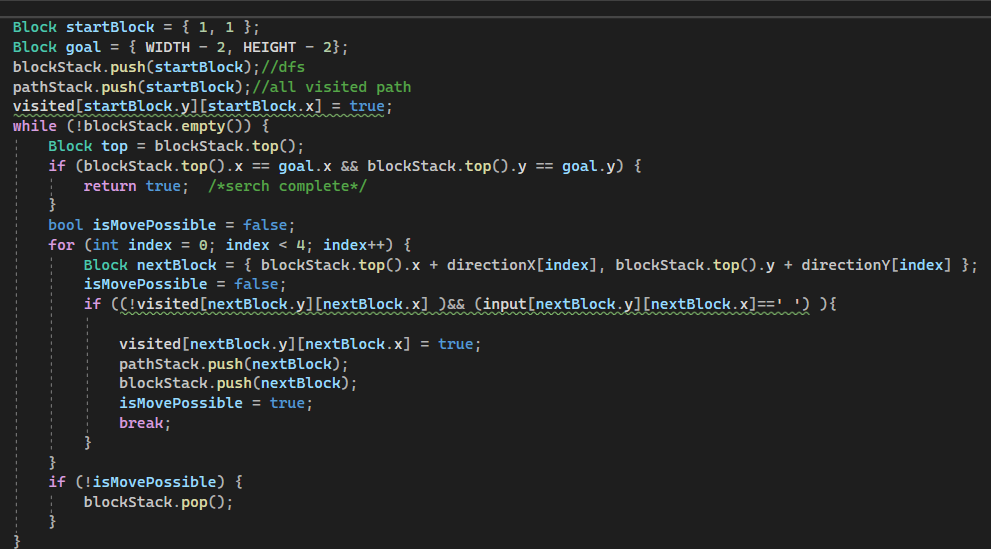
13주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20202106 이름: OSHIMA ASUKA

1. startBlock(1,1)부터 탐색을 시작하여 goal(WIDTH-2,HEIGHT-2)에 도달하면 탐색이 끝난다. 스택을 blockStack: 최소탐색경로를 저장하는 스택과 pathStack: 방문하는 모둔 경로를 저장하는 스택을 만들었다. 먼저 visited(방문 여부를 확인하는 배열)을 false로 초기화했고, 미로 시작점은 true로 했다. Start위치에서 상하좌우 방향에 대해 이동여부와 방문여부를 확인하여 방문하기 전이고 벽이 아닌 공백인 경우에는 visited를 true로 하여 방문을 마친 상태로 하고 각 스택의 저장한다. 또 현재 있는 위치에서 이동 가능하는 여부를 검사하는 isMovePossible를 true로 한다. 만약 isMovePossible이 false인 경우 blockStack에서 pop를 하여 삭제한다. 시간 복잡도는 모드의 방을 방문하는 가능이 있으며 O(WIDTH\*HEIGHT)가 되고 공간 복잡도도 O(WIDTH\*HEIGHT)이다. 이는 실습을 하기 전과 같다.



그림

1. BFS는 방문하지 않은 어떤 노드부터 인접할 모든 노드를 방문하기 때문에 DFS보다 방문한 노드가 많아지고 시간이 걸리지만 최적 경로를 찾기 위해서는 좋은 방법이다. DFS는 경로를 찾는 것에 대해서 는 충분하고 좋은 방법이지만 그 경로가 최적이라고 말할 수 없다. 만약 미로의 경로가 하나가 아닌 경우 최적이 아닌 경로를 찾는 가능성이 있다. 하지만 노드를 저장하는 메모리 양은 BFS보다 작다 점이 장점이다.