

14-10-2020

MANU.N.Y

IBM18CS053

AVL Tree

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 → child → data)

node = left rotate (node)

if (balance > 1 && data < node → child → data)

node = right rotate (node)

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

node → lchild = left Rotate (node → lchild)

node = left Rotate (node)

return node

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 \rightarrow lchild is NULL or root \rightarrow rchild is NULL

~~{ if root \rightarrow lchild is NULL or root \rightarrow rchild is NULL~~

{ temp = root \rightarrow lchild? root \rightarrow lchild : root \rightarrow rchild

if temp is NULL

temp = root

root = NULL

else

root = temp

free(temp) }

else

{ temp = min Value Node (root \rightarrow rchild)

root \rightarrow data = temp \rightarrow data

root \rightarrow rchild = Delete (root \rightarrow rchild, temp \rightarrow data)

}

}

if root is NULL

return root

root \rightarrow height = 1 + max(height (root \rightarrow rchild), height (root \rightarrow lchild))

balance = getBalance (root);

if (balance > 1 & getBalance (root \rightarrow lchild) < 0)

root \rightarrow lchild = left Rotate (root \rightarrow lchild)

root = right Rotate (root)

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

root = right Rotate (root)

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

root = left Rotate (root);

if (balance < -1 and getBalance (root \rightarrow rchild) > 0)

root \rightarrow rchild = right Rotate (root \rightarrow rchild);

root = left Rotate (root)

return root.