

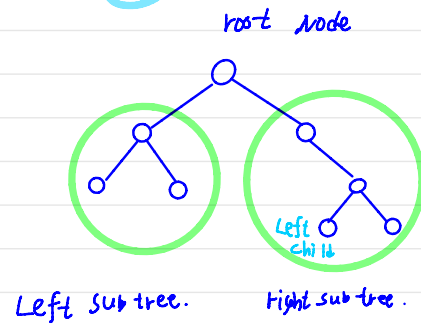
node(노드) : 트리구성요소
 parent node (부모노드)
 root node / root : 반드시 1개
 Siblings node (형제노드, 같은부모)
 edge (간선) : 노드 간 연결
 child node (자식노드)

height (가장 긴 루트-경로 길이)
 branch (node & node 간 연결)
 degree of a node (각 노드의 자식노드 개수)
 degree of tree (트리 최대차수)
 Leaf (= terminal node) (자식이 없는 노드 : degree 0)
 Internal node (degree > 0)
 Ancastors (자신과 root 상 존재 노드)

트리 문제 (problem)
 자식노드 불규칙 트리 경우, node 탐색엔임! / 하나의 노드가 2개의 자식노드 가지는 이진트리

이진 탐색 트리 (Binary search tree)

모든 트리가 이진 아니다! 모든 노드 차수가 최대 2인 형태.
 일반트리 → 이진트리 변환? 가능하다!!



Next page...

L1: n 개 node, degree가 k 인 tree. Child를 가리키는 pointer n 개 중 $(n(k-1)+1)$ 개가 null

proof) N 개 node

branch $(n-1)$ 개 모든 node pointer $(n \cdot k)$ 개



branch 수 만큼 pointer 가 채워지고 나머지는 null

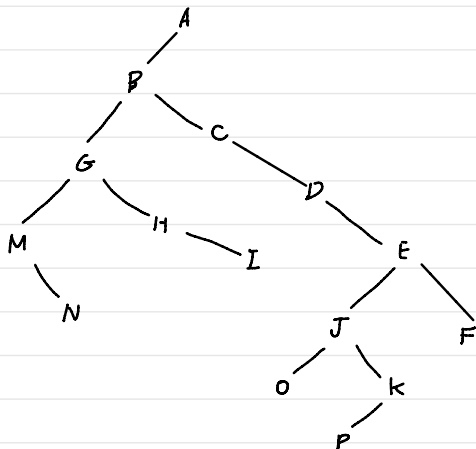
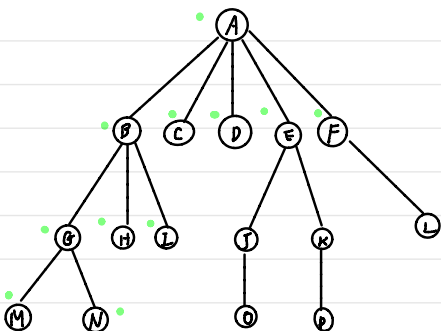
$$\therefore n \cdot k - (n-1) = \underline{n(k-1)+1}$$

Then: node 가든 data 저장, 하위 node 가리키는 pointer 저장:

data	
Child	sibling

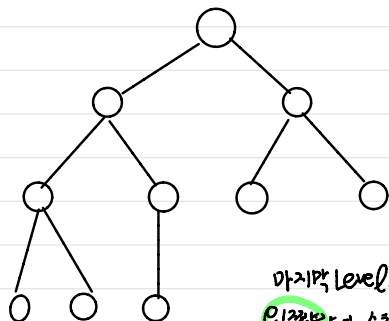
1) 첫 번째 자식노드를 왼쪽 서브트리 (left sub tree)

2) 형제노드는 오른쪽 서브트리 (right subtree)에 위치:

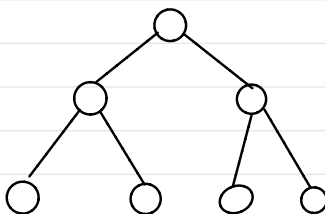


완전 이진트리
(Complete Binary tree)

진 이진트리 (Full Binary tree)



마지막 level의 노드들은
왼쪽부터 순차적으로 채워져
있음



각 level 노드가 완전히 채워져 있음.

모든 leaf 노드는 같은 level에 있어야 함

마지막 level 노드 개수는 최대가 되어야 함.

Binary tree. Level $i \rightarrow$ node 최대 2^{i-1}

Height K 인 binary tree 최대 node 수는 $(2^K - 1)$ 개

Node 수가 n 개 complete binary tree height? \rightarrow log 함수

L2 - Binary tree, degree 0 인 node (leaf 수) degree가 2인 node 수보다 1개 많음.

$$n_0 = n_2 + 1$$

(증명) n : 총 node 수 B : branch 수 n_0 : leaf node 수 n_1 : degree 1인 node 수

n_2 : degree 2인 node 수

총 node 수 $n = n_0 + n_1 + n_2$

branch 각 node 의 degree 합

$$B = n_1 + 2n_2$$

$$n = n_0 + 2n_1 + 1$$

$$n_1 + 2n_2 + 1 = n_0 + n_1 + n_2$$

$$n = n_0 + n_1 + n_2$$

$$1 + n_2 = n_0$$

총 node 수 branch 수보다 1개 많음.

$$n = B + 1$$

binary tree traversal (순회) - 재귀호출을 통해 재!!

모든 node 에 대하여 중복, 누락없이 일일해서 특정 작업수행

Inorder : Left - Node - Right

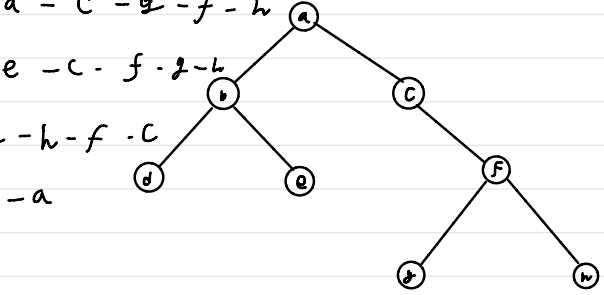
preorder : Node - Left - Right

postorder : Left - Right - node

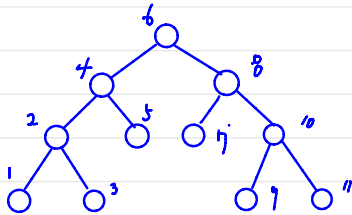
In order d - b - e - a - c - g - f - h

pre order a - b - d - e - c - f - g - h

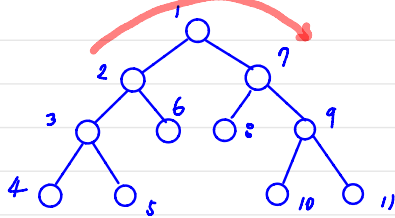
postorder d - e - b - j - h - f - c



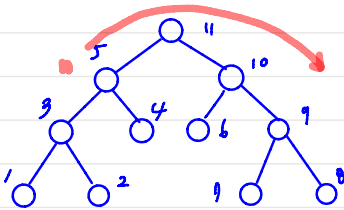
1.) $\sum_{i=1}^n \hat{z}_i^L$ left - n - right



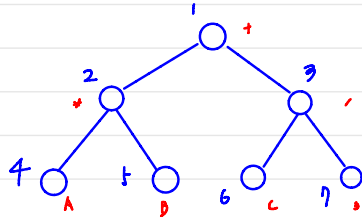
2) 전위 순회 N - left - right



3) 후위 순회 left - right - **N**



27/04/2021 (level-order Traversal) ...



level 노드는 root 부터 동일 level에서는

Left rim.