13주차 예비보고서

전공 : 경영학과 학년 : 4학년 학번 : 20190808 이름 : 방지혁

1. DFS와 BFS의 시간 복잡도를 계산하고 그 과정을 설명한다.

DFS는 그래프의 모든 vertex를 방문하고, 각 vertex에서 adjacent한 edges를 탐색한다. 각 정점을 한 번씩 방문하기에 O(V)의 시간복잡도가 소요되고, 각 간선을 한 번씩 확인하기에 O(E)라고 볼 수 있다. 총 시간복잡도는 O(V + E)가 된다. **BFS**도 마찬가지다. 각 정점을 큐에 넣고 빼는 작업은 O(V), 각 간선을 확인하는데 O(E)가 소요되기에 최종적으로는 O(V + E)가 된다. 이는 모두 인접 리스트를 사용할 때이며, 인접 행렬을 사용하면 O(V^2)이 된다.

2. 자신이 구현한 자료구조 상에서 DFS와 BFS 방법으로 실제 경로를 어떻게 찾는지 설명한다. 특히 DFS 알고리즘을 iterative한 방법으로 구현하기 위한 방법을 생각해보고 제시한다.

DFS

2주차 실습까지 미로를 직접 만든 구조체의 2차원 배열로 표현했다. 해당 자료구조는 4방향 벽 정보를 Boolean 변수로 저장하고 있다. 그렇지만 이를 인접 리스트 형태로 변환해야 겠다는 생각이 들었다. 각 셀은 그래프의 vertex가 되고, 벽이 없는 인접 셀의 경우 연결시키면 된다.

DFS의 경우 iterative하게 하면 미로 크기가 커질 경우 stack overflow가 발생할 수 있기에 iterative하게 구성해보도록 한다. 방문 체크 배열도 추가하여 무한 루프를 방지해야 한다. 또한, 현재까지의 탐색 경로를 저장할 스택도 필요하고 부모를 저장하는 배열도 추가하여 최종적으로 도착지점에 도달하면 역추적하여 최종 경로를 그릴 수 있도록 해야 한다.

알고리즘에 대해 설명하자면 우선 앞선 자료구조들을 초기화시키고 시작점과 목표점을 설정해야 한다. 시작점을 스택에 넣고 방문 표시를 해준다. 그리고 스택이 비기전까지 루프를 도는데 목표 지점에 도달하면 종료하고, 방문하지 않는 인접 노드를 스택에 넣는다. 넣을 인접노드가 없을시 스택에서 빼 다시 이전 노드로 돌아간다. 이렇게 계속 루프를 돌린다. 그리고 끝나면 부모를 저장하는 배열을 통해 최종 경로를 역추적한다.

BFS의 경우 큐를 사용하는 것이 큰 차이다. 큐에 시작점을 넣고 루프를 돌린다. 큐의 가장 앞에 있는 노드를 꺼내 목표 지점에 도달했는지 검사하고 모든 인접 노드를 큐에 넣는다. 이게 DFS와 다르다. 왜냐하면 BFS는 말그대로 너비우선탐색이기 때문이다. 그리고 루프가 끝나면 최종 경로를 역추적한다.