S2 섬의 개수 (Baekjoon 4963)

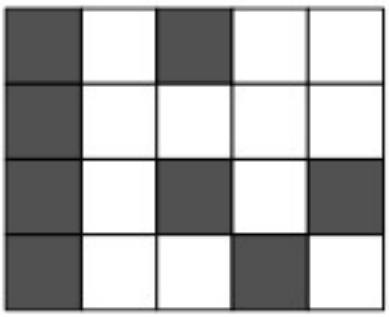
categories: 그래프 이론, 그래프 탐색, 깊이 우선 탐색

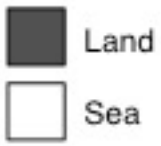
시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	53048	26775	19227	49.315%

문제

정사각형으로 이루어져 있는 섬과 바다 지도가 주어진다.

섬의 개수를 세는 프로그램을 작성하시오.





입력

입력은 여러 개의 테스트 케이스로 이루어져 있다.

각 테스트 케이스의 첫째 줄에는 지도의 너비 w와 높이 h가 주어진다.

w와 h는 50보다 작거나 같은 양의 정수이다.

둘째 줄부터 h개 줄에는 지도가 주어진다. 1은 땅, 0은 바다이다.

입력의 마지막 줄에는 0이 두 개 주어진다.

출력

각 테스트 케이스에 대해서, 섬의 개수를 출력한다

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S2

섬의 개수 (Baekjoon 4963)

categories: 그래프 이론, 그래프 탐색, 깊이 우선 탐색

예제입력

```
1 1
0
2 2
0 1
1 0
3 2
1 1 1
1 1 1
5 4
1 0 1 0 0
1 0 0 0 0
1 0 1 0 1
1 0 0 1 0
5 4
1 1 1 0 1
1 0 1 0 1
1 0 1 0 1
1 0 1 1 1
5 5
1 0 1 0 1
0 0 0 0 0
1 0 1 0 1
0 0 0 0 0
1 0 1 0 1
0 0
```

예제출력

```
0
1
1
3
1
9
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S2 섬의 개수 (Baekjoon 4963)

categories: 그래프 이론, 그래프 탐색, 깊이 우선 탐색

Answer(1/2)

```python
from pprint import pprint
width, height = -1,-1
directions = [(-1,0),(1,0),(0,-1),(0,1),(-1,-1),(1,-1),(-1,1),(1,1)]
# Up, Down, Left, Right, Left Up, Left Down, Right Up, Right Down
need_to_find = 0
found = []
dfs_search_count = 0

def search_next_start_node(graph):
    for i in range(height):
        for j in range(width):
            if graph[i][j] == 1 and (i,j) not in found:
                return (i,j)

    return (-1,-1)

def dfs(graph, start_node:tuple):
    global found, dfs_search_count

    stack = [start_node]
    visited = [[False]*width for _ in range(height)]

    while stack:
        current_node = stack.pop()
        if not visited[current_node[0]][current_node[1]]:
            visited[current_node[0]][current_node[1]] = True
            found.append((current_node[0],current_node[1]))

            for i in range(len(directions)):
                next_x = current_node[0] + directions[i][0]
                next_y = current_node[1] + directions[i][1]
                if 0 <= next_x < height and 0 <= next_y < width:
                    if graph[next_x][next_y] != 0 and not visited[next_x][next_y]:
                        stack.append((next_x,next_y))

    dfs_search_count += 1
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S2

섬의 개수 (Baekjoon 4963)

categories: 그래프 이론, 그래프 탐색, 깊이 우선 탐색

Answer(2/2)

```
def solve():
    global need_to_find,found,dfs_search_count
    world_map = []
    need_to_find = 0
    dfs_search_count = 0
    found = []
    for i in range(height):
        list_to_append = list(map(int, input().split(" ")))
        need_to_find += list_to_append.count(1)
        world_map.append(list_to_append)


    while len(found) != need_to_find:
        search_result = search_next_start_node(world_map)
        if search_result != (-1,-1):
            dfs(world_map, search_result)


    print(dfs_search_count)

while(True):
    width, height = map(int, input().split(" "))
    if width == 0 and height == 0:
        break

    solve()
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering