[ 삼성 아이디어 ] - 발상

<<13460 구슬 탈출 2 >>

-> 어려운 문제다

Dfs/bfs에 while 문을 넣어서 구를 때 까지 구르게 하는게 포인트고

while (MAP[ry + dy[dir]][rx + dx[dir]] == '.') {

ry += dy[dir];

rx += dx[dir];

if (ry == O.y && rx == O.x) {

res = cnt;

break;

}

}

이런식으로

또한 같은 구멍에 들어갔을 때 거리가 먼것을 빽으로 한칸 넣는것이 포인트 인 듯 하다 .

if (ry == by && rx == bx) {

어쩌고 저쩌고

<<12100 2048(easy) >>

구슬 탈출 2 와 마찬가지로

Dfs, bfs 변형이지만

조금 쉬운 느낌

<< 3190 뱀 >>

뱀의 몸통의 처음과 끝

"""처음과 끝""" ->덱을 쓰자 .

<< 14499 주사위굴리기 >>

무난

<< 14500 테트로미노 >>

블럭이 dfs 순환을 한 모양이라지만

개인적으로 그렇게 발상이 나지 않는다

따라서 노가다를 뛰자

하드코딩 및 2중 포문 떡칠.

<< 14502 연구소 >>

일단 나는 DFS를 통해 맵에 벽을 세우는 조합을 만들었고, 벽은 3개를 반드시 설치해야하기 때문에, 벽 3개를 세운 조합 하나가 완성되면, 바이러스를 퍼트리고 (바이러스를 퍼트리는 부분은 bfs방식을 사용했다.) 안전지대가 얼마나 있는지 체크한다. 그 다음 다른 조합들과 비교하며 안전지대가 가장 많이 생기는 경우를 찾는 식으로 코드를 구성했다

<- 대표적인 샘숭 마인드 같다

<< 14503 로봇 청소기 >>

그냥 시뮬레이션이다

<< 14683 감시 >>

내 생각대로 하면 맞긴 맞을 건데 코드가 늘어지기 쉬울 것 같다.

동 서 남 북

각각 짜고

Cctv 1. -> 동

Cctv 2 -> 동 서

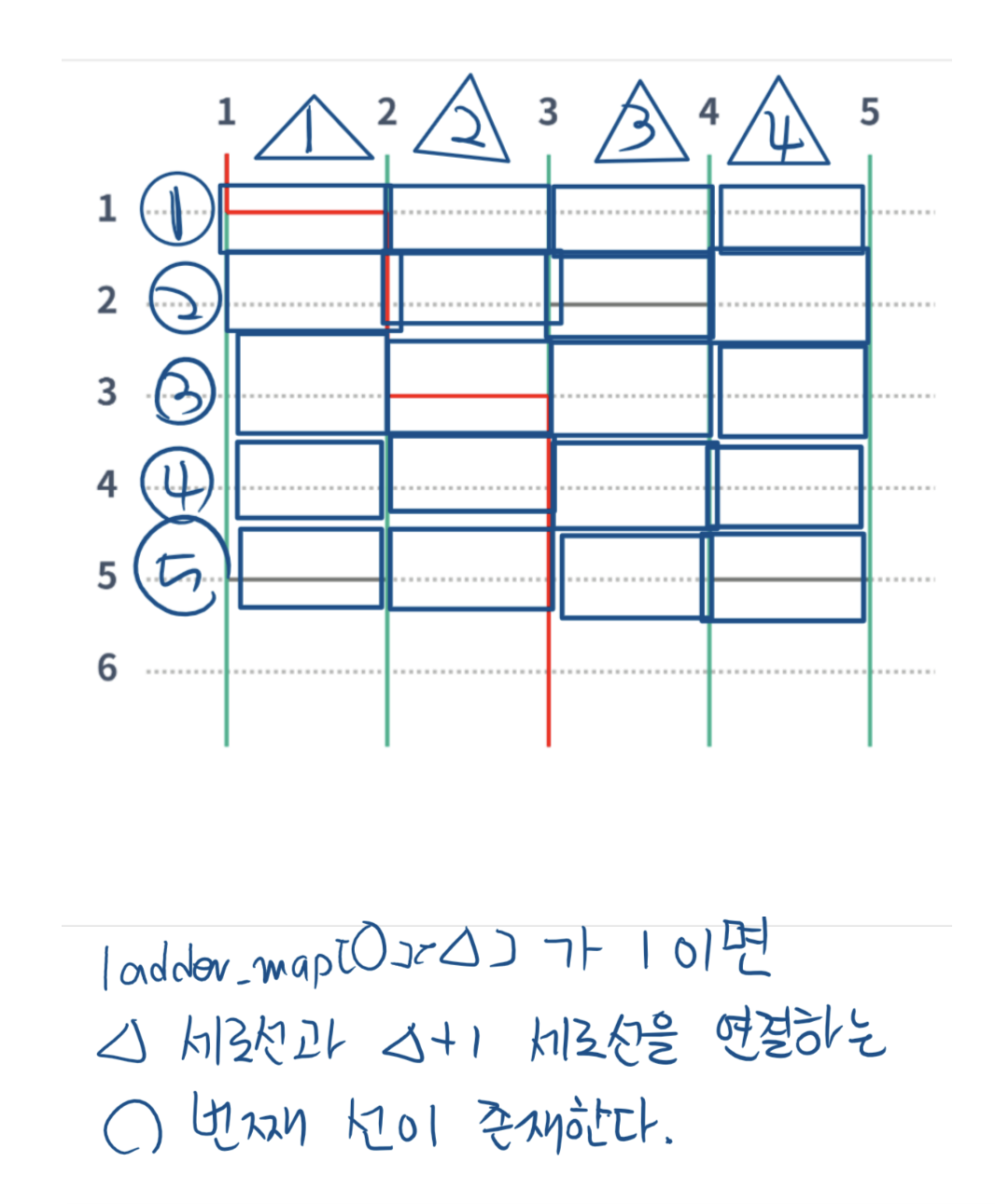
이런식으로 짜는 게 좋을 거 같다 . !

그럼 동서남북 코드를 한번쓰고

모든 cctv에 재사용 할 수 있다 !

https://donggoolosori.github.io/2020/11/20/boj-15683/

<< 15684 사다리 조작 >>



<<23290 마법사 상어와 복제 >>

문제에서 **물고기가 사라지고 그 자리에 물고기의 냄새가 남는다**는 문구를 보고 물고기와 냄새를 같은 구조체로 두고 단지 물고기인지 냄새인지 구분하는 값을 넣어주려 했다. 하지만 문제를 푸는 도중에 하나의 좌표에서 물고기와 냄새를 구분하고 제거해야 하는 작업이 빈번했다. 따라서 이러한 구현 방식은 **최적화에 굉장히 불리했고 구현도 까다로웠다.**

**물고기만 저장하는 배열과 냄새를 따로 두어야** 구현적인 면과 성능적인 면에서 이득을 볼 수 있다. 물고기 배열의 경우에는 **물고기의 방향을 나타내는 정수**를 넣고, 냄새의 경우에는 **남은 지속 시간**을 정수로 넣어주고 매 턴마다 1씩 감소시켜주면 된다. 이처럼 두 배열 모두 **정수형 벡터**로 표현이 가능하다.

**시계 방향 90도 회전**

**배열 전체를 회전**

int arr[N][N], tmp\_arr[N][N];

for(int i=0; i<N; i++){

for(int j=0; j<N; j++) tmp\_arr[j][N-1-i] = arr[i][j];

}

**반시계 방향 90도 회전**

int arr[N][N], tmp\_arr[N][N];

for(int i=0; i<N; i++){

for(int j=0; j<N; j++) tmp\_arr[N-1-j][i] = arr[i][j];

}