번 도시이며, 제일 서쪽에 있는 도시는 N번 도시이다
- M개 이하의 도시를 지나는 여행을 계획하려 한다
- 여행 경로는 반드시 1번 도시에서 시작해서 N번 도시에서 끝나야 한다
- 도시 번호가 증가하는 순서대로만 이동
- 최대한 맛있는 기내식만 먹으면서 이동
- 입력으로 a번 도시에서 b번 도시로 갔을 때 기내식의 점수가 주어진다.

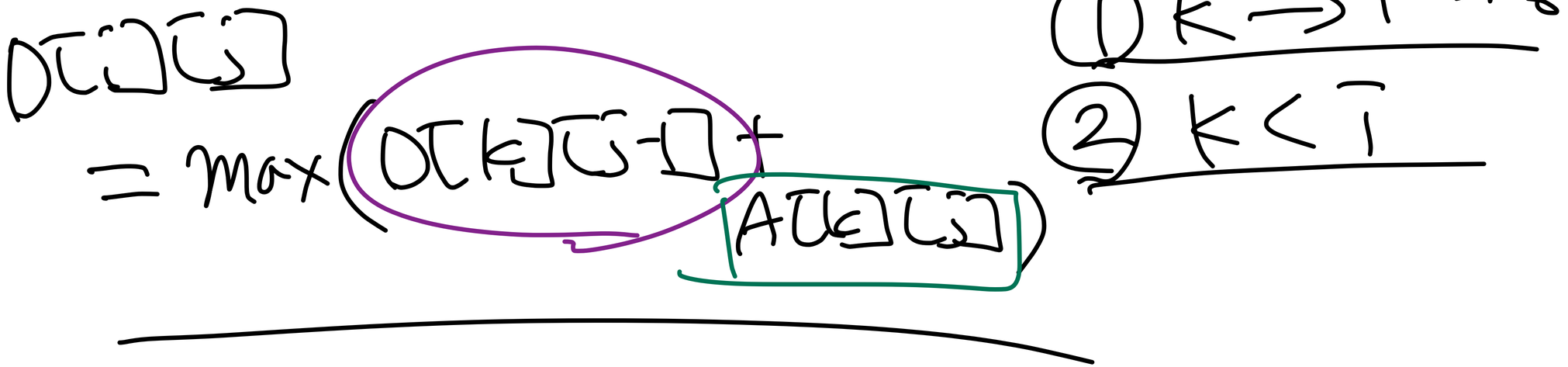


여행

https://www.acmicpc.net/problem/2157

- D[i][j] = i번 도시에 j개의 도시를 거쳐서 도착했을 때, 기내식의 최대값

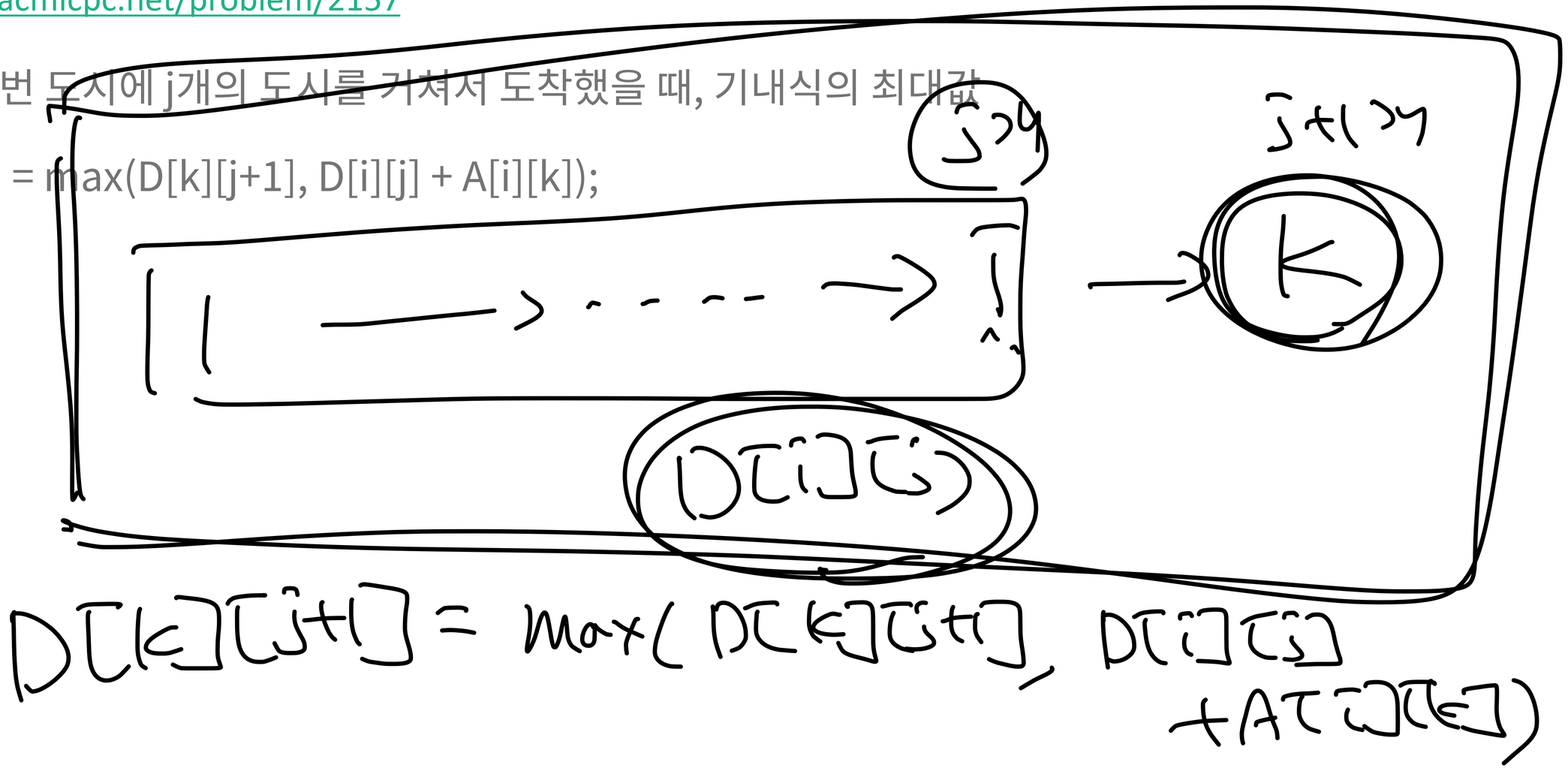




여행

https://www.acmicpc.net/problem/2157

- D[i][j] = i번 도시에 j개의 도시를 거쳐서 도착했을 때, 기내식의 최대값
- D[k][j+1] = max(D[k][j+1], D[i][j] + A[i][k]);



여행

https://www.acmicpc.net/problem/2157

- 소스: http://codeplus.codes/66db0c89e76f4eeca5356251d8154579

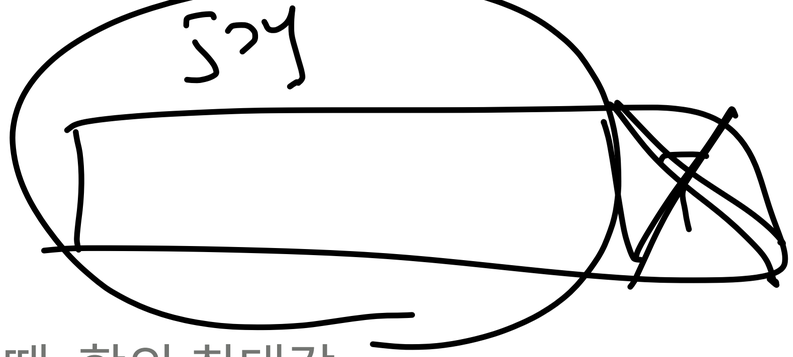
구간 나누기

https://www.acmicpc.net/problem/2228

- N(1≤N≤100)개의 수로 이루어진 1차원 배열이 있다
- 이 배열을 M(1≤M≤N/2 올림)개의 구간으로 나눠서 구간에 속한 수들의 총 합이 최대가 되도록 하려 한다
- 단, 다음의 조건들이 만족되어야 한다
  1. 각 구간은 한 개 이상의 연속된 수들로 이루어진다.
  2. 서로 다른 두 구간끼리 겹쳐있거나 인접해 있어서는 안 된다.
  3. 정확히 M개의 구간이 있어야 한다. M개 미만이어서는 안 된다.

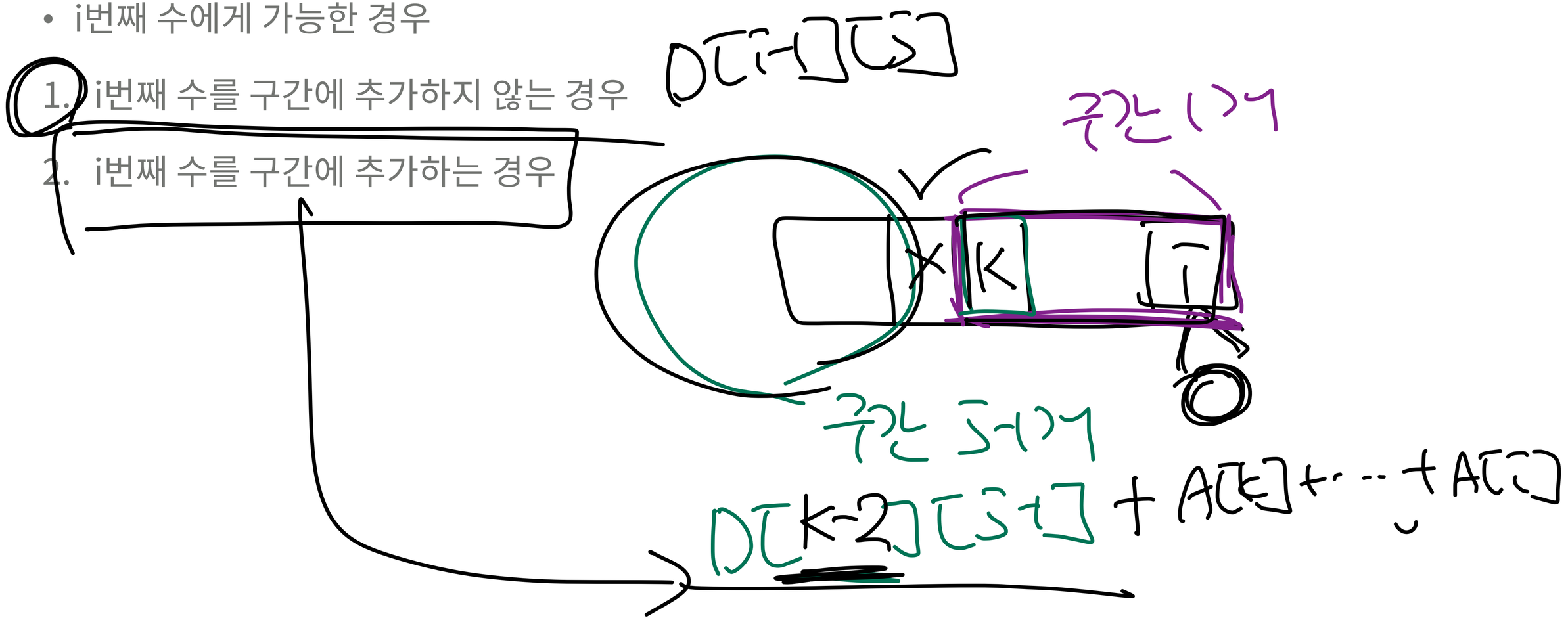
구간 나누기

https://www.acmicpc.net/problem/2228



- D[i][j] = i개의 수를 j개의 구간으로 나누었을 때, 합의 최대값
- i번째 수에게 가능한 경우

1. i번째 수를 구간에 추가하지 않는 경우
2. i번째 수를 구간에 추가하는 경우



구간 나누기

https://www.acmicpc.net/problem/2228

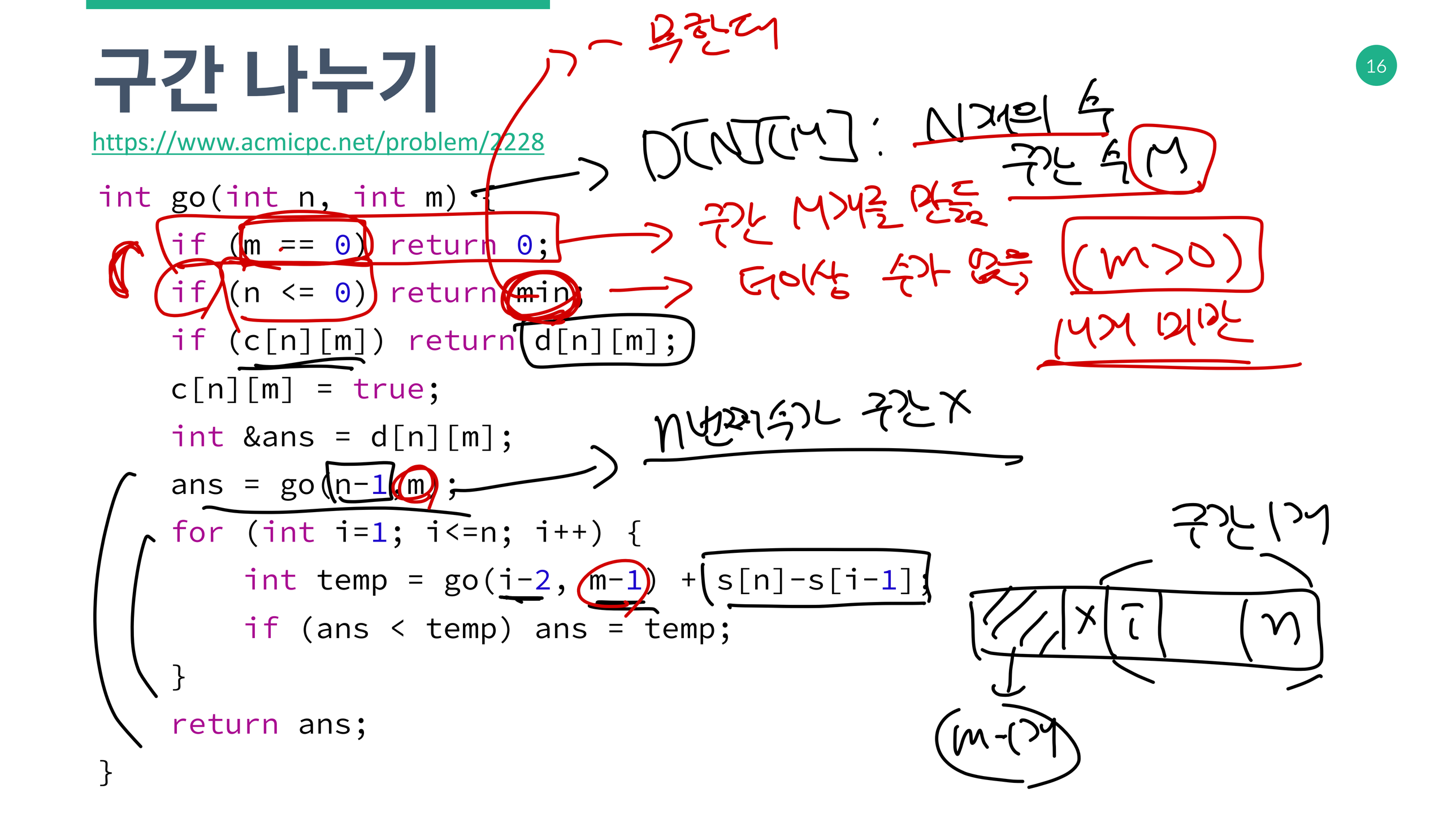
- D[i][j] = i개의 수를 j개의 구간으로 나누었을 때, 합의 최대값
- i번째 수에게 가능한 경우

1. i번째 수를 구간에 추가하지 않는 경우
    - 구간의 수: 변하지 않음 j
    - i-1개의 수를 j개의 구간으로 나누어야 함
    - D[i-1][j]
2. i번째 수를 구간에 추가하는 경우
    - i번째 수를 새로운 구간에 추가해야 함
    - 어디서부터 구간에 추가해야 할지 결정해야 함. (k번째 수 부터 구간에 추가)
    - D[k-2][j-1] + (A[k] + … + A[i]) (k-2인 이유는 붙어있으면 안되기 때문)

구간 나누기

https://www.acmicpc.net/problem/2228


```
int go(int n, int m) {
    if (m == 0) return 0;
    if (n <= 0) return min;
    if (c[n][m]) return d[n][m];
    c[n][m] = true;
    int &ans = d[n][m];
    ans = go(n-1, m);
    for (int i=1; i<=n; i++) {
        int temp = go(i-2, m-1) + s[n]-s[i-1];
        if (ans < temp) ans = temp;
    }
    return ans;
}
```



구간 나누기

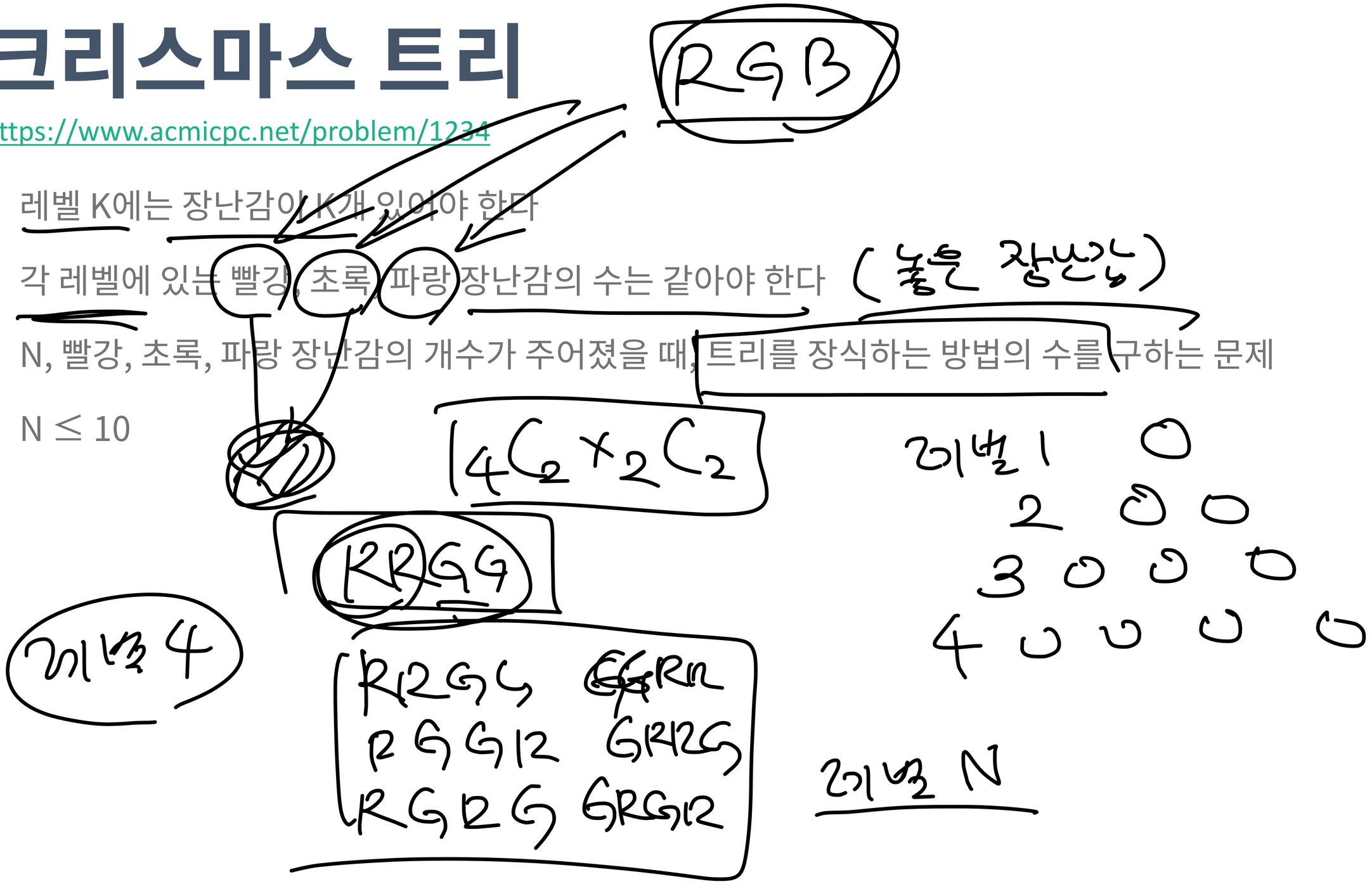
https://www.acmicpc.net/problem/2228

- 소스: http://codeplus.codes/19d3e81788214bdaa4b5b7c16771002d

크리스마스 트리

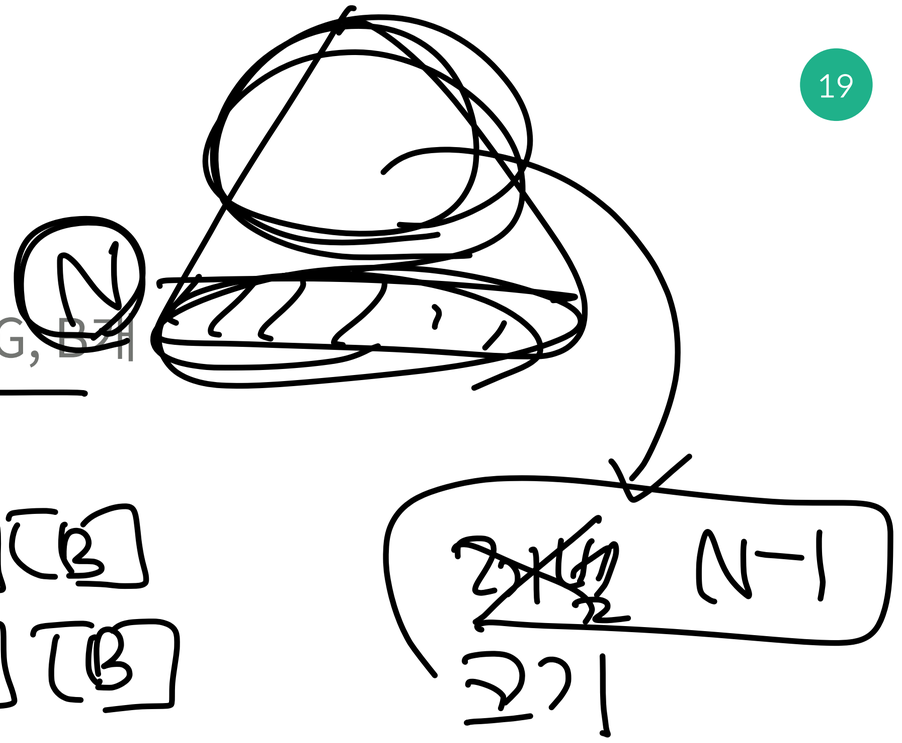
https://www.acmicpc.net/problem/1234

- 레벨 K에는 장난감이 K개 있어야 한다
- 각 레벨에 있는 빨강, 초록, 파랑 장난감의 수는 같아야 한다
- N, 빨강, 초록, 파랑 장난감의 개수가 주어졌을 때, 트리를 장식하는 방법의 수를 구하는 문제
- $N \leq 10$



크리스마스 트리

https://www.acmicpc.net/problem/1234



- D[N][R][G][B] = 크기가 N인 트리, 가지고 있는 장난감의 수 R, G, B개

① 색을 1개만 사용

① 빨: D[N-1][R-N][G][B]

② 초: D[N-1][R][G-N][B]

③ 파: D[N-1][R][G][B-N]

② 색을 2개 사용 (N이 2의 배수)

① 빨초: D[N-1][R-N/2][G-N/2][B] × NCN/2

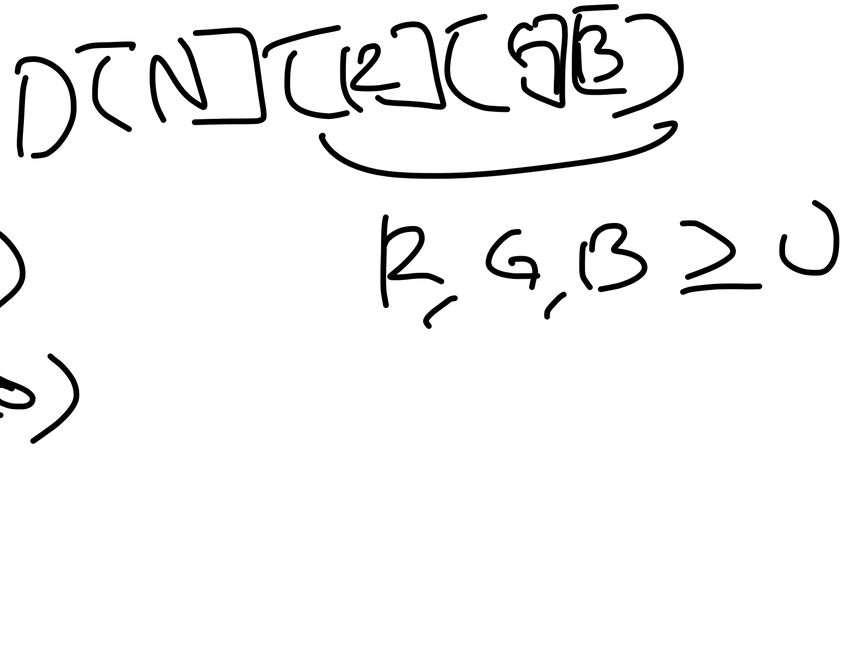
② 초파: D[N-1][R][G-N/2][B-N/2] × NCN/2

③ 빨파: D[N-1][R-N/2][G][B-N/2] × NCN/2

크리스마스 트리

https://www.acmicpc.net/problem/1234

- 색 1개를 사용하는 경우
- D[N-1][R-N][G][B]
- D[N-1][R][G-N][B]
- D[N-1][R][G][B-N]



크리스마스 트리

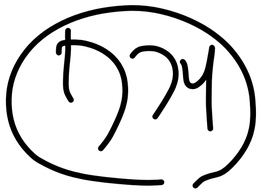
https://www.acmicpc.net/problem/1234

- 색 2개를 사용하는 경우
- D[N-1][R-N/2][G-N/2][B] * Comb[N][N/2]
- D[N-1][R-N/2][G][B-N/2] * Comb[N][N/2]
- D[N-1][R][G-N/2][B-N] * Comb[N][N/2]
- Comb[N][R] = N개 중에서 R개를 뽑는 경우의 수

nCR

크리스마스 트리

https://www.acmicpc.net/problem/1234



- 색 3개를 사용하는 경우
- D[N-1][R-N/3][G-N/3][B-N/3] * Comb[N][N/3] * Comb[N-N/3][N/3]

크리스마스 트리

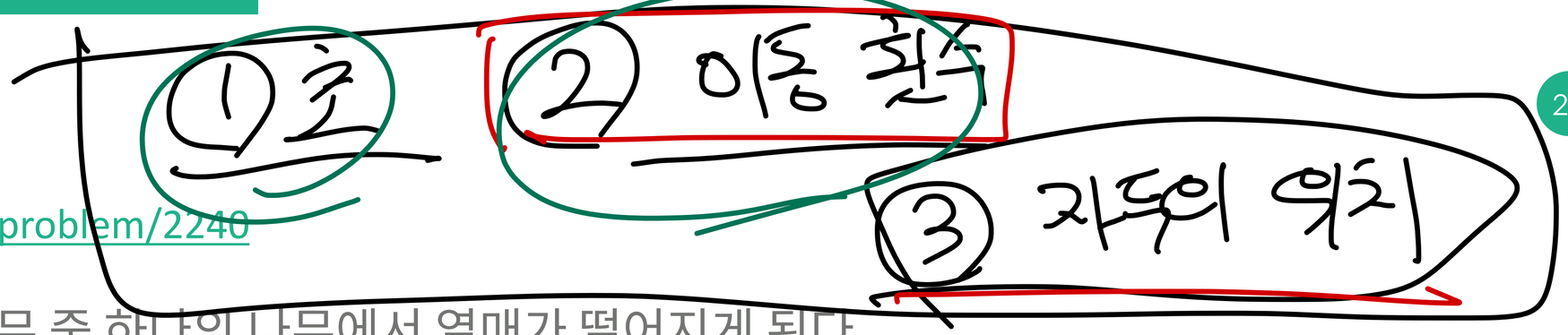
https://www.acmicpc.net/problem/1234

- 소스: http://codeplus.codes/6ce84c1242f94ecc86adebe6ea8b5d06

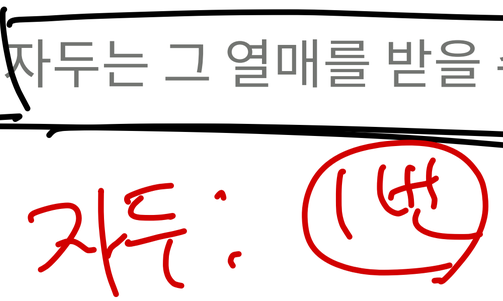
Top Down

자두나무

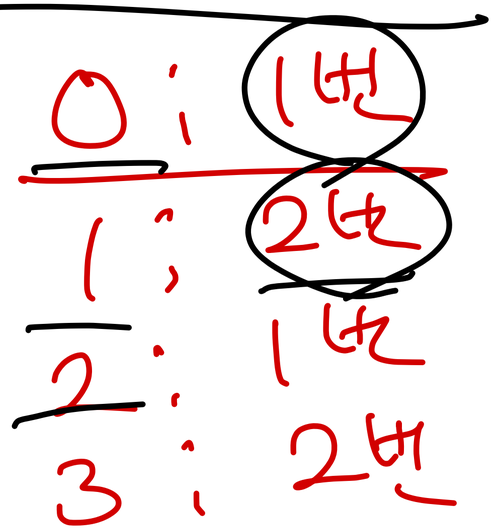
https://www.acmicpc.net/problem/2240



- 매 초마다, 두 개의 나무 중 하나의 나무에서 열매가 떨어지게 된다
- 만약 열매가 떨어지는 순간, 자두가 그 나무의 아래에 서 있으면 자두는 그 열매를 받을 수 있다
- 열매는 T(1 ≤ T ≤ 1,000)초 동안 떨어지게 된다
- 자두는 최대 W(1 ≤ W ≤ 30)번만 움직이고 싶어 한다
- 매 초마다 어느 나무에서 열매가 떨어질지에 대한 정보가 주어졌을 때, 자두가 받을 수 있는 열매 개수 최대값



2개



자두나무

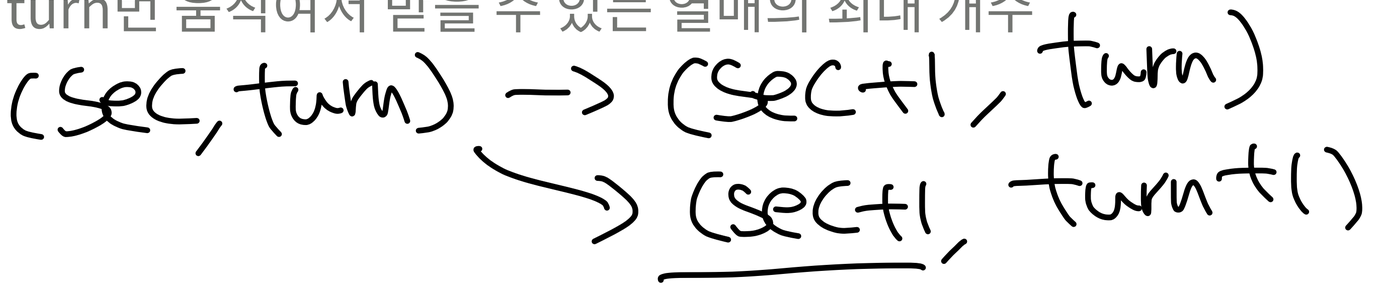
https://www.acmicpc.net/problem/2240

- D[sec][turn] = sec에 turn번 움직여서 받을 수 있는 열매의 최대 개수
- $1 \le sec \le T$
- $0 \le turn \le W$
- 지금 위치 = turn % 2 + 1번 나무

자두나무

https://www.acmicpc.net/problem/2240

- D[sec][turn] = sec에 turn번 움직여서 받을 수 있는 열매의 최대 개수
- 움직이지 않는 경우
- 움직이는 경우
- 움직이는 경우와 상관없이 위치만 같으면 열매를 받을 수 있다



자두나무

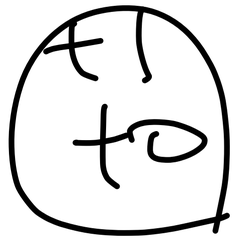
https://www.acmicpc.net/problem/2240

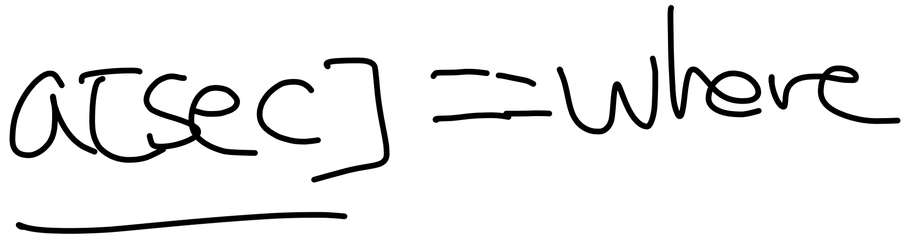
- D[sec][turn] = sec에 turn번 움직여서 받을 수 있는 열매의 최대 개수
- 움직이지 않는 경우
  - D[sec+1][turn]
- 움직이는 경우
  - D[sec+1][turn+1]

자두나무

https://www.acmicpc.net/problem/2240

- D[sec][turn] = sec에 turn번 움직여서 받을 수 있는 자두의 최대 개수
- 움직이지 않는 경우
  - D[sec+1][turn]
- 움직이는 경우
  - D[sec+1][turn+1]

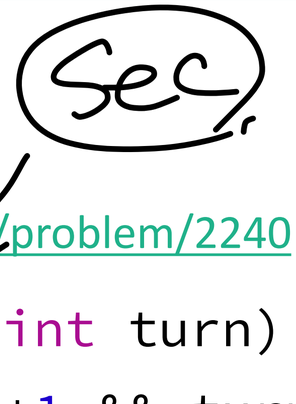


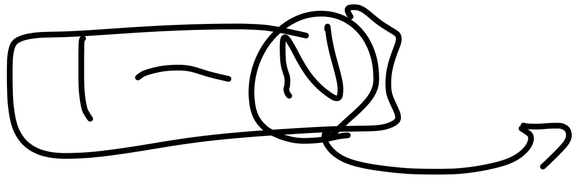


- where = turn % 2 + 1
- 열매를 받을 수 있는 경우는 a[pos] == where

자두나무

https://www.acmicpc.net/problem/2240



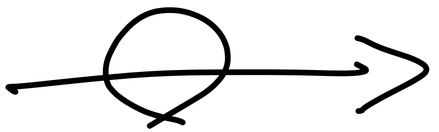


```
int go(int pos, int turn) {
    if (pos == n+1 && turn <= m) return 0;
    if (turn > m) return 0;
    if (d[pos][turn] != -1) {
        return d[pos][turn];
    }
    int where = turn % 2 + 1;
    d[pos][turn] = max(go(pos+1, turn), go(pos+1, turn+1)) + (where
== a[pos] ? 1 : 0);
    return d[pos][turn];
}
```

자두나무

https://www.acmicpc.net/problem/2240

- 가장 처음에 호출해야 하는 값은 2개이다.
- 1에서 시작하는 경우
  - go(1, 0)
- 2에서 시작하는 경우
  - go(1, 1)



```
memset(d,-1,sizeof(d));
printf("%d\n",max(go(1,0), go(1,1)));
```

자두나무

https://www.acmicpc.net/problem/2240

- 소스: http://codeplus.codes/1d9d8dd8e95f4248bd93a1ef7e74f755

숫자 박스

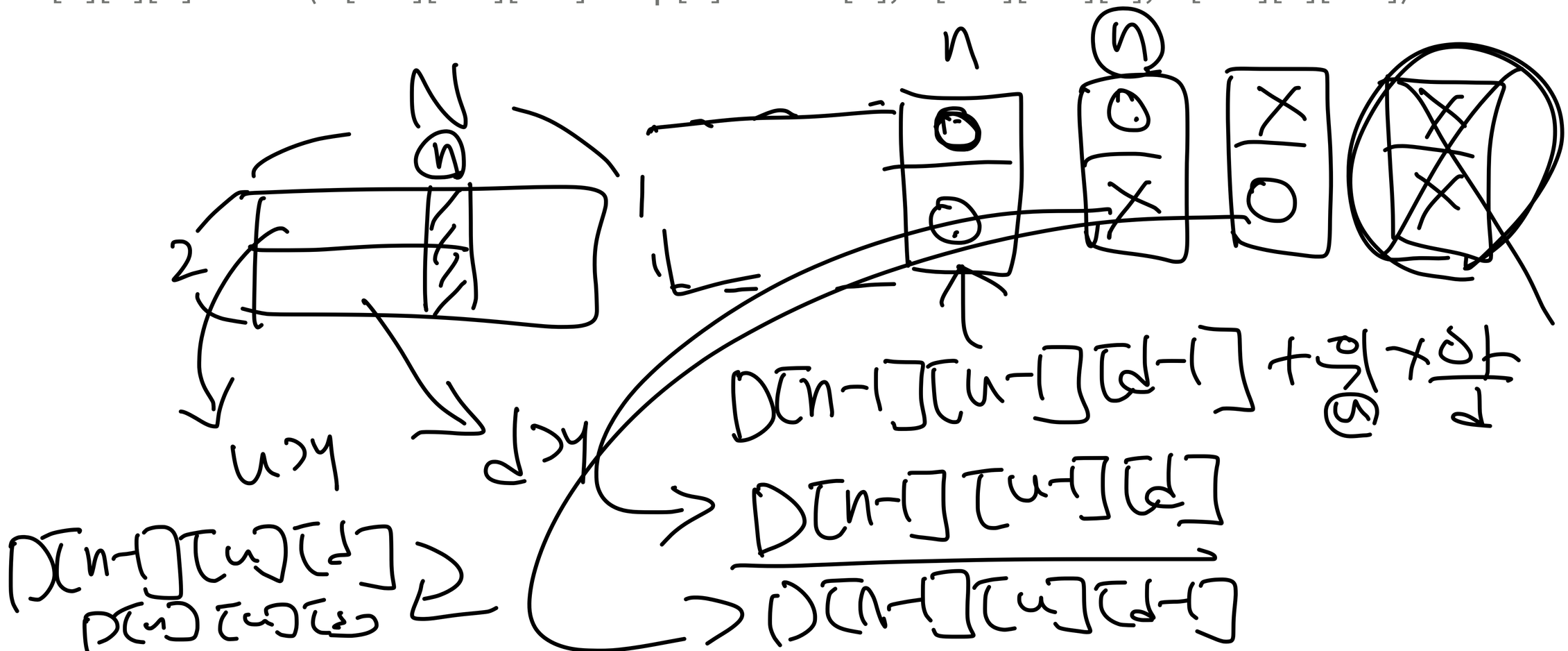
https://www.acmicpc.net/problem/1983

- 두 개의 행과 N개의 열로 이루어진 숫자 박스가 있다.
- 각 칸에는 0이 아니면서, -10 이상, 10 이하인 정수가 하나씩 써있다.
- 수는 순서를 유지하면서, 좌우로 움직일 수 있다.
- 숫자 박스의 값은 위 아래 써 있는 수를 곱하는 것이다.
- 순서를 적절히 이동시켜 숫자 박스 값의 최댓값을 구하는 문제

숫자 박스

https://www.acmicpc.net/problem/1983

- D[n][u][d] = 총 n칸을 채웠고, 위쪽의 수를 u개, 아래쪽의 수를 d개 사용했을 때 최대값
- D[n][u][d] = max(D[n-1][u-1][d-1] + Up[u]*Down[d], D[n-1][u-1][d], D[n-1][u][d-1])



숫자 박스

https://www.acmicpc.net/problem/1983

- 소스: http://codeplus.codes/514a0c1b48b24e4f8ed315e68b2ac013

즐거운 단어

https://www.acmicpc.net/problem/2922

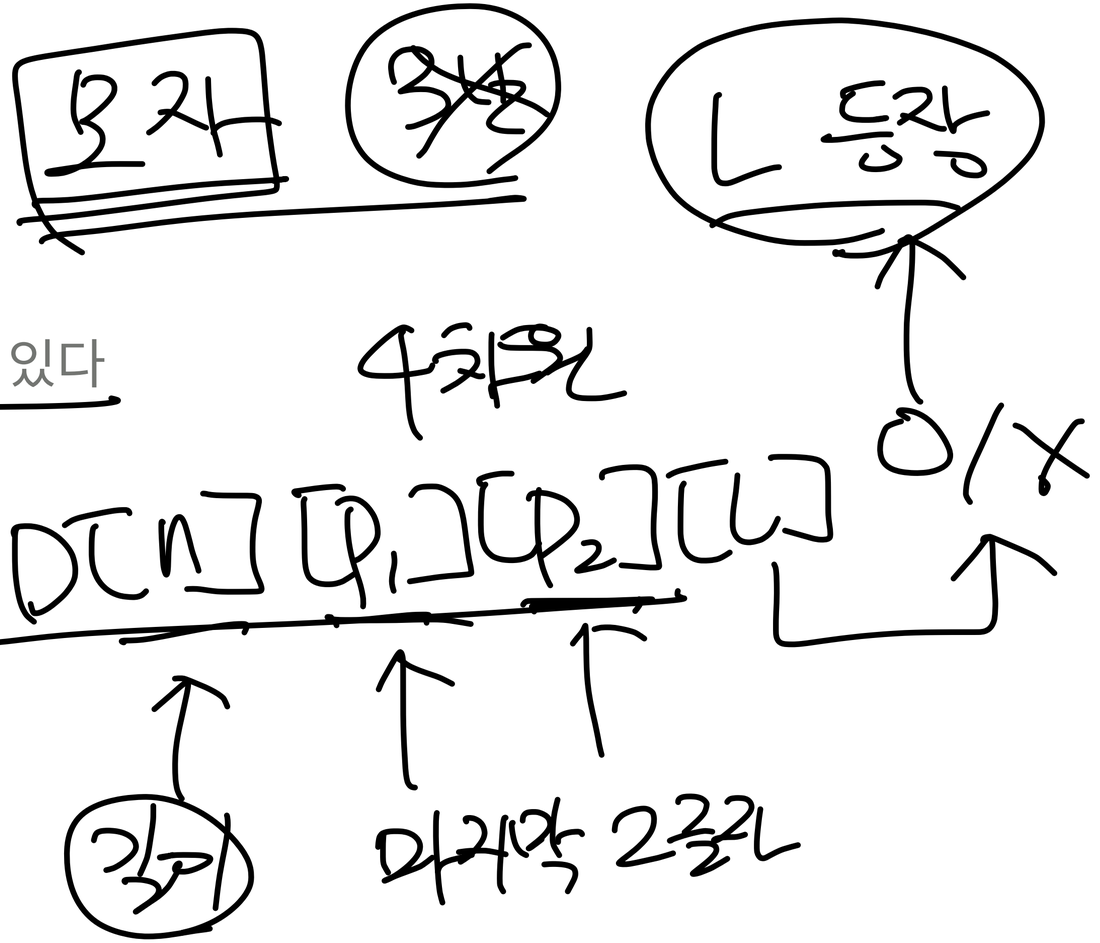
- 모음이 연속해서 3번, 자음이 연속해서 3번 나오지 않아야 한다
- L을 반드시 포함해야 한다
- 빈칸을 알파벳으로 바꿔서 즐거운 단어를 만들 수 있는 경우의 수를 세는 문제
- V__K의 경우 정답은 10

즐거운 단어

https://www.acmicpc.net/problem/2922

- 빈 칸을 바꿀 수 있는 경우는 총 3가지가 있다

1. 모음으로 바꾼다
2. L로 바꾼다
3. L을 제외한 다른 자음으로 바꾼다



즐거운 단어

https://www.acmicpc.net/problem/2922

- 마지막 2글자와 L이 나왔는지 아닌지를 기록해서 다이나믹을 만들 수 있다.

- D[N][P1][P2][L] = N번째 글자까지로 만들 수 있는 즐거운 단어의 개수
- P1: 전 글자, P2: 전전 글자, L: L이 나왔는지 아닌지

즐거운 단어

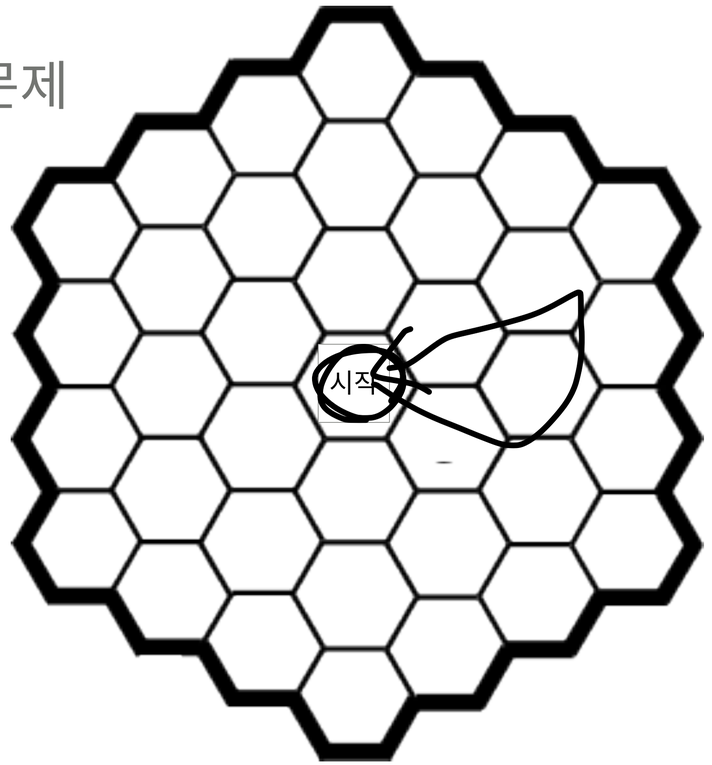
https://www.acmicpc.net/problem/2922

- 소스: http://codeplus.codes/c4a91916eaca44c3ab708a0ba3c8d5bb

미로에 갇힌 상근

https://www.acmicpc.net/problem/5069

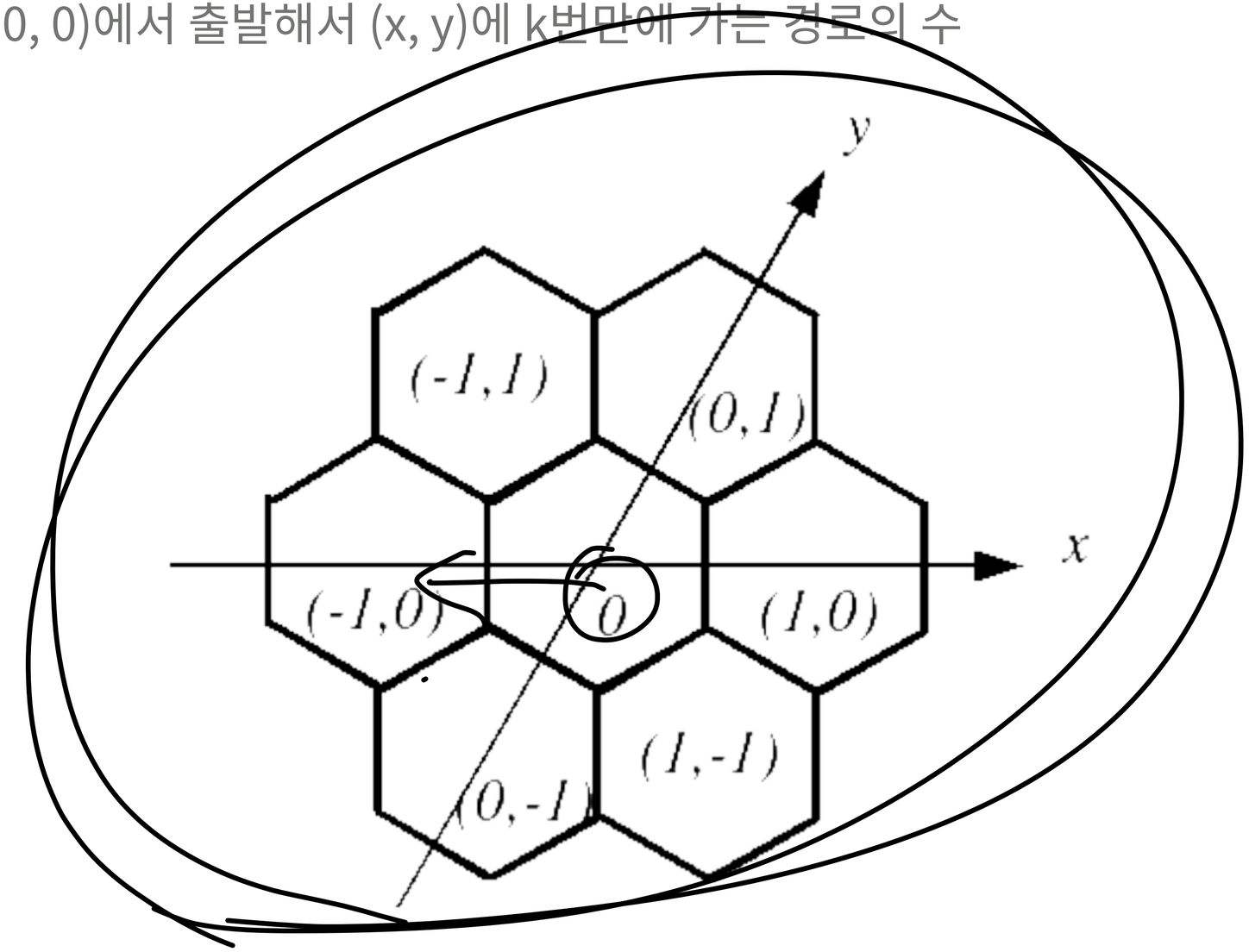
- 육각형 모양의 방이 계속해서 붙어있는 미로가 있다.
- 상근이가 있는 방에서 이동을 시작한다.
- N번 이동해서 원래 있던 방으로 이동하는 방법의 개수를 구하는 문제
- $1 \le N \le 14$



미로에 갇힌 상근

https://www.acmicpc.net/problem/5069

- D[k][x][y] = (0, 0)에서 출발해서 (x, y)에 k번만에 가는 경로의 수



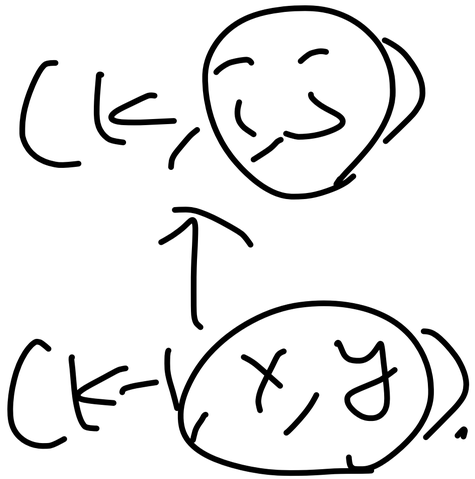
미로에 갇힌 상근

https://www.acmicpc.net/problem/5069

- 음수가 나올 수 있기 때문에
- D[k][x][y] = (14, 14)에서 출발해서 (x, y)에 k번만에 가는 경로의 수

미로에 갇힌 상근

https://www.acmicpc.net/problem/5069




```
d[0][14][14] = 1;
for (int k=1; k<=14; k++) {
    for (int i=0; i<m; i++) {
        for (int j=0; j<m; j++) {
            for (int l=0; l<6; l++) {
                int x = i+dx[l];
                int y = j+dy[l];
                if (0 <= x && x < m && 0 <= y && y < m) {
                    d[k][i][j] += d[k-1][x][y];
                }
            }
        }
    }
}
```

미로에 갇힌 상근

https://www.acmicpc.net/problem/5069

- 소스: http://codeplus.codes/b7e4e0ef106d461990dec22ff79ff382

돌다리 건너기

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

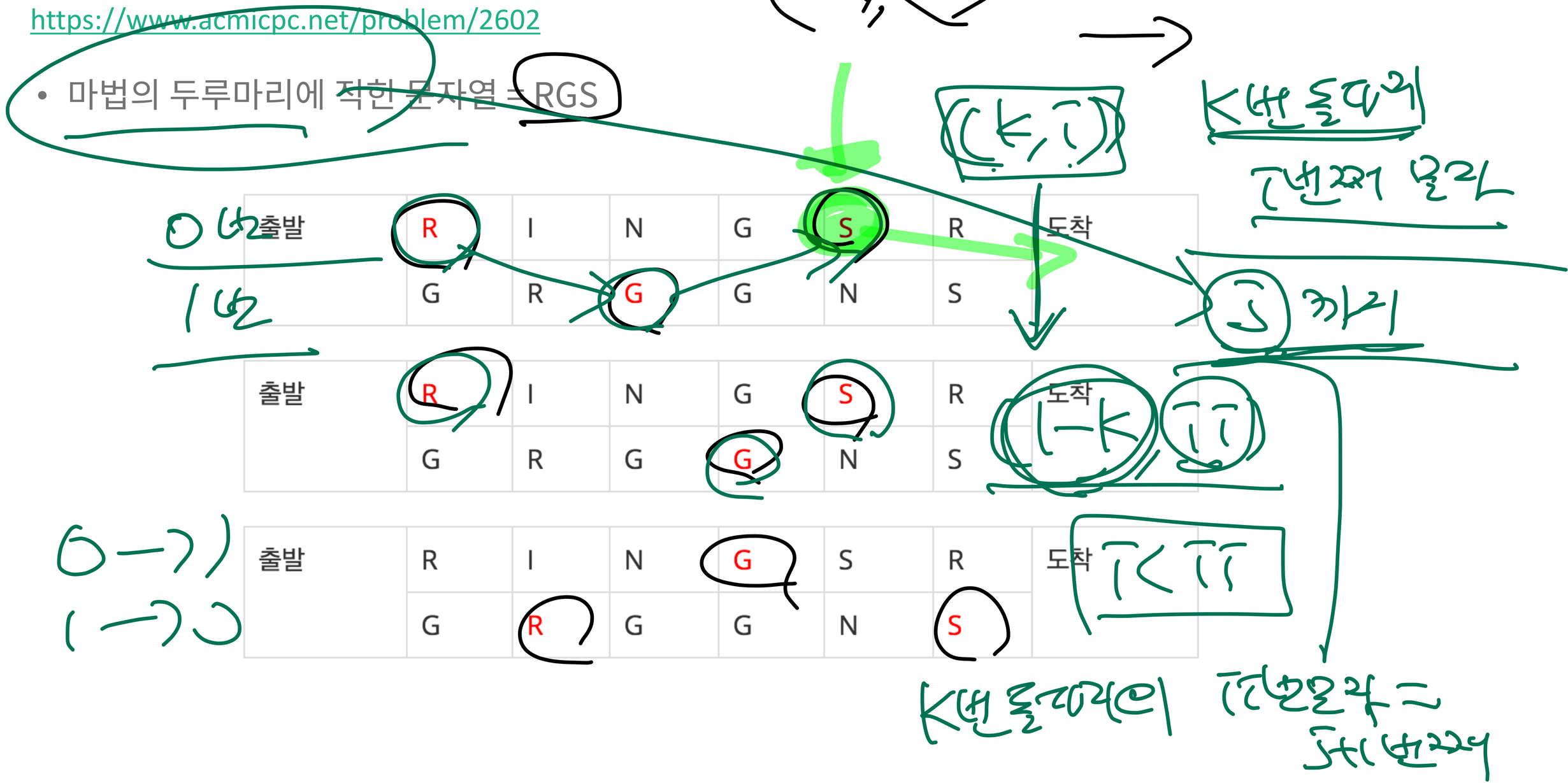
- 길이가 N인 다리가 있고, 다음 조건을 만족하면서 출발에서 도착으로 가는 경우의 수를 구하려고 한다
  1. 마법의 두루마리에 적힌 문자열의 순서대로 모두 밟고 지나가야 한다
  2. 악마의 돌다리(위)와 천사의 돌다리(아래)를 번갈아 가면서 밟아야 한다
  3. 한 칸 이상 오른쪽으로 전진해야 한다.

출발	R	I	N	G	S	R	도착
	G	R	G	G	N	S	

돌다리 건너기

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

- 마법의 두루마리에 적힌 문자열 = RGS



돌다리 건너기

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

RGS

- 불가능한 경우

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	

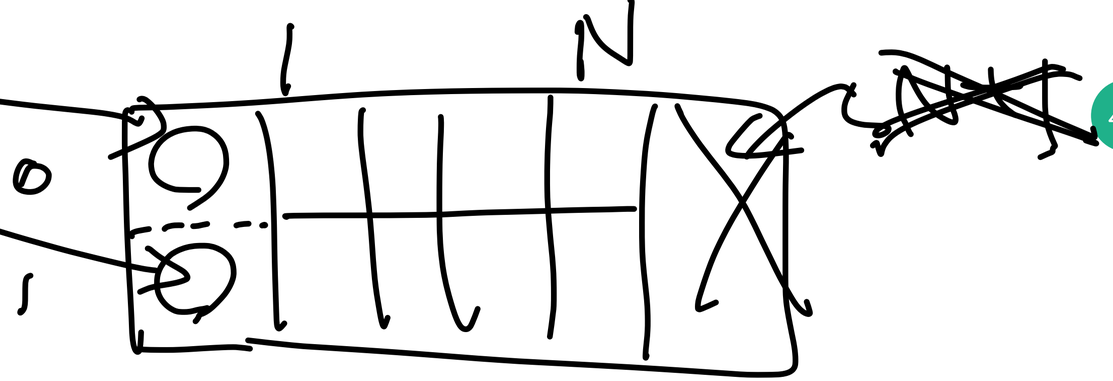
돌다리 건너기

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];
                }
            }
        }
    }
}
```

마방이 움직이기

두개에 대해

(1-k, l) → (k, i)

j-1 → j

(k, i)

마방이 움직이기 전

마방이 움직인 후

결국: d[k][i][m]

k=0, 1

1 ≤ i ≤ n

돌다리 건너기

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

- 소스: http://codeplus.codes/47c4830aa5264d0ebc9938b0ccebaa3e