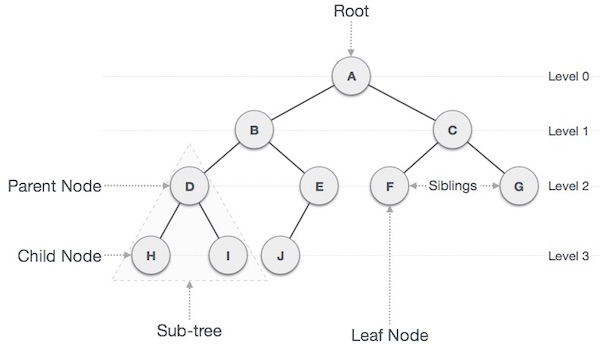
1. 트리

[**노드**](https://ko.wikipedia.org/wiki/%EB%85%B8%EB%93%9C)**들이**[**나뭇가지처럼**](https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0)**연결된**[**비선형**](https://ko.wikipedia.org/wiki/%ED%8F%AC%EC%9D%B8%ED%84%B0_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D)) **계층적인**[**자료 구조**](https://ko.wikipedia.org/wiki/%EC%9E%90%EB%A3%8C_%EA%B5%AC%EC%A1%B0)https://yoongrammer.tistory.com/68

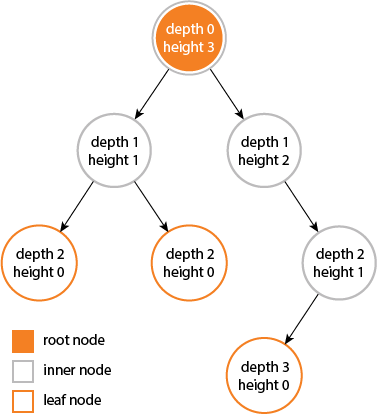
**(나무를 거꾸로 뒤집어 놓은 모양과 유사)**

2. 특징 및 장단점

하나의 루트 노드와 0개 이상의 하위 노드

계층적, 순차적X -> 비선형, 재귀적,

무방향, 효과적인 탐색

 (응용) 컴퓨터의 directory

탐색 : O(logn) / 불균형 : O(N)

삽입 : O(logn)

삭제 : O(logn)

3. 용어 설명

형제노드(Sibling Noed) : 같은 부모를 가진

외부노드, 단말노드, 리프노드 : 자식 없는

내부노드, 비단말노드, 가지노드 : 자식 하나 이상

깊이 : 루트에서 어떤 노드까지의 간선 수

높이 : 노드에서 리프노드까지 가장 긴 간선 수

https://velog.io/@kimdukbae/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0-%ED%8A%B8%EB%A6%AC-Tree

노드(Node) : 기본 요소

간선(Edge) : 연결선

루트노드(Root Node) : 최상위 노드

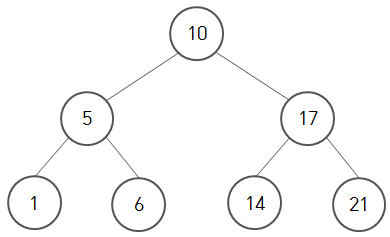
부모노드(Parent Node) : 자식노드를 가진 노드

자식노드(Child Node) : 부모노드의 하위노드

4. 종류

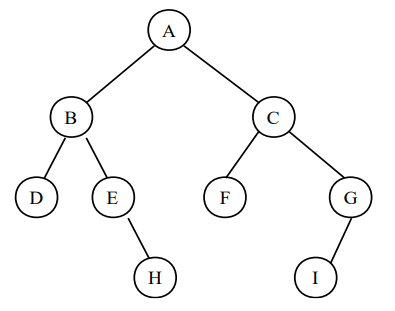
나. **이진탐색트리** : 왼쪽 자식은 나보다 작고

오른쪽 자식은 나보다 크다



모든 노드는 중복된 값 없음

가. 이진트리 : 각각의 노드가 최대 두 개의 자식 노드를 가짐

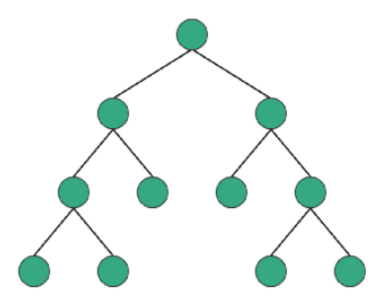


For 정렬된 배열 -> 이진탐색? 순서 무작위

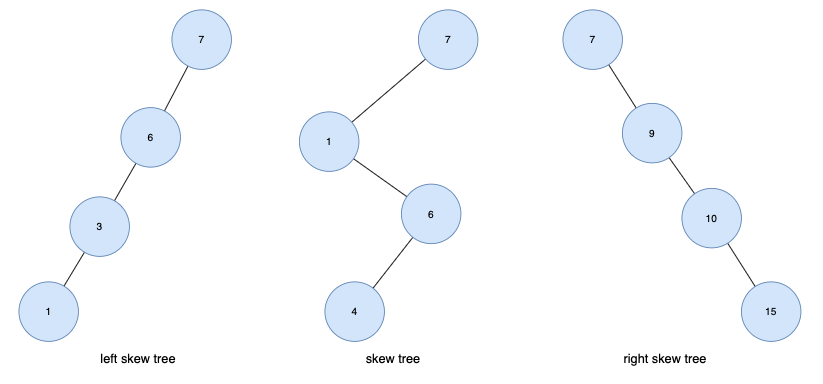
∵ 편향 발생

라. 전이진트리(Full) : Leaf Node 제외 **모든 노드의 차수가 2개**

가위이(가) 표시된 사진

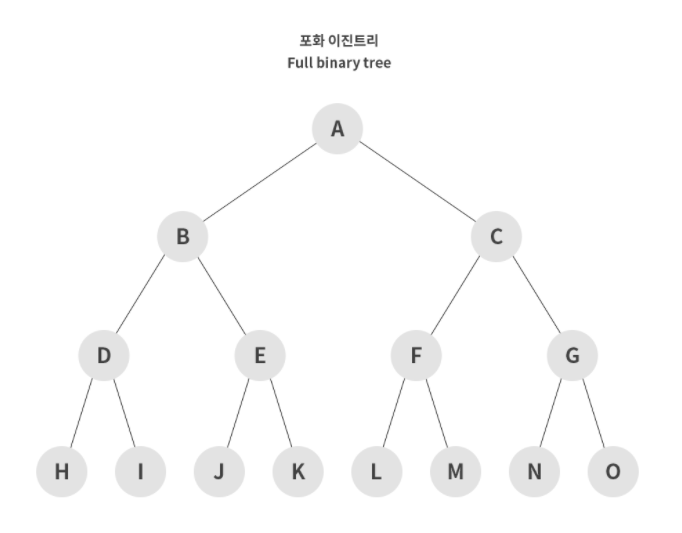
자동 생성된 설명

다. 편향트리 : 모든 노드들이 하나의 자식

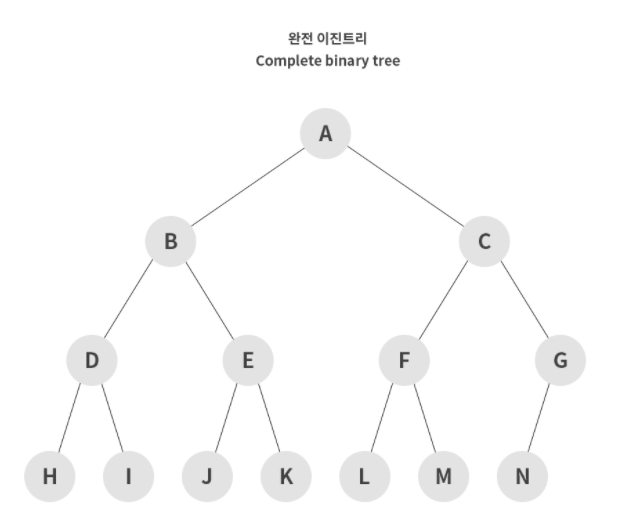


바. 포화이진트리(perfect) : 리프레벨 꽉 차,

모두 2개자식, 동일한 깊이(level), 노드 수 개



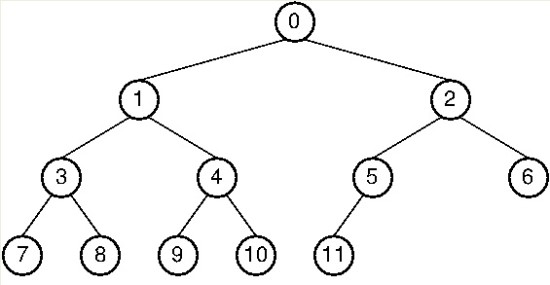
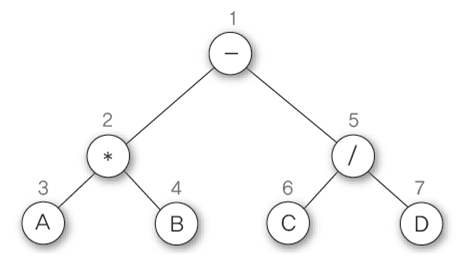
마. 완전이진트리(Complete) : 모든 노드가 **왼쪽부터 차례대로** 생성 **(힙)**

가위이(가) 표시된 사진

자동 생성된 설명

5. 트리 순회 (tree traversal)

노드를 언제 볼거야? -> 먼저 볼거야 : 전위 / 가운데 볼거야 : 중위 / 나중에 볼거야 : 후위

****

**←**

**A\*B-C/D를 만들려면?**

가. 전위순회 (Preorder) : DFT, depth-first traversal / 노드 – 왼 – 오

전위 순회 : 0->1->3->7->8->4->9->10->2->5->11->6 / -\*AB/CD

나. 중위순회 (Inorder) : 이진탐색트리(BST)에서 오름/내림차순 / 왼 – 노드 - 오

중위 순회 : 7->3->8->1->9->4->10->0->11->5->2->6 / A\*B-C/D

다. 후위순회 (Postorder) : 트리 삭제 / 왼 – 오 – 노드

후위 순회 : 7->8->3->9->10->4->1->11->5->6->2->0 / AB\*CD/-

6. 어디에 쓸까?

가. 계층적 데이터 저장 (파일, 폴더)

나. 힙 다. 데이터베이스 인덱싱