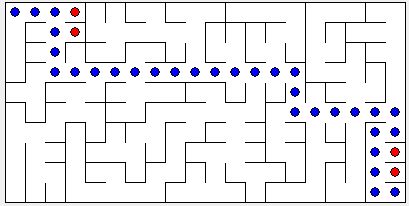
13주차 예비보고서

전공 : 컴퓨터공학 학년 : 2학년 학번 : 20151616 이름 : 최승환

1. DFS와 BFS 알고리즘에 대해 조사하고 간략히 요약한다.

Depth-First Search를 이용하여 미로를 만들 수 있다. 알고리즘은 아래와 같다.

미로찾기의 가장 간단한 알고리즘 중 하나로 DFS 알고리즘이 있다. DFS 알고리즘은 자신의 현재 위치를 기준으로 상,하,좌,우를 확인하며 그 중 가는 것이 가능한 곳이 있으면 그 좌표로 바로 이동한다. 재귀의 방식을 이용하면 이 단순한 알고리즘을 이용해 모든 가능한 경로를 가는 것이 가능하다.

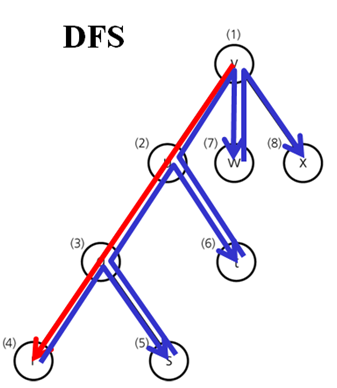


위 이미지에서 붉은색으로 표시된 부분이 막다른 길을 방문한 후 돌아가는 것이다. 만약 특정 좌표에 도착했을 때 아무 길로도 이동이 불가능하면 두 경우 중 하나이다. 목적지에 도착했거나 막다른 길에 봉착한 것이다. 목적지에 도착한 경우 바로 종료시켜주면 된다.

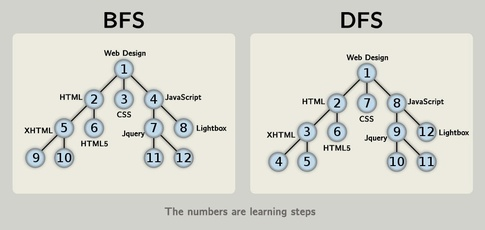
막다른 길에 도착된 경우에는 재귀를 통해 처음 잘못된 선택을 한 위치로 돌아갈 수 있다. 이 곳에서 다음 경로로 이동하는 방식을 채택한 후 다시 길을 따라 간다. 이를 통해 미로를 통과하는 모든 경우의 수를 돌아보게 되고 이 방식들 중에서 목적지에 도착하는 경우를 출력하면 된다.

2. 미로 문제에서 DFS, BFS를 수행하기 위한 자료구조를 설계하고 이의 공간복잡도를 보인다.

DFS는 Depth-First-Search로 트리 자료구조를 이용했을 때 트리의 깊이가 깊은 곳부터 먼저 탐색을 한다는 뜻이다. 이 경우 모든 경우의 수를 차례대로 검색하게 되고, 깊이가 깊은 가지수부터 탐색하게 된다. 이 경우 공간복잡도는 미로에 도착하기까지 모든 좌표만큼의 깊이를 탐색하게 되고 이 경우 공간복잡도는 이 노드들의 개수, 이 경우 O(최단경로의 좌표개수)가 된다. 시간복잡도의 경우 미로의 구조에 따라 크게 다를 수 있다.



BFS의 경우 Breath-First-Search로 같은 깊이에 있는 다른 노드를 먼저 검색하게 된다. 이 경우 공간복잡도는 DFS에 비해 커지게 되고 한 좌표에 대한 가짓수를 a라고 하면 공간복잡도는 약 O(an)이라고 할 수 있다.



3. 설계한 자료구조에서 DFS, BFS를 어떻게 수행할지 간략히 보인다.

일반적으로 미로에서 길을 찾을 때 다음 좌표를 찾을 때 다음과 같은 방법을 쓴다. 다음의 방법을 쓰면 미로를 탐색할 때 이와 같은 방법을 사용할 수 있다.

int dx[4] = {1,0,-1,0};

int dy[4] = {0,1,0,-1};

for(i=0;i<4;++i)

{

search(현 x좌표+dx[i],현 y좌표+dx[i])

}

이를 이용하면 상 하 좌 우를 단 하나의 반복문만을 이용해 탐색할 수 있다.