AVL tree

Prajwal PatilN
1BM17CS062
Sec 'A'

```
struct node* insert(struct node * r, int data)
{ if (r==NULL)
 { struct node*n;
   n=new struct node;
   n->data = data;
   r=n;
 r-> left = r-> right = NULL;
   r -> height = 1;
     return r;
 }
 else
 { if (data < r->data)
    r -> left = insert(r->left, data)
 else
   r-> right = insert (r-> right, data);
 }
 r->height = calheight(r);
   if (bf(r) ==2 && bf(r->left) ==1)
    {
       r = LL rotation(r);
    }
 else if (bf (r) = =2&& bf(r-> left ==-1)
     {
        r= Lr rotation(r);
     }
   return r
 }
```

Deletion

```
struct node* deleteNode(struct node*P, int data
{
if(P->left==NULL && P->right==NULL)
 {
   if (P== this -> root)
      this ->root = NULL;
        delete P;
        return NULL;
      }
 struct node *t;
 struct node *q;
 if (P-> data < data)
  {
    P-> right = delete Node(P-> right, data);
  }
else if (P-> data > data)
 {
   P-> left = delete node(P-> left, data);
  }
 else
  {
    if (P-> left != NULL)
 {
   q = insuc(P-> right);
     P-> data = q-> data;
     P-> right = delete Node(P->right, q->data);
   }
 }
 if (bf(P) == 2 && bf(P-> left) == 1)
 {
   P = LL rotation(P);}
```

```
elseif (bf(P)==2 && bf(P-> left==-1)
{
 P = LR rotation (P);}
elseif (bf(P) ==-2 && bf(P-> right ==1)
{
  P= RL   rotation (P);
}
elseif (bf(P) ==-2 && bf(P-> right) ==0)
 {
  P =LL rotation (P);
 }
   return P;
}
```