

양

백준 3184

202021018 오윤주

문제

2 3184번 제출 맞은 사람 슷코딩 재채점 결과 채점 현황 내 제출 난이도 기여 강의▼ 질문 검색

양 성공 출처 다국어

★ 한국어 ▼

2 실버 II

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	5439	3430	2611	63.652%

문제

미키의 뒷마당에는 특정 수의 양이 있다. 그가 폭 잠든 사이에 배고픈 늑대는 마당에 들어와 양을 공격했다.

마당은 행과 열로 이루어진 직사각형 모양이다. 글자 '.' (점)은 빈 필드를 의미하며, 글자 '#'는 울타리를, 'o'는 양, 'v'는 늑대를 의미한다.

한 칸에서 수평, 수직만으로 이동하며 울타리를 지나지 않고 다른 칸으로 이동할 수 있다면, 두 칸은 같은 영역 안에 속해 있다고 한다. 마당에서 "탈출"할 수 있는 칸은 어떤 영역에도 속하지 않는다고 간주한다.

다행히 우리의 양은 늑대에게 싸움을 걸 수 있고 영역 안의 양의 수가 늑대의 수보다 많다면 이기고, 늑대를 우리에서 쫓아낸다. 그렇지 않다면 늑대가 그 지역 안의 모든 양을 먹는다.

맨 처음 모든 양과 늑대는 마당 안 영역에 존재한다.

아침이 도달했을 때 살아남은 양과 늑대의 수를 출력하는 프로그램을 작성하라.

8 8

```

#####
#..O...#
#.####.#
#.#v.#.#
#.#.O#O#
#O.##. .#
#.v..v.#
#####

```

늑대 1마리, 양 3마리

9 12

```

#####
#.OO#...#v#
#..O#.#.#.#
#...#O#...#
#.#v#O####.#
#...#v#...#
#...v#v#####
.#####VV.O#
.....#####

```

늑대 5마리, 양 3마리

구현

```
R, C = map(int, input().split()) // 크기 입력받기
```

```
dx = [-1, 1, 0, 0]
```

```
dy = [0, 0, -1, 1]
```

```
answer = [0, 0]
```

```
graph = []
```

```
for _ in range(R):
```

```
    graph.append(list(map(str, input()))) // 그래프에 넣어주기
```

```
for i in range(R):
```

```
    for j in range(C):
```

```
        if graph[i][j] != '#':
```

```
            bfs(j, i)
```

```
// 울타리가 아닌 부분 bfs로 확인
```

구현

```
def bfs(x, y):  
    q = []  
    sheep, wolf = 0, 0  
  
    if graph[y][x] == 'o':  
        sheep += 1  
    elif graph[y][x] == 'v':  
        wolf += 1  
    q.append((x, y))  
    graph[y][x] = '#'
```

// 처음 양과 늑대 마릿수 세어주기

// 시작점, 울타리 생성

구현

```
while len(q) > 0:
    newx, newy = q.pop()
    for i in range(4):
        nx = newx + dx[i]
        ny = newy + dy[i]
        if 0 <= nx < C and 0 <= ny < R:
            if graph[ny][nx] != '#':
                if graph[ny][nx] == 'o':
                    sheep += 1
                elif graph[ny][nx] == 'v':
                    wolf += 1
                q.append((nx, ny))
                graph[ny][nx] = '#'
    if sheep > wolf:
        answer[0] += sheep
    else:
        answer[1] += wolf
```

// 리스트에서 하나씩 빼서

// 울타리가 아닌 부분을

// 하루가 지난 후 양과 늑대 마릿수 세기

// 확인 한 부분 울타리로 바꾸기

// 양이 늑대보다 많으면 양은 살아남았기에 양 결과값에 추가 아니라면
늑대 결과값에 추가

감사합니다.