4963번: 섬의 개수

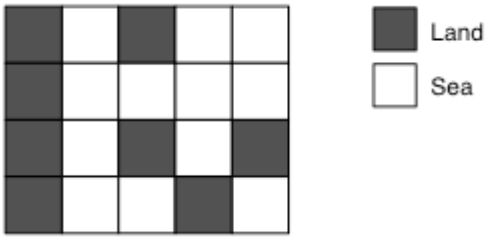
섬의 개수

한국어 원문

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	5859	2839	2144	49.039%

문제

정사각형으로 이루어져 있는 섬과 바다 지도가 주어진다. 섬의 개수를 세는 프로그램을 작성하시오.



한 정사각형과 가로, 세로 또는 대각선으로 연결되어 있는 사각형은 걸어갈 수 있는 사각형이다.

두 정사각형이 같은 섬에 있으려면, 한 정사각형에서 다른 정사각형으로 걸어서 갈 수 있는 경로가 있어야 한다. 지도는 바다로 둘러쌓여 있으며, 지도 밖으로 나갈 수 없다.

문제 유형 : 그래프 BFS, DFS 연결 요소 세기

접근 방법 : 너무나 정형적인 연결 요소 세는 문제 였다. 다른 점은 연결 요소가 대각선 까지 8 방향까지 인접 노드 이며, 평소에 풀던 문제와 n,m 맵핑 방식이 달랐다. 이건 내가 문제를 제대로 읽지 않은 문제이다. DFS가 코드가 짧아 더 좋아하지만, 연습을 위해 BFS로 풀었다.

배열이나 리스트로 맵을 만들면 계속 인덱스 에러가 뜬다. 꼼꼼히 끝까지 확인해야 한다.

```
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

class Pair{
int x;
int y;

Pair(int x, int y){
this.x = x;
this.y = y;
}
}

public class Main_4963{

private static final int[] dx = {0, 0, 1,-1, 1, 1,-1,-1};
private static final int[] dy = {1,-1, 0, 0,-1, 1,-1, 1};

public static void main(String[] args){

Scanner sc = new Scanner(System.in);



while (true) {
int m = sc.nextInt();
int n = sc.nextInt();
if (n == 0 && m == 0) {
break;
}

int[][] a = new int[n][m];
for (int i=0; i<n; i++) {
for (int j=0; j<m; j++) {
a[i][j] = sc.nextInt();
```

```
}
}
int cnt = 0;
int result[][] = new int[n][m];

for(int i=0; i<n; i++){
for(int j=0; j<m; j++){
if(result[i][j]==0 && a[i][j]==1){
bfs(i,j,a,result,++cnt,m,n);
}
}
}

System.out.println(cnt);
}
}

private static void bfs(int y,int x,int[][] map,int[][] result,int cnt,int m,int n){
Queue<Pair> queue = new LinkedList<Pair>();
queue.add(new Pair(x,y));
result[y][x] = cnt;
int nx = 0;
int ny = 0;
while(!queue.isEmpty()){
Pair pair = queue.remove();
result[pair.y][pair.x] = cnt;
for(int i=0; i<8; i++){
nx = pair.x + dx[i];
ny = pair.y + dy[i];

if(0<=nx && nx<m && 0<=ny && ny<n){
if(map[ny][nx]==1 && result[ny][nx]==0){
result[ny][nx] = cnt;
queue.add(new Pair(nx,ny));
}
}
}
}
}
```

```
}


}


from collections import deque as queue
n = int()
m = int()
cnt = int()
q = queue()
dx=[1,0,-1,0,1,1,-1,-1]
dy=[0,1,0,-1,1,-1,1,-1]
def isInRange(x,y):
  return x>=0 and x <m and y >=0 and y<n
def BFS(x,y):
  q.append((x,y))
  while q :
  tx, ty =q.popleft();
  land[tx][ty] = 0
  for i in range (8):
  nx = tx+dx[i]
  ny = ty+dy[i]
  if isInRange(nx,ny) and land[nx][ny]: q.append((nx,ny))
  return



while (1):
  n,m = map(int,input().split())
  q.clear()
  cnt =0
  if not m and not n : break
  land = [[int(i)for i in input().split() ]for i in range(m)]
  for i in range(n) :
  for j in range(m) :
  if land[j][i] :
  BFS(j,i)
  cnt+=1
  print(cnt)
```

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.

예제 입력

```
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
```

예제 출력

```
0
```

1
1
3
1
9

힌트

출처

ACM-ICPC > Regionals > Asia > Japan > Domestic Contest > 2009 Domestic Contest B번

- 문제를 번역한 사람: baekjoon

링크

- TJU Online Judge

알고리즘 분류

보기