

Part.II 탐색기반 알고리즘의 설계

V2017117 강현민



서강대학교 영상대학원
SOGANG UNIVERSITY GRADUATE SCHOOL OF MEDIA

두더지 굴(S)

문제

두더지 굴(S)

정올이는 땅속의 굴이 모두 연결되어 있으면 이 굴은 한 마리의 두더지가 사는 집이라는 사실을 발견하였다.

정올이는 뒷산에 사는 두더지가 모두 몇 마리인지 궁금해졌다. 정올이는 특수 장비를 이용하여 뒷산의 두더지 굴을 모두 나타낸 지도를 만들 수 있었다.

이 지도는 직사각형이고 가로 세로 영역을 0 또는 1로 표현한다. 0은 땅이고 1은 두더지 굴을 나타낸다. 1이 상하좌우로 연결되어 있으면 한 마리의 두더지가 사는 집으로 정의할 수 있다.

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]

[그림 2]는 [그림 1]을 두더지 굴로 번호를 붙인 것이다. 특수촬영 사진 데이터를 입력받아 두더지 굴의 수를 출력하고, 각 두더지 굴의 크기를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오.

입력

첫 번째 줄에 가로, 세로의 크기를 나타내는 n 이 입력된다. (n 은 30이하의 자연수)
두 번째 줄부터 n 줄에 걸쳐서 n 개의 0과 1이 공백으로 구분되어 입력된다.

출력

첫째 줄에 두더지 굴의 수를 출력한다. 둘째 줄부터 각 두더지 굴의 크기를 내림차순으로 한 줄에 하나씩 출력한다.

입력 예	출력 예
7	
0 1 1 0 1 0 0	
0 1 1 0 1 0 1	3
1 1 1 0 1 0 1	9
0 0 0 0 1 1 1	8
0 1 0 0 0 0 0	7
0 1 1 1 1 1 0	
0 1 1 1 0 0 0	

두더지 굴(S)

풀이

- 문제에서 주어진 입력 예를 그래프로 나타내면 아래 그림과 같다.
- 배열의 (0,0)부터 순차탐색을 진행하면서 1이라면 깊이우선탐색을 이용하여 모든 연결된 점을 방문하고 특정 값으로 체크한다.
- 마지막에 깊이우선탐색을 실행한 횟수가 두더지의 수가 되고, 각 두더지 굴의 크기는 다른 배열에 저장해 둔 다음 마지막에 `std::sort()`를 이용하여 정렬한다.
- 여기에 사용되는 알고리즘은 지뢰찾기, 뽕요뽕요 등의 게임에 많이 활용되는 방법으로서, flood fill이라고도 한다.

입력 예	대응되는 그래프
7 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	

두더지 굴(S)

알고리즘 실행 과정

- 두더지 = 2
- 두더지의 값은 2부터 시작. 1은 주인 없는 굴이므로 2부터 증가함.
- (0,0)에서 탐색을 시작. 원소가 0이므로 통과
- Size배열은 각 두더지 집의 크기를 저장할 배열

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

Size	0	0	0	0	0	0
index	0	1	2	3	4	5

두더지 굴(S)

알고리즘 실행 과정

- (0,1)을 탐색, 원소의 값이 1이므로 dfs를 이용하여 상, 하, 좌, 우로 연결된 그래프를 모두 탐색하여 2로 수정.
- 방문한 정점의 수인 7을 size[2]에 기록하여 크기를 저장하고, 두더지 값 1 증가.

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

Size	0	0	0	0	0	0
index	0	1	2	3	4	5

두더지 굴(S)

알고리즘 실행 과정

- 두더지 = 3
- (0,2)를 탐색, 원소의 값이 원래 1이었으나 2로 바뀌었으므로, 이미 다른 두더지의 굴에 도착되었음. 따라서 그냥 통과.

0	2	2	0	1	0	0
0	2	2	0	1	0	1
2	2	2	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

Size	0	0	7	0	0	0
index	0	1	2	3	4	5

두더지 굴(S)

알고리즘 실행 과정

- 두더지 = 3
- (0,3), (0,4)는 모두 패스, (0,5)에서 다시 1이 등장하므로 이 점을 기준으로 다시 dfs로 flood fill을 수행하면 상, 하, 좌, 우의 모든 칸들이 3으로 바뀐.
- Size[3]을 방문한 정점의 수인 8로 채우고, 두더지의 값 1 증가.

0	2	2	0	1	0	0
0	2	2	0	1	0	1
2	2	2	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

Size	0	0	7	0	0	0
index	0	1	2	3	4	5

두더지 굴(S)

알고리즘 실행 과정

- 두더지 = 4
- (0,6)부터 (4,0)까지는 1이 하나도 없으므로 모두 패스. (4,1)에서 1이 등장하므로 이 칸으로부터 dfs로 모든 영역을 4로 채움.
- 방문한 정점의 수를 Size[4]에 기록, 두더지 값은 5가 됨.

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

Size	0	0	7	8	0	0
index	0	1	2	3	4	5

두더지 굴(S)

알고리즘 실행 과정

- 두더지 = 5
- 마지막 칸까지 1의 값이 없으므로 모든 작업 종료.
- 세 마리 두더지가 있었고, 각 굴의 크기는 7, 8, 9임을 알 수 있다.

0	2	2	0	3	0	0
0	2	2	0	3	0	3
2	2	2	0	3	0	3
0	0	0	0	3	3	3
0	4	0	0	0	0	0
0	4	4	4	4	4	0
0	4	4	4	0	0	0

Size	0	0	7	8	9	0
index	0	1	2	3	4	5