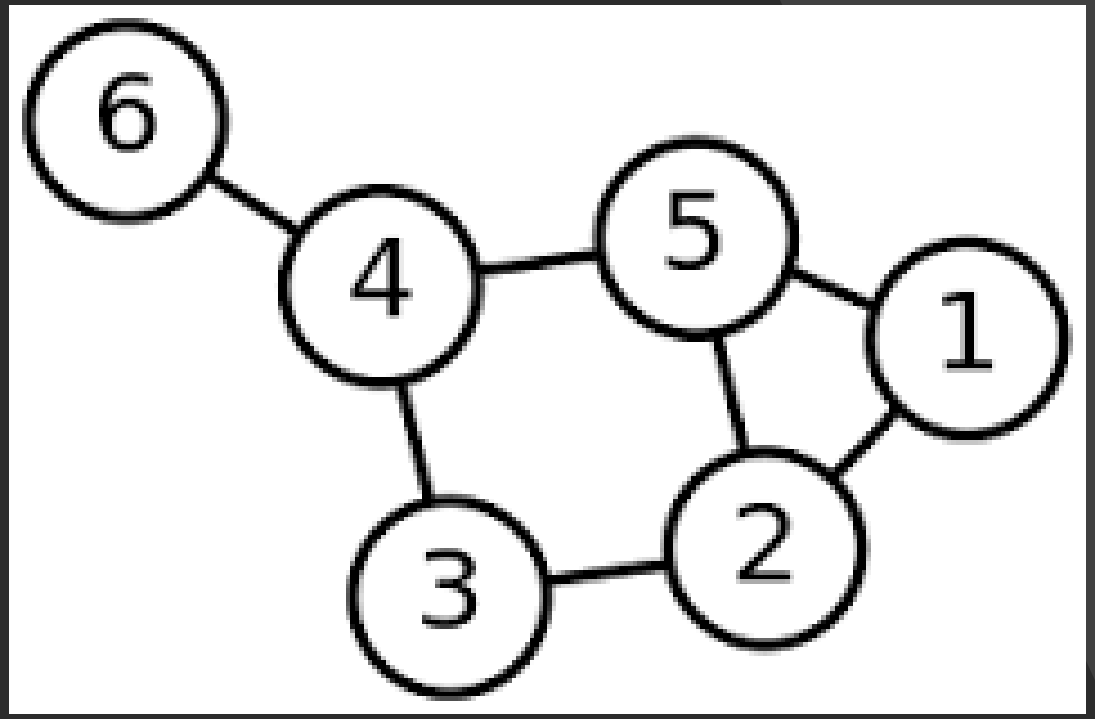
3 주차

DFS & BFS

그래프

- 노드와 노드를 연결하는 간선을 하나로 모아놓은 자료구조
- 방향그래프와 무방향그래프로 나뉨
- 사이클 가능, 자체 간선도 가능
- 루트 노드, 부모-자식의 개념이 없음
- 순회 방법 : DFS, BFS
- 간선의 수가 제한되어있지 않음

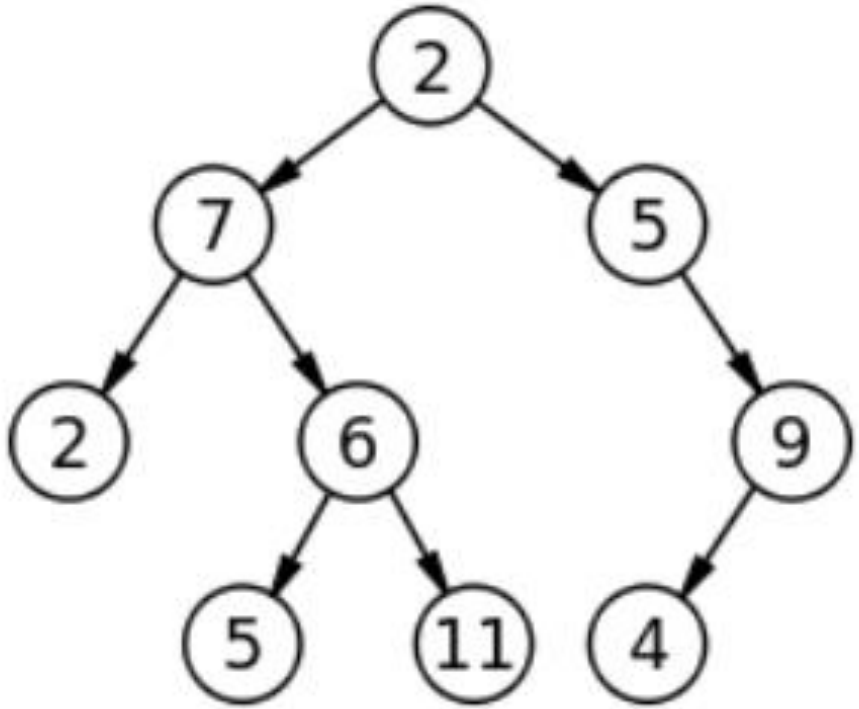


그래프 용어

- 정점, 간선
- 무향그래프/유향그래프
- 인접 정점
- 정점의 차수
- 사이클

트리

1. 연결 그래프이다. (컴포넌트가 하나이다.)
2. 방향을 무시하였을 때, 싸이클이 존재하지 않는다.
3. 트리의 간선 개수는 반드시 n – 1이다.
4. 계층관계 (부모-자식관계)
5. 순회 방법 : DFS, BFS, 전위순회, 중위순회, 후위순회



그래프의 구현방법(1) - 인접리스트

- 가장 일반적인 방법
- vector, arraylist를 이용해서 구현
- 어떤 노드에 인접한 노드들을 쉽게 찾을 수 있다.
- 모든 간선의 수를 O(N+E)안에 알 수 있다.
- 간선의 존재 여부를 알기 위해선 정점의 차수만큼의 시간이 필요

```
vector<int> graph[50];

graph[2].push_back(3);
graph[3].push_back(2);
```

그래프의 구현방법(2) – 인접행렬

- 간선이 많은 밀집 그래프의 경우 사용
- 두 정점을 연결하는 간선의 여부를 O(1)만에 알 수 있다.
- 정점의 차수는 O(N)만에 알 수 있다.
- 어떤 노드에 인접한 노드를 찾기 위해선 모든 노드를 순회해야한다.
- 그래프에 존재하는 모든 간선의 수는 O(N^2)만에 알 수 있다.

```
bool graph[50][50];

graph[2][3] = true;
graph[3][2] = true;
```

DFS(깊이 우선 탐색)

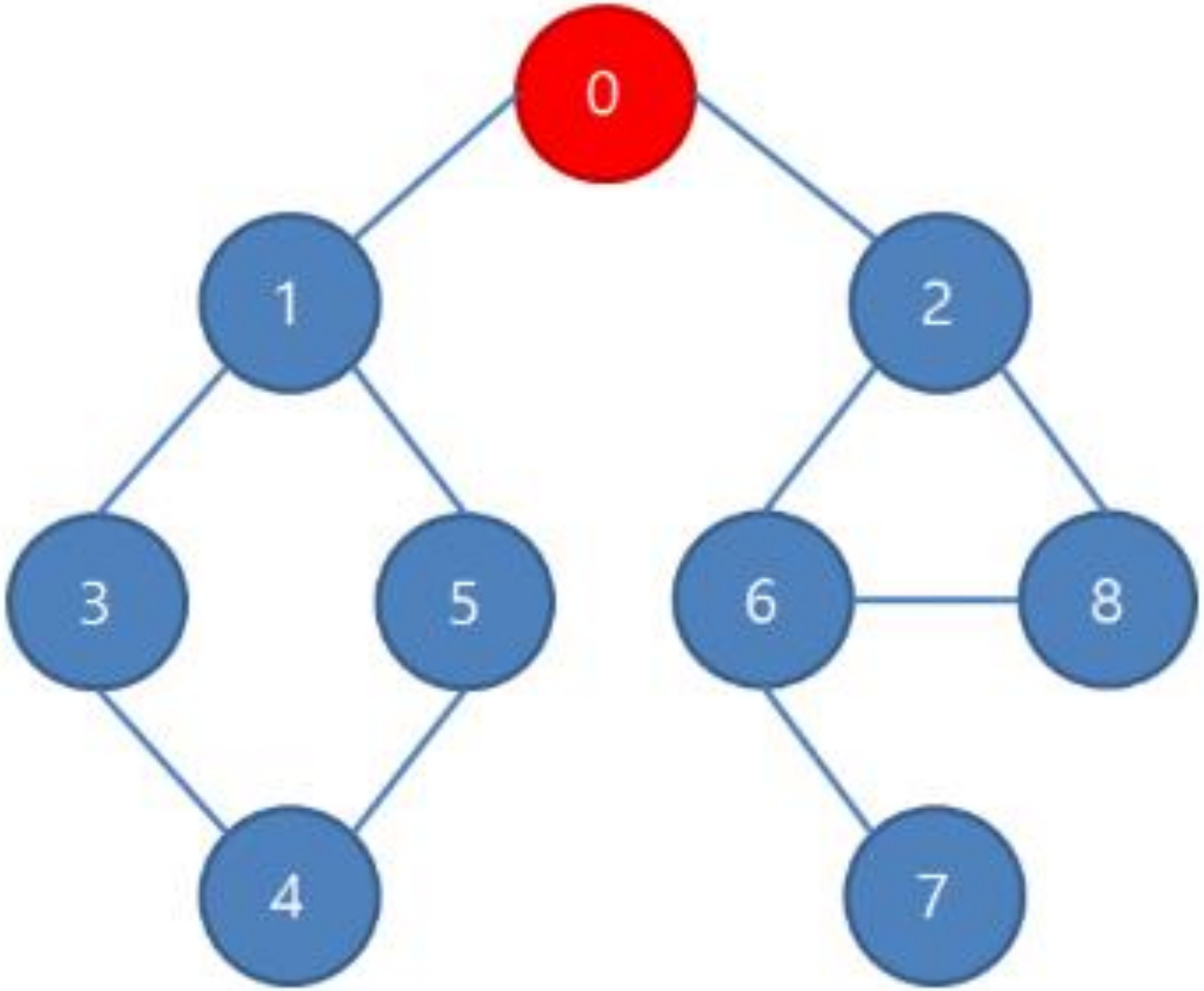
- 루트 노드(혹은 다른 임의의 노드)에서 시작해서 다음 분기(branch)로 넘어가기 전에 해당 분기를 완벽하게 탐색하는 방법
- 재귀를 이용해서 구현
- 모든 노드를 방문하고자 하는 경우 사용
- 시간 복잡도 : O(N)

```
1    void dfs(int curr){
2        visited[curr] = true;
3        cout << "node " << curr << " visited" << endl;
4        for(int next: adj[curr])
5            if(!visited[next]) dfs(next);
6    }
```

BFS(넓이 우선 탐색)

- 루트 노드(혹은 다른 임의의 노드)에서 시작해서 인접한 노드를 먼저 탐색하는 방법
- Queue를 이용해서 구현
- 두 노드 사이의 최단 경로 혹은 임의의 경로를 찾고 싶을 때 사용
- 시간 복잡도 : O(N)

```cpp
// 너비 우선 탐색
void bfs(){
    vector<bool> visited(N, false); // 방문 여부를 저장하는 배열
    queue<int> Q;
    Q.push(0);
    visited[0] = true;
    // 탐색 시작
    while(!Q.empty()){
        int curr = Q.front();
        Q.pop();
        cout << "node " << curr << " visited" << endl;
        for(int next: adj[curr]){
            if(!visited[next]){
                visited[next] = true;
                Q.push(next);
            }
        }
    }
}
```


0
1
2
3
5
6
8
4
7

DFS실습(1) – 연결 요소의 개수

- https://www.acmicpc.net/problem/11724

DFS실습(2) – 유기농 배추

- https://www.acmicpc.net/problem/1012

BFS실습(1) – 촌수계산

- https://www.acmicpc.net/problem/2644

BFS실습(2) – 미로탐색

- https://www.acmicpc.net/problem/2178

백트래킹(1) – 암호 만들기

- https://www.acmicpc.net/problem/1759

백트래킹(2) – 알파벳

- https://www.acmicpc.net/problem/1987