



완전탐색 알고리즘 (DFS)

완전탐색

정의 - 가능한 모든 경우를 전부 시도하여 정답을 찾아내는 방법. for & if 문을 활용하여 처음부터 끝까지 탐색하는 방법이 있고, 대표적으로 그래프 탐색으로서 DFS(깊이 우선 탐색) 과 BFS(너비 우선 탐색) 이 있다.

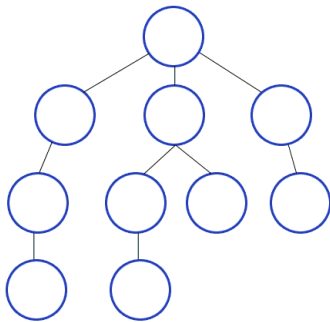
그래프 탐색

정의 - 하나의 정점으로부터 시작하여 차례대로 모든 정점들을 한 번씩 방문하는 탐색방법

DFS

정의 - 하나의 정점으로부터 가장 깊숙한 곳 까지 들어갔다가, 더 이상 들어갈 곳이 없으면 최정점으로 다시 돌아가서 다음 브랜치를 전과 같은 방법을 통하여 탐색한다. 여기서 주의할점으로는 최정점으로 돌아가기 위해 방문하였음을 체크해 주는것이 중요하다.

DFS의 탐색과정



1. 1번 노드(시작 노드)를 방문한다.
 - a. 방문한 노드는 방문했다고 표시한다.
2. 1번 노드와 인접한 노드들을 차례로 순회한다.
 - a. 1번 노드와 인접한 노드가 없다면 종료한다.
3. 1번 노드와 이웃한 노드 2번 노드를 방문했다면, 1번노드와 인접한 또 다른 노드를 방문하기 전에 2번노드의 이웃 노드들을 전부 방문해야 한다.
 - a. 2번노드를 시작 정점으로 DFS를 다시 시작하여 2번노드의 이웃 노드들을 방문한다.
4. 2번노드의 분기를 전부 완벽하게 탐색했다면 다시 1번노드에 인접한 정점들 중에서 아직 방문이 안 된 정점을 찾는다.
 - a. 즉, 2번노드의 분기를 전부 완벽하게 탐색한 뒤에야 1번노드의 다른 이웃 노드를 방문할 수 있다는 뜻이다.
 - b. 아직 방문이 안 된 정점이 없으면 종료한다.
 - c. 있으면 다시 그 정점을 시작 정점으로 DFS를 시작한다.

타겟 넘버 그래프


```

        answer++;
    }
    return;
}
// 재귀함수 : 만약 더 파고들 깊이가 있다면, 다음 깊이를 처음 인자에 세팅을 해주고 다음 인자로는 +로 가냐, -로 가냐 로 구분지어서 각각 보내준다.
dfs(level + 1, sum + numbers[level]);
dfs(level + 1, sum - numbers[level]);
}

dfs(0, 0);

return answer;
}

console.log(solution([1, 1, 1, 1, 1], 3));

```