ADS LAB-5

1.) Create node with key, height values along with Node left, Node right.

2.) Create rightRotate, LeftRotate, getBalance functions

```
rightRotate (Node y) {
        Node x = y.left;
        Node T2 = x.right;
        x.right = y;
        y.left = T2;
        // Update x and y heights
        return x;
    }
leftRotate (Node x) {
        Node y = x.right;
        Node T2 = y.left;
        y.left = x;
        x.right = T2;
        // Update x & y heights
        return y;
    }
getBalance {
        return height(N.left) - height(N.right);
    }
```

3.) Insert {

```
if (node == null)
        return node(key);
if (key < node.key)
        node.left = insert(node.left, key);
else if (key > node.key)
        node.right = insert(node.right, key);
else
    return node;
```

```
// Update node height.
// Below code is for updating heights of every other node
if ( getBalance(Node) > 1  && node.left.key > key)
        return rightRotate(node);

if( getBalance(Node) < -1  && node.right.key < key)
        return leftRotate

if ( getBalance(Node) > 1  && key > node.left.key)
        node.left = leftRotate(node.left);
        return rightRotate(node);

if( balance < -1  && key < node.right.key)
        node.right = rightRotate(node.right)
        return leftRotate(node);

return node;

}
```

4.) deleteNode {

~~Similar to insert, recursively~~

```
if ( root == null )
        return root;

if(key < root.key)
        deleteNode ( root.left, key)

if ( key > root.key)
        deleteNode ( root.right, key)

else {
    if( root.left == null || root.right == null)
        {
          Node temp = null;
          if (temp = root.left)
                temp = root.right;
          else  temp = root.left;
          if (temp == null)
                temp = root
                root = null
        }
    else
        root = temp
else{
temp = minValueNode(root.right)
root.key = temp.key;
root.right = deleteNode(root.right, temp.key) }
```

5.) Update heights after delete, of every other node similar to insert.