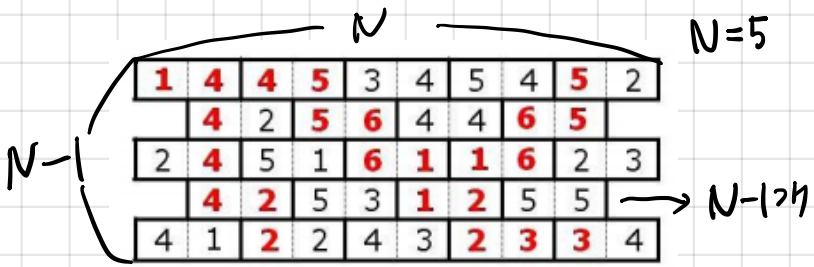
BOJ 5213 과외맨 BFS

구현해야할 것이 굉장히 많은 문제



한 타일에서 다른 타일로 넘어가려면, 두 타일이 인접해야 한다. 또, 같은 변을 공유하는 조각에 쓰여 있는 숫자가 같아야 한다.

과외맨은 반대편으로 넘어가기 위해서 첫 줄의 가장 첫 타일에서 마지막 줄의 가장 마지막 타일로 이동하는 가장 짧은 경로를 찾으려고 한다.

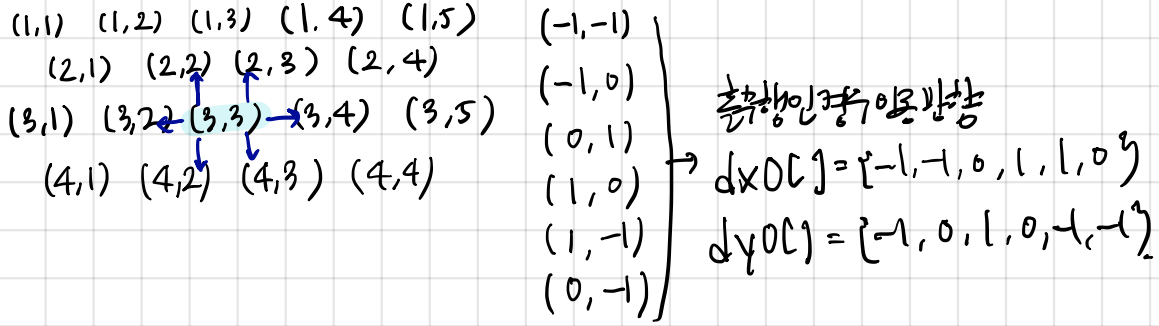
타일은 row-major order에 의해서 번호가 매겨져 있으며, 첫 번째 줄의 첫 타일의 번호는 1, 마지막 타일의 번호는 N이다. 두 번째 줄에서 첫 타일의 번호는 N+1이고, 마지막 타일의 번호는 2*N-1이다.

첫 줄의 첫 타일로만 과외맨이 들어갈 수 있고, 마지막 줄의 마지막 타일위에 과외 노트가 놓여져 있다.

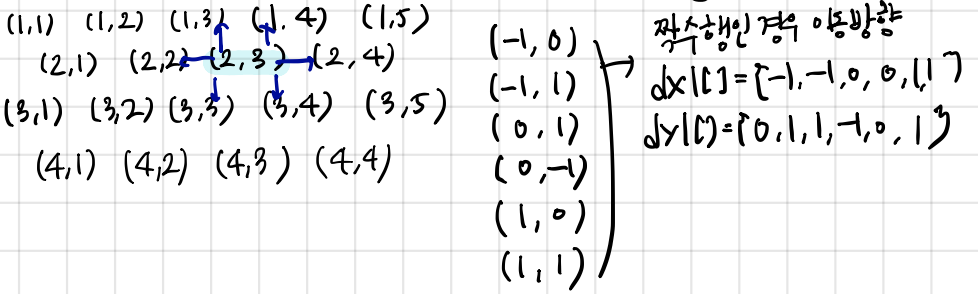
마지막 줄의 마지막 타일로 이동할 수 없는 경우가 존재할 수 있다. 이 경우에는 번호가 가장 큰 타일로 이동하면 된다.

최소경로를 지나온 타일출력

① 현재칸이 홀수행인경우 (혹은 짝수) 이동 방향



② 현재칸이 홀수행인 경우 (혹은 짝수) 이동방향



③ int a[500][500][2] 0 : 타일 앞부분 1 : 타일 뒷부분

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n - 1; j++) {
        cin >> a[i][j][0];
        cin >> a[i][j][1];
    }
    if (i % 2 == 0) {
        cin >> a[i][n - 1][0];
        cin >> a[i][n - 1][1];
    }
}
```

홀수행, 짝수행 타일입력주의

④ pair<int,int> from[500][500] → 타일(i,j)가 어디에서 왔는지

⑤ bool ok(int x, int y) → (x,y)가 짝수행인지 홀수행인지
bool go(int x1, int y1, int x2, int y2)
→ (x1, y1) → (x2, y2) 로 갈수 있는지

⑥ 알맞은 도로아가씨 (몇번째타일인지 반환해줘서, 스택)

```
while (check[x][y] == false) { //마지막 타일로 이동하지 못한 경우
    //번호가 가장 큰 타일을 찾는다
    y -= 1;
    if (y < 0) {
        x -= 1;
        y = n - 1;
        if (x % 2 == 1) { //짝수 행인 경우 열 개수 한개 더 빼줌
            y -= 1;
        }
    }
}

cout << dist[x][y] << '\n';
stack<pair<int, int>> s;

while (!(x == 0 && y == 0)) {
    s.push(make_pair(x, y));
    auto p = from[x][y];
    x = p.first;
    y = p.second;
}
```

```
int num(int x, int y) { //(x, y)가 몇 번째 타일인지
    int ans = x / 2 * (n * 2 - 1);
    if (x % 2 == 1) {
        ans += n;
    }
    ans += y + 1;
    return ans;
}
```