



백준 2667번

단지번호 붙이기

- 문제 설명 :

정사각형 모양의 지도에 1은 집, 0은 집이 없는 곳을 의미함

집들로 연결된 집합을 단지 하나로 정의하고 그 수를 구해야함

단, 대각선상에 집이 있는 경우는 연결된 것이 아님

- 입력

첫 줄에는 지도의 크기 n 이 입력, 그 다음 줄에는 각각 n 개의 0 또는 1이 입력 됨

- 출력

첫 줄에는 총 단지수를 출력하고 각 단지내 집의 수를 오름차순으로 정렬하여 출력

• 문제 설명 :

0	1	1	0	1	0	0
0	1	1	0	1	0	1
1	1	1	0	1	0	1
0	0	0	0	1	1	1
0	1	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	0	0	0

<그림 1>

0	1	1	0	2	0	0
0	1	1	0	2	0	2
1	1	1	0	2	0	2
0	0	0	0	2	2	2
0	3	0	0	0	0	0
0	3	3	3	3	3	0
0	3	3	3	0	0	0

<그림 2>

주어진 행렬에서 단지 수와 단지 내에 속하는 집의 수를 구해야 함

예를 들어서 위 그림에서 색칠된 구역들이 각각 하나의 단지임

단지의 수는 3개이고 각 단지에 속한 집의 수는 시계방향으로 7, 8, 9임

- 문제 풀이 : DFS

2차원 배열의 각 자료 값을 살피면서 자료의 값이 1일 경우

해당 위치에서 상하, 좌우로 이동하며 DFS 를 실행해서 자료의 값을 탐색한다

단, 인덱스를 벗어나거나 값이 0일 경우 실행하지 않는다

중복 탐색을 방지하기 위해 탐색한 자료의 값을 0으로 변경한다

• 문제 풀이 : DFS 예시

CNT 0

0	1	0	0
1	1	0	0
0	0	1	1
0	0	1	1

CNT 0

0	0	0	0
1	1	0	0
0	0	1	1
0	0	1	1

dfs(0, 1)

CNT 0

0	0	0	0
1	0	0	0
0	0	1	1
0	0	1	1

dfs(0, 1)
dfs(1, 1)

CNT 0

0	0	0	0
0	0	0	0
0	0	1	1
0	0	1	1

dfs(0, 1)
dfs(1, 1)
dfs(1, 0)

• 문제 풀이 : DFS 예시

CNT 0

0	0	0	0
0	0	0	0
0	0	1	1
0	0	1	1

dfs(0, 1)
dfs(1, 1)

CNT 0

0	0	0	0
0	0	0	0
0	0	1	1
0	0	1	1

dfs(0, 1)

CNT 1

0	0	0	0
0	0	0	0
0	0	1	1
0	0	1	1

CNT 1

0	0	0	0
0	0	0	0
0	0	1	1
0	0	1	1

> 2667번

- 소스 코드 :

- `#define _CRT_SECURE_NO_WARNINGS`
- `#include <iostream>`
- `#include <algorithm>`

- `using namespace std;`

- `int map[26][26]; //지도`
- `int n = 0; //지도의 크기`

- `int dirx[4] = { -1, 1, 0, 0 }; //좌우탐색`
- `int diry[4] = { 0, 0, 1, -1 }; //상하탐색`

- `int cnt = 0; //집의 갯수`

> 2667번

- 소스 코드 :

- `int house[625] = { 0, };` //최대단지의 갯수
- //좌표 범위 확인을 위한 함수
- `int check(int x, int y, int n) {`
- `return x >= 0 && x < n && y >= 0 && y < n;`
- `}`
- `void dfs(int i, int j) {`
- `map[i][j] = 0;` //탐색한 자료값은 0으로 변경
- `house[cnt]++;`

- 소스 코드 :

- //상하좌우 탐색

- for (int k = 0; k < 4; k++) {

- int x = i + dirx[k];

- int y = j + diry[k];

- if (check(x, y, n) && map[x][y] == 1) {

- dfs(x, y);

- }

- }

- }

> 2667번

- 소스 코드 :

- `int main() {`

- `//지도의 크기입력`

- `scanf("%d", &n);`

- `//지도에 자료입력`

- `for (int i = 0; i < n; i++) {`

- `for (int j = 0; j < n; j++) {`

- `scanf("%1d", &map[i][j]);`

- `}`

- `}`

> 2667번

- 소스 코드 :

- //각 자료값을 확인

- for (int i = 0; i < n; i++) {

- for (int j = 0; j < n; j++) {

- //자료 값이 1이면 dfs 수행

- if (map[i][j] == 1) {

- dfs(i, j);

- //dfs수행이 완료 되면 카운트

- cnt++;

- }

- }

- }

> 2667번

- 소스 코드 :
- //출력하기전 정렬
- `sort(house, house + cnt);`
- //모든 자료값 확인 후 출력
- `cout << cnt << endl;`
- `for (int i = 0; i < cnt; i++) {`
- `cout << house[i] << endl;`
- `}`
- `}`

> 2667번

• 채점 결과 :

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
47879614	haddoli9	2667	맞았습니다!!	2024 KB	8 ms	C++17 / 수정	1260 B	10초 전



끝!