

## 13주차 예비보고서

전공: 컴퓨터공학과

학년: 2학년

학번: 20231632

이름:

Jumagul Alua

1. 그래프는 일반적으로 두가지로 나눌 수 있다.

정점을 리스트로 나타내는 Adjacency List와 정점 간의 연결을 2차원 배열로 나타내는 Adjacency Matrix이 있다.

- 깊이 우선 탐색을 하는 DFS는 최대 멀리까지 시작하고 켜다다 다시 돌아와 그 점에서 다시 탐색을 계속한다. 인접 리스트에는 각 정점을 한번만 방문하고 간선을 한번 확인하는 것이라 정점 수를  $M$ 이라 하고 간선 수를  $N$ 이라 했을때 시간 복잡도는  $O(M+N)$ 이 된다. 인접 행렬에는 리스트와 비슷하게 각 정점을 한번 방문하고, 각 장점을 다른 장점과 확인하는 작업을 한다. 각 정점에 대해  $M$ 번을 확인하여,  $O(M^2)$ 의 시간복잡도를 갖는다.
- 그래프의 너비 우선으로 탐색하는 BFS 탐색도 DFS와 유사한 작업을 하므로써, 정점 수를  $M$ 이라 하고 간선 수를  $N$ 이라고 하여 인접 리스트에는  $O(M+N)$  시간 복잡도가 되며, 인접 행렬에도 똑같이  $O(M^2)$ 의 시간복잡도를 갖는다.

2. DFS:

DFS를 반복적 방법으로 구현하려면 스택을 이용할 것이다. 여기서는 스택으로 deque를 사용하여 경로를 추적하고, 방문한 노드를 기록하기 위해 2차원 deque를 사용할 예정이다. 일단, 미로 시작점을 선언해주고, 현재 위치를 찾아서, 가능한 모든 방향으로 이동을 해본다. 각 이동이 유효한지 확인하고, 방문하지 않은 위치이면 스택에 넣고 기록하는 작

업을 할 것이다. 그리고 목표 위치에 도달하면 경로를 저장하고 탐색을 스택이 비워질 때 까지 계속해준다.

BFS:

BFS를 구현하려면 큐를 사용하여 탐색을 관리할 수 있다. `deque`를 큐로 사용하여 각 단계에서 방문할 노드를 추적할 예정이다. BFS 탐색을 할 때 DFS를 구현한 원리로 탐색을 한다. 하지만, DFS에서 스택을 이용해 해준다면, BFS에는 큐 안에서 이동하면 비워질 때 까지 계속해서 해준다는 차이거 있다.