

14-10-2020

AVL Tree

LAB-5

Nitish.N.Banakar

18M18CS065

Insert (Node\* node, int data):

if node == NULL

return newNode(data)

else if data < node->data

node->lchild = insert (node->lchild, data)

else if data > node->data

node->rchild = insert (node->rchild, data)

node->height = 1 + max(height (node->lchild), height (node->rchild));

balance = getBalance (node)

if (balance < -1 && data > node->rchild->data)

node = leftrotate (node) # left rotate

if (balance > 1 && data < node->lchild->data)

node = rightrotate (node)

if (balance > 1 && data > node->lchild->data)

node->lchild = leftRotate (node->lchild)

node = leftRotate (node)

return node

(1)

Nitish

Delete (Node\* root, int data)

if (root == NULL)  
return root

if (data < root->data)

root->lchild = Delete (root->lchild, data)

else if (data > root->data)

root->rchild = Delete (root->rchild, data)

else {

if root->lchild is NULL or root->rchild is NULL

{ temp = root->lchild ? root->lchild : root->rchild

if temp is NULL

temp = root

root = NULL

else

root = temp

Free (temp) }

else {

temp = minValueNode (root->rchild)

root->data = temp->data

root->rchild = Delete (root->rchild, temp->data)

}

}

if root is NULL

return root

14-10-2020

root  $\rightarrow$  height =  $1 + \max(\text{height}(\text{root} \rightarrow \text{rchild}), \text{height}(\text{root} \rightarrow \text{lchild}))$   
balance = getBalance(root);

if (balance > 1 && getBalance(root  $\rightarrow$  lchild) < 0)  
    root  $\rightarrow$  lchild = leftRotate(root  $\rightarrow$  lchild)  
    root = rightRotate(root)

if (balance > 1 and getBalance(root  $\rightarrow$  lchild) >= 0)  
    root = rightRotate(root)

if (balance < -1 and getBalance(root  $\rightarrow$  rchild) <= 0)  
    root = leftRotate(root);

if (balance < -1 and getBalance(root  $\rightarrow$  rchild) > 0)  
    root  $\rightarrow$  rchild = rightRotate(root  $\rightarrow$  rchild)  
    root = leftRotate(root)

return root

(3)

Notish