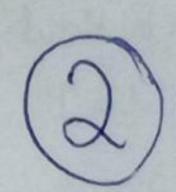
Rutageet Ritik Pout Advanced Data Structure 1BMIKCEIS! Batch-5 14/10/20 Program - 4 Insertion: -Struct Mode# cost insert (node & head, int val) if (head == NULL) return rewrode (val) if Chead -> val > val)
head -> left = twood insert ( head > left, val) else if (head -> val < val) 'read -> right = insurt ( head -> right, val) return node int made = height (head -> left) - height (head -> right) if (balance > 1) if ( head > left > key > val) else beft right rotate else sif C balance < - 1) if Chiad -> right -> key > kal)
right left rotate else right rotate. Use. return node



```
Deletion: -
struct node * delete ( node & root, int ky) {
    if (root == null)
          return root
     if (kmy < root > kmy)
root > left = deleter (root > left, kmy)
     else it ( key > root -> key)
         root -> right = delete (root -> right, key)
            if ((root > left == NULL) | (root > right == NULL) R
node * temp = root > left ? root > left: root > right
                if (temp = = NULL) &
                      femile = soot
                     200f = NULL
                    Aroot = #temp
             z free (temp)
                nodea temp = min Valuenale ( root -> right)
             root -> key = temp -> key
root -> right = delete (root -> right, temp > key)
       14 (2004 = = NOTP)
             return root
        root > height = 1+ max Chaight (root > left)

height (root > right)
       int balance = get balance (root)
```

3

if ( balance > 1)

if ( get balance ( root > left) > 0)

heft heft rotate case. // right rotate

else heft right case. // right rotate

glesif ( balance < -1)

if ( get balance ( root > right) > 0).

right left case. // left rotate

else return root

glese return root

Bressing