

AVL-Insert(~~Node~~ node, key):

IBM18CSD31
-pendum

if node == NULL

return newNode(key)

if (key < node → key)

node → left = AVL-Insert(node → left, key)

if (key > node → key)

node → right = AVL-Insert(node → right, key)

else

return node

// get height

node → height = 1 + max(height(node → left),
height(node → right))

// get balance of the node

// Left
Left case { if (balance > +1 and key < node → left → key)
return rightRotate(node)

// Right
Right case { if (balance < -1 and node → right → key < key)
return leftRotate(node)

// Left
Right case { if (balance > +1 and key > node → left → key)
node → left = leftRotate(node → left)
return rightRotate(node)

// Right
Left case { if (balance < -1 and key < node → right → key)
node → right = rightRotate(node → right)
return leftRotate(node)

AVL Delete :

IBM18CS031

Handwritten

deleteNode(root, key):

if (root == NULL)
return root

else if (key < root->key)

root->left = deleteNode(root->left, key)

else if (key > root->key)

root->right = deleteNode(root->right, key)

else

{ //node with only one child

if (root->left == NULL) ||

{ (root->right == NULL)

temp = root->left ? root->left : root->right

//No child

if (temp == NULL)

temp = root

root = NULL

else

root = temp

} free(temp)

else {

temp = minValueNode(root->right)

root->key = temp->key

root->right = deleteNode(root->right,
temp->key)

}

(12)


```
if (root == NULL)
    return root
```

IBH18CS051
ifunilunif

```
// get balance of root
```

```
// Left Left Case
```

```
if (balance > 1 && getBalance(root->left) >= 0)
    return rightRotate(root)
```

```
// Left Right Case
```

```
if (balance > 1 and getBalance(root->left) < 0)
    root->left = leftRotate(root->left)
    return rightRotate(root)
```

```
// Right Right Case
```

```
if (balance < -1 and getBalance(root->right) <= 0)
    return leftRotate(root)
```

```
// Right Left Case
```

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

```
// Finally
```

```
return root
```