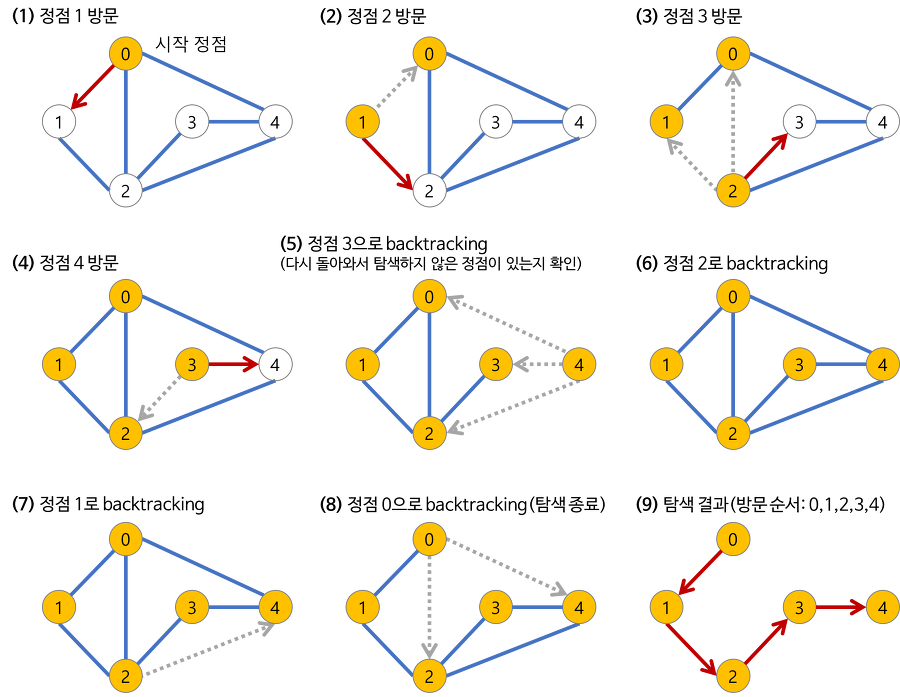
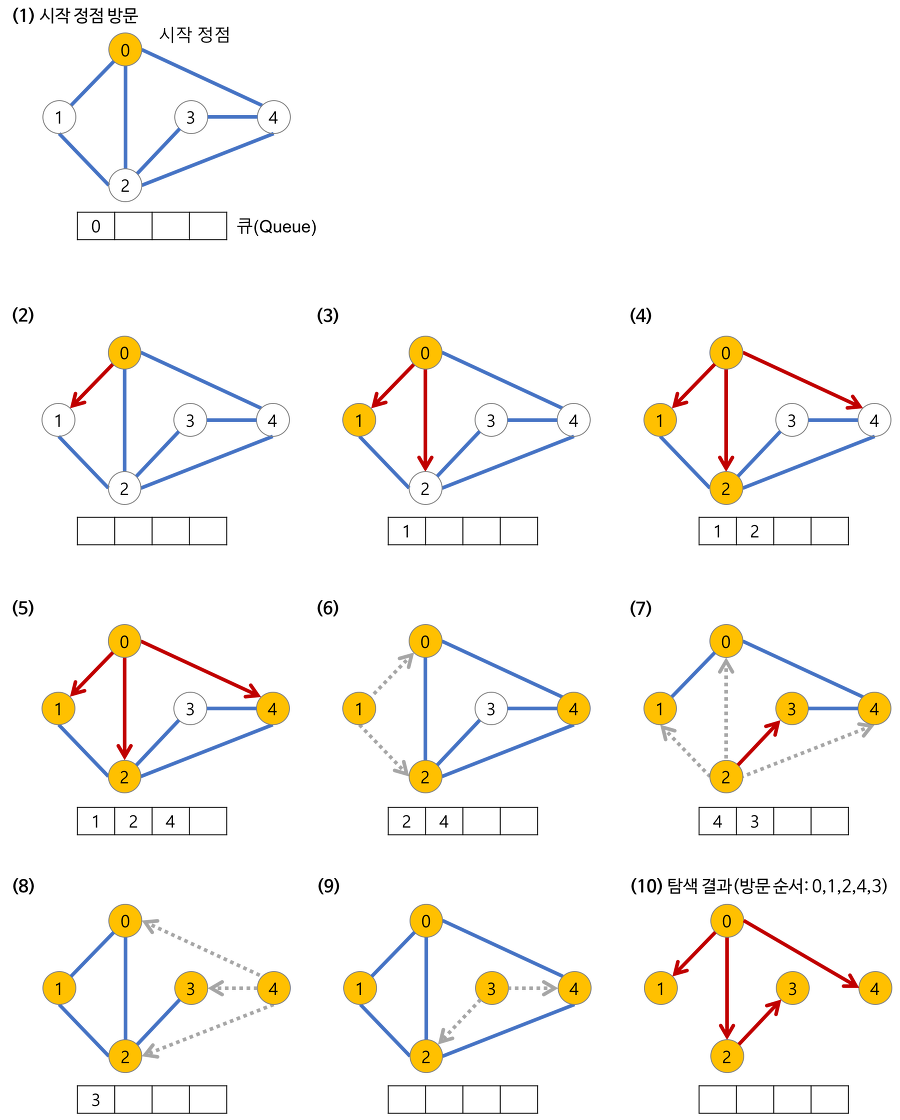
**학과: 컴퓨터공학과 학년: 2 학번: 20191657 이름:최세은**

1. **DFS와 BFS 알고리즘에 대해 조사하고 간략히 요약한다.**

DFS(Depth First Search): 깊이 우선 탐색. 한 루트 노드에서 시작해서 다음 branch로 넘어가기 전 해당 분기를 완벽하게 탐색하는 방법이다. 따라서 넓게 탐색하기 전에 깊게 탐색하는 방법이다.



BFS(Breadth First Search): 너비 우선 탐색. 루트 노드에서 시작해 인접한 노드를 먼저 탐색하는 방법이다. 따라서 깊게 탐색하기 전에 넓게 탐색하는 방법이다.



1. **미로문제에서 DFS, BFS를 수행하기 위한 자료구조를 설계하고 이의 공간복잡도를 보인다.**

미로 문제에서 DFS, BFS를 사용하면 미로를 어떻게 빠져나가는지 알아낼 수 있다. 미로의 각 방들을 노드, 미로를 이동하는 것을 노드를 간선으로 연결하는 것으로 볼 수 있기 때문이다. 따라서 DFS를 수행하기 위해 자료구조로 stack을, BFS를 수행하기 위해 자료구조로 queue를 사용한다. 각 노드 안에는 해당 노드의 x, y 좌표가 들어있으며, 미로에서는 상하좌우로만 움직일 수 있으므로 이동할 수 있는 좌표도 포함한다. 이동할 수 있는 좌표는 배열로 구현한다. 따라서 이중 노드 배열을 사용하므로 공간복잡도는 O(n \* m)이다.

1. **설계한 자료구조에서 DFS, BFS를 어떻게 수행할지 간략히 보인다.**

DFS: 시작 위치부터 가능한 최대로 내려가며 stack에 방문한 위치의 좌표를 포함하는 노드를 push 한다. 그러다 더 이상 내려갈 수 없다면 오른쪽으로 가능한 최대로 내려가며 노드들을 또 push 한다. 이렇게 최대한 아래로-최대한 오른쪽으로 이동하는 것을 반복해서 더 이상 이동할 수 없을 경우 stack의 노드를 pop하며 이전 노드부터 이동을 재개하고 이러한 과정을 도착할 때까지 반복한다.

BFS: 시작 위치부터 오른쪽으로 최대한 이동하며 DFS와 마찬가지로 방문한 위치의 좌표를 포함하는 노드를 push 한다. DFS와는 다르게 최대한 오른쪽으로-최대한 아래로 이동하는 것을 반복하며 queue에 이동할 때마다 노드를 push 해주고, 더 이상 이동할 수 없을 때는 pop하며 이전 노드부터 이동을 재개하고 이를 도착할 때까지 반복한다.