실험 PRJ-2 미로(Maze) 2주차 예비보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**1. 목적**

DFS와 BFS 알고리즘에 대해서 서술한다. 또한 제작한 미로에서 두 방식을 사용해서 어떻게 경로를 찾아낼 수 있는지 서술한다.

**2. DFS와 BFS 알고리즘**

DFS 알고리즘과 BFS 알고리즘은 Graph 형태에서 모든 vertices를 방문하는 알고리즘 중 한 종류이다. 이때 DFS는 깊이 우선 탐색, BFS는 넓이 우선 탐색이다.

DFS 알고리즘은 갈 수 있는 최대 깊이의 vertex를 먼저 방문하는 형식으로 진행된다. 이는 현재 이동한 vertex가 연결되어 있는(edge) 다른 vertices들을 이전에 모두 방문하기 전까지는, 계속해서 연결되어 있는 다른 vertex로 이동하는 형태이다. 이후에 최대 깊이의 vertex로 오기 이전 vertex로 이동해, 다시 DFS 형태를 진행하는 것으로 진행된다. 이때 DFS 알고리즘에서는 탐색을 진행하며 이전까지 방문한 vertices를 stack에 저장함을 통해서, 이전 vertex에 대한 정보를 저장한다. 그리고 모든 vertices가 stack에 들어왔을 경우에, 모든 연결을 확인해주며 DFS를 종료하는 형태이다.

DFS와 반대로 BFS는 현재 vertex에서 방문할 수 있는(연결되어 있는) 모든 vertices를 방문하고, 이후에 다음 연결로 넘어가는 형식으로 진행된다. 이 방식 또한 stack을 통해서 방문한 vertices를 저장하며, 이전 vertex에 대한 정보를 저장한다. 이후에는 두 가지 방법으로 다음 vertex를 방문할 수 있는데, 현재 같은 width, 예를 들자면, A->B A->C A->D 라고 할 때, B, C, D 에서 맨 처음 방문한 B로 먼저 BFS를 수행할 것인지, 마지막에 방문한 D로 먼저 BFS를 수행할 것인지 선택할 수 있다. 이와 같은 방식으로 BFS를 수행하면, stack에 모든 vertex가 저장되어 있다.

**3. DFS와 BFS를 통해 미로의 경로를 찾아내는 방법**

우선 기본적으로 미로의 경로를 찾아내는 방법은, 위에서 설명한 모든 vertices를 방문하는 것과는 약간의 차이가 존재한다. 우리가 원하는 위치의 vertex를 찾아내기만 하면, 종료할 수 있기 때문이다.

이를 바탕으로 우선 DFS를 사용한 방식으로는, 원하는 위치의 vertex를 찾아내기 이전까지 DFS를 수행하며, 각 위치에서 가능한 이동은 벽이 없는 오른쪽, 위, 왼쪽, 아래 방향 총 4방향이 가능하다. 이를 통해서 각 vertex마다 4개의 연결이 있는 형태로 DFS를 수행해주면, stack과 함께, 미로의 경로를 찾아낼 수 있다. 추가로 2. 에서 설명한 stack과 함께 새로운 stack을 정의해주어야 한다. 우리가 실제로 필요한 stack은(새롭게 정의해주는 stack) 경로만 존재하면 되므로, 최대 깊이의 node에 대해서는 stack에서 삭제를 수행해주며 방문해주면 경로를 알아낼 수 있다.

BFS를 사용한 방식 또한 위의 DFS와 같이 두 종류의 stack을 바탕으로 저장해낼 수 있다.