

2667 단지번호붙이기

2021년 4월 5일 월요일 오후 5:51

단지번호붙이기

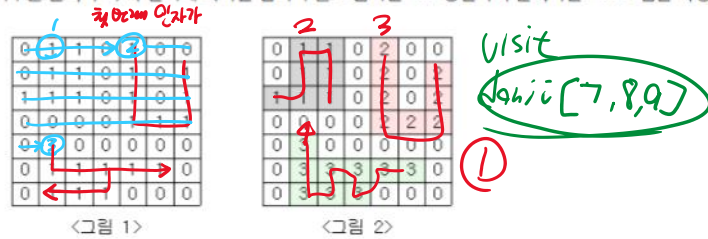
Silver I

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	79587	32550	20596	39.070%

문제

<그림 1>과 같이 정사각형 모양의 지도가 있다. 1은 집이 있는 곳을, 0은 집이 없는 곳을 나타낸다. 철수는 이 지도를 가지고 연결된 집의 모임인 단지를 정의하고, 단지에 번호를 붙이려 한다. 여기서 연결되었다는 것은 어떤 집이 좌우, 혹은 아래위로 다른 집이 있는 경우를 말한다. 대각선상에 집이 있는 경우는 연결된 것이 아니다. <그림 2>는 <그림 1>을 단지별로 번호를 붙인 것이다. 지도를 입력하여 단지수를 출력하고, 각 단지에 속하는 집의 수를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오.



입력

첫 번째 줄에는 지도의 크기 N (정사각형이므로 가로와 세로의 크기는 같으며 $5 \leq N \leq 25$)이 입력되고, 그 다음 N 줄에는 각각 N 개의 자료(0 혹은 1)가 입력된다.

출력

첫 번째 줄에는 총 단지수를 출력하시오. 그리고 각 단지내 집의 수를 오름차순으로 정렬하여 한 줄에 하나씩 출력하시오.

예제 입력 1 복사

```
7
0110100
0110101
1110101
0000111
0100000
0111110
0111000
```

예제 출력 1 복사

```
3
7
8
9
```

```
from collections import deque
n = int(input())
# 2차원 map 생성
town = [list(map(int, list(input())) for i in range(n))]
visit = [[0]*n for i in range(n)]
dx = [0,1,0,-1]
dy = [-1,0,1,0]
# 집 갯수 배열
num = []
# bfs
def bfs(town, visit, sx, sy, order):
    queue = deque()
    visit[sx][sy] = order;
    x, y = sx, sy
    queue.append([x,y])
    result = 0
    # queue가 계속될 동안
    while queue:
        temp = queue.popleft()
        for i in range(4):
            x = temp[0] + dx[i]
            y = temp[1] + dy[i]
            # 밖의 아예 들어간 경우
```

bfs

```

temp = queue.popleft()
for i in range(4):
    x = temp[0] + dx[i]
    y = temp[1] + dy[i]
    # 범위 안에 들어갈 경우
    if 0 <= x < n and 0 <= y < n:
        # 밟는 곳이 0이면 무시
        if town[x][y] == 0:
            continue
        # 방문을 하지 않았던 곳일 경우
        elif visit[x][y] == 0:
            visit[x][y] = order;
            result+=1
            queue.append([x,y])
    return result+1
# order = 1부터 시작 -> 1이 마을이므로 헛갈리지 않기 위해서
order = 2
# 순회하며 만날 경우 bfs를 시작
for i in range(n):
    for j in range(n):
        if town[i][j] != 0 and visit[i][j] == 0:
            num.append(bfs(town, visit, i, j, order))
            order+=1
print(len(num))
num.sort()
for i in num:
    print(i)

```

계속모의