PAGE NO.:
DATE 6 2 22

Name : Gaurav Singh
Roll.no : 70
Batch : Cyber Security

DSA - T.A.4

Q1] write an algorithm function to delete the smallest element from a binary search Tree. Assume the tree is already constructed

→ Function to find the min value (struct node* node)

```
{ struct node * newleft = node;
while (newleft && newleft -> left /NULL)
    newleft = newleft -> Left;
    return newleft;
}
```

→ Function to delete the smallest element

Gaurav Singh Roll.No 70

```
struct node * delete smallest (struct
                node* root, int key)
{
    if (root == NULL)
        return root,

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

else
    {
    if (root -> left == NULL)

    { struct node * Temp = root -> right;
      free (root);
       return temp;
    }

else if (root -> right == NULL)
    {
      struct node* temp = root -> left;

      free (root);
       return temp;
    }
```

Gaurav Sing roll.no: 70

```
struct node* temp =
                minvalue (root->right);

    root ->key = temp -> key;
    root -> right = delete Smallest (
                root.-> right , temp -> key);

    }
    return root;
    }
```

(Q2) write a C function to find minimum and maximum element in a given Binary Search tree.

→

```
int max_Element (struct node* root)
    {
        if (root == NULL)
            return MIN;
        int root_data = root -> data ;
        int left root data = maxEle (root -> left);
        int right root data = maxEle (root -> right);
```

Gaurav Singh roll.no70

```
if (left root data > root data)
    rootdata = leftrootdata;

if (right rootdata > root data)
    rootdata = right.root data;

return rootdata;
}
```

→ int minEle (struct node * root)
{ if (root == NULL)
return max;

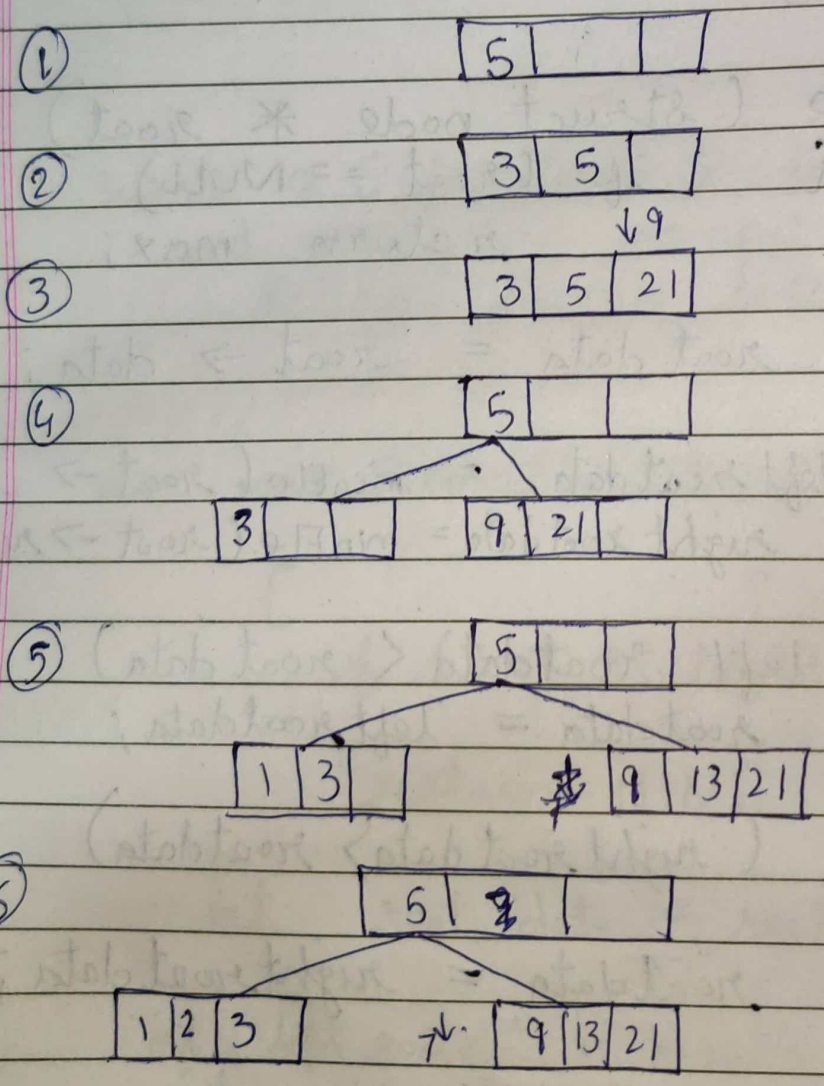
```
int root data = root -> data;

int leftrootdata = minEle (root -> left);
int rightrootdata = minEle (root -> right);

if (left root data < root data)
    rootdata = leftrootdata;

if (right root data > root data)

    rootdata = rightroot data;

return rootdata;

}
```

Gaurav Singh roll.no: 70

Q3] Construct a B' tree of order 4 given values - 5, 3, 21, 9, 1, 13, 2, 7, 10, 12, 4, 8
Delete 7.

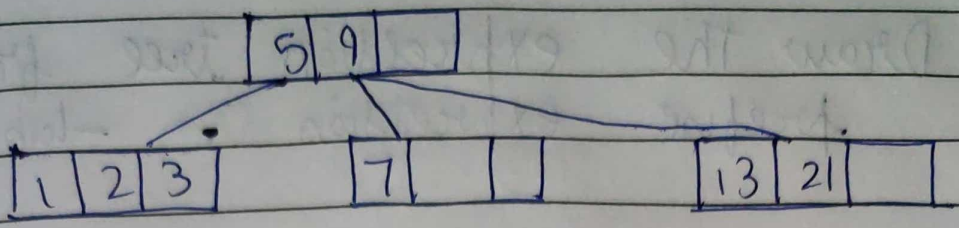
order = 4

max keys = 4 - 1 = 3

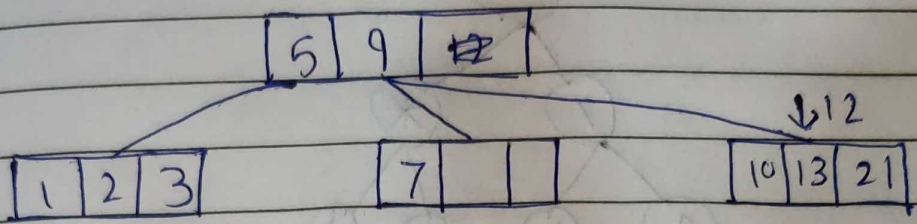


Gaurav Sing roll.no: 70

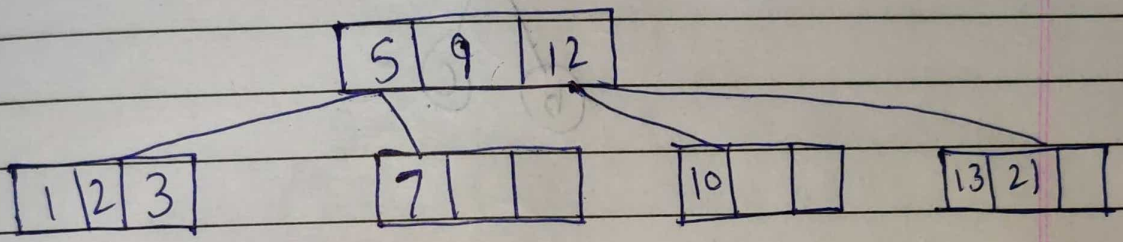
(7)



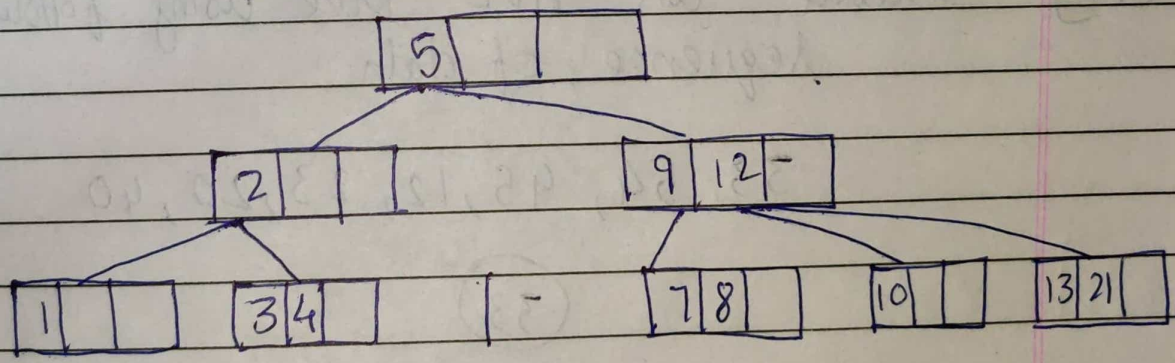
(8)



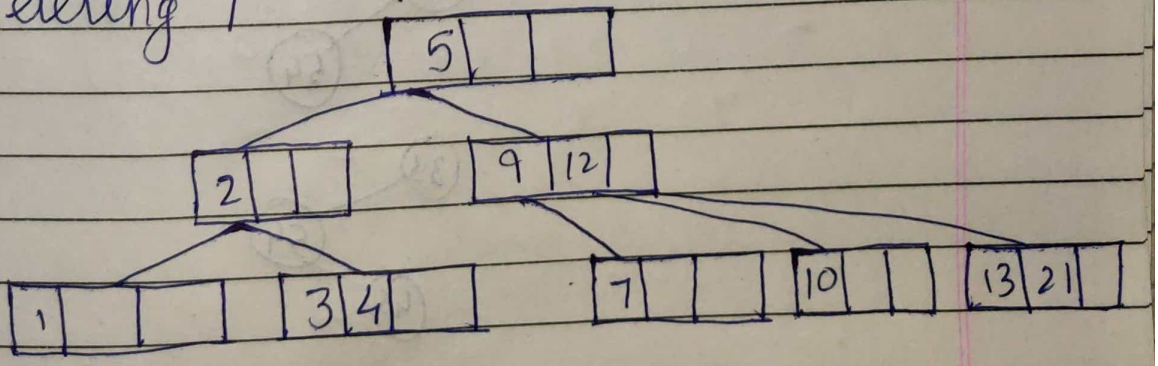
(9)



(10)

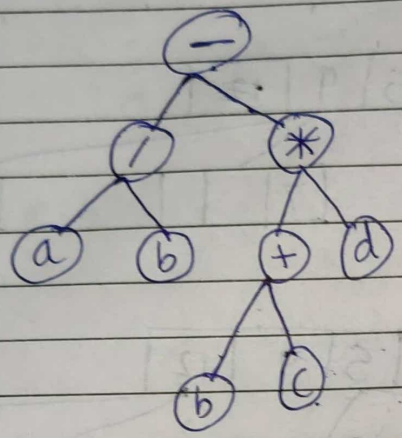


Deleting 7



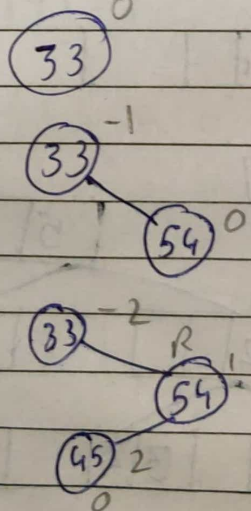
Gaurav Singh roll no: 70

Q4] Draw the expression tree from given prefix expression :- -lab * +bcd

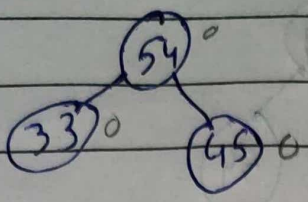


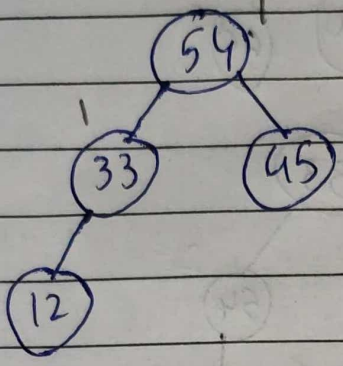
Q5] Create an AVL Tree using following sequence of data

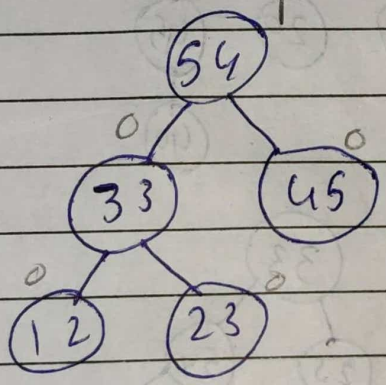
33, 54, 45, 12, 23, 20, 40

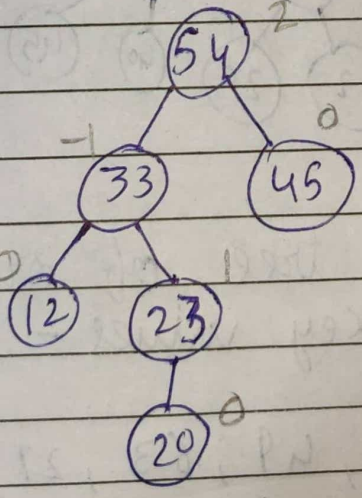


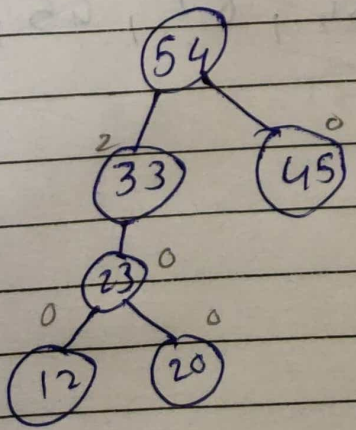
Gaurav Singh roll.no :70



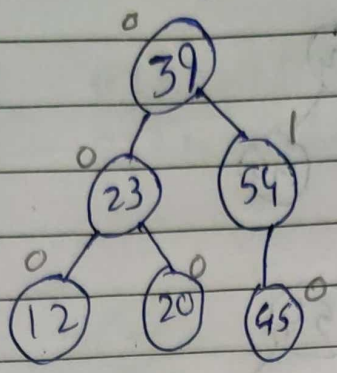


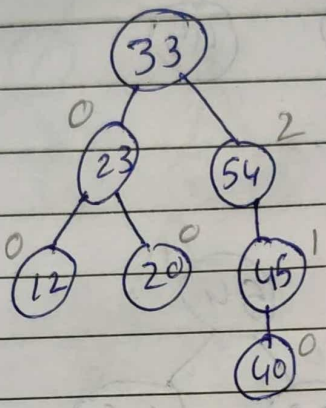


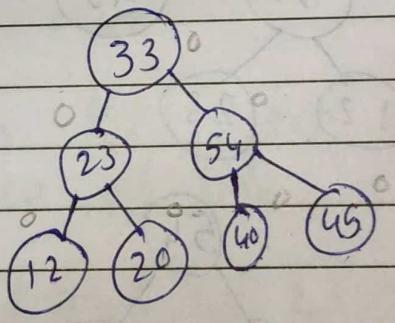




Gaurav Singh roll.no : 70



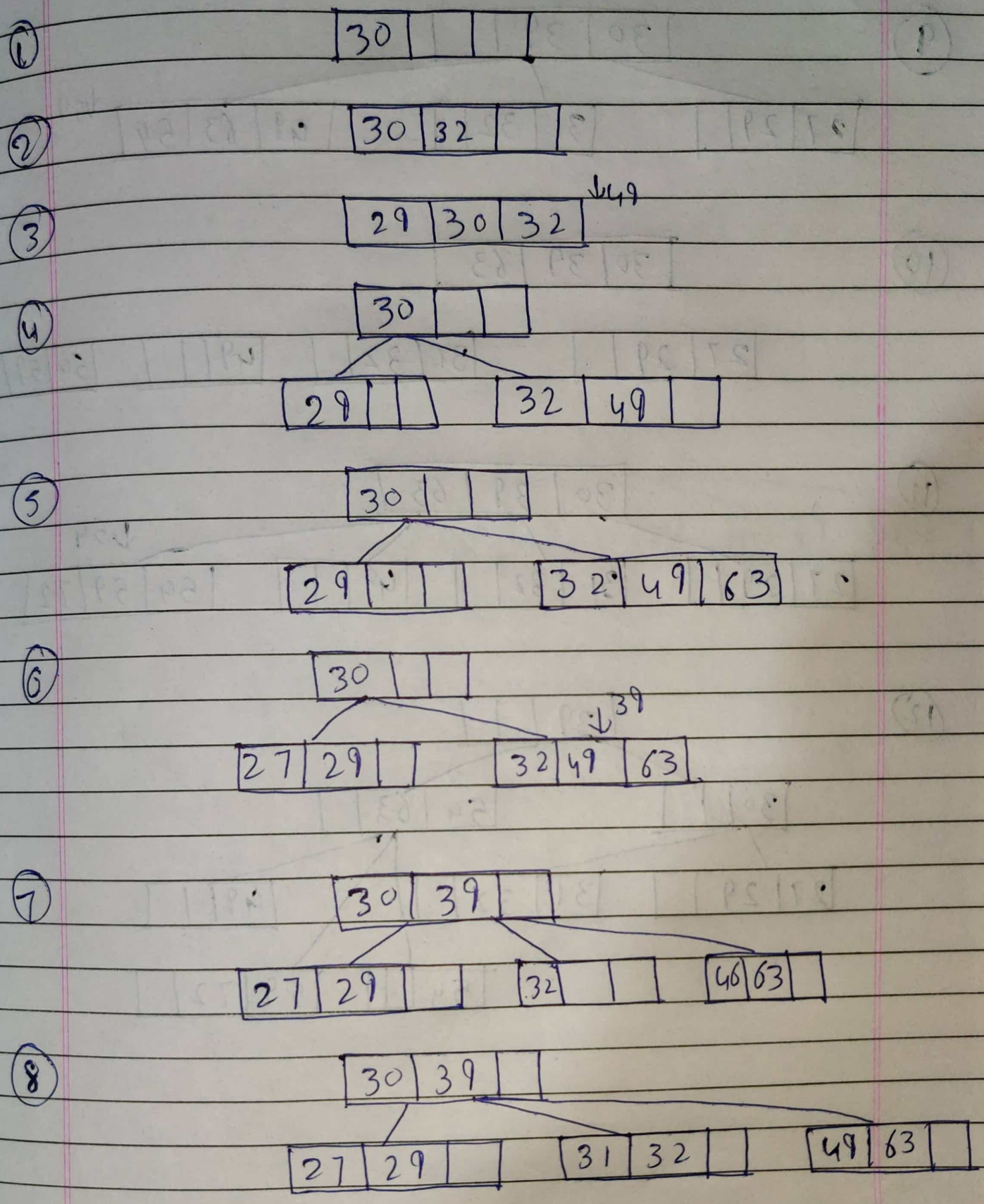




Q6] Create a B tree of order 4 by inserting following key values -

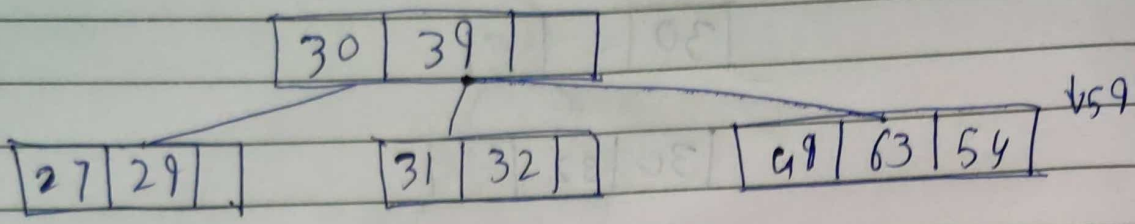
30, 32, 29, 49, 63, 27, 39, 31, 54, 59, 72, 54, 67, 45, 18.

Gaurav Singh roll. no 70

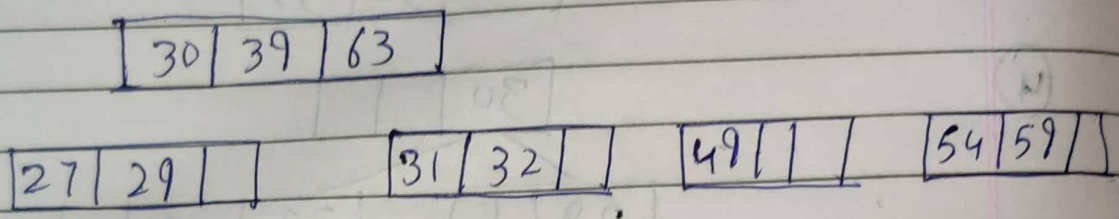


Gaurav Sing roll no: 70

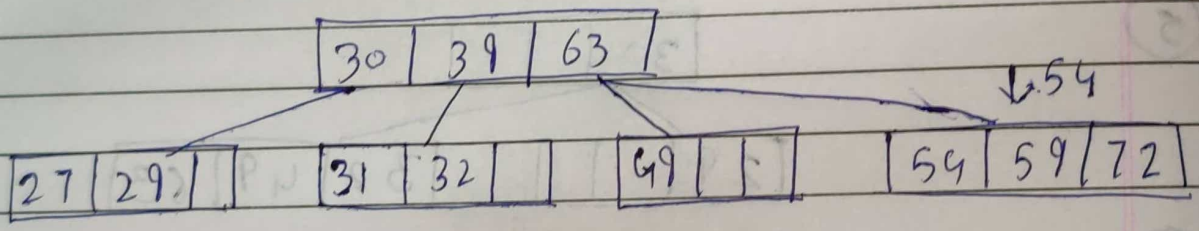
(9)



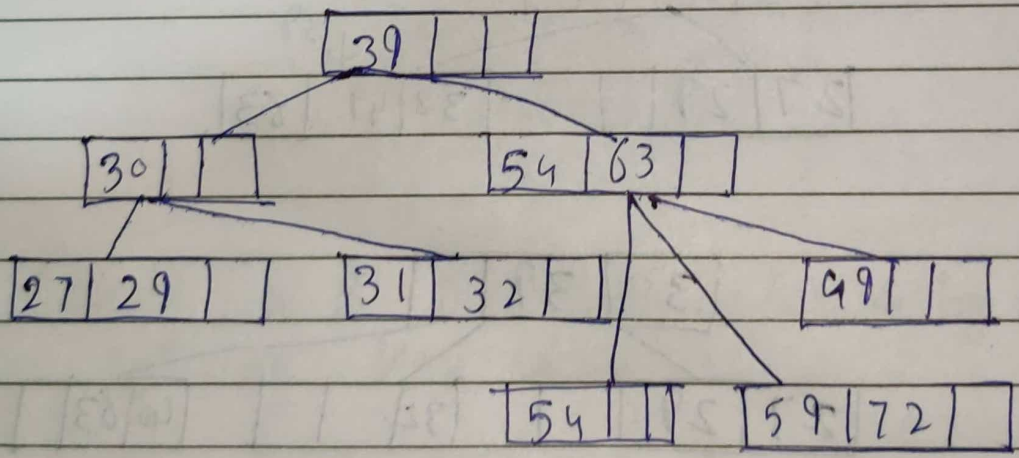
(10)



(11)

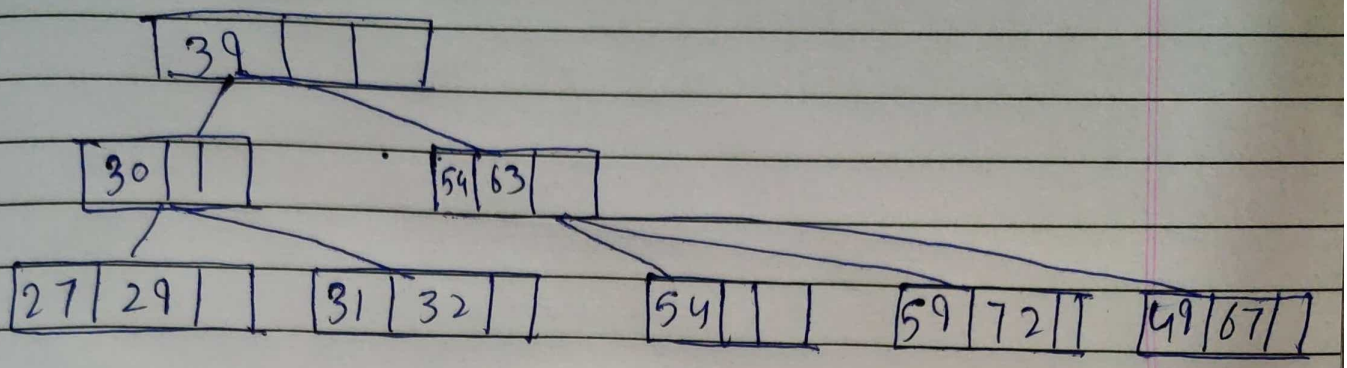


(12)



Gaurav Singh roll.no 70

(13)



(14)