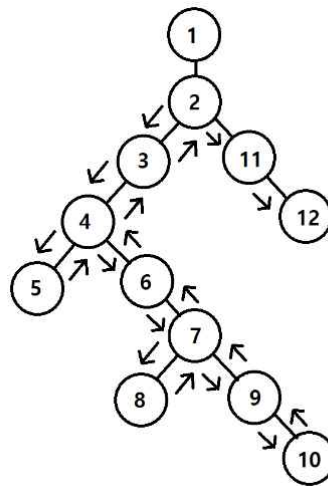
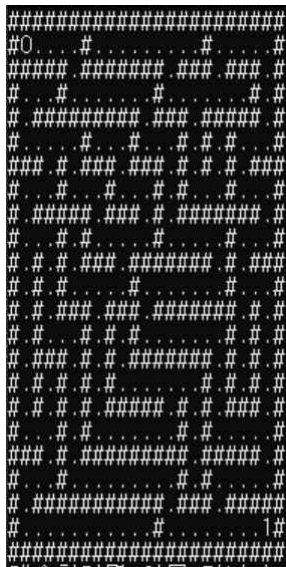


문제. 앞에서 다루지 않은 문제 중 DFS와 BFS로 풀 수 있는 문제를 각각 한가지 씩 예를 들어 설명하시오. 관련된 그래프 구조도 그리시오.

미로 생성은 DFS를 이용하여 구현할 수 있다. DFS는 루트 노드에서 시작하여 다음 branch로 가기 전 그 branch를 완전히 탐색하는 방법이다. 즉 한방향으로 모든 탐색을 마친 후 backtracking을 이용하여 갈림길로 돌아와 다른 방향으로 탐색이 시작된다. 이러한 방식으로 모든 노드를 방문하여 탐색하는데 유리하다. 복잡한 미로를 만들기는 다음 과정을 통해 진행된다.

1)  $n \times m$ 의 2차원 배열을 벽으로 초기화한다. 2) 시작점을 정해 빈공간으로 바꾸고, visit을 표시한다. 3) 4방향중 무작위로 방향을 정해 빈 공간으로 바꾼다. 4) 그 위치를 기준으로 다시 방문하지 않은 공간에 방문하여 빈 공간으로 바꾼다. 5) 더이상 진행 불가능한 경우 3의 과정으로 돌아가 다시 진행한다. 이 방식으로 맵 전체를 채우는 단일 경로의 완벽한 미로를 만들 수 있다.



BFS로 풀 수 있는 대표적인 문제는 미로찾기이다. BFS는 Level order로 가까운 순서대로 찾는 방법이다. 이러한 탐색 특징으로 인해 DFS와 달리 Optimal하다는 특징이 있다.(cost가 1일 때) DFS는 미로에서 출발지에서 목적지까지의 경로를 찾을 순 있지만 최단 경로는 보장이 안도니다. 그에 반해 BFS는 인접 노드를 먼저 탐색하여 미로에서 최단 경로를 찾을 수 있다. 미로에서 최단 경로 찾기는 다음 과정을 통해 이루어진다. 1) Queue에 시작 노드를 넣는다. 2) Queue에 노드 하나를 꺼낸다. 3) 현 위치에서 뻗어 나갈 수 있는 방향을 찾고, 처음 방문하는 경우에만 최단

거리를 기록한다. 4) queue에 기록된 노드를 넣는다. 5) queue가 빌때까지 2부터 반복한다. 이 과정을 통해 미로에서 최단 경로를 찾을 수 있다.

출발	0	0	0	0	0	0
0	1	1	1	1	1	0
0	0	0	1	0	0	0
1	1	0	1	0	1	1
1	1	0	1	0	0	0
1	0	0	0	1	0	0
1	0	1	0	0	0	도착

