13주차 예비보고서

전공: 아트&테크놀로지학과 학년: 4학년 학번: 20191048 이름: 김도솔

**1.**

DFS는 깊이 우선으로 그래프를 탐색하는 알고리즘이다. 재귀적인 방식으로 동작하며, 더 이상 진행할 수 없는 경우 되돌아가서 다른 경로를 탐색한다. 그래프의 모든 노드를 방문하기 위해서는 최악의 경우 그래프의 모든 간선을 한 번씩만 방문하면 된다. BFS는 너비 우선으로 그래프를 탐색하는 알고리즘이다. 시작 노드에서부터 차례대로 인접한 노드를 탐색하며, 더 이상 방문하지 않은 노드를 큐에 저장하고 처리한다. DFS와 마찬가지로 그래프의 모든 노드를 방문하기 위해서는 최악의 경우 그래프의 모든 간선을 한 번씩만 방문하면 된다. 따라서 두 알고리즘의 시간 복잡도는 동일하며 노드 수를 V, 간선 수를 E라고 할 때, 각 노드와 간선을 한 번씩만 방문하기 때문에 O(V + E)이다.

**2.**

⚫ 12주차 실습에서 구현한 자료구조: 2차원 벡터

⚫ DFS(Depth-First Search)로 경로 찾기: DFS는 깊이 우선 탐색으로, 스택(Stack) 자료구조를 사용하여 구현할 수 있다. DFS 알고리즘을 iterative한 방법으로 구현하려면 스택을 명시적으로 사용하여 방문할 노드를 관리하는 방식을 사용하면 된다. 예를 들어 만약 연결된 노드 w가 아직 방문되지 않았다면, dfs 함수를 재귀적으로 호출하여 w->vertex 노드부터 DFS를 진행하는 식이다. 구현 방법은 다음과 같다.

1. 시작점과 도착점의 좌표를 설정한다.
2. visited라는 2차원 배열을 사용하여 노드의 방문 여부를 기록한다.
3. 스택을 이용하여 DFS를 구현한다.
4. 스택에 시작점의 좌표를 넣고, 스택이 비어있지 않은 동안 다음 작업을 반복한다.

스택에서 노드를 꺼낸다.

현재 노드의 좌표를 확인하고, 도착점에 도달했는지 확인한다.

현재 노드를 방문 처리한다.

현재 노드와 인접한 미방문 노드가 있다면 스택에 넣는다.

1. 만약 도착점에 도달했다면 경로를 찾은 것으로 true를 반환한다. 그렇지 않으면 false를 반환한다.

⚫ BFS(Breadth-First Search)로 경로 찾기: BFS는 너비 우선 탐색으로, 큐(Queue) 자료구조를 사용하여 구현할 수 있다. 구현 방법은 다음과 같다.

1. 시작점과 도착점의 좌표를 설정한다.
2. visited라는 2차원 배열을 사용하여 노드의 방문 여부를 기록한다.
3. 큐를 이용하여 BFS를 구현한다.
4. 큐에 시작점의 좌표를 넣고, 큐가 비어있지 않은 동안 다음 작업을 반복한다.

큐에서 노드를 꺼낸다.

현재 노드의 좌표를 확인하고, 도착점에 도달했는지 확인한다.

현재 노드를 방문 처리한다.

현재 노드와 인접한 미방문 노드가 있다면 큐에 넣는다.

1. 만약 도착점에 도달했다면 경로를 찾은 것으로 true를 반환한다. 그렇지 않으면 false를 반환한다.