13주차 예비보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20211558 이름 : 윤준서

**1. 시간 복잡도와 공간 복잡도**

**- DFS -**

미로의 모든 칸을 하나의 노드로 가정했을 때, 노드의 수는 WIDTH \* HEIGHT 이다. 그리고 각 노드에서 연산을 실행할 때 상, 하, 좌, 우 네 방향을 모두 진행하므로 4 \* (노드의 수) 이다. 따라서 전체 시간 복잡도는 O(WIDTH \* HEIGHT) 이다.

방문 배열 visit와 스택 m\_block 모두 최악의 경우 모든 노드를 다뤄야 하므로 필요한 공간의 수는 WIDTH \* HEIGHT 이다. 따라서 전체 공간 복잡도는 O(WIDTH \* HERIGHT) 이다.

**- BFS -**

미로의 모든 칸을 하나의 노드로 가정했을 때, 노드의 수는 WIDHT \* HEIGHT 이다. 그리고 각 노드에서 연산을 실행할 때 상, 하, 좌, 우 네 방향을 모두 진행하므로 4 \* (노드의 수) 이다. 따라서 전체 시간 복잡도는 O(WIDTH \* HEIGHT) 이다.

방문 배열 visit와 큐 Q 모두 최악의 경우 모든 노드를 다뤄야 하므로 필요한 공간의 수는 WIDTH \* HEIGHT 이다. 따라서 전체 공간 복잡도는 O(WIDTH \* HEIGHT) 이다.

**2. 경로를 찾는 과정**

**- DFS -**

***1.*** 2차원 방문 벡터 visit를 초기화 한다. 모든 값을 false로 초기화 하는데, maze를 차례로 방문하여 벽인 경우 true로 수정한다. 따라서 크기 또한 maze와 같다.

***2.*** 두 개의 스택 m\_block, d\_path를 초기화한다. 이 때 d\_path는 최종 경로를 저장하는 스택이다. 즉 주 탐색의 주체는 m\_block이므로 시작점의 좌표 (1, 1)을 m\_block에 추가한다.

***3.*** 이동 방향을 나타내는 배열을 생성한다. x\_move = {1, -1, 0, 0}, y\_move = {0, 0, 1, -1}로 정의한다.

***4.*** 본격적으로 DFS 탐색을 시작한다. while문을 통해 m\_block 스택이 빌 때까지 탐색을 진행한다.

***5.*** 모든 탐색 경로를 저장하는 ad\_path에 현재 경로를 저장한다.

***6.*** 적절한 경로를 찾았을 때 이를 d\_path에 저장하고 true를 반환한다.

***7.*** 다음으로 이동 가능 여부를 나타내는 변수 possible을 생성한다. 다음 네 방향으로 이동하는 좌표 (newX, newY)를 가지는 노드 next를 생성한다. 이를 m\_block 스택에 추가하고 해당 위치를 방문했음을 나타내기 위해 visit을 수정한다. 이동이 가능함을 나타내므로 possible = true 로 수정한다.

***8.*** 만약 possible == false인 경우, 즉 이동이 불가능한 경우 현재 위치를 스택에서 제거한다. 따라서 이전 위치로 다시 돌아와 다른 방향으로 탐색을 재게한다.

**- BFS -**

***1.*** 2차원 방문 벡터 visit를 초기화 한다. 모든 값을 false로 초기화 하는데, maze를 차례로 방문하여 벽인 경우 true로 수정한다. 따라서 크기 또한 maze와 같다.

***2.*** 큐 Q를 초기화 하고, 시작점의 좌표 (1, 1)를 추가한다. 각 노드 마다 부모 노드를 생성하여 parent를 생성, 초기화 한다. 이 때 시작점의 부모 노드의 좌표를 (-1, -1)로 설정한다.

***3.*** 이동 방향을 나타내는 배열을 생성한다. x\_move = {1, -1, 0, 0}, y\_move = {0, 0, 1, -1}로 정의한다.

***4.*** 그리고 두 경로를 초기화하는데, 모든 방문 경로를 저장하는 all\_visit, 최단 경로를 저장하는 a\_path를 초기화한다.

***5.*** 본격적으로 BFS 탐색을 시작한다. while문을 통해 큐가 완전히 빌 때까지 탐색을 진행한다. 진행 중 큐의 맨 앞 원소를 제거한다. 현재 위치를 all\_visit에 추가하고, 경로를 추적한다.

***6.*** 경로 추적이 완료되면 다음으로 네 방향으로 이동하는 새로운 좌표 (newX, newY)를 가지는 노드 next를 생성한다. 이 노드를 큐에 추가하고 해당 위치를 방문했음을 남기기 위해 visit를 수정한다. 그리고 기존 노드를 부모 노드로써 저장한다.