14/10/20

ADS - Lab - 4

Abhishikat Soni
1BM18CS004
5A

Pseudocode / Algorithm

1. Insert new node using recursion, as while backtracking we will be able to check whether they are balanced or not
2. Make height, h = 1
3. When new node added, parent node height get incremented by 1.
4. As mentioned in Step 1, every ancestor height will get updated while backtracking to the root.
5. Check balance factor (height of (left subtree - right subtree))
6. If balance factor (bf) = 1 ⇒ balanced
7. bf > 1 ⇒ left height more than right ⇒ Need to rotate.
8. If curr_node < new_node ⇒ Then it is left-left case otherwise left-right case.
9. If bf < -1 ⇒ Not balanced ⇒ Need Rotation (Right-Right Case or Right-left case)
10. If new_node < curr_node ⇒ Right-Right Case otherwise Right-left case.

For deletion

1. We will do normal BST deletion.
2. Now, curr_node must be one of the ancestors of deleted node.
   Update height of curr_node.
3. Calculate balance factor (bf) of curr_node.
4. Rest Step same as Step 5 to 10.