Parul® University

# Data Structure

**Prof. Vijya Tulsani,** Assistant Professor
IT and Computer Science

Parul® University

# Data Structure

**Prof. Vijya Tulsani,** Assistant Professor
IT and Computer Science

Parul®
University

DIGITAL LEARNING CONTENT

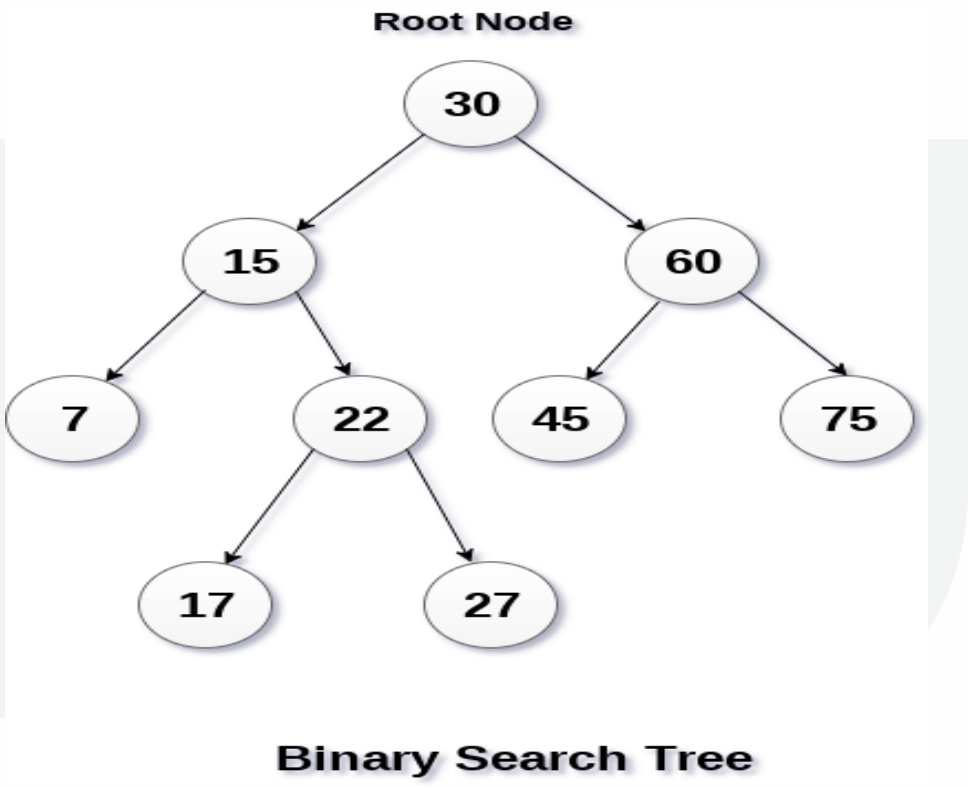
# CHAPTER-3

# Non-Linear Data Structure


# Binary Search Tree

- Binary Search tree can be defined as a class of binary trees, in which the nodes are arranged in a specific order. This is also called ordered binary tree.
- In a binary search tree, the value of all the nodes in the left sub-tree is less than the value of the root.
- Similarly, value of all the nodes in the right sub-tree is greater than or equal to the value of the root.
- This rule will be recursively applied to all the left and right sub-trees of the root.

# Binary Search Tree




Binary Search Tree


# Advantages of using binary search tree

- Searching become very efficient in a binary search tree since, we get a hint at each step, about which sub-tree contains the desired element.
- The binary search tree is considered as efficient data structure in compare to arrays and linked lists. In searching process, it removes half sub-tree at every step. Searching for an element in a binary search tree takes o(log2n) time. In worst case, the time it takes to search an element is 0(n).
- It also speed up the insertion and deletion operations as compare to that in array and linked list.

# Operations on Binary Tree

| SN | Operation | Description |
|---|---|---|
| 1 | Searching in BST | Finding the location of some specific element in a binary search tree. |
| 2 | Insertion in BST | Adding a new element to the binary search tree at the appropriate location so that the property of BST do not violate. |
| 3 | Deletion in BST | Deleting some specific node from a binary search tree. However, there can be various cases in deletion depending upon the number of children, the node have. |

# Insertion

- Lexically Ordered Binary tree
- In Lexically ordered Binary Tree, the creation of a BT could be based on information associated with each node. This ordering could be numerical (either ascending or descending), or it could be a list of names to be kept in lexicographical order.
- That is, the left subtree of the tree is to contain nodes whose associated names are lexically less than the name associated with the root node of the tree. Similarly, the right subtree of the tree is to contain nodes whose associated names are lexically greater than the name associated with the root node of the tree.

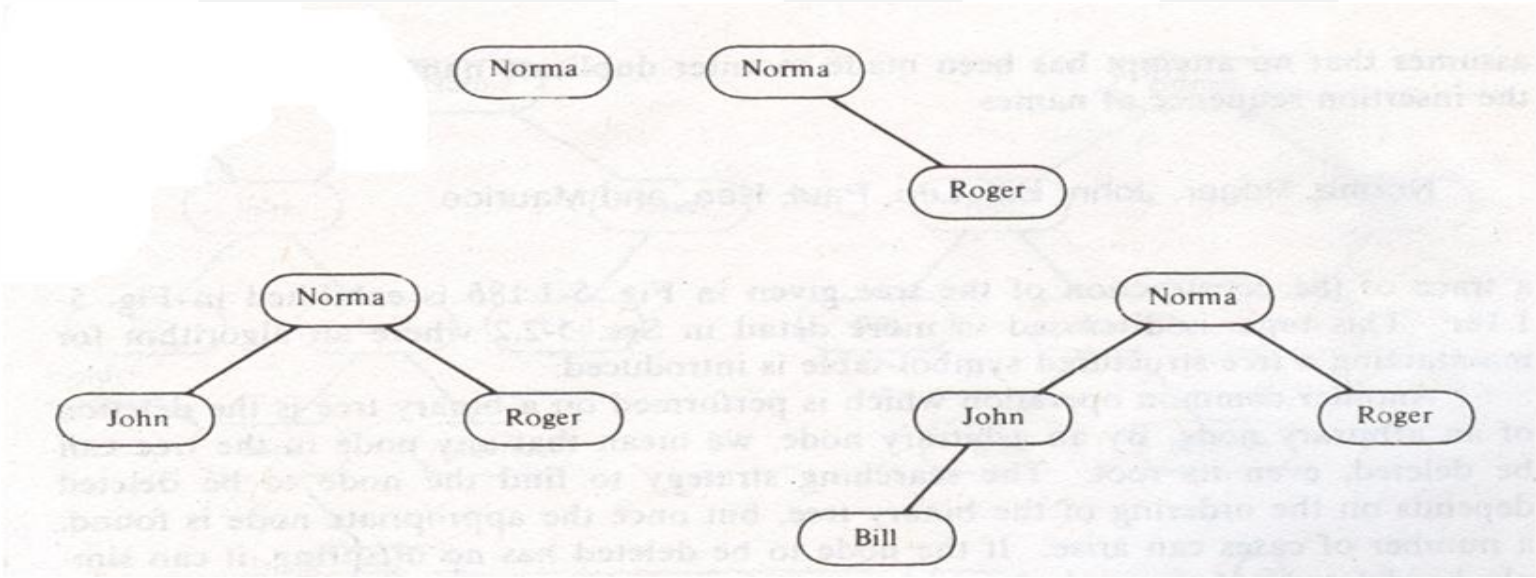

# Insertion

- Given preorder traversal of a binary search tree, construct the BST.

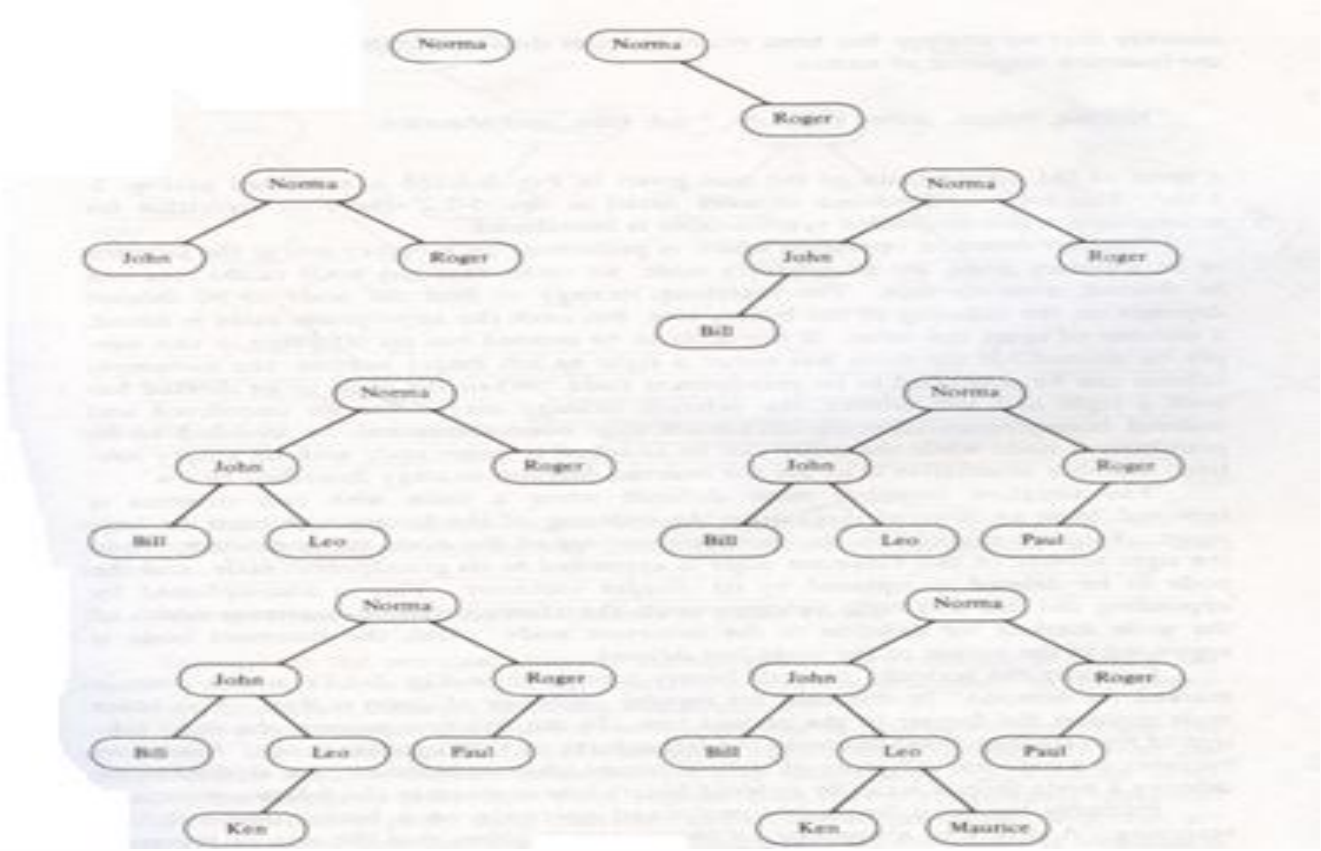
- For example, if the given traversal is
- {10, 5, 1, 7, 40, 50}.
- {7, 4, 12, 3, 6, 8, 1, 5, 10}

# Insertion

- Construct the Lexically Ordered Binary Tree for Norma, Roger, John, bill, Leo, Paul, Ken and Maurice



# Insertion - Norma, Roger, John, bill, Leo, Paul, Ken and Maurice



# Deletion

- Deletion of an arbitrary node from the given binary tree. The searching strategy to find the node to be deleted depends on the ordering of the binary tree, if the node is found then,

Case 1 : If the node to be deleted has no offspring, it can simply be deleted.

Case 2 : If the node has either a right or left empty subtree, the nonempty subtree can be appended to its grandparent node.

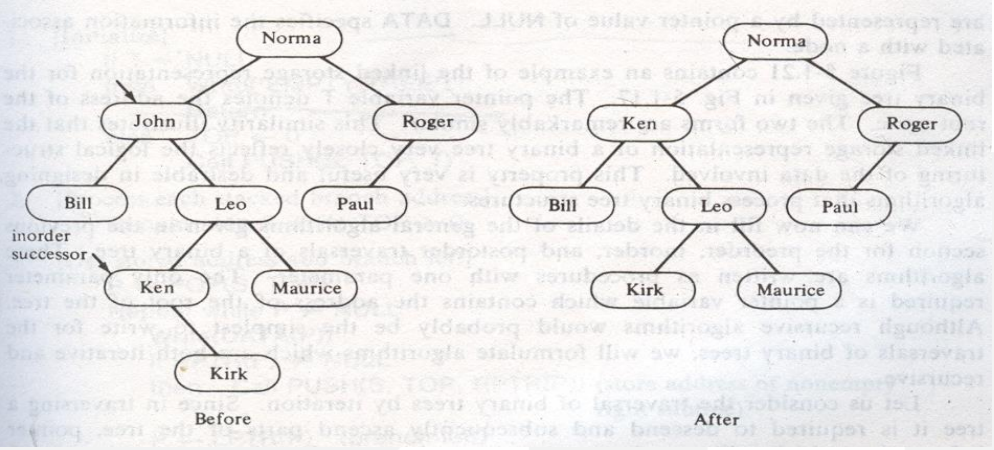
Case 3 : When the node to be deleted has both right and left subtrees, then deletion strategy may differ for unordered and ordered binary tree.

# Deletion

- For an unordered tree, one subtree can be attached to its grandparent node while the other can be attached to some node with an empty subtree.
- For an ordered tree,

1. Obtain the inorder successor of the node to be deleted.
2. Then the right subtree of this successor node is appended to its grandparent node.
3. The node to be deleted is replaced by its inorder successor.
4. The successor node is appended to the parent of the node just deleted.

# Deletion




- Suppose “John” is marked for deletion.
- Then find inorder successor of “John”.
- Bill, John,Ken,kirk,leo,maurice,norma,paul,roger
- The inorder successor of “John” is “Ken”.
- The right subtree of Ken becomes the left subtree of Ken’s parent (Leo).
- Ken becomes the new left offspring of John’s parent (Norma).

# Search

- Case 1: If the tree is non ordered Binary tree then traversed using any traversal method – Preorder, Inorder, Postorder, converse Preorder, converse inorder, converse Postorder until the appropriate node is found.
- Case 2: If the tree is lexically ordered tree, then the number of comparisons are reduced than the non ordered Binary tree because compare the search element with the root node, if it is lexically greater than the root node then examined only right sided binary subtree. The search continue until either the item I found or an empty subtree is encountered. If the search ends with an empty subtree, then the item is not present in the tree.
- Apply the same method for left subtree, if the search element is lexically less than the root node.