

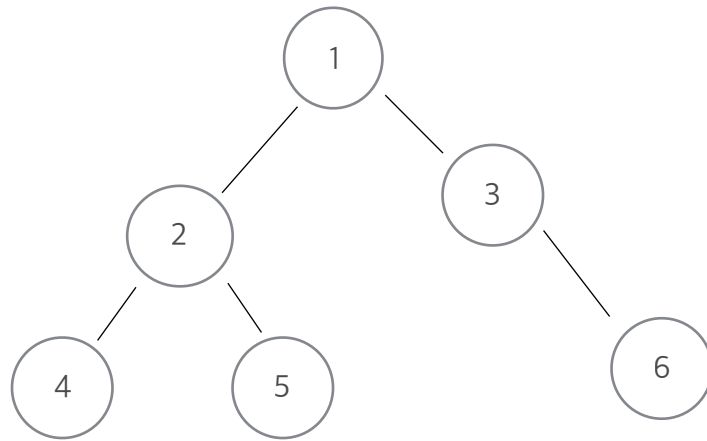
Tree

이번에는 Tree에 대해서 배워봅시다. 이번 시간에서는 bi-directed Graph 에서의 Tree만 언급할 것입니다. 물론 배운 내용중에 있는 Trie는 directed Graph 이지만 bi-directed Graph 에서의 Tree를 이해했다면, 이해하는데에 큰 어려움은 없을거라 판단됩니다.

Tree 의 정의는 다음과 같습니다.

- 임의의 서로 다른 두 Vertex에 대해 경로는 유일하다. 혹은 단 하나만 존재한다.
- 모든 Vertex는 Connected 되어 있다.

여기서 Connected란, 임의의 서로 다른 두 Vertex $\{u, v\}$ 에 대해서 그러한 경로가 하나 이상 존재함을 의미합니다. 이것을 통해 알 수 있는 것은 Tree 의 Vertex 개수를 N 이라고 한다면 Tree의 Edge의 개수는 $N - 1$ 임이 보장 됩니다. 다음 그림을 통하여 예시를 들어보겠습니다.



2 에서 3으로 가는 경로는 2 -> 1 -> 3으로 유일하고 마찬가지로 다른 모든 두 Vertex에 대한 Path 역시 전부 하나 밖에 없습니다. 그리고 모든 어떤 두 Vertex를 잡던 Path는 항상 존재합니다. 그리고 Tree 에서의 Vertex는 Node(노드)라고도 합니다. 이 그래프를 위 아래로 뒤집어 보면 어떻게 될까요? 마치 나무처럼 보입니다. 그래서 Tree라고 이름 붙였을지도 모릅니다.

이제 Tree 의 용어에 대해서 설명하겠습니다. Tree는 Level(레벨), 혹은 Depth(깊이) 라는 개념이 존재합니다. 위 그래프에서 가장 맨 위에 있는 노드(1)가 레벨 0이고 그 노드로 부터 뻗어나가는 두개의 노드(2, 3)가 레벨 1, 그리고 그 두개에 연결되어 있는 세개의 노드(4, 5, 6)가 레벨 2가 되겠습니다. 이때 Level 0 인 단 하나의 가장 맨 위의 노드를 Root(루트) 라고합니다. 여기서 눈치 채셨겠지만 Level은 Root로 부터의 거리를 의미합니다. 즉, Root가 1 이라면 6은 1->3->6 이므로 레벨은 2가 됩니다. 만약 Root가 2가 된다면 6은 레벨이 3이 되겠지요.

트리의 노드간의 관계도 설명할 필요가 있습니다. 자신과 연결되어 있으면서 자신보다 Level이 정확히 1 작은 노드를 자신의 Parents(부모)라고 합니다. 그리고 자신에게 연결되어 있는 레벨이 정확히 1큰 노드들을 Child(자식) 라고 합니다. 그리고 같은 부모를 공유하는 노드들을 Sibling(형제) 라고 합니다. 나중에 LCA라는 개념과 그것을 찾는 알고리즘을 배우게 된다면 Ancestor(조상)이라는 개념 역시 알아두면 좋지만 알아두는 수준에서 넘어가도록 합시다.

그리고 Leaf(리프) 라는 것이 있습니다. 자식이 0개인 노드를 의미합니다. 위 그래프에서는 4, 5, 6이 되겠네요.

트리의 종류에는 여러가지가 있으며 Binary Tree(자식이 최대 2개인 트리)가 유명하며, 이러한 종류는 나중에 차근차근 배워가도록합니다. 지금은 트리의 정의와 용어만 알아두시면 됩니다.