

DATA STRUCTURES & ALGORITHMS

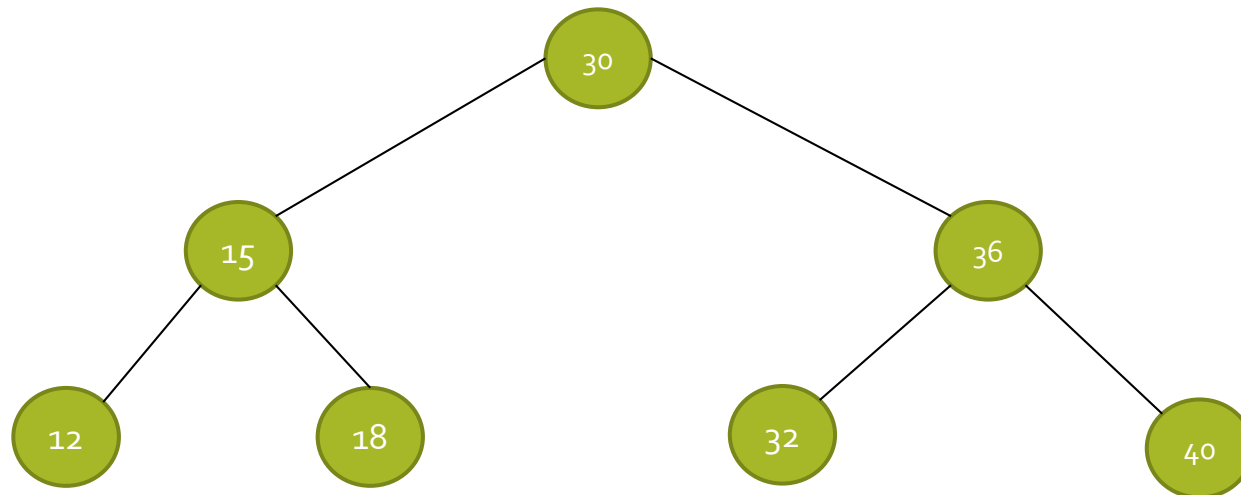
Trees : BST

Instructor: Engr. Laraib Siddiqui

Binary Search Tree

Binary search tree is a data structure that quickly allows us to maintain a sorted list of numbers.

- The left sub-tree of a node has a key less than or equal to its parent node's key.
- The right sub-tree of a node has a key greater than to its parent node's key.



Searching and insertion in BST

Suppose an **ITEM** with the root node **N** of the tree. The following algorithm finds the location of **ITEM** in the binary search tree, or inserts **ITEM** as a new node in its appropriate place in the tree.

- a) Compare **ITEM** with the root node **N** of the tree.
 - **ITEM** \leq **N**, proceed to the **left child** of **N**
 - **ITEM** $>$ **N**, proceed to the **right child** of **N**
- b) Repeat step (a) until one of the following occurs:
 - We meet a node **N** such that **ITEM** = **N**. In this case the search is **successful**.
 - We meet an **empty** subtree, which indicates that the search is **unsuccessful**, and we **insert ITEM** in place of the empty subtree.

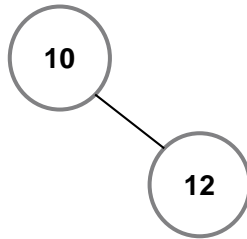
Example: Insertion in BST

10,12,5,4,20,8,7,15,13

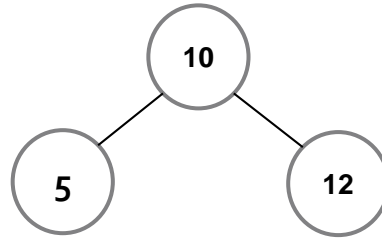
Insert 10



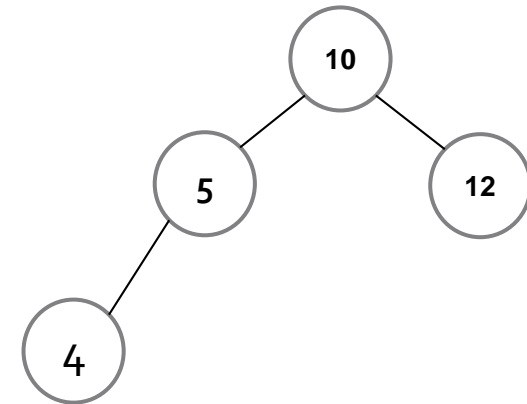
Insert 12, $12 > 10$, will be added to right



Insert 5, $5 < 10$, will be added to left

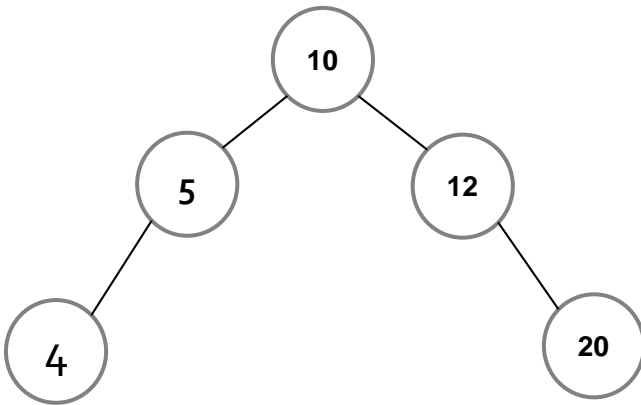


Insert 4, $4 < 10$ & $4 < 5$, will be added to left

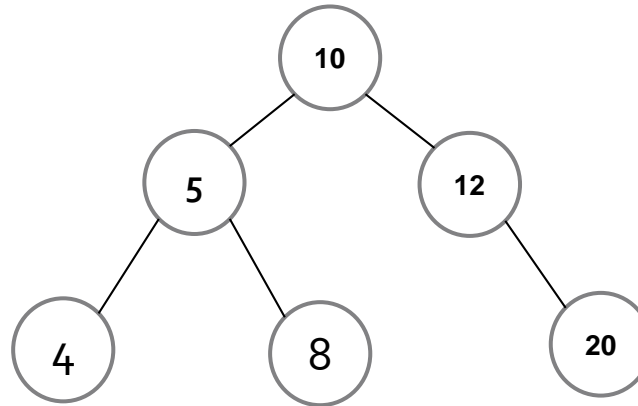


Example: Insertion in BST

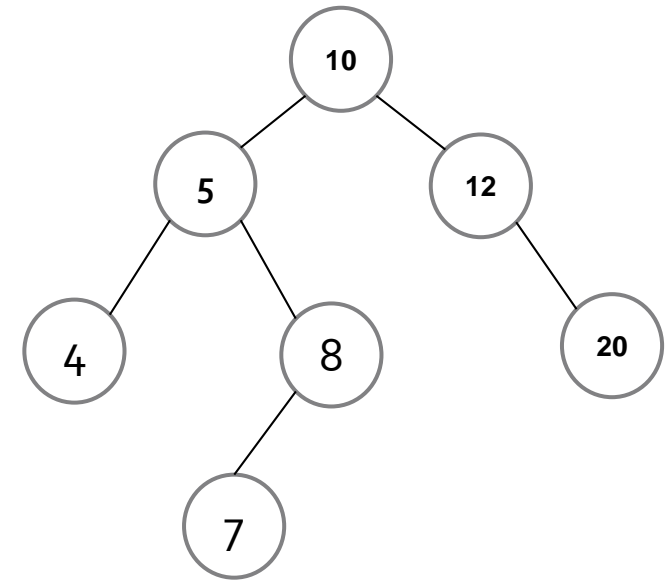
10,12,5,4,20,8,7,15,13



Insert 20, $20 > 10$ & $20 > 12$, will be added to right



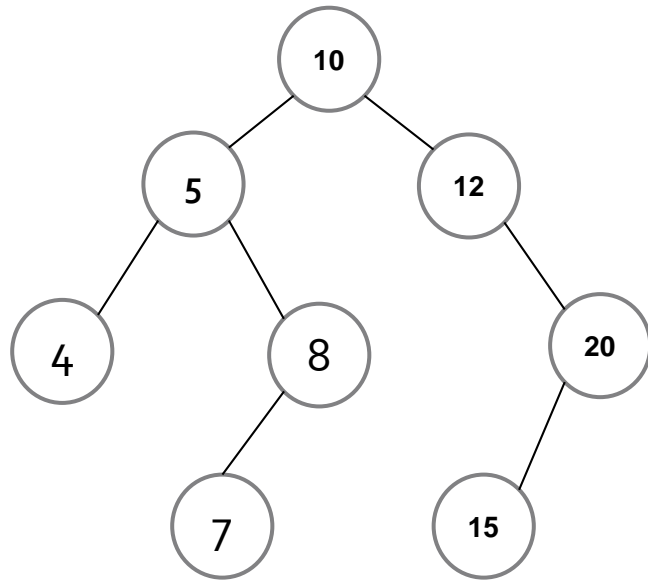
Insert 8, $8 < 10$ & $8 > 5$, will be added to right



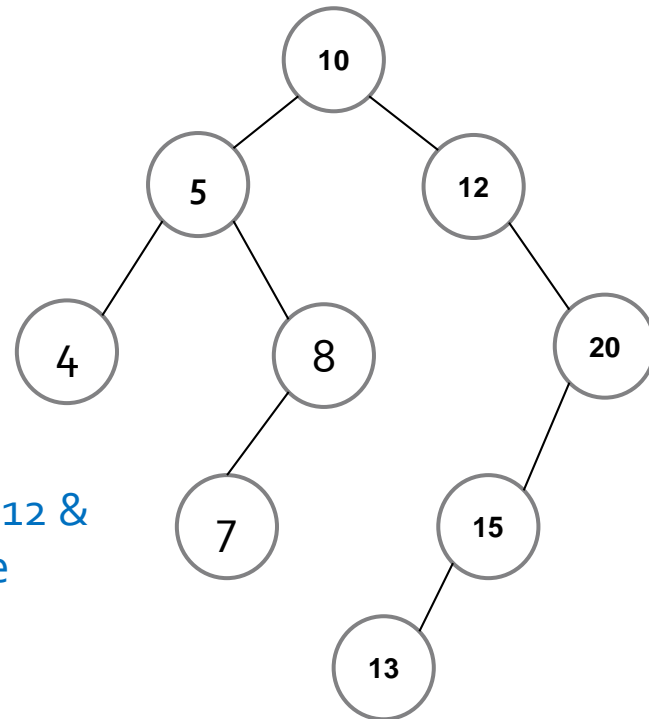
Insert 7, $7 < 10$ & $7 < 8$, will be added to left

Example: Insertion in BST

10,12,5,4,20,8,7,15,13



Insert 15, $15 > 10$ & $15 > 12$ & $15 < 20$, will be added to left



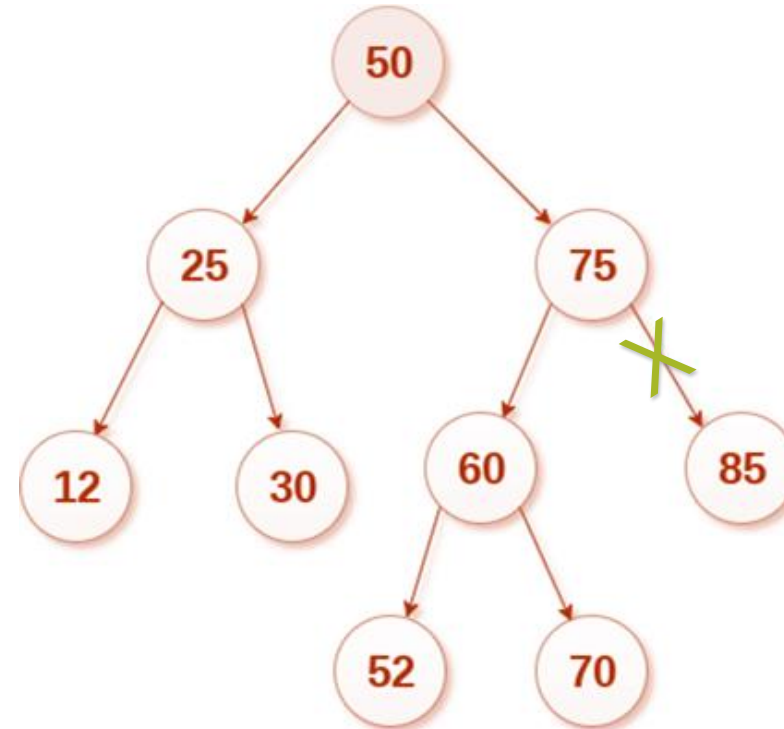
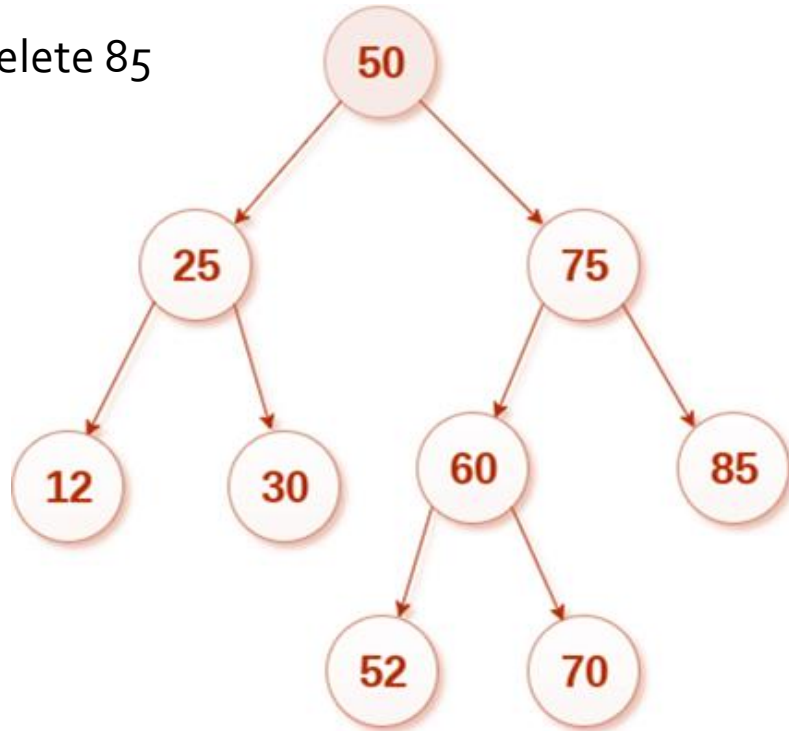
Insert 13, $13 > 10$ & $13 > 12$ & $13 < 20$ & $13 < 15$, will be added to left

Deleting in a BST

- a) Find the desired element.
 - It does not exist
 - It exists in the left subtree
 - It exists in the right subtree
- b) Replace the desired element with its successor (if any)
 - Node is a leaf node – Remove the node
 - Node has left/right subtree – Successor of the node replace the node.
 - Node has both left and right subtrees – Successor which can replace the node, can either be the largest value in the left subtree or the smallest value in the right subtree.

Deleting in a BST

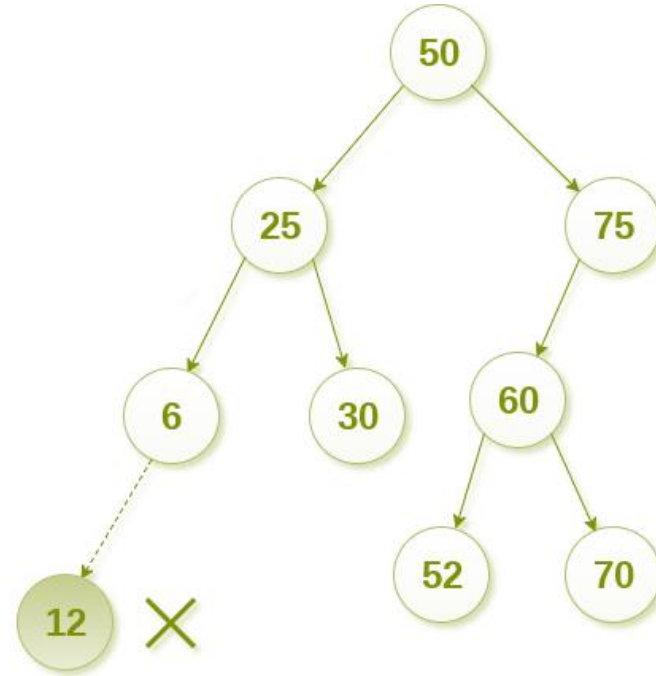
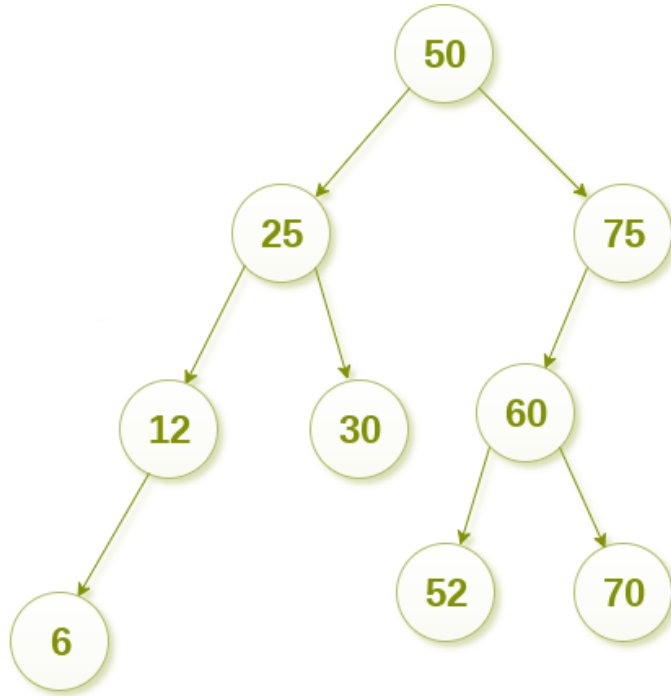
Delete 85



Node is a leaf node – Remove the node

Deleting in a BST

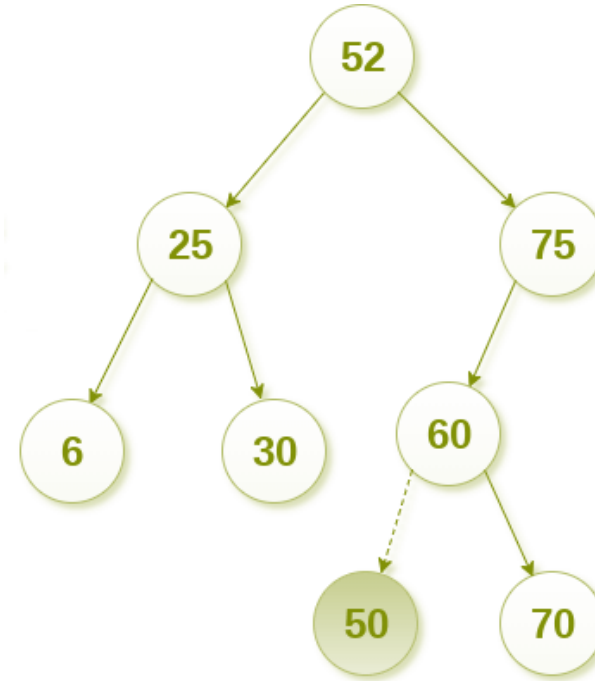
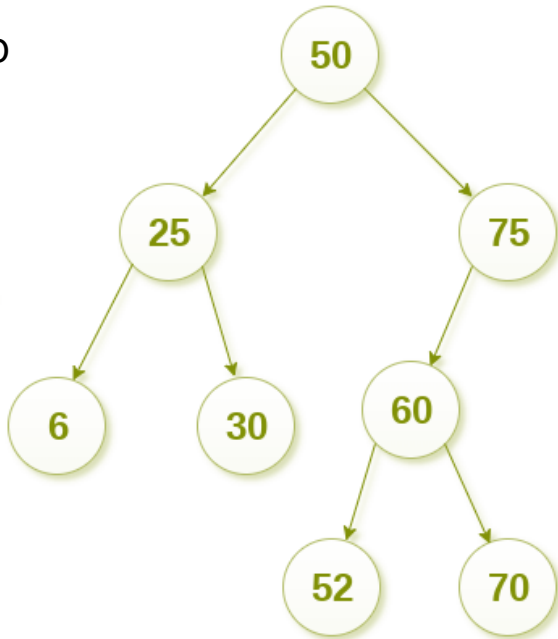
Delete 12



Node has left/right subtree – Successor of the node replace the node.

Deleting in a BST

Delete 50



Node has both left and right subtrees – Successor which can replace the node, can either be the largest value in the left subtree or the smallest value in the right subtree.

Complexity

	Binary search tree
Access	$O(n)$
Search	$O(n)$
Deletion	$O(n)$
Insertion	$O(n)$

Binary Search Tree

What happens when you Insert elements in ascending order?????

Example

Insert: 1, 2, 4, 6, 8 into an empty BST

Issue: *Lack of Balance*

Solution: *Balanced binary search trees (BBST)*

