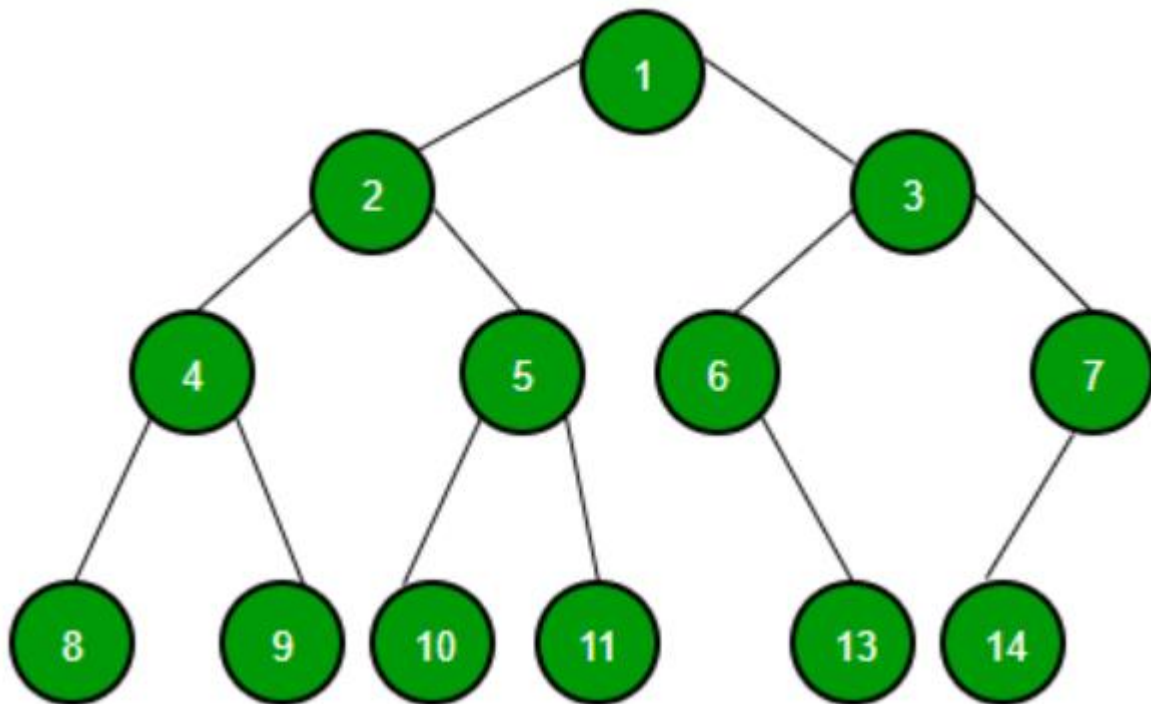


<Tree>

- 예제 코드 : <https://github.com/limes22/algorithm/blob/main/go/tree/tree.go>

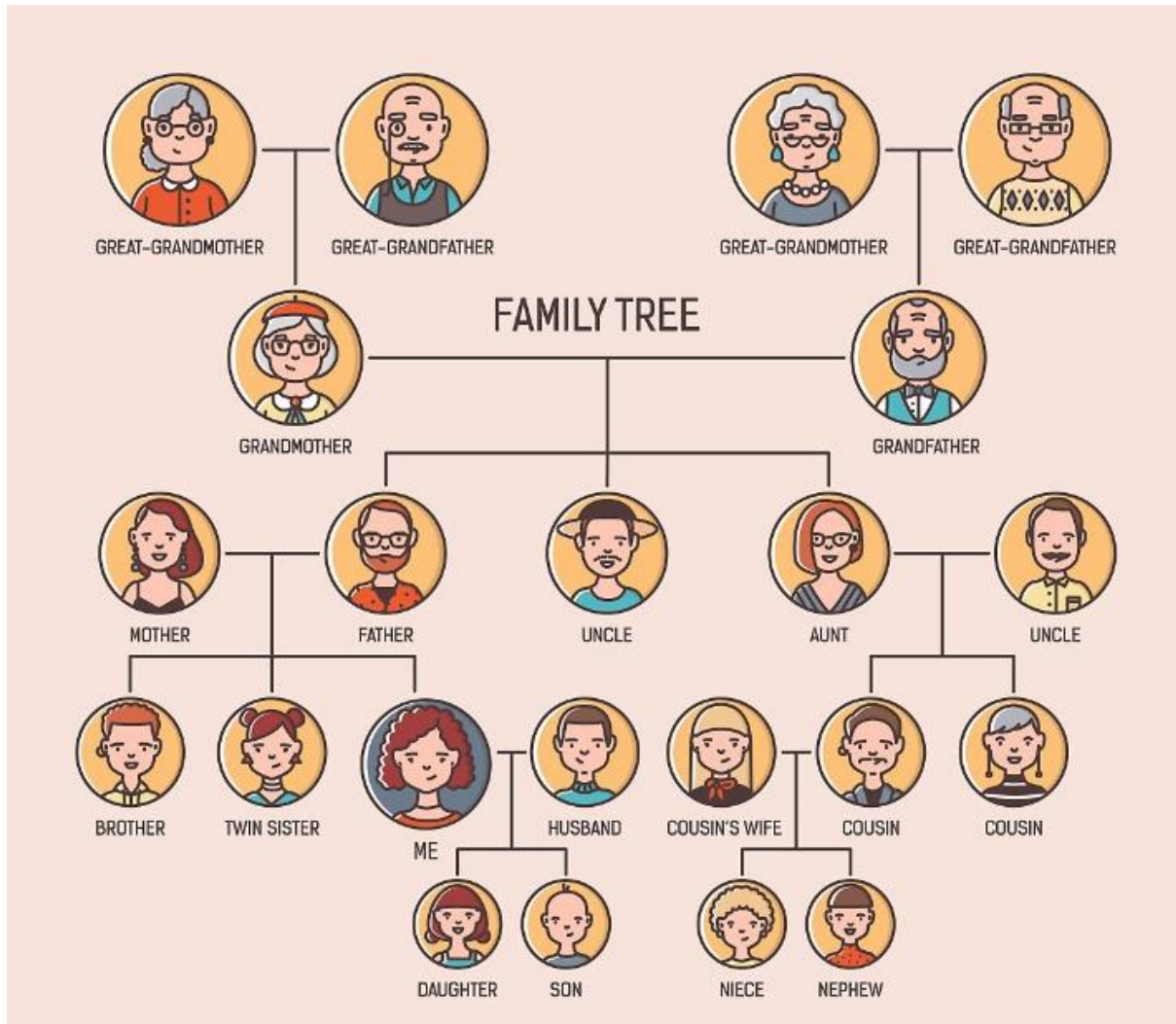
Node와 Edge로 이루어진 자료구조

1. 특징



트리는 값을 가진 노드(Node)와 이 노드들을 연결해주는 간선(Edge)으로 이루어져있다.
그림 상 데이터 1을 가진 노드가 루트(Root) 노드다.
모든 노드들은 0개 이상의 자식(Child) 노드를 갖고 있으며 보통 부모-자식 관계로 부른다.

아래처럼 가족 관계도를 그릴 때 트리 형식으로 나타내는 경우도 많이 봤을 것이다.
자료구조의 트리도 이 방식을 그대로 구현한 것이다.



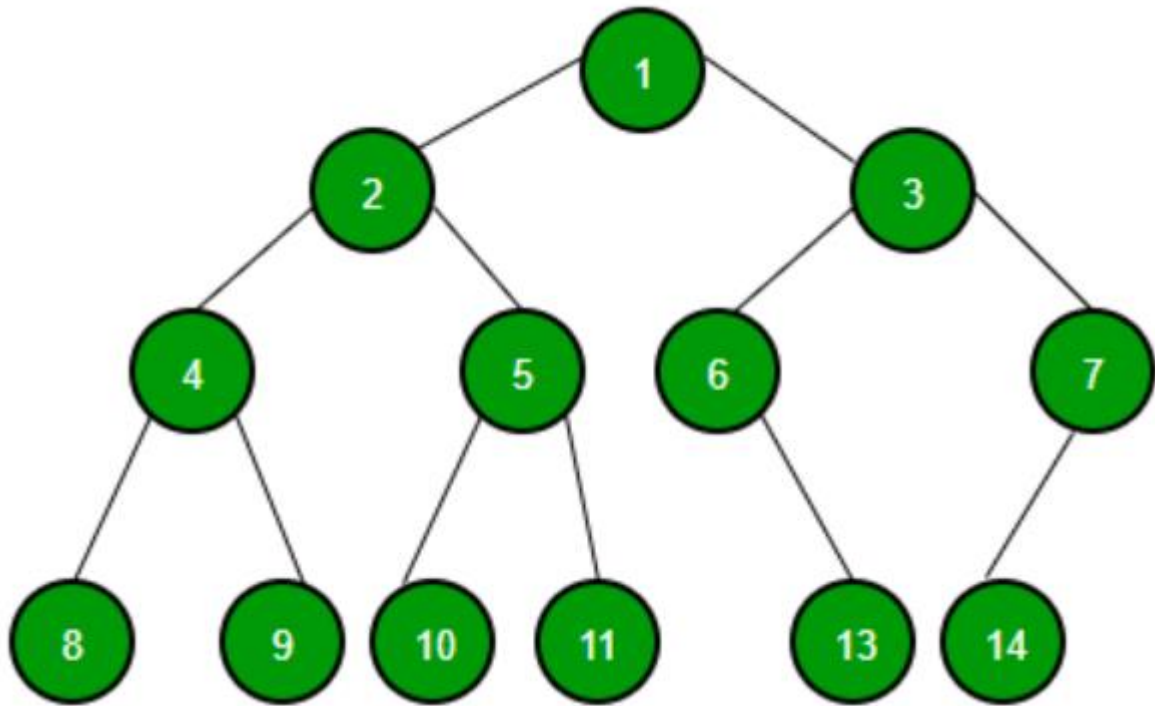
트리는 몇 가지 특징이 있다.

- 트리에는 사이클이 존재할 수 없다. (만약 사이클이 만들어진다면, 그것은 트리가 아니고 그래프다)
- 모든 노드는 자료형으로 표현이 가능하다.
- 루트에서 한 노드로 가는 경로는 유일한 경로뿐이다.
- 노드의 개수가 N 개면, 간선은 $N-1$ 개를 가진다.

가장 중요한 것은, 그래프와 트리의 차이가 무엇 인가인데, 이는 사이클의 유무로 설명할 수 있다.

2. 트리 순회 방식

트리를 순회하는 방식은 총 4 가지가 있다.



1) 전위 순회(pre-order)

각 루트(Root)를 순차적으로 먼저 방문하는 방식이다.

(Root → 왼쪽 자식 → 오른쪽 자식)

1 → 2 → 4 → 8 → 9 → 5 → 10 → 11 → 3 → 6 → 13 → 7 → 14

2) 중위 순회(in-order)

왼쪽 하위 트리를 방문 후 루트(Root)를 방문하는 방식이다.

(왼쪽 자식 → Root → 오른쪽 자식)

8 → 4 → 9 → 2 → 10 → 5 → 11 → 1 → 6 → 13 → 3 → 14 → 7

3) 후위 순회(post-order)

왼쪽 하위 트리부터 하위를 모두 방문 후 루트(Root)를 방문하는 방식이다.

(왼쪽 자식 → 오른쪽 자식 → Root)

8 → 9 → 4 → 10 → 11 → 5 → 2 → 13 → 6 → 14 → 7 → 3 → 1

4) 레벨 순회(level-order)

루트(Root)부터 계층 별로 방문하는 방식이다.

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 13 → 14

<Reference>

1. <https://github.com/gyoogle/tech-interview-for-developer/blob/master/Computer%20Science/Data%20Structure/Tree.md>