

DSA LAB Assignment

Name:

Ahmad Hussain

Reg no#

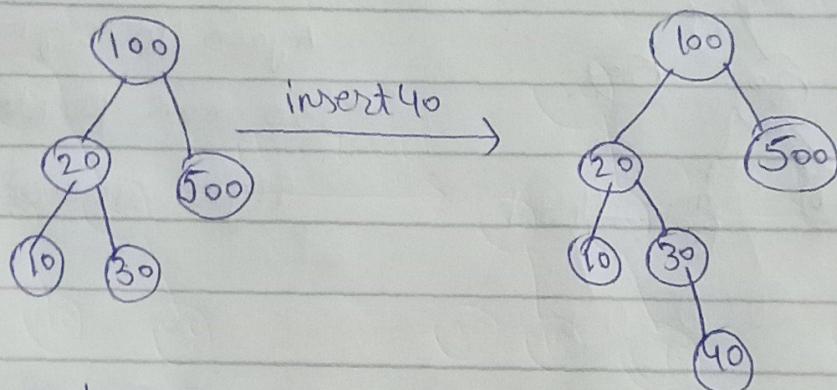
SP22-BCS-008

Submitted to:

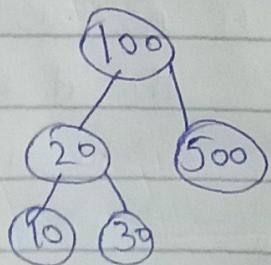
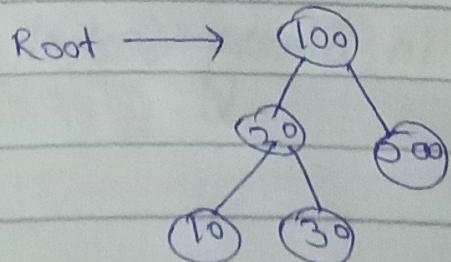
Mam Yasmeen Jani

Comsat University Islamabad, Vehari
Campus

① Insertion in binary search tree (BST)

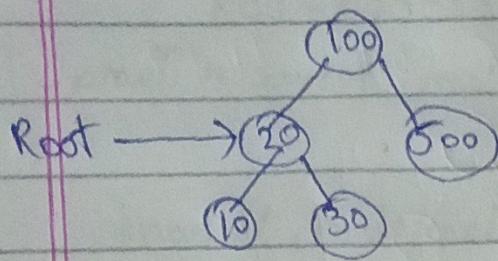
Illustration

Consider the following BST.

 $X = 40$ (the node to be inserted)Step 1: comparing X with Root Node.

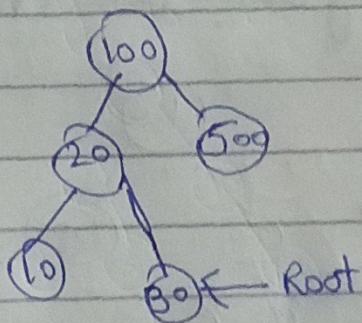
since 100 is greater than 40 move point to the left child (20)

Step 2: Comparing X with left child of root node.



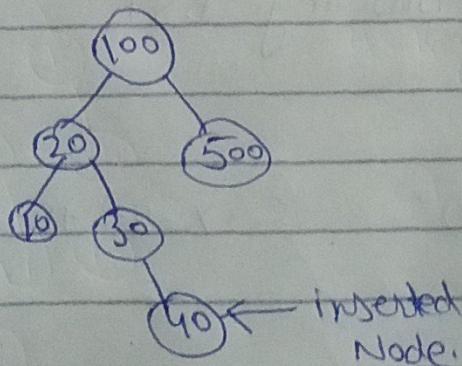
since 20 is less than 40
move pointer to the
Right child (30)

Step 3: Comparing X with the right child of 20



Again 40 is greater than
30 move pointer to the
right side of 30.

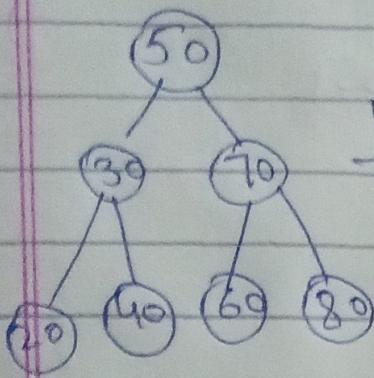
Step 4: insert item to the right of 30.



As 40 is greater than 30,
thus it will be inserted
to right side of 30.

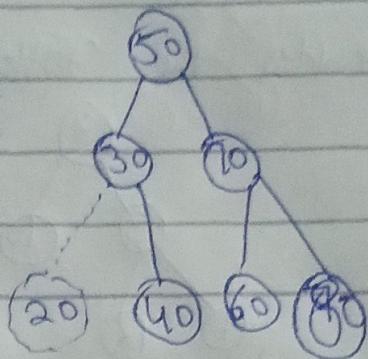
② Deletion in Binary Search Tree (BST)

Case ① Delete a leaf node in BST.



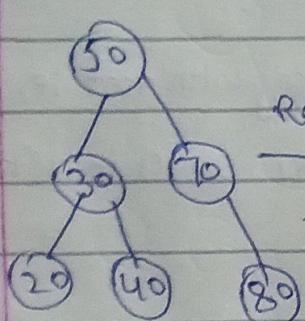
Delete Node 20

Assign node
to null



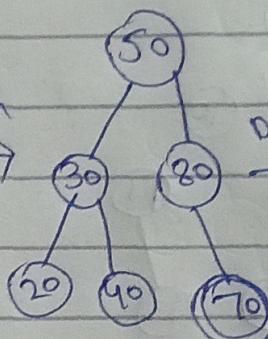
After Deleted Node 20

Case ② Delete a Node with single child in BST

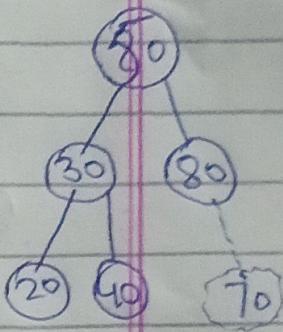


Delete Node 70

Replace 70 with
80

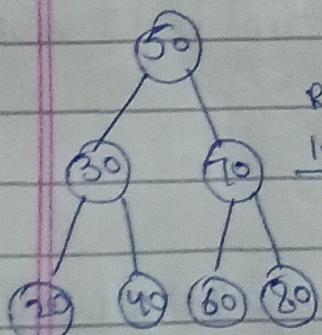


Delete 70



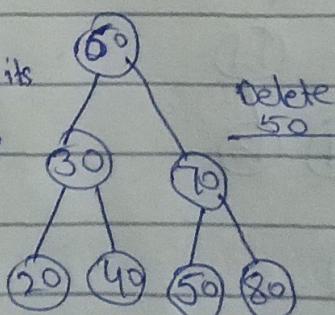
After deleting
Node 70

Case ③ Delete a Node with Both children in BST

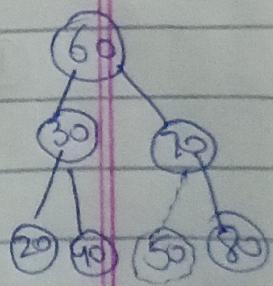


Delete Node 50

Replace 50 with its
In order Successor



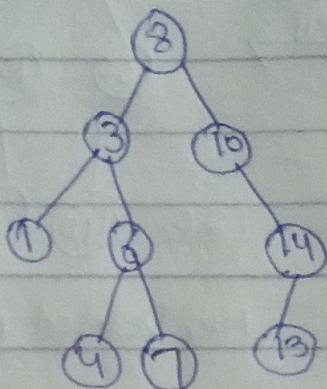
Delete 50



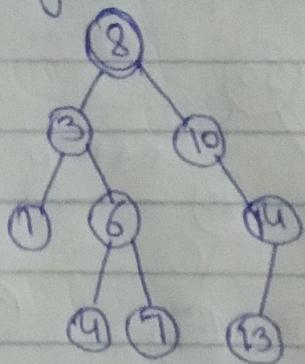
After deleting
Node 50

③ Searching in Binary Search Tree (BST)

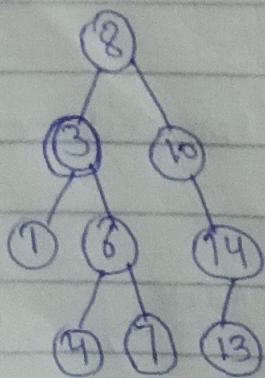
Consider the following BST



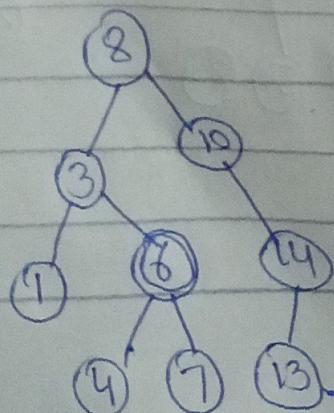
Key is 6.



compare key with root ie, 8
as $6 < 8$, search in left subtree
of 8.



As key (6) is greater than 3,
search in the right subtree
of 3.

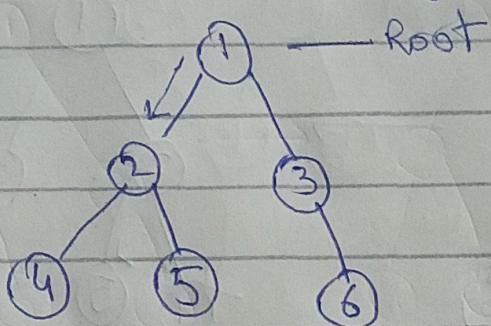


As 6 is equal key (6),
So we have found
the Key.

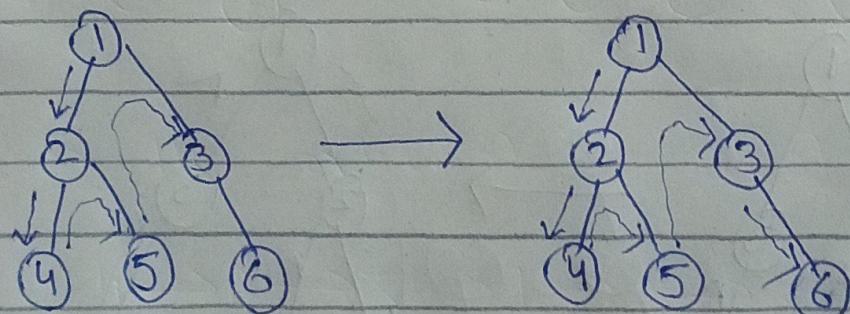
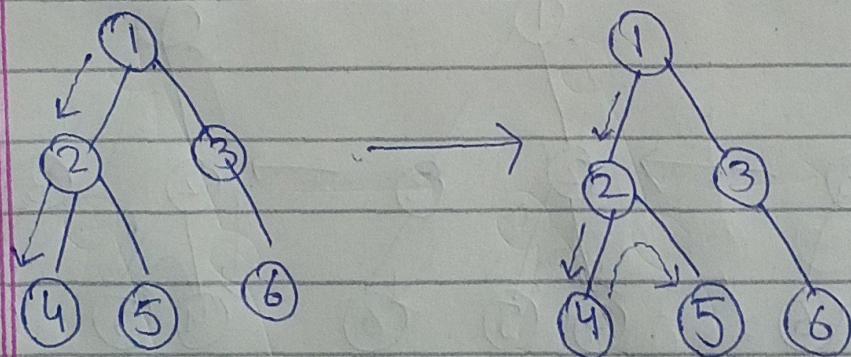
④ Pre Order Traversal in Binary Search Tree

- i) Visit the root Node.
- 2) Traverse the left subtree recursively.
- 3) Traverse the right subtree recursively.

Consider the following tree.



Visit the root than left subtree and the right subtree.

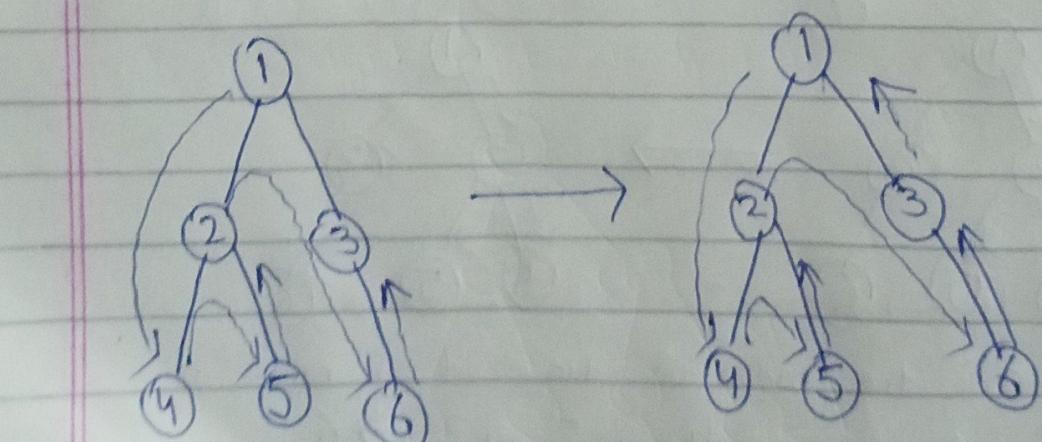
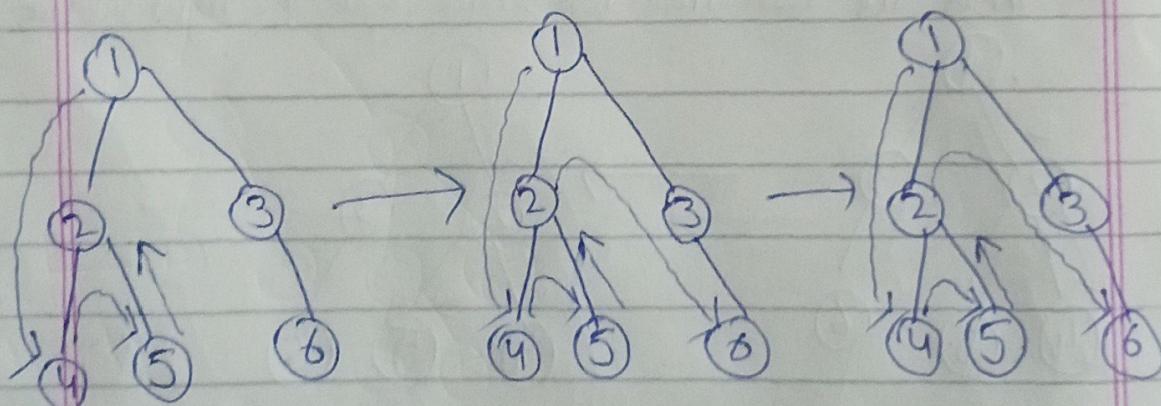
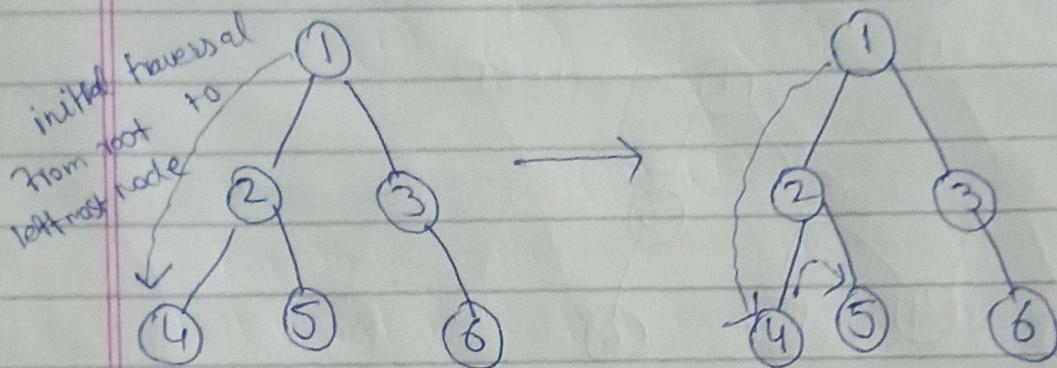


So the order of traversal of node is

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 6$

⑤ Post order Traversal of Binary Search tree (BST)

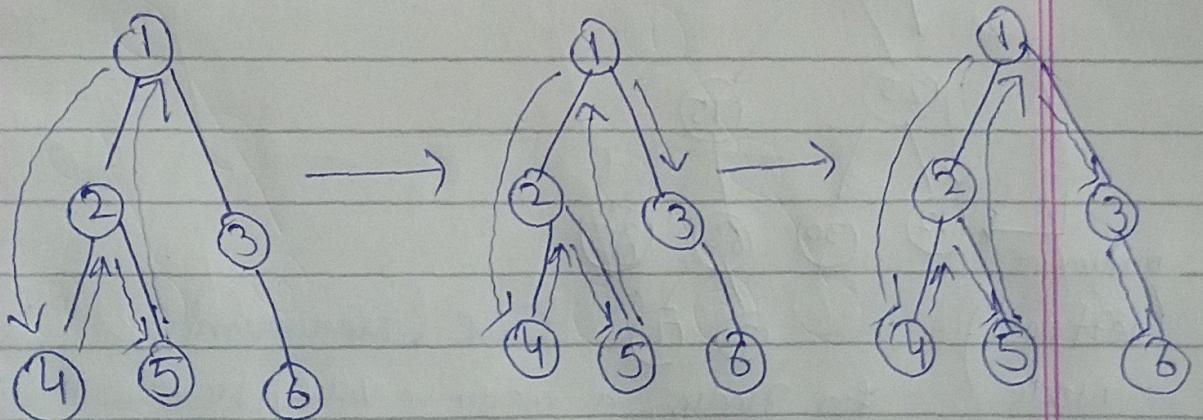
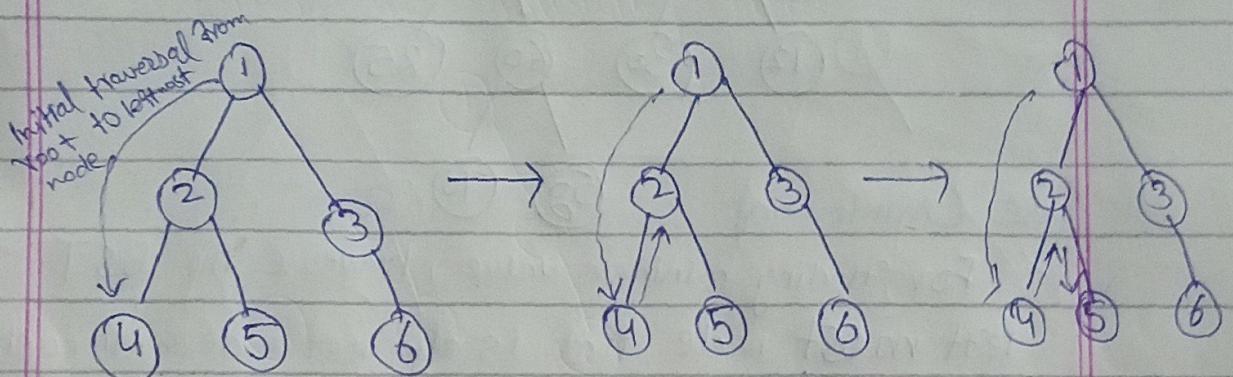
- 1) Follow step 2 to 4 until root \neq Null.
- 2) Post order (root \rightarrow left)
- 3) Post order (root \rightarrow right)
- 4) Write root \rightarrow data,
- 5) End loop.



So the order of traversal of nodes is
 $4 \rightarrow 5 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 1$

⑥ Inorder traversal of Binary Search Tree (BST)

- 1) Follow step 2 to 4 until $\text{root} \neq \text{NULL}$.
- 2) Inorder ($\text{root} \rightarrow \text{left}$)
- 3) Write $\text{root} \rightarrow \text{data}$.
- 4) Inorder ($\text{root} \rightarrow \text{right}$)
- 5) End loop

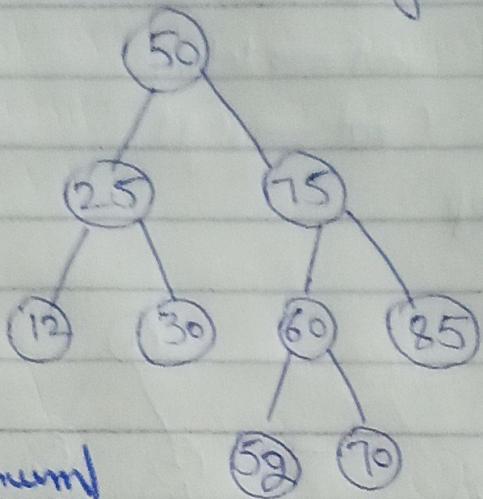


So the order of traversal of nodes is

4 → 2 → 5 → 1 → 3 → 6

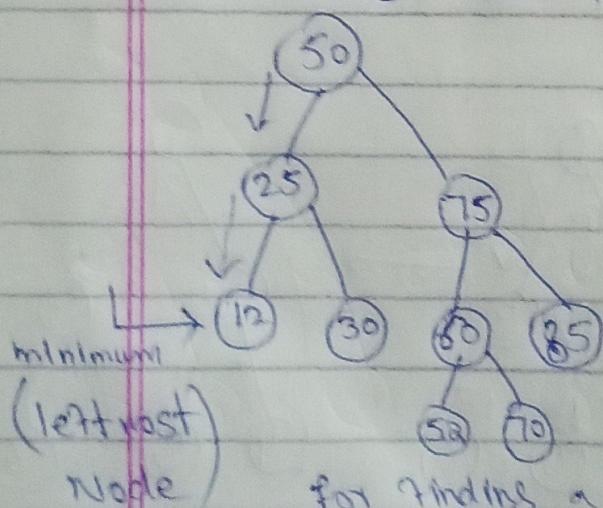
① finding a minimum and maximum value in binary Search Tree (BST)

Let we have a binary search tree.



Case (Minimum)

For finding minimum value, traverse the BST left most node that is also leaf node with no child left



Case (Maximum)

for finding a maximum value, traverse the BST to the right most node that is also leaf Node with no child left

