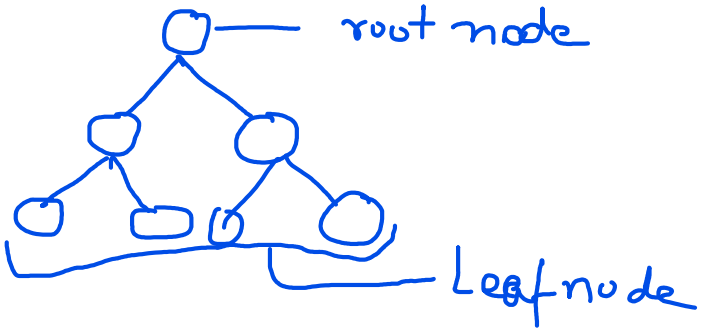
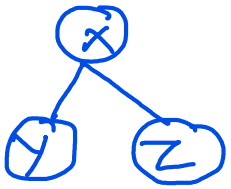
Tree

height



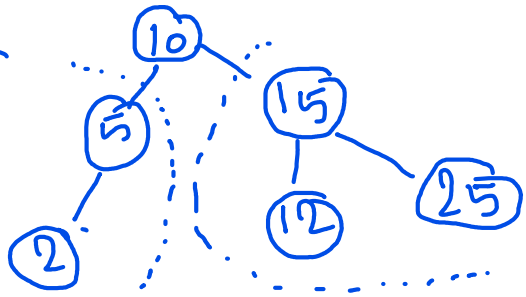
Binary Search Tree — BST



$x > y$
$x > z$ } condition

Smaller Value in Left subtree
Larger " " Right.

Ex

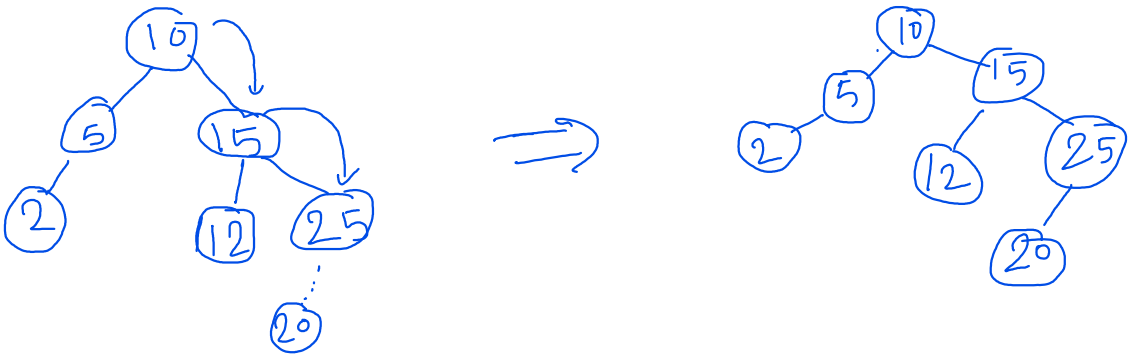


Searching time — is perporational to height of tree

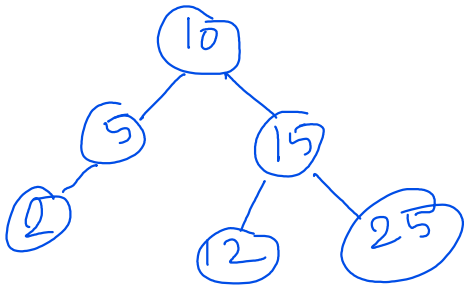
Balanced BST — $O(\log n)$
Degenrate BST — $O(n)$

Operations — Insertion, Deletion

insert 20 in above example —

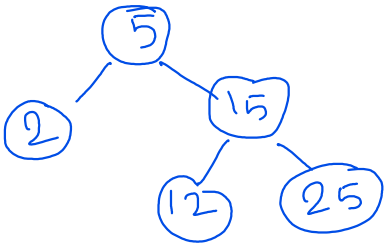


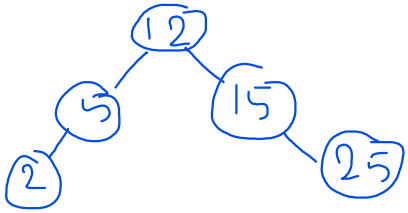
Delete Leaf node - 20 Now tree is



Now Delete 10 from ~~the~~ tree

Replace 10 with Largest value from Left Subtree or Replace 10 with Smallest value from Right Subtree





Tree Traversal

inorder

(A - ((C / 5) * 2)) + ((D * 5) % 4)

Preorder

+ - A * / C 5 2 % * D 5 4

POST-order

A C 5 / 2 * - D 5 * 4 % +