SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

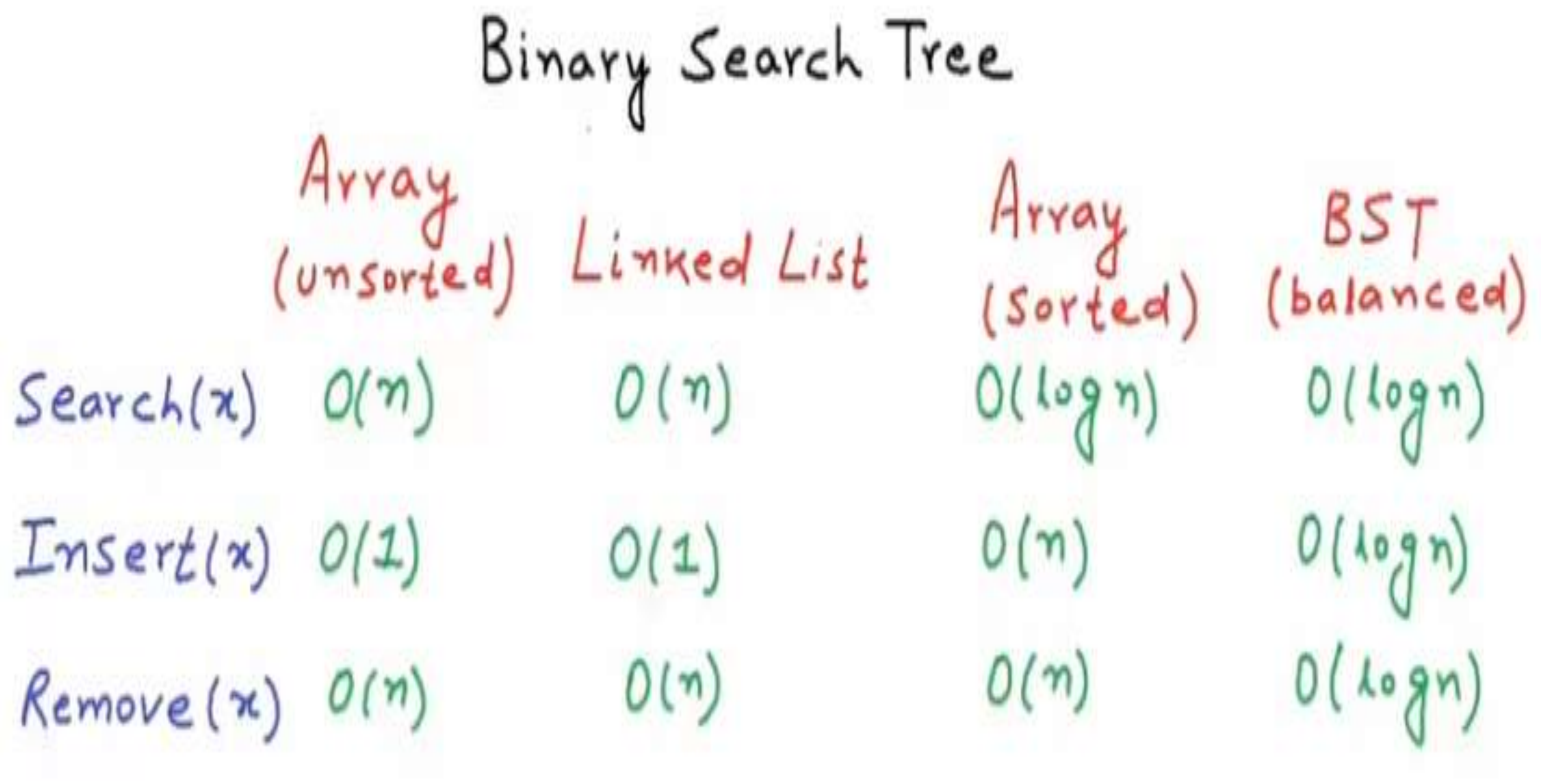
COURSE NAME : 23ITT102 _Programming in C and Data Structure

I YEAR/ II SEMESTER

UNIT V

Topic: *Binary Search Tree*

Need of Binary Search Tree

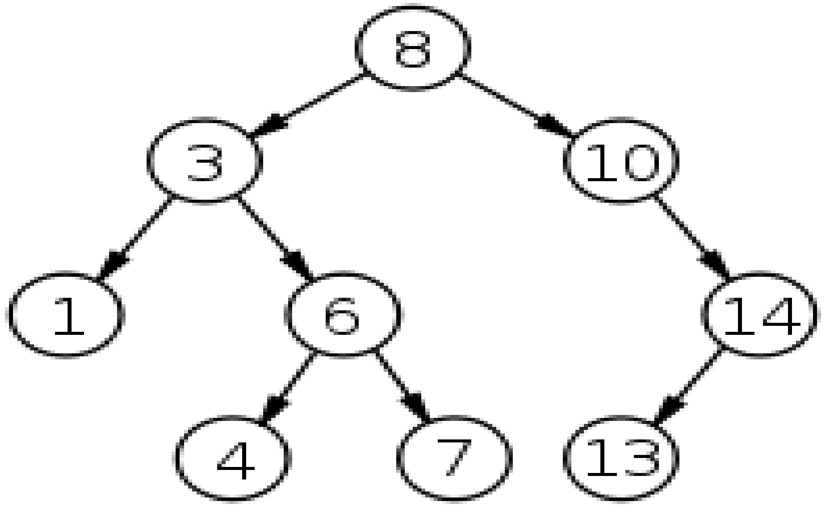


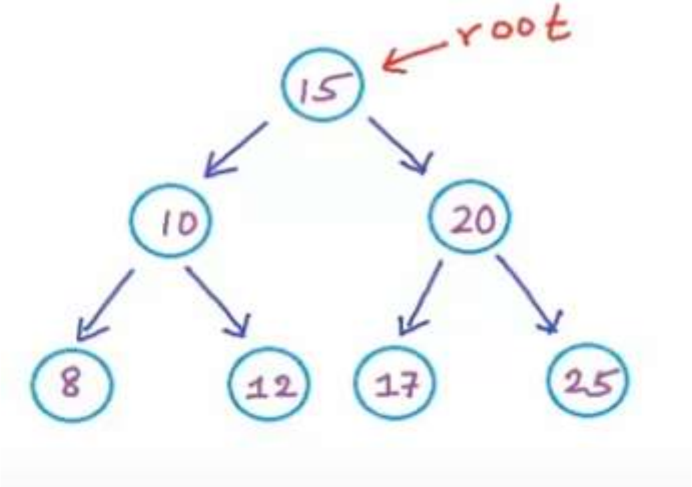
Binary Search Tree

	Array (unsorted)	Linked List	Array (sorted)	BST (balanced)
Search(x)	O(n)	O(n)	O(log n)	O(log n)
Insert(x)	O(1)	O(1)	O(n)	O(log n)
Remove(x)	O(n)	O(n)	O(n)	O(log n)

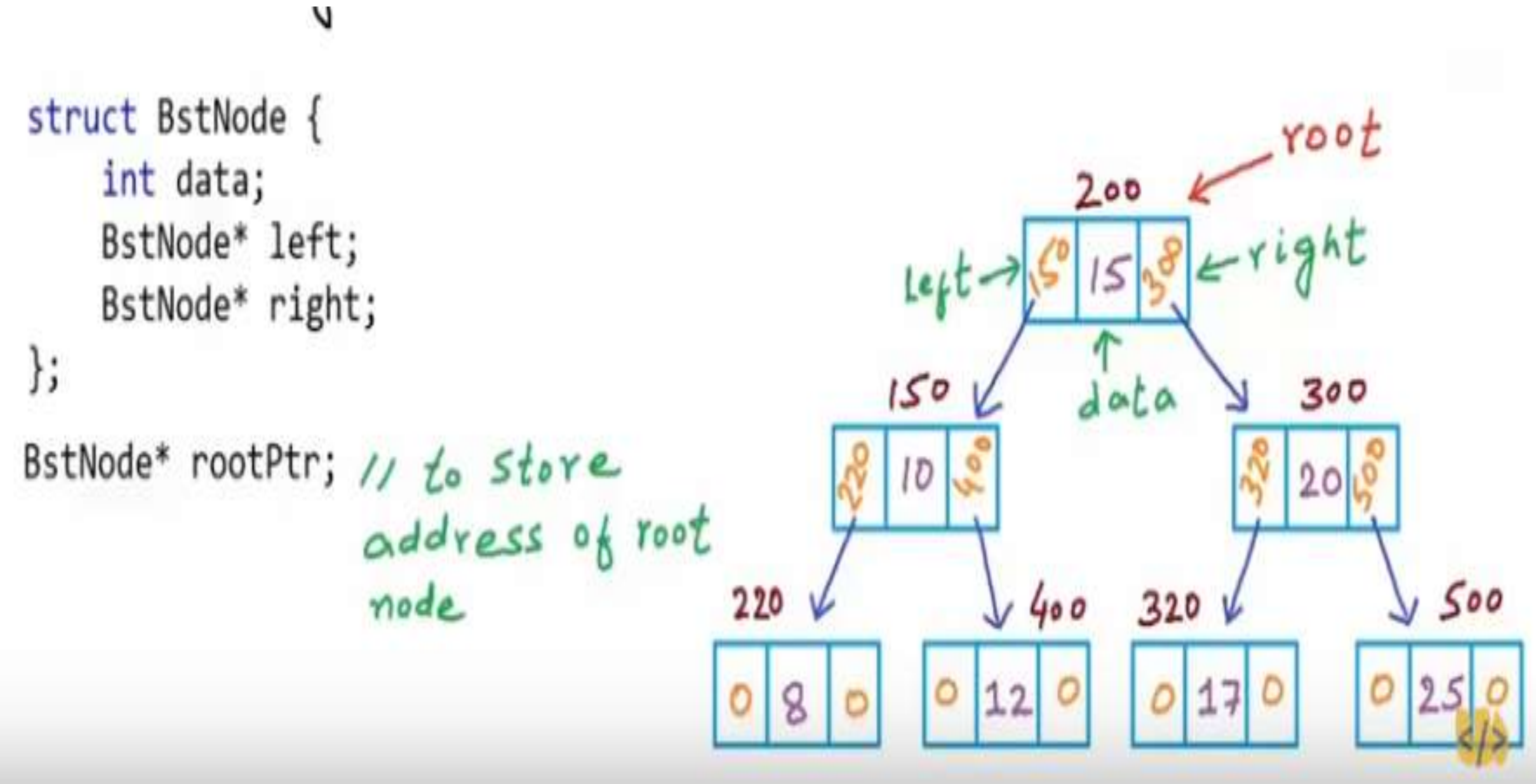
Binary Search Tree

- Binary Tree in which for every node X in the tree, the values of all the keys in its left sub tree are smaller than the key value in X and the values of all the keys in the right sub tree are larger than the key value in X





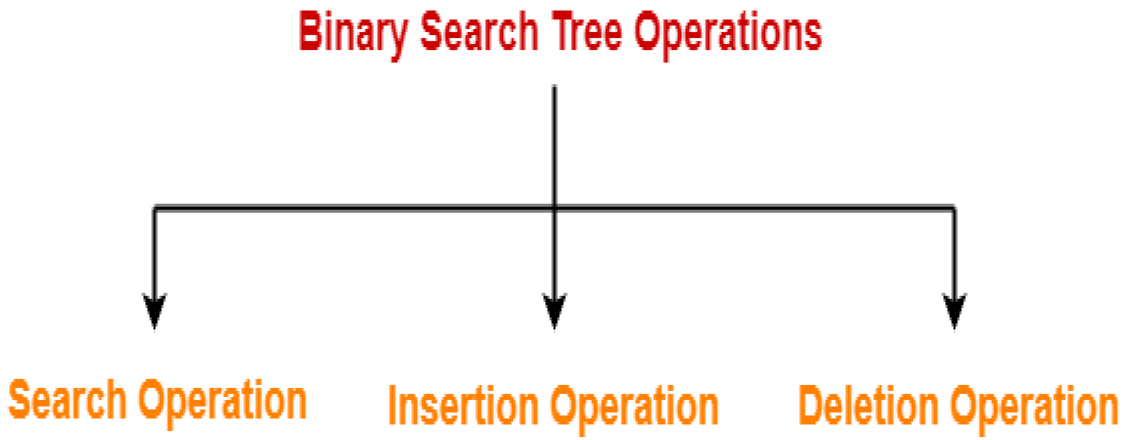
Binary Search Tree



```
struct BstNode {
    int data;
    BstNode* left;
    BstNode* right;
};

BstNode* rootPtr; // to store
                     address of root
                     node
```

Binary Search Tree



Binary Search Tree Operations

Search Operation

Insertion Operation

Deletion Operation

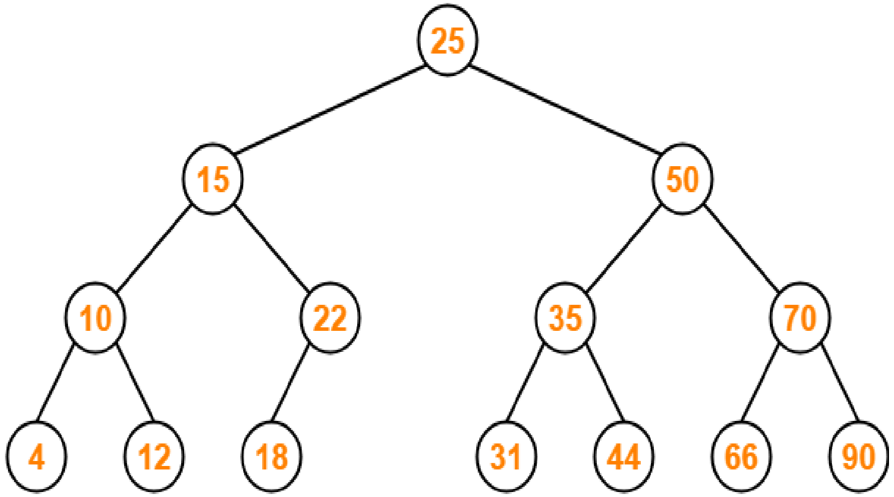
.

<u>1. Search Operation-</u>Search Operation is performed to search a particular element in the Binary Search Tree

Binary Search Tree -Search Operation

- For searching a given key in the BST,
- Compare the key with the value of root node.
- If the key is present at the root node, then return the root node.
- If the key is greater than the root node value, then recur for the root node's right sub tree.
- If the key is smaller than the root node value, then recur for the root node's left sub tree.

Consider key = 45 has to be searched in the given BST



Binary Search Tree

Binary Search Tree-Insertion Operation

Insertion Operation-

Insertion Operation is performed to insert an element in the Binary Search Tree.

Rules-

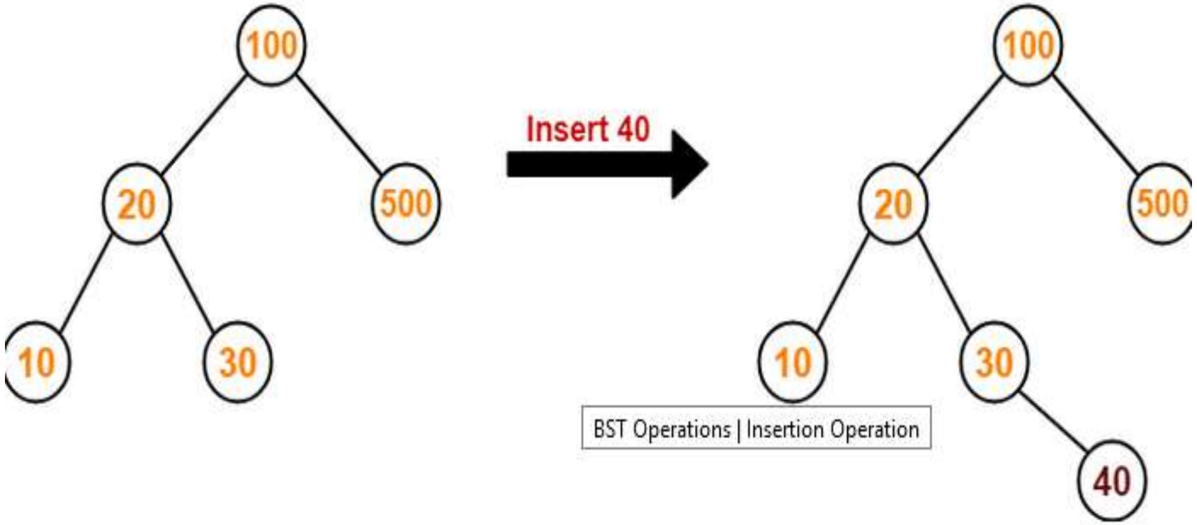
The insertion of a new key always takes place as the child of some leaf node.

For finding out the suitable leaf node,

Search the key to be inserted from the root node till some leaf node is reached.

Once a leaf node is reached, insert the key as child of that leaf node.

- Consider the following example where key = 40 is inserted in the given BST-



Assessment -1

1. ---------- type of traversal of binary search tree outputs the value in sorted order

2. A binary search tree is generated by inserting in order the following integers:

50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24

A. (4, 7)

B. (7, 4)

C. (8, 3)

D. (3, 8)

Binary Search Tree-Deletion Operation

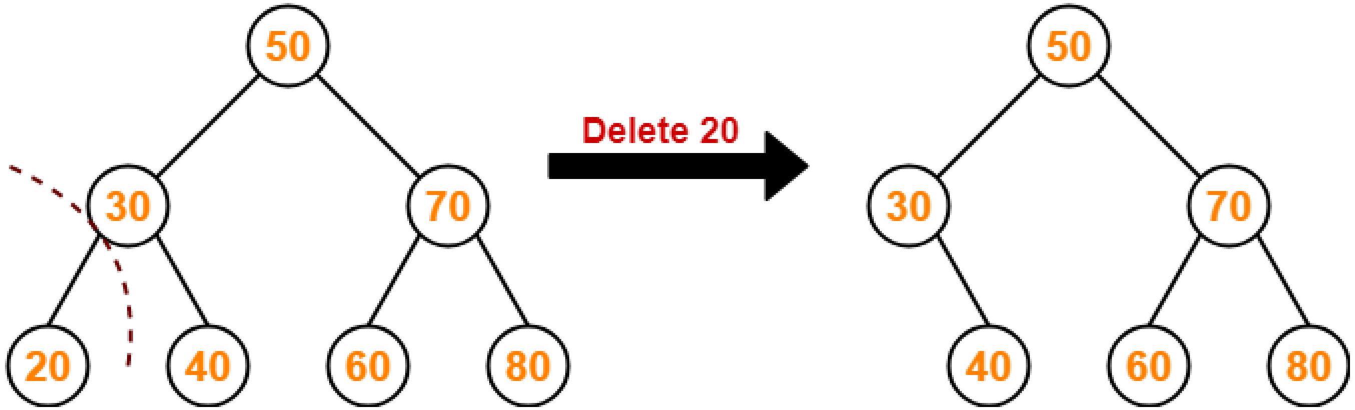
Deletion Operation-

Deletion Operation is performed to delete a particular element from the Binary Search Tree.

Case-01: Deletion Of A Node Having No Child (Leaf Node)-

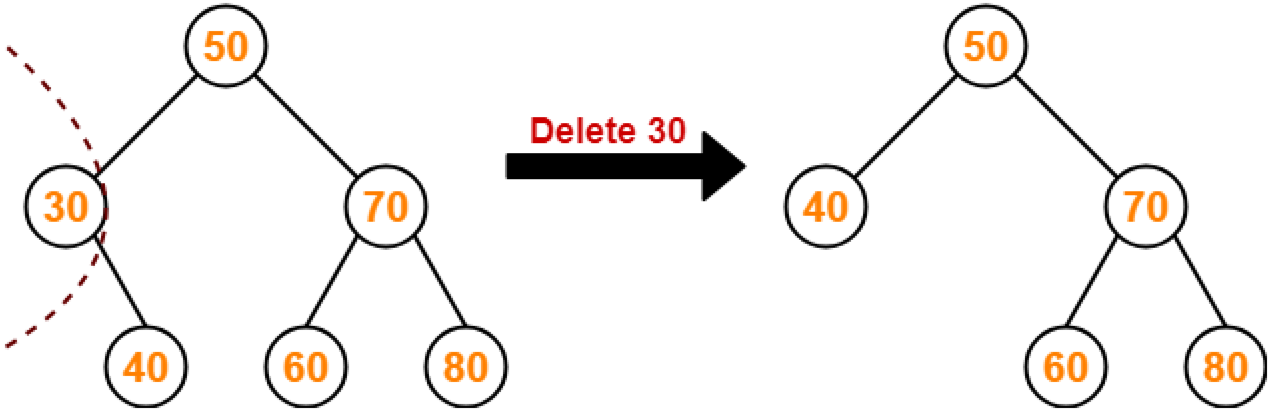
Just remove / disconnect the leaf node that is to deleted from the tree.

Consider the following example where node with value = 20 is deleted from the BST-



Binary Search Tree-Deletion Operation

- **Case-02: Deletion Of A Node Having Only One Child-**
- Just make the child of the deleting node, the child of its grandparent.
- **Example-**
- Consider the following example where node with value = 30 is deleted from the BST-



Binary Search Tree-Deletion Operation

Case-02: Deletion Of A Node Having Two Children-

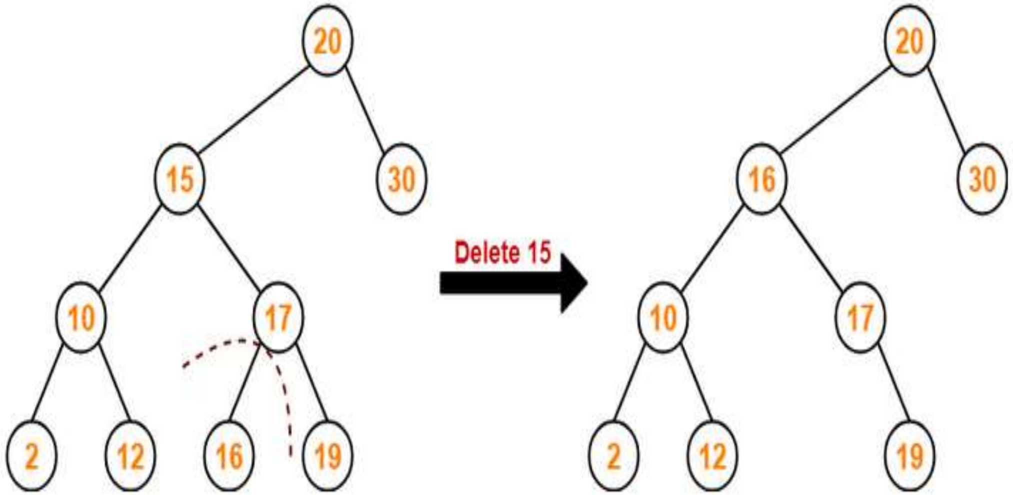
A node with two children may be deleted from the BST in the following two ways-

Method-01:

- Visit to the right subtree of the deleting node.
- Pluck the least value element called as inorder successor.
- Replace the deleting element with its inorder successor.

Example-

Consider the following example where node with value = 15 is deleted from the BST-



Binary Search Tree-Deletion Operation

Method-02:

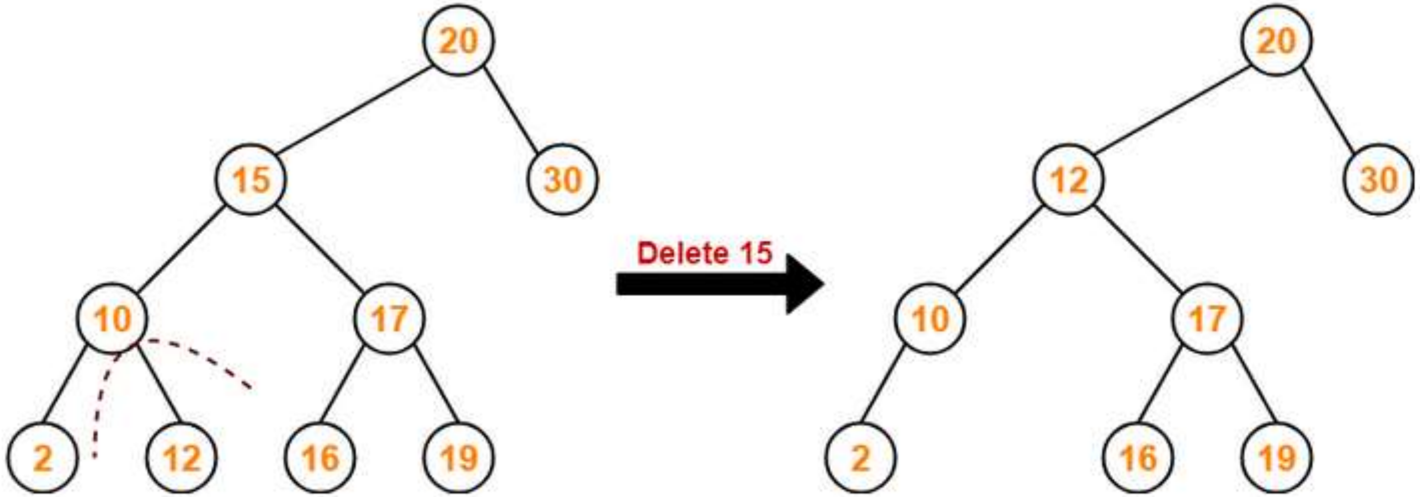
Visit to the left sub tree of the deleting node.

Pluck the greatest value element called as in order predecessor.

Replace the deleting element with its in order predecessor.

Example-

Consider the following example where node with value = 15 is deleted from the BST-



References

1. M. A. Weiss, “Data Structures and Algorithm Analysis in C”, Pearson Education, 2nd Edition, 2002.
2. A. V. Aho, J. E. Hopcroft and J. D. Ullman, “Data Structures and Algorithms”, Pearson Education, 2nd Edition, 2007
3. Ashok Kamthane, " Data Structures Using C ", Pearson Education, 2nd Edition, 2012.
4. Sahni Horowitz, “Fundamentals of Data Structures in C”Universities Press; Second edition 2008

Thank You