Binary Search & Trees

Learn Vid Fun...

Operations on BST -

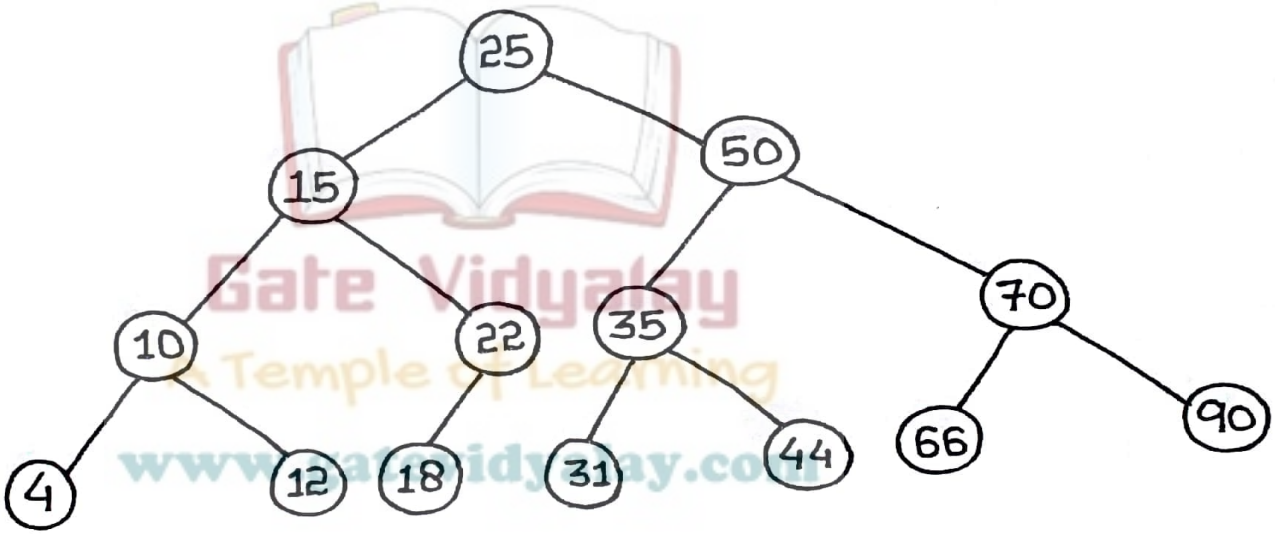
The following operations are performed on a Binary Search Tree -

Search Operation -

To search a given key in Binary Search Tree, we first compare it with root. If the key is present at root, we return root. If the key is greater than root's key, we recur for right subtree of root node otherwise we recur for left subtree.

Example-

Search for 45 in the BST-



Step-01: Start at the root. As $45 > 25$, so search in right subtree.

Step-02: As $45 < 50$, so search in 50's left subtree.

Step-03: As $45 > 35$, so search in 35's right subtree.

Step-04: As $45 > 44$, so search in 44's right subtree.

But 44 has no subtrees. So, 44 is not present in the BST.

Insertion Operation -

A new key is always inserted at leaf. We start searching a key from root till we hit a leaf node. Once a leaf node is found, the new node is added as a child of the leaf node.

Example -

Step-01: Start at root node 100. As $40 < 100$, so search in 100's right subtree.

Step-02: As $40 > 20$, so search in 20's right subtree.

Step-03: As $40 > 30$ (leaf node), so add 40 to 30's right subtree.

Deletion Operation-

Deleting a node from Binary Search Tree gives rise to following 3 cases-

Case-I: Deleting a node with no child (leaf node)

Case-II: Deleting a node with one child

Case-III: Deleting a node with two children

Case-I: Deleting a leaf node-

It is very simple. Just remove the leaf node from the tree.

Example-

Case-II: Deleting a node with one child-

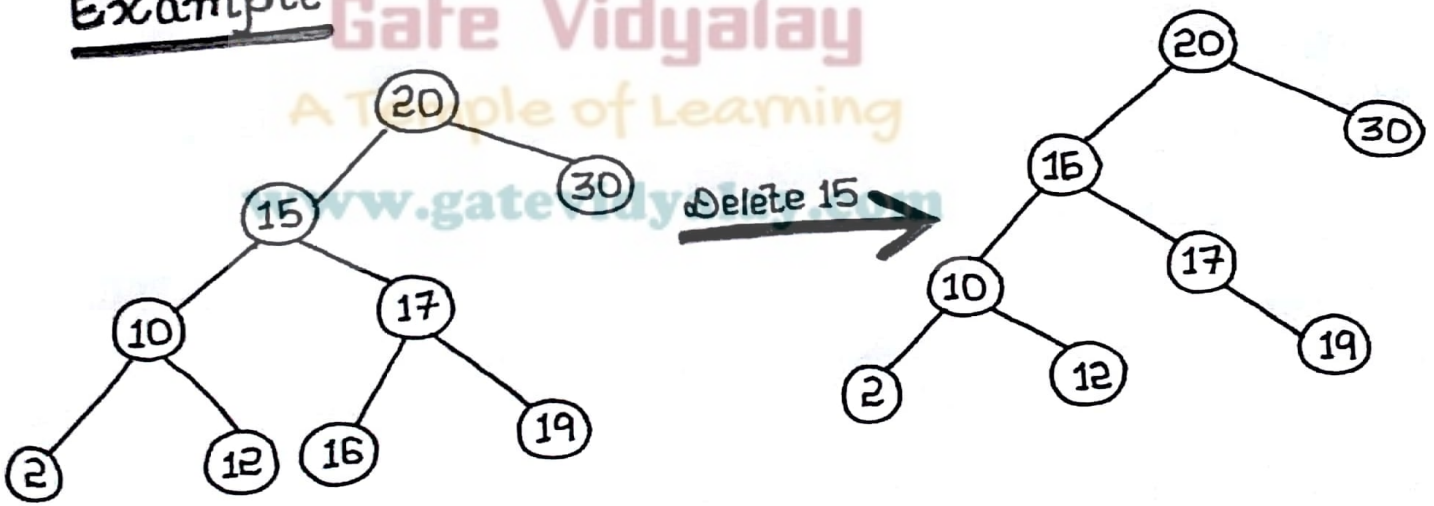
Just make the child of the deleting node, the child of the grandparent.

Example-

Case-III: Deleting a node with 2 children-

Method-I: Go to the right subtree of the deleting node, pluck the least element called inorder successor and replace with the deleting node.

Example-



Method-II:

Go to the left subtree of the deleting node, pluck the greatest element called inorder predecessor and replace with the deleting node.

Example-