1

Advanced Data Structure
Batch - 5

Rutazeet Ritik Rout
1BM18CS151
14/10/20

Program - 4

Insertion :-

```
struct node*
      insert (node * head, int val)
   {
        if (head == NULL)
            return newnode (val)
        if (head → val > val)
            head → left = ~~head~~ insert( head → left, val)
        else if (head → val < val)
            head → right = insert ( head → right, val)
        else
              return node

        int bal = height (head → left) - height (head → right)

        if (balance > 1)
        {
            if ( head → left → key > val)
              ~~left,~~ left rotate
            else
                left right rotate
        }
        else ~~*~~ if ( balance < - 1)
        {
            if (head → right → key > val )
                right left rotate
            else
                right right rotate.
        }
         else.
             return node.
   }
```

Rutazeet

Deletion:-

```
struct node * delete (node * root, int key) {
{  if (root == null)
       return root
   if (key < root -> key)
      root -> left = delete (root -> left, key)
   else if (key > root -> key)
      root -> right = delete (root -> right, key)
   else {
        if((root -> left == NULL) || (root -> right == NULL)){
           node * temp = root -> left ? root -> left : root -> right
           if (temp == NULL) {
                temp = root
                root = NULL
           }
           else
              *root = *temp
           free (temp)
        }
        else {
           node* temp = minValuenode (root -> right)
           root -> key = temp -> key
           root -> right = delete (root -> right, temp -> key)
        }
   }
   if (root == NULL)
       return root
   root -> height = 1+ max (height (root -> left),
                            height (root -> right))

   int balance = get balance (root)
```

Rutazeet

```
if (balance >1)
{
if(getbalance(root → left) > 0)
     Left left ~~rotate~~ case.    // right rotate
 else
     Left right case.   //right rotate
 }
 elseif ( balance < -1)
 {
    if (getbalance (root → right) > 0)
      right left case.          // left rotate
    else
      right right case          // left rotate.
 }
 else
      return root
 }
}.
```