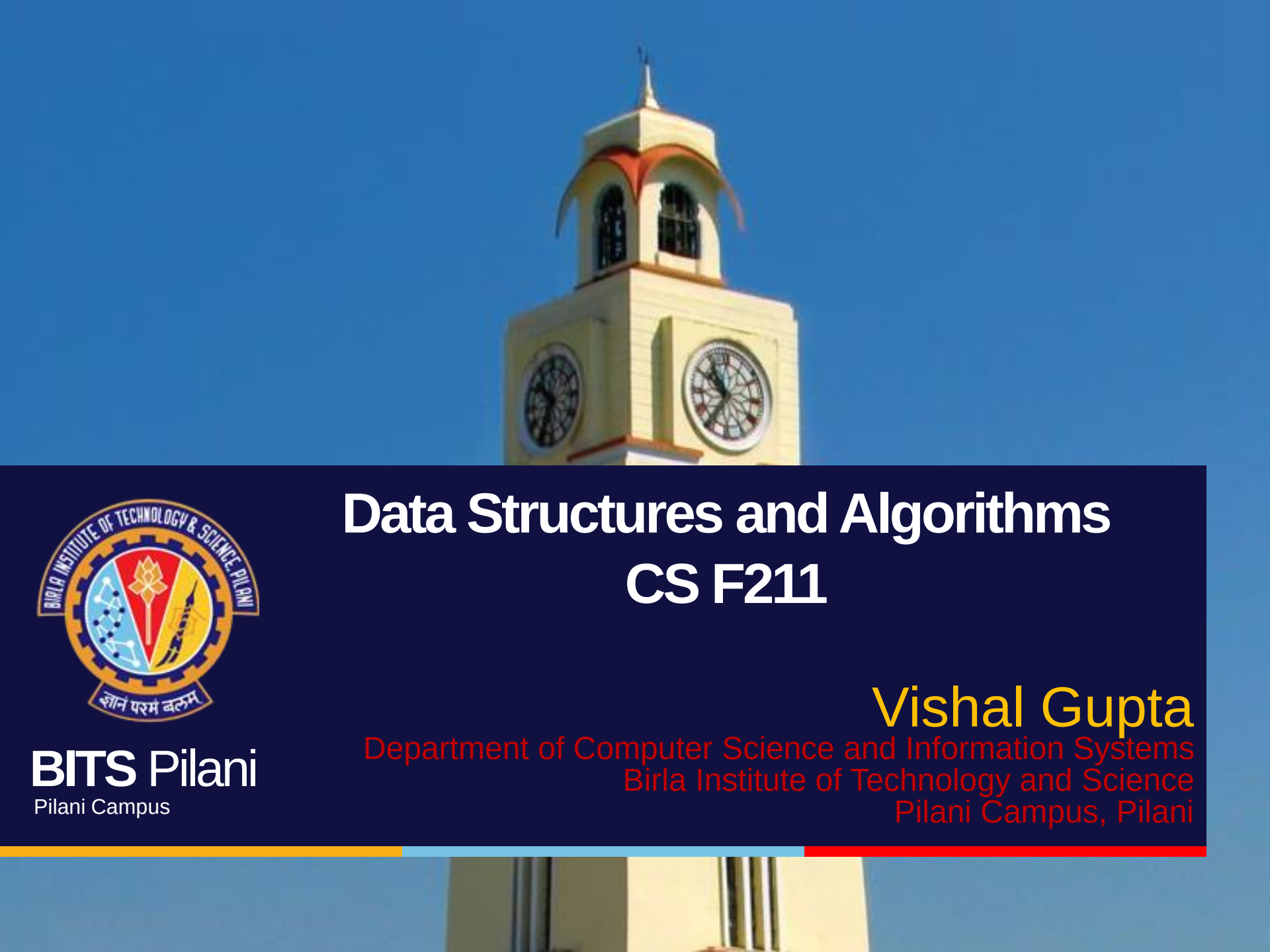


Data Structures and Algorithms
CS F211

Vishal Gupta

Department of Computer Science and Information Systems
Birla Institute of Technology and Science
Pilani Campus, Pilani

BITS Pilani
Pilani Campus

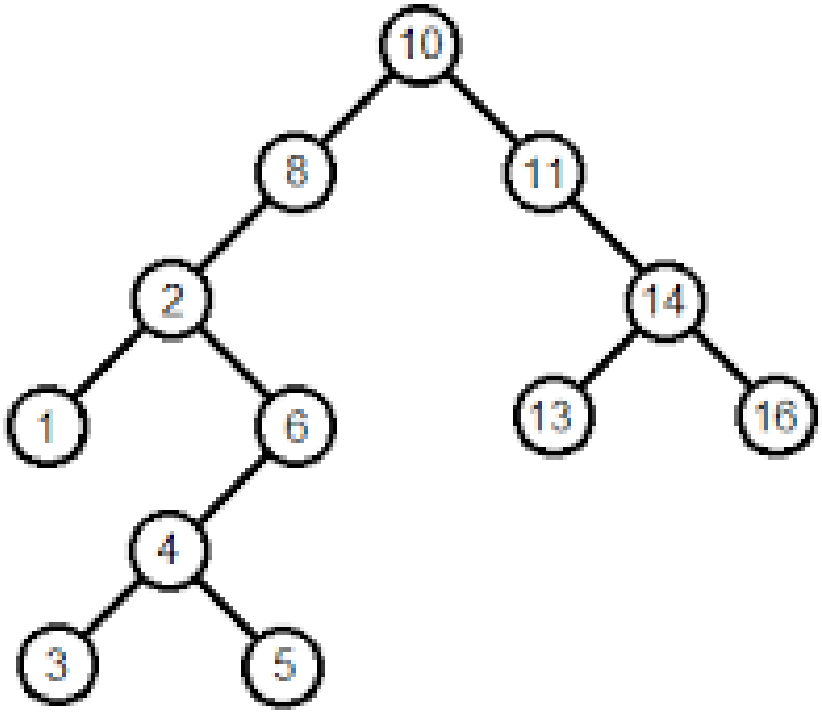
innovate achieve lead

BITS Pilani

Pilani Campus

Binary Search trees

Pseudo code for INORDER

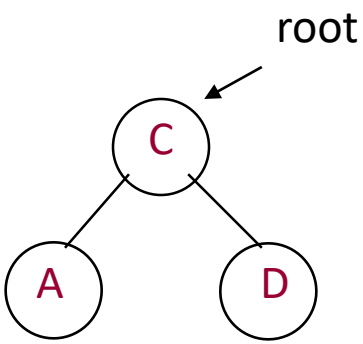


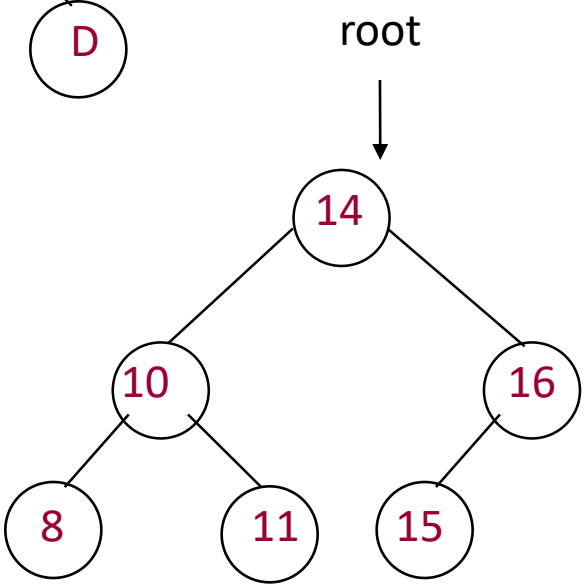
Binary Search Trees

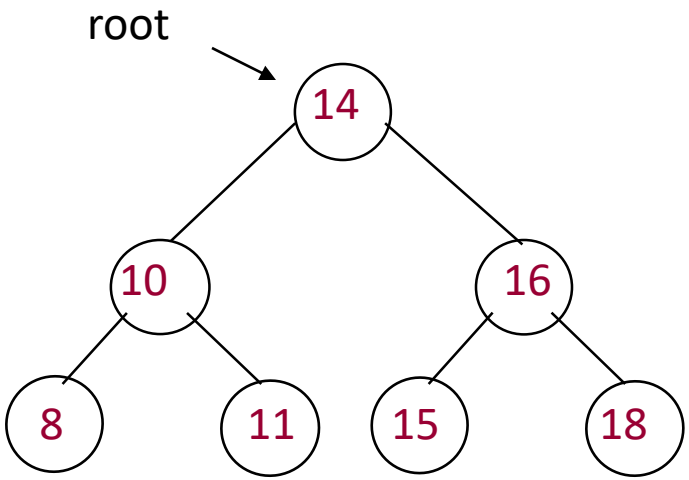
A **Binary Search Tree (BST)** is a binary tree with the following properties:

- The key of a node is *always greater* than the keys of the nodes in its left subtree
- The key of a node is *always smaller* than the keys of the nodes in its right subtree

Binary Search Trees: Examples





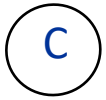


Building a BST

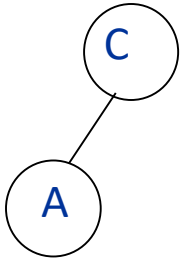
Build a BST from a sequence of nodes read one a time

Example: Inserting **C A B L M** **(in this order!)**

1) Insert C

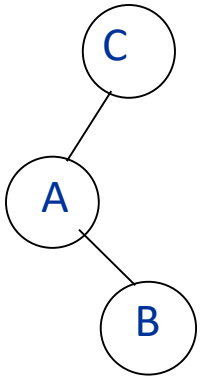


2) Insert A

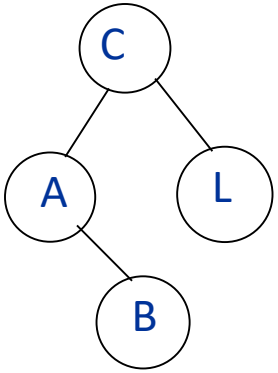


Building a BST

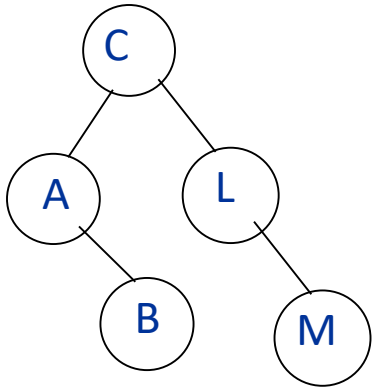
3) Insert B



4) Insert L



5) Insert M

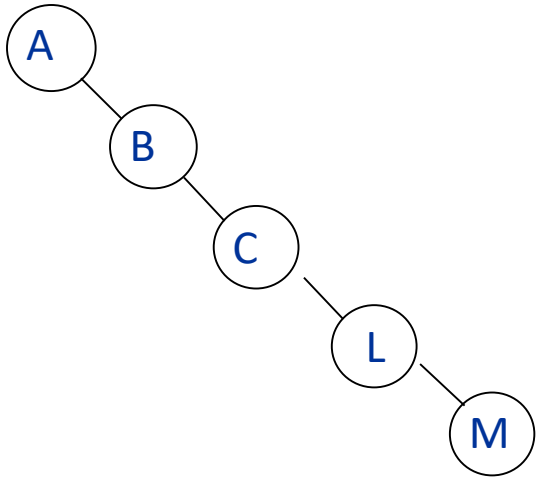


Building a BST

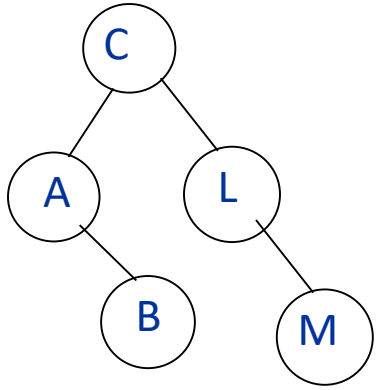
Is there a unique BST for letters A B C L M ?

NO! Different input sequences result in different trees

Inserting: A B C L M

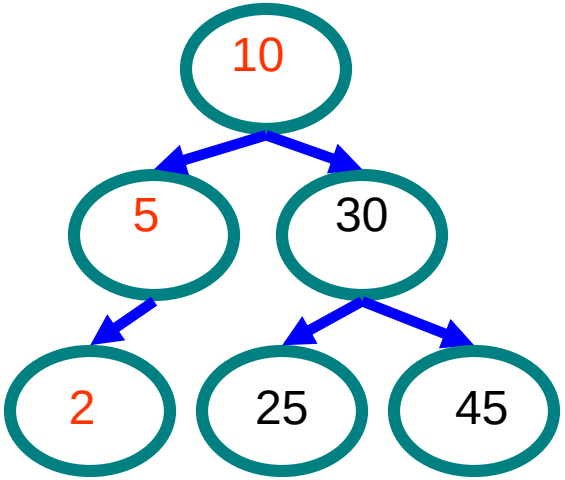


Inserting: C A B L M



Example Binary Searches

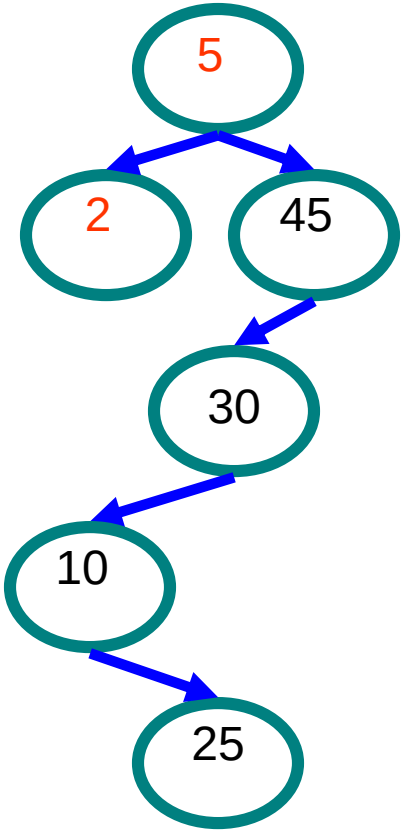
- **Find (2)**



10 > 2, left

5 > 2, left

2 = 2, found



5 > 2, left

2 = 2, found

Recursive Search of Binary Tree

```
Node Find( Node n, Value key) {
    if (n == null)                  // Not found
        return( n );
    else if (n.data == key)         // Found it
        return( n );
    else if (n.data > key)          // In left subtree
        return Find( n.left );
    else                            // In right subtree
        return Find( n.right );
}
```

Complexity of Search

- Running time of searching in a BST is proportional to the height of the tree.

If n is the number of nodes in a BST, then

- **Best Case** – O(logn)
- **Worst Case** – O(n)

Binary Search Tree - Insertion

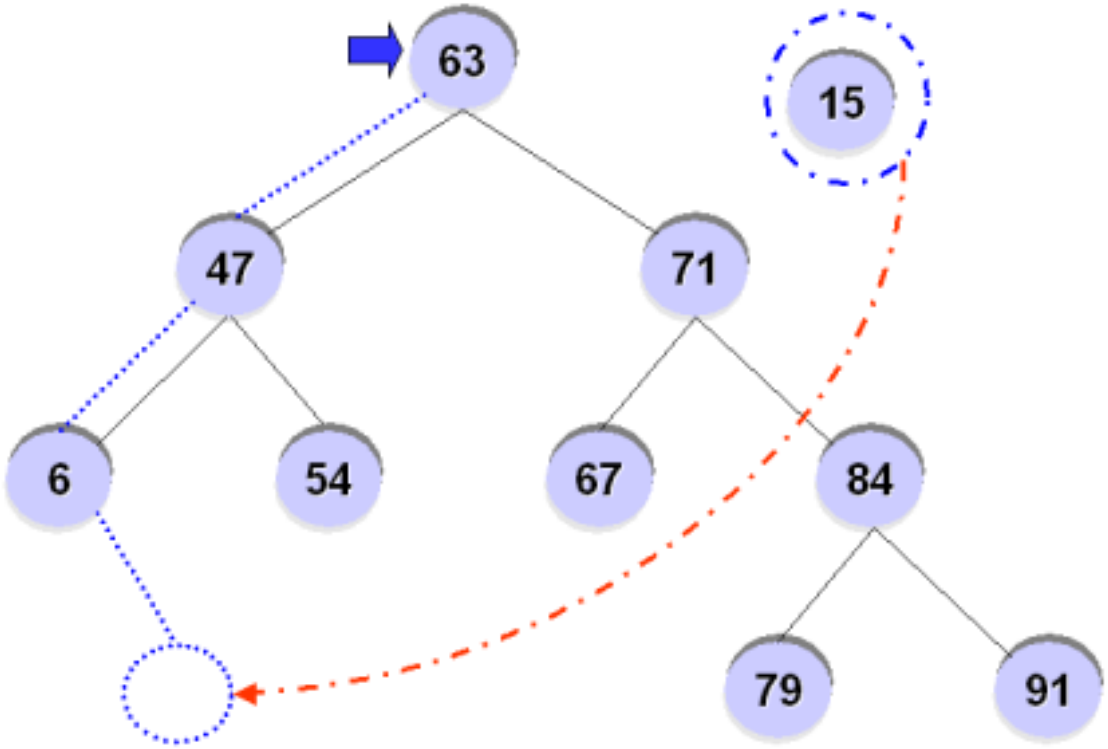
Insert Algorithm

- If value we want to insert < key of current node, we have to go to the left subtree
- Otherwise we have to go to the right subtree
- If the current node is empty (not existing) create a node with the value we are inserting and place it here.

Insertion - Example

innovate achieve lead

For example, inserting ’15’ into the BST?



Binary Search Tree - Deletion

There are 3 possible cases

Case1 : Node to be deleted has no children

→ **We just delete the node.**

Case2 : Node to be deleted has only one child

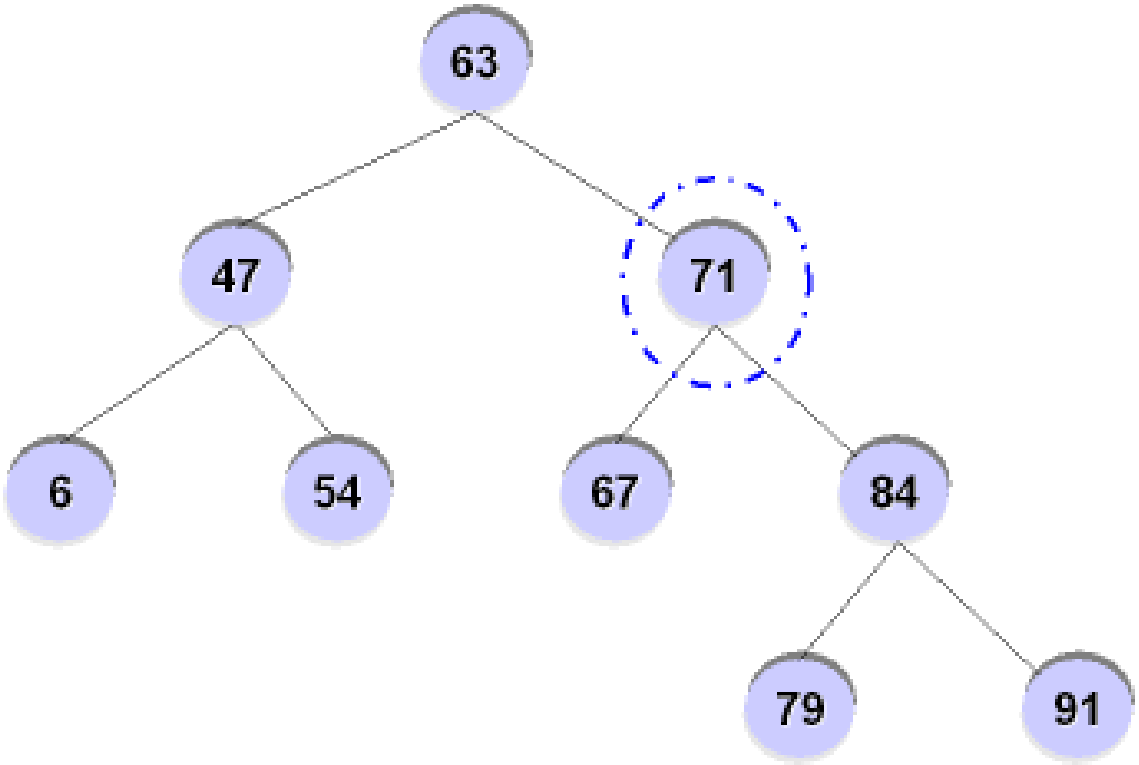
→ **Replace the node with its child and make the parent of the deleted node to be a parent of the child of the deleted node**

Case3 : Node to be deleted has two children

Binary Search Tree - Deletion

innovate achieve lead

Node to be deleted has two children



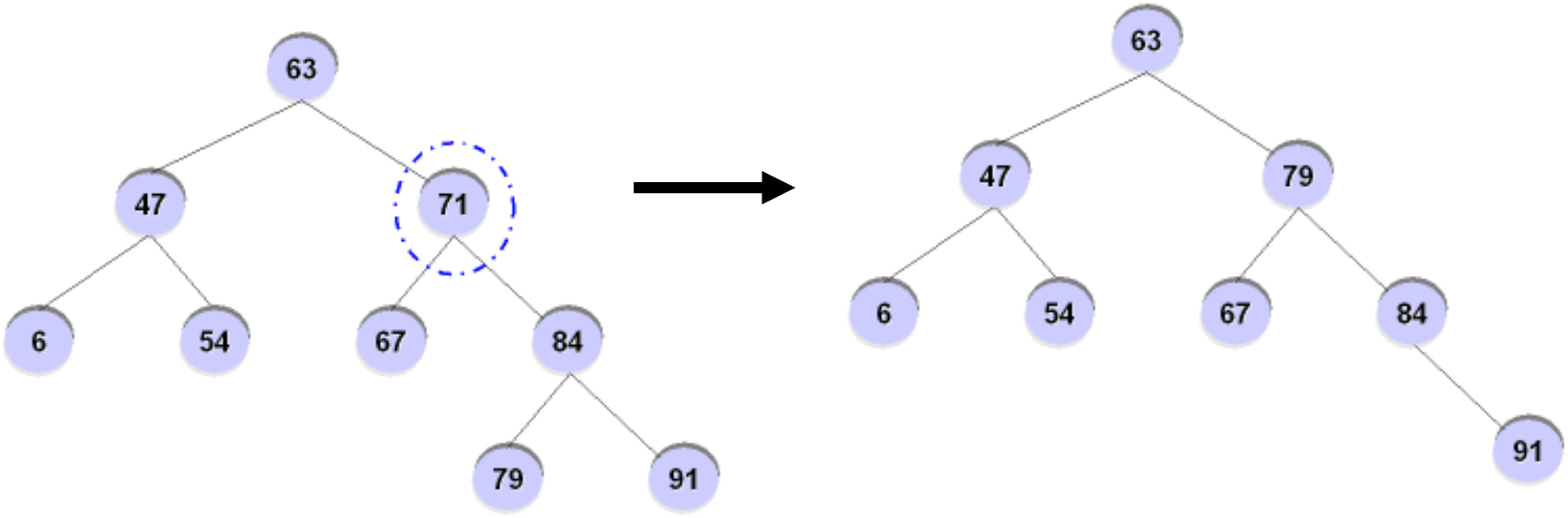
Binary Search Tree - Deletion

Node to be deleted has two children

Steps:

- Find minimum value of right subtree
- Delete minimum node of right subtree but keep its value
- Replace the value of the node to be deleted by the minimum value whose node was deleted earlier.

Binary Search Tree - Deletion



Convert the following into a pseudo code.


```
Node deleteNode(Node root, int valueToDelete) {
  if root = null
    return node
  if root.value < valueToDelete
    deleteNode(root.right, valueToDelete)
  if root.value > valueToDelete
    deleteNode(root.left, valueToDelete)
  else
    if (isLeafNode(root))
      return null

    if (root.right == null)
      return root.left
    if (root.left == null)
      return root.right

    else
      minValue = findMinInRightSubtree(root)
      root.value = minValue
      removeDuplicateNode(root)
      return root
```