

MD YASEEN AHMED  
13M19CS404

## → Insertion : [ AVL TREES ]

insert (node, Key)

if !node

return newNode(Key)

if Key < node → Key

node → left = insert (node → left,  
Key)

if Key ≥ node → Key

node → right = insert (node → right,  
Key)

else

return node

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



Balance(Node)

int bal = getBalance(Node)

if bal > 1 and Key < node->left  
->Key  
return rotateRight(Node)

if bal < -1 and Key > node->right  
->Key  
return leftRotate(Node)

if bal > 1 and Key > node->left->key  
node->left = rotateLeft(node->left)  
return rotateRight(Node)

if bal < -1 and Key < node->right->Key  
node->right = rotateRight(node->right)  
return rotateLeft(Node)

return Node



→ Deletion:

Delete (root, key)

if !root  
return root

if key < root->key  
root->left = delete(root->left,  
key)

if key > root->key  
root->right = delete(root->  
right, key)

else  
{

if !root->left || root->right  
root = (root->left) ?  
root->left :  
root->right



else

temp = minVal(root->right)

root->key = temp->key

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

if !root

return root

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

return Balance(root)

}