그래프 탐색이란

하나의 정점으로부터 시작하여 차례대로 모든 정점들을 한 번씩 방문하는 것

Ex) 특정 도시에서 다른 도시로 갈 수 있는지 없는지 탐색

깊이 우선 탐색(DFS, Depth-First Search)

깊이 우선 탐색이란

시작 노드(임의 지정 노드)에서 시작해서 다음 분기(branch)로 넘어가기 전에 해당 분기를 완벽하게 탐색하는 방법

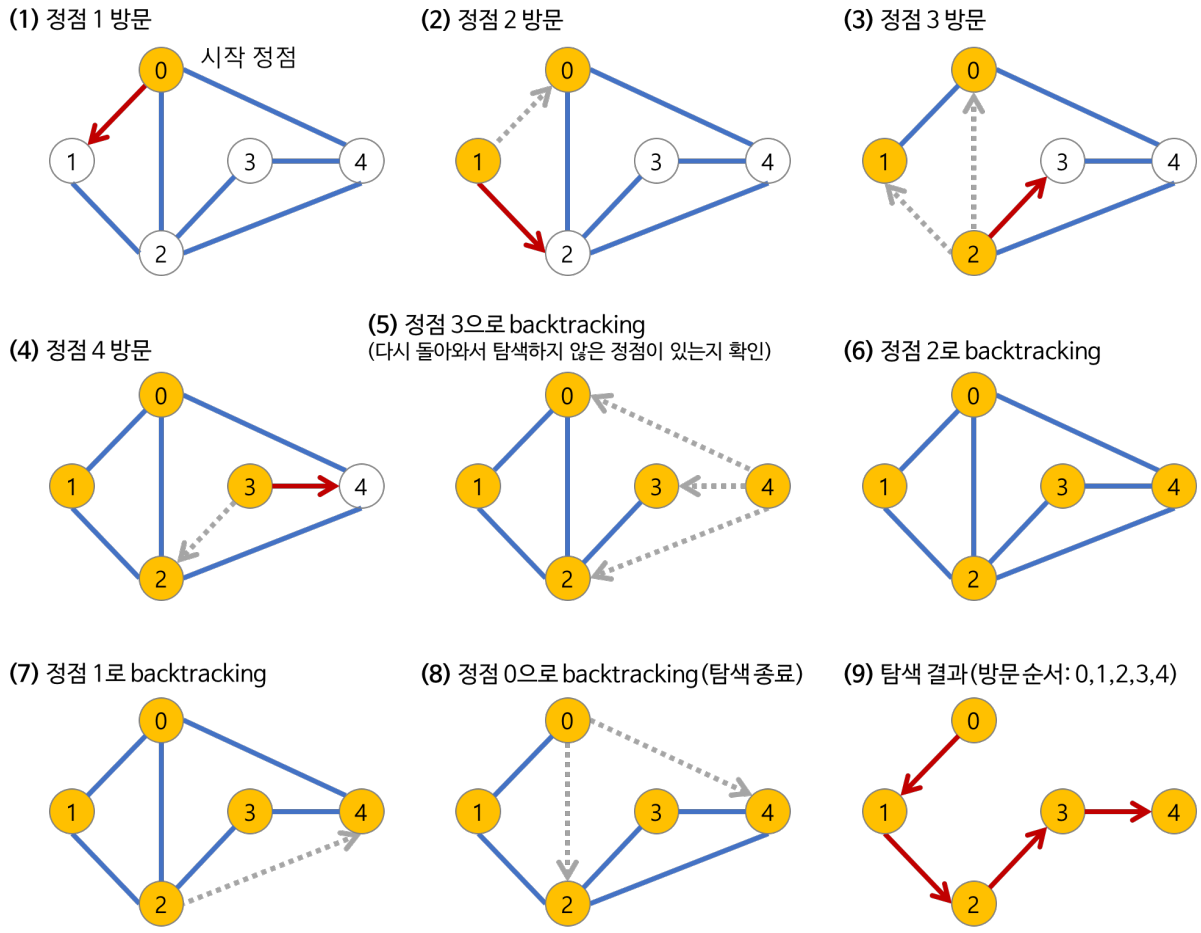
미로를 탐색할 때 한 방향으로 갈 수 있을 때까지 계속 가다가 더 이상 갈 수 없게 되면 다시 가장 가까운 갈림길로 돌아와서 이곳으로부터 다른 방향으로 다시 탐색을 진행하는 방법과 유사하다.

깊이 우선 탐색(DFS)의 특징

자기 자신을 호출하는 순환 알고리즘의 형태를 가지고 있다.

트리 순회는 모두 DFS의 한 종류이다.

이 알고리즘을 구현할 때 그래프 탐색의 경우 어떤 노드를 방문했었는지 여부를 반드시 검사 해야 한다는 것이다. 이를 검사하지 않을 경우 무한 재귀에 빠질 위험이 있다.



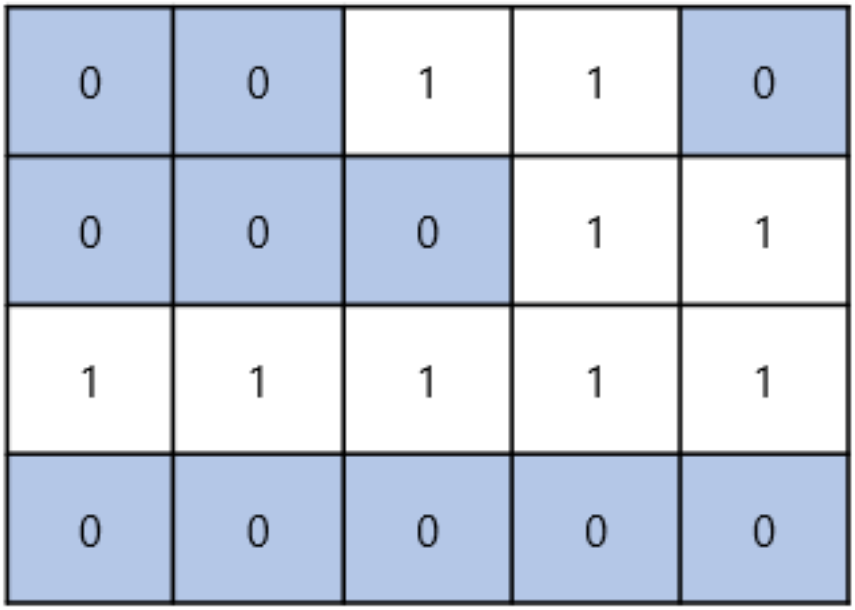
```python
def dfs(graph,v,visited):
    visited[v] = True
    print(v,end=' ')
    for i in graph[v]:
        if not visited[i]:
            dfs(graph,i,visited)

graph = [
    [],
    [2,3,8],
    [1,7],
    [1,4,5],
    [3,5],
    [3,4],
    [7],
    [2,6,8],
    [1,7]
]
visited = [False] * 9
dfs(graph,1,visited)
```

깊이 우선 탐색(DFS)의 시간 복잡도

DFS는 그래프(노드의 수: N, 간선의 수: E)의 모든 간선을 조회한다.
인접 리스트로 표현된 그래프: O(N+E)
인접 행렬로 표현된 그래프: O(N^2)

즉, 그래프 내에 적은 숫자의 간선만을 가지는 희소 그래프(Sparse Graph) 의 경우 인접 행렬보다 인접 리스트를 사용하는 것이 유리하다.

백준 DFS 문제 음료수 얼려먹기
N x M 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1 로 표시된다. 구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다. 이때 얼음 틀의 모양이 주어졌을 때 생성되는 총아이스크림의 개수를 구하는 프로그램을 작성하시오.

- 첫 번째 줄에 얼음 틀의 세로 길이 N 과 가로길이 M 이 주어진다. (1 <= N, M ,= 1,000)
- 두 번째 줄부터 N + 1 번째 줄까지 얼음 틀의 형태가 주어진다.
- 이때 구멍이 뚫려있는 부분은 0, 그렇지 않은 부분은 1 이다.
- 

0	0	1	1	0
0	0	0	1	1
1	1	1	1	1
0	0	0	0	0

TIP) 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1 이므로 위의 얼음틀 예시에서는 총 3 개의 아이스크림이 생성된다. 이 문제는 DFS 를 통해 '0'인 값이 상, 하, 좌, 우로 연결되어 있는 노드를 묶어 구할 수 있다.