

ADS Lab ③ - AVL Trees.

Name - Nimit Sajal -  
USN - 18M18CS063  
Date - 14/10/20

Insert

```
int insert (node * head, int val) {
```

```
    if (head == null)
```

```
        return new node (val);
```

```
    if (head->val > val)
```

```
        head->left = insert (head->left, val);
```

```
    else if (head->val < val)
```

```
        head->right = insert (head->right, val);
```

```
    else
```

```
        return node;
```

```
    int bf = height (head->left) - height (head->right)
```

```
    if (balance > 1)
```

```
    { if (head->left->key > val)
```

```
        left rotate;
```

```
    else
```

```
        left right rotate;
```

```
    }.
```

```
    else if (balance < -1)
```

```
    { if (head->right->key > val)
```

```
        right left rotate;
```

```
    else
```

```
        right rotate;
```

```
    }.
```

```
    else
```

```
        return node;
```

```
}
```

*[Signature]*

## Delete:

```
int delete (Node * root, int key) .  
{  
    if (root == null) .  
        return root;  
    if (key < root->key)  
        root->left = deleteNode (root->left, key);  
    else if (key > root->key)  
        root->right = deleteNode (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 = deleteNode (root->right, temp->key);  
        }  
    }  
}
```

Name - Nimit Sagar  
USN - IBM18CS063  
Date - 14/10/20 - wed.

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

```
root->height = 1 + max(height(root->left), height(root->right));
```

```
int balance = getBalance(root);
```

```
if (balance > 1 && getBalance(root->left) < 0)
{
    root->left = leftRotate(root->left);
    return rightRotate(root);
}
```

```
if (balance < -1 && getBalance(root->right) > 0)
    return leftRotate(root);
```

```
if (balance < -1 && getBalance(root->right) > 0)
{
    root->right = rightRotate(root->right);
    return leftRotate(root);
}
```

```
return root;
```

```
}
```

*[Signature]*