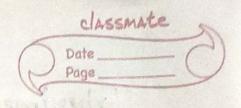


Experiment 5

Ques! - Program to search for anode with a given key in BST i) Iterative function tunc search (Node * root, int key) Node * curr = root, * parent = null while (cur; = null fl curi -) data ! = Rey) parent = curr if key < curr - data Curr= curr -> left Curr = curr -> right if parent = null if curr = anull KEY NOT FOUND else if key < parent > data Given key is left made of parcent nade Given key is right node of node with key



ii) Recursive Function

func search (Node* root, int key, Node*parent)

if (root == null)

KEY NOT FOUND

if root - data = Rey

f if (parent = = nuce)

THE NODE WITH KEY IS ROOT NODE

else if Rey < parent data

GIVEN KEVISLEFT NODE OF NODE WITH KEY

Given key is RIGHT mode of node with key

if key < root > data

return search (root-) left, key, root)

retwen search (root-right, key, root)

Quesa:- Retworms the successory of a mode se in a BST if it exits and NIII, if x has the largest key in the tree.

func find Min (Node * root)

while (root -> left)

root = root -> left

return root

```
func successor (Node * root, Node * succ, int key)
             if root = null
                succ = null
             if root > data = key
            f if root > right
             succ = findMin(root - right)
           else if key < root > data
        fuccessor (root → left, succ, key)
                    Hacurust a printer to mode.
              successor (root-right, succ, key)
Ques 3:- Insert new value into a BST
         I wite [2] = NULL or and I [2] = NULL
        Node * insert (Node * node, int key)
          if (node = = NULL)
          return new Node (key)
        if key < mode → key
mode → left = insent (node → left, key)
        else if Rey > node > key
node > right = insert (node + right, key)
```

