ASSIGNEMENT- 2

AVL TREE

AVL_Insert(k) :-

Insertion steps –

1. Find element in tree if element is present in tree then throw exception that element is already present.
2. If element is not present in tree, then find the appropriate position of inserted element and maintain a stack which contain pointers of nodes which comes in path during the insertion.
3. After insertion balance factor of inserted node is 0 and we find the parent of inserted element.
    a. If inserted element is in left subtree, then balance factor of parent node increases by 1.
    b. If inserted element is in right subtree, then balance factor of parent node decreases by 1.
4. After modifying balance factor of parent node
    a. If balance factor of parent node is 0 means size of tree is not increases so we no need to traverse to root, we simply exit.
    b. If balance factor is +1 or -1 then height of tree increases but this node is balanced so we need to go upper level by popping new element from stack. Popped element is new parent and we again check in which the child node is left or right and according to that we modify the balance factor of node and so on.
    c. If balance factor is +2 or -2 means the node is unbalanced now we need to balance the tree.
        i. If the balance factor of parent node and child node have same sign, then it need single rotation
            1. IF BF(parent)=2 and BF(child)=1 then it is LL unbalanced
            2. IF BF(parent)=-2 and BF(child)=-1 then it is RR unbalanced
        ii. If balance factor of parent node and child node have different sign, then it needs double rotation.
            1. IF BF(parent)=-2 and BF(child)=1 then it is RL unbalanced
            2. IF BF(parent)=2 and BF(child)=-1 then it is RL unbalanced

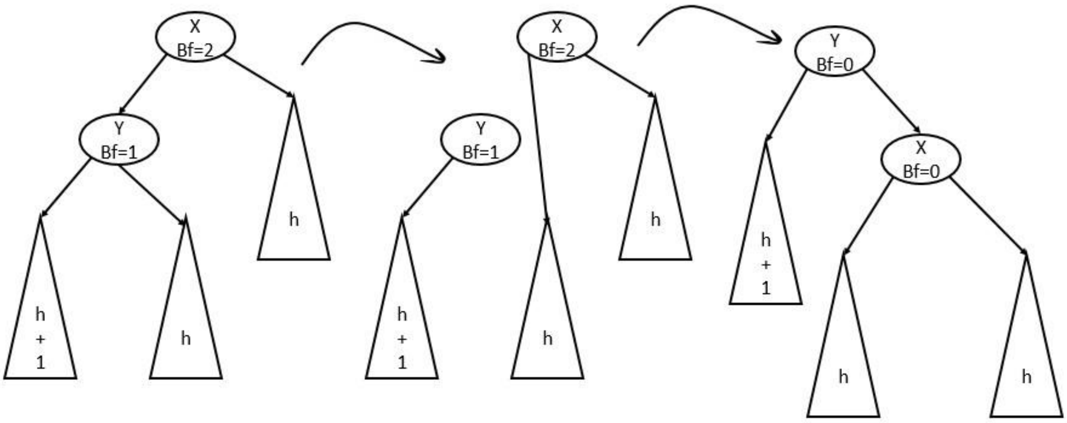
5. Once we do rotation we can safely assume that there is no error and we not need to propagate above level so we can return now.

LL_rotation(node):

If balance factor of node is +2 and balance factor of child node is 1 then tree is LL unbalance. For removing LL unbalance we follow following steps.

Steps –

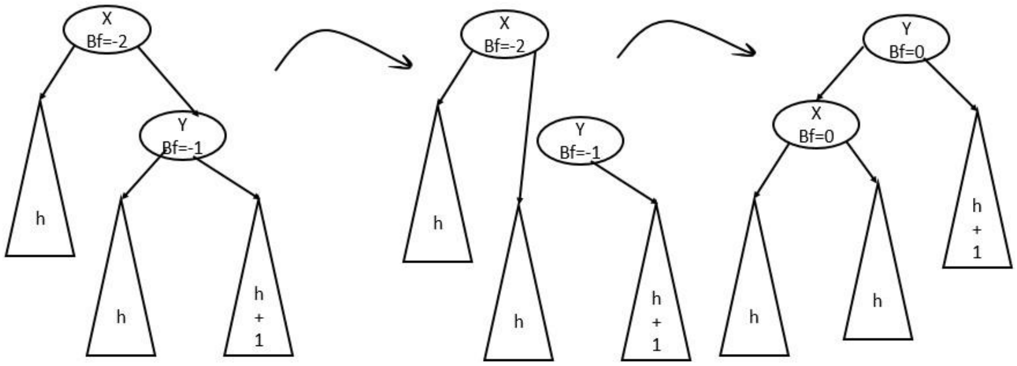
1. Find the left child of node (let lchild)
2. Node left child = lchild right child
3. Lchild right child = node.
4. Now lchild become the root pointer of new subtree
5. We modify the balance factor of node as 0 and balance factor of lchild =0 and return the lchild.



RR_Rotation:

If balance factor of node is -2 and balance factor of child node is -1 then tree is RR unbalance. For removing RR unbalance we follow following steps.

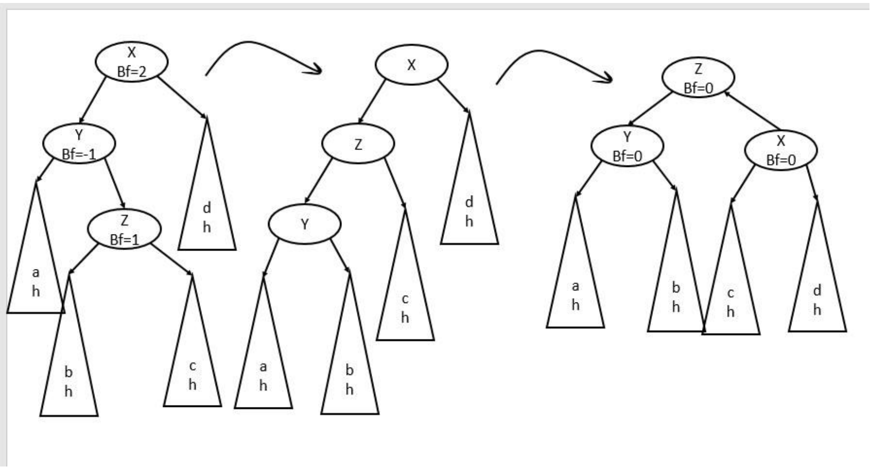
1. Find the right child as rchild.
2. Node right child = rchild left child
3. Rchild left child =node
4. Now rchild become root of subtree.
5. We modify the balance factor of node as 0 and balance factor of lchild =0 and return the rchild.



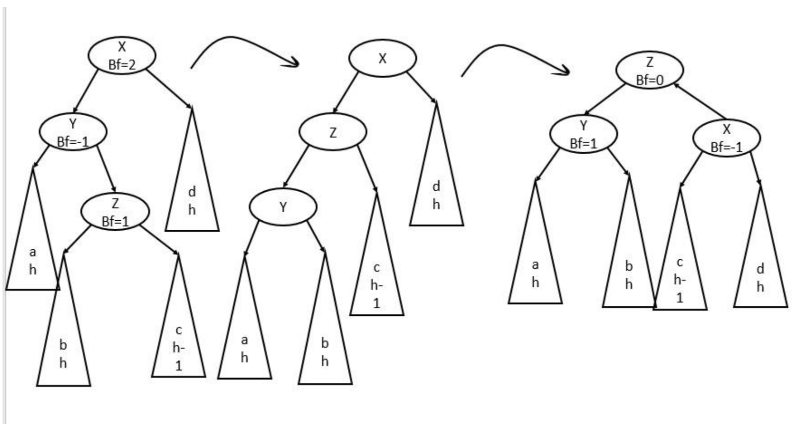
LR_Rotation

If balance factor of node is +2 and balance factor of left child is -1 then it is LR unbalanced tree. For removing LR unbalance we follow the following steps.

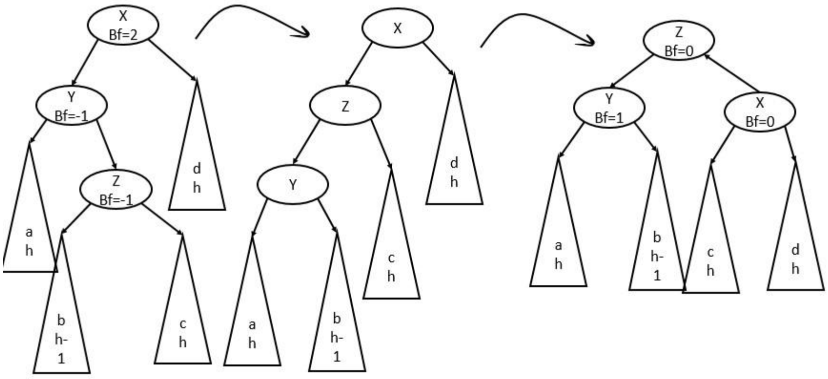
Case 1: BF(lrnode)=0



Case 2: BF(lrchild)=1



Case 3: BF(lrchild)=-1

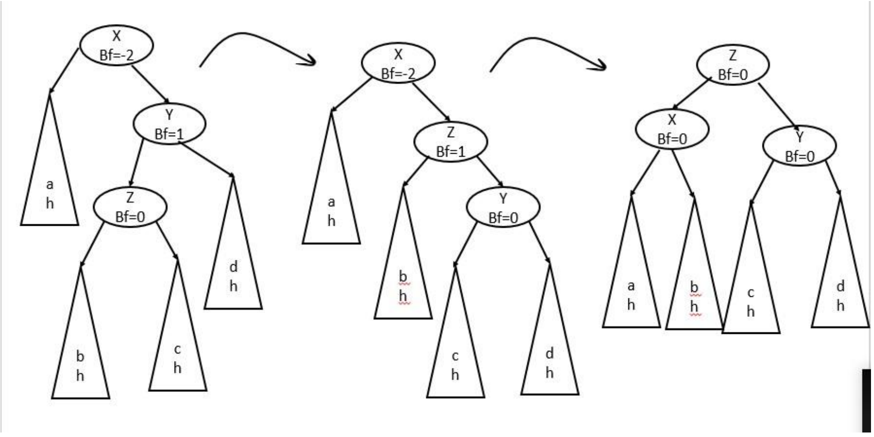


RL_Rotation:

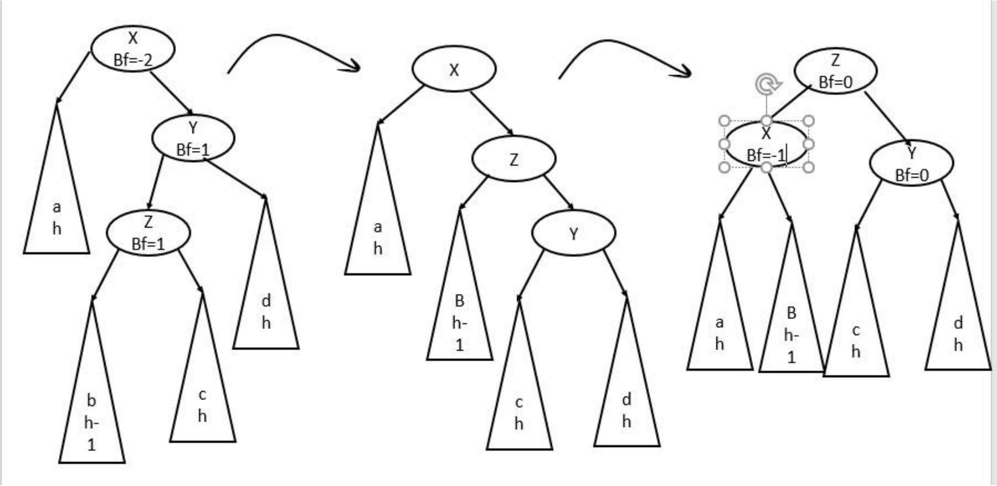
If balance factor of node is -2 and balance factor of right child is 1 then it is RL unbalanced tree. For removing RL unbalance we follow the following steps.

Let right child of node is called rchild and left child of rchild is called lrchild

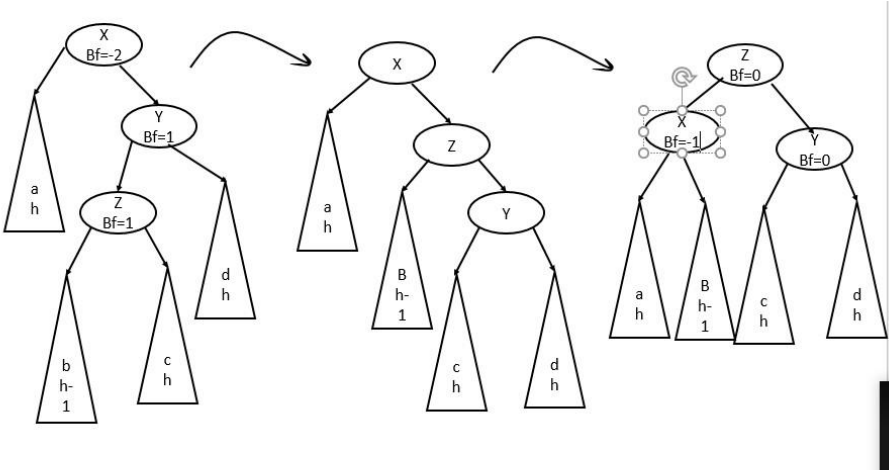
Case 1. BF(lrchild)=0



Case 2. BF(lrchild)=1



Case 3. BF(lrchild)=-1



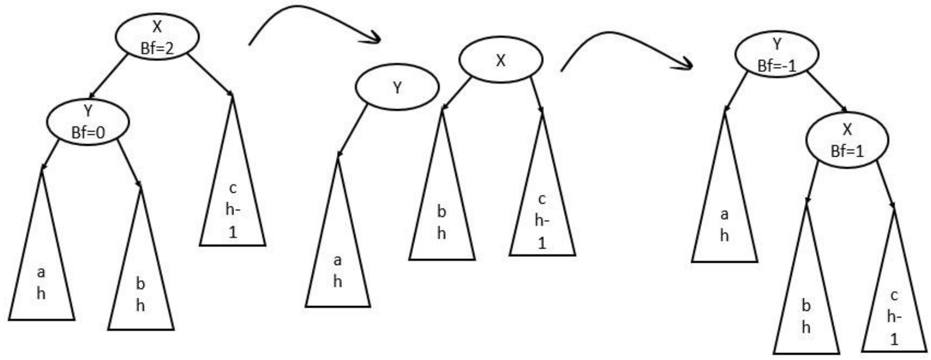
Delete(k):

Steps –

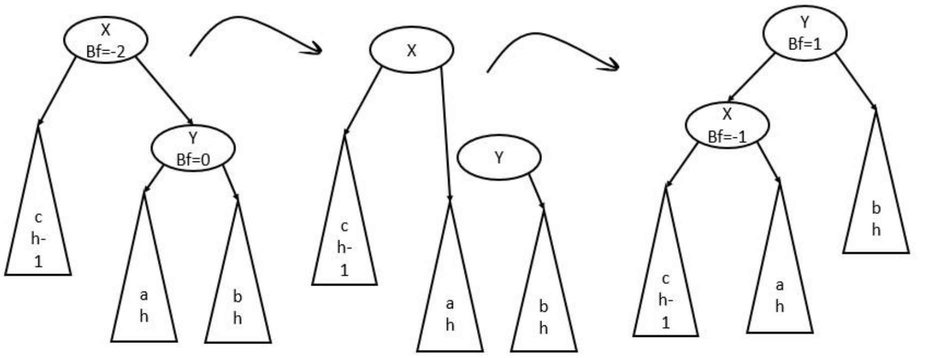
1. Search the element in tree if element is not present in tree then throw the error element is not present in tree.
2. If element is present in tree, then we find the deleted node of tree.
    a. If deleted element have two subtrees, then find the successor element of node and swap the key value of successor node to deleted node.
    b. New deleted element is successor node of tree.
3. We maintain a stack which contain the elements which is ancestor node of deleted node.
4. Now deleted node have only one child or no child

5. If deleted node have no child, then we pop a node from stack and if deleted node is left then make left as NULL or right as NULL.
6. If deleted node have only one node.
    a. If deleted node is in left subtree, then left child of parent =child of deleted node
    b. If deleted node is in right subtree, then right child of parent =child of deleted node

7. If deleted node is left subtree, then BF(parent) decreases by 1 otherwise increases by 1.
8. If balance factor is +1 or -1 means high of tree not decrease so no need to balance and we can simply return.
9. If balance factor is not +2 or -2 then height of size decreases
    a. If BF(parent)=2 and BF(parent->Lchild)=1 => LL_Rotation
    b. If BF(parent)=2 and BF(parent->Lchild)=-1 => LR_Rotation
    c. If BF(parent)=2 and BF(parent->Lchild)= 0



    d. If BF(parent)=-2 and BF(parent->Lchild)=-1 => RR_Rotation
    e. If BF(parent)=+2 and BF(parent->Lchild)=-1 =>RL Rotation
    f. If BF(parent)=-2 and BF(parent->Lchild)= 0

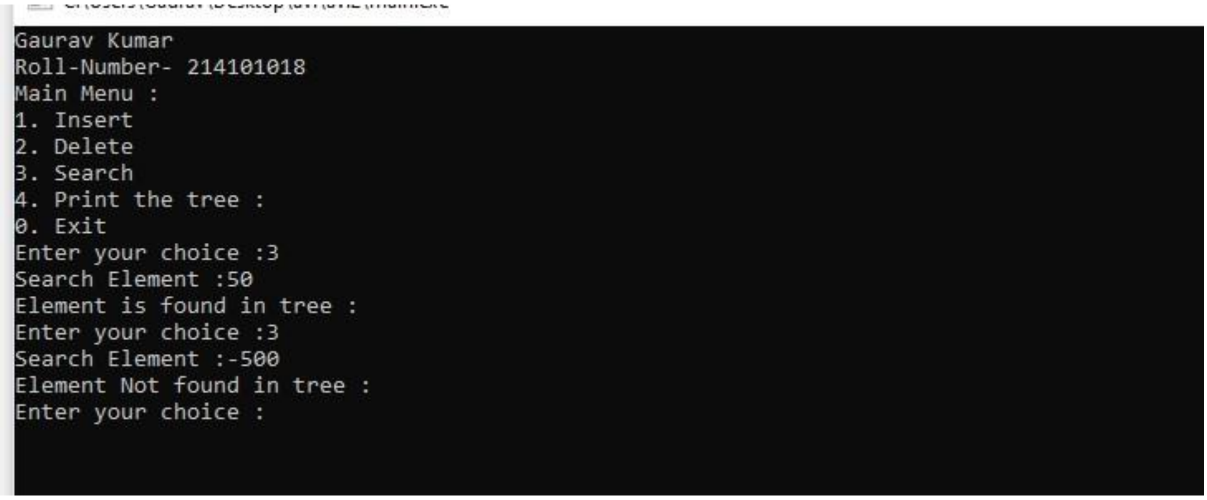


10. Once we do rotation we can safely assume that there is no error and we not need to propagate above level so we can return now.

Search(k):

For finding the element in tree we traverse the tree from top to bottom. We start from root node if search key= root key then element is present in tree. If search key is < root key, then we find the element in left subtree and if search key is greater than root key then we find element in right subtree. We search until we not reach at the leaf node.

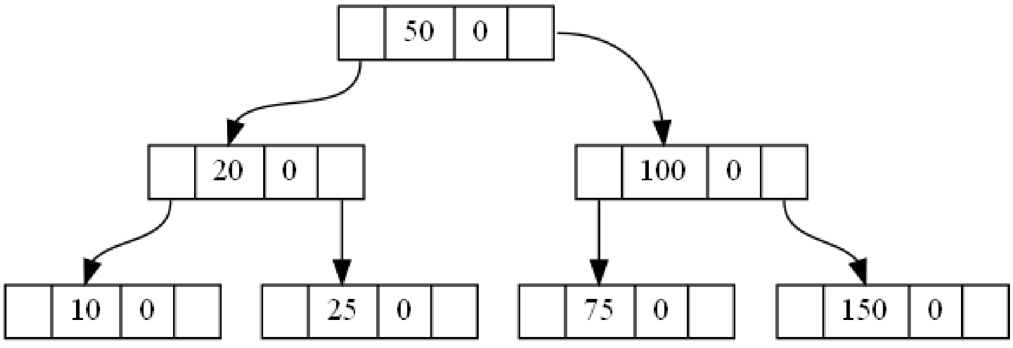
- If element is present in tree, then return true.
- If element is not present in tree, then return false.
- Search element 50 return true means Element present in tree
- Search -500 return false means element is not present in tree



```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :3
Search Element :50
Element is found in tree :
Enter your choice :3
Search Element :-500
Element Not found in tree :
Enter your choice :
```

Print():

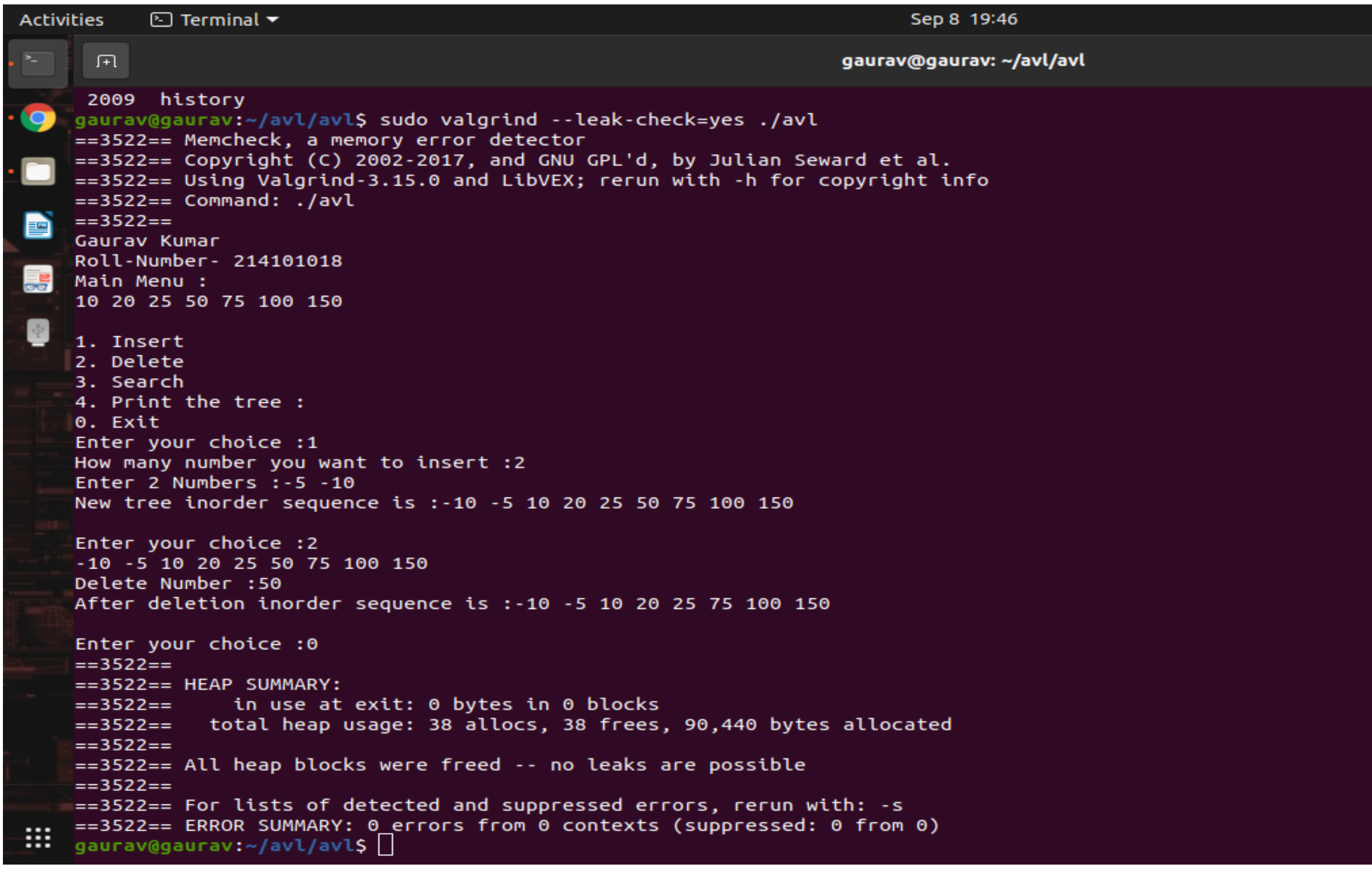
In print function we ask .dot file name for create .dot file .I traverse the tree in in-order traversal and add edge in filename.dot file which is used for create a png image.



rooted at 50

Memory leak –

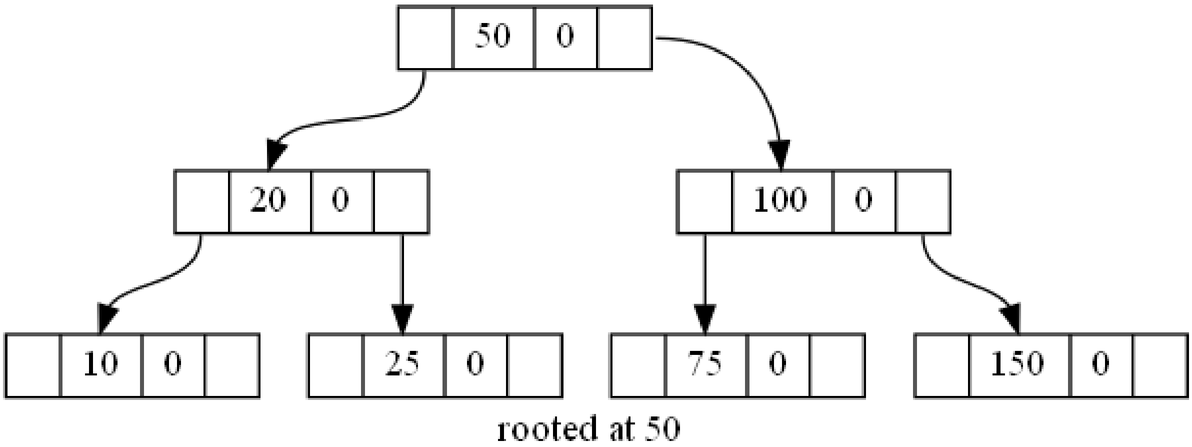
No memory leak found during



```
 2009  history
gaurav@gaurav:~/avl/avl$ sudo valgrind --leak-check=yes ./avl
==3522== Memcheck, a memory error detector
==3522== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3522== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==3522== Command: ./avl
==3522==
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
10 20 25 50 75 100 150

1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :2
Enter 2 Numbers :-5 -10
New tree inorder sequence is :-10 -5 10 20 25 50 75 100 150

Enter your choice :2
-10 -5 10 20 25 50 75 100 150
Delete Number :50
After deletion inorder sequence is :-10 -5 10 20 25 75 100 150

Enter your choice :0
==3522==
==3522== HEAP SUMMARY:
==3522==     in use at exit: 0 bytes in 0 blocks
==3522==   total heap usage: 38 allocs, 38 frees, 90,440 bytes allocated
==3522==
==3522== All heap blocks were freed -- no leaks are possible
==3522==
==3522== For lists of detected and suppressed errors, rerun with: -s
==3522== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
gaurav@gaurav:~/avl/avl$ 
```

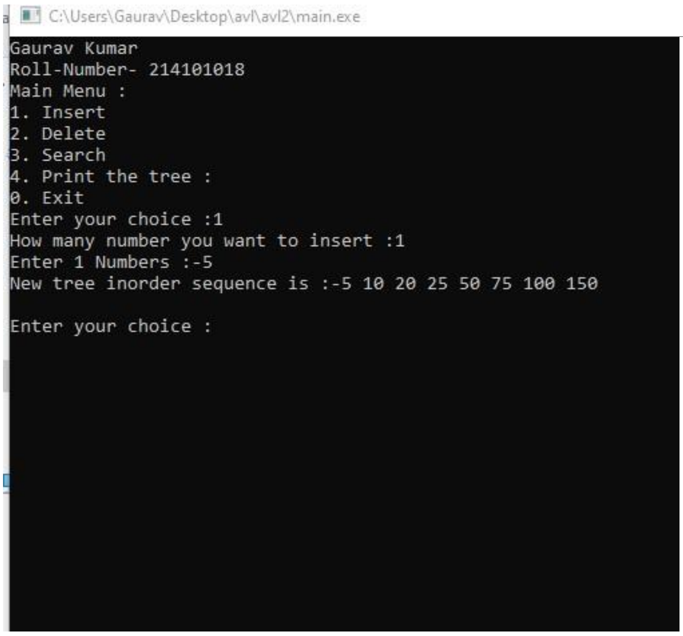
Test Cases-

Insertion- 20,50,100,10,25,150,75

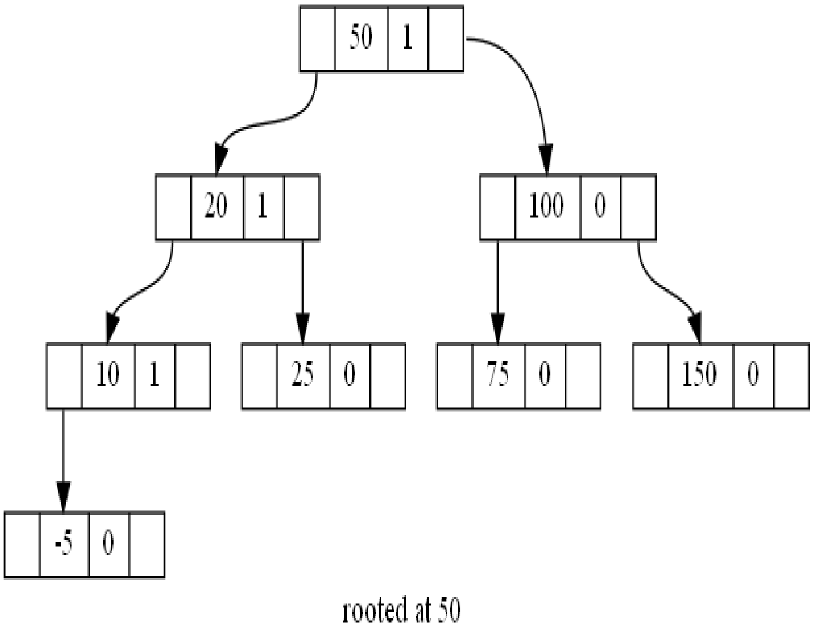


- Insert two element -10 and -5 (LL unbalance)

  Insert -5

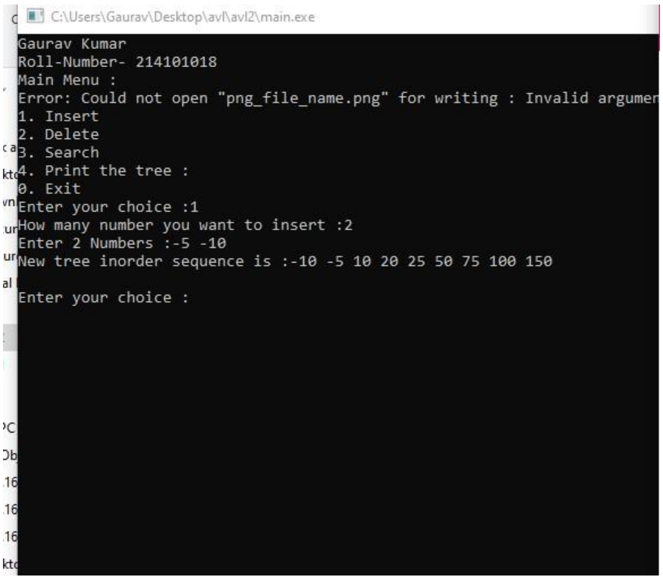


```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :-5
New tree inorder sequence is :-5 10 20 25 50 75 100 150

Enter your choice :
```

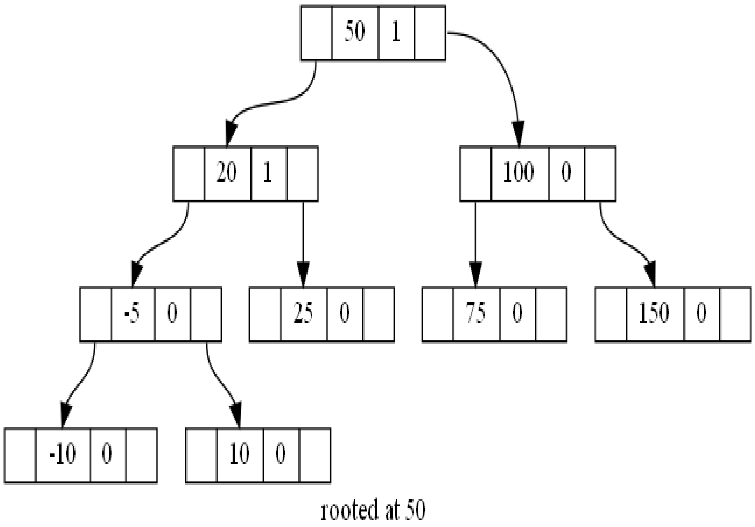


Insert -10

when we insert -10 then node 10 will unbalance



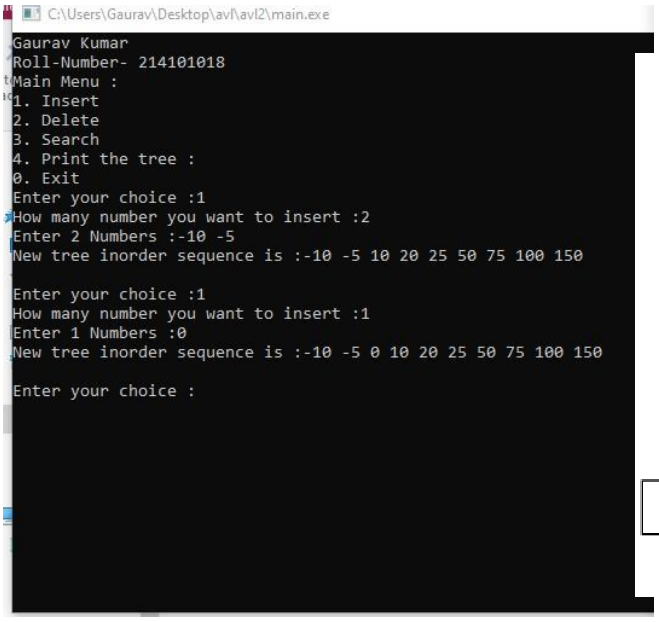
```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
Error: Could not open "png_file_name.png" for writing : Invalid argumen
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :2
Enter 2 Numbers :-5 -10
New tree inorder sequence is :-10 -5 10 20 25 50 75 100 150

Enter your choice :
```



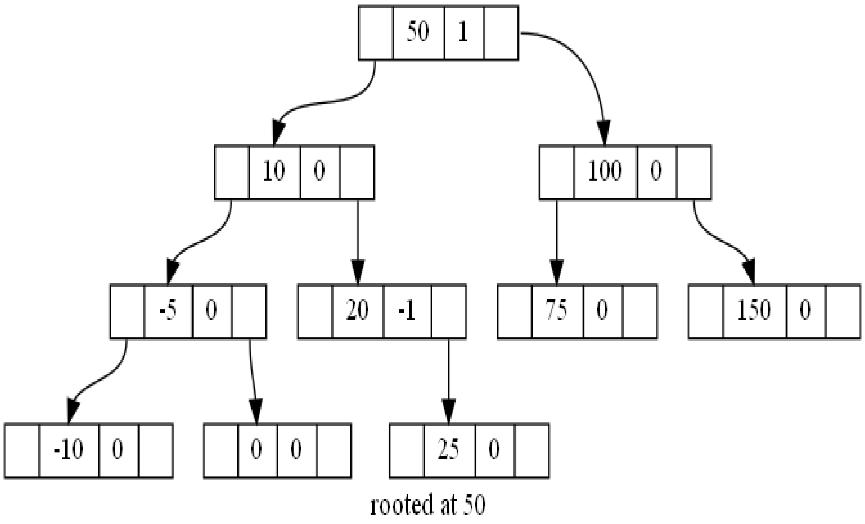
- LR unbalance

Insert 0

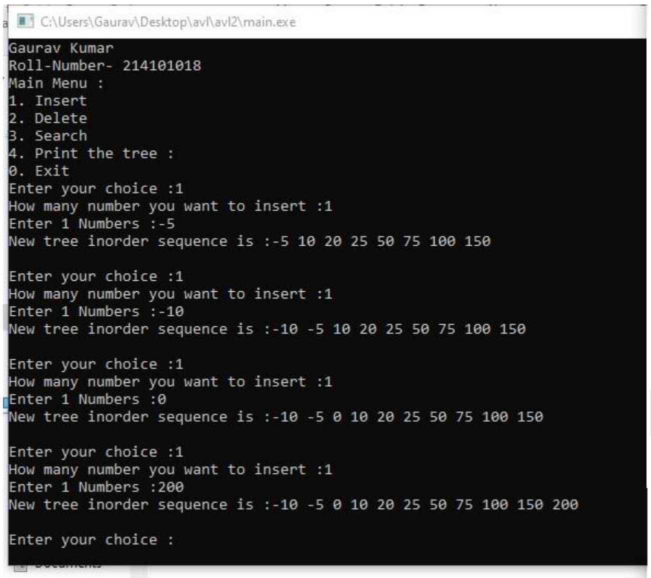
When we insert 0 in tree. 20 will unbalance (LR_rotation)



```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :2
Enter 2 Numbers :-10 -5
New tree inorder sequence is :-10 -5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :0
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150

Enter your choice :
```

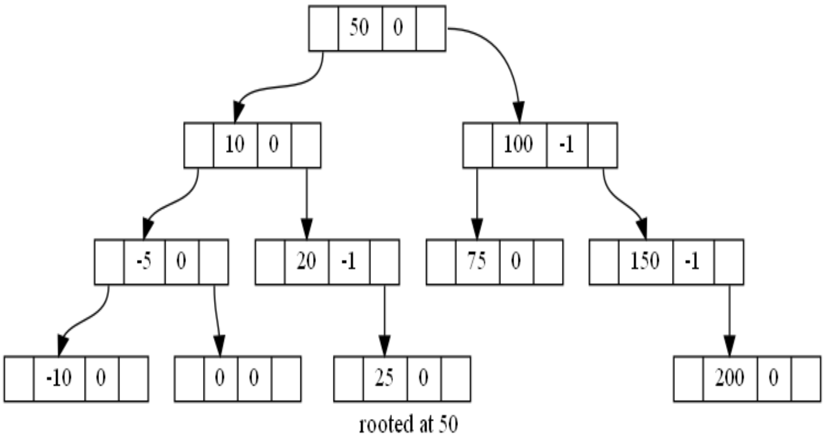


- RR unbalance
  Insert 200

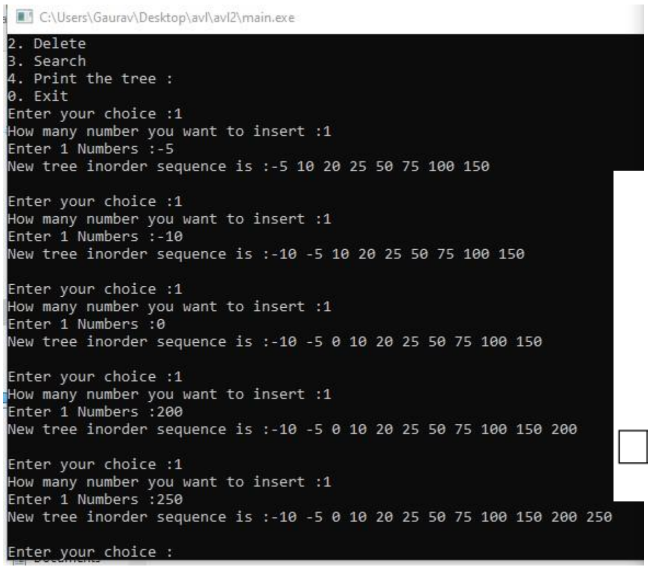


C:\Users\Gaurav\Desktop\avl\avl2\main.exe

```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :-5
New tree inorder sequence is :-5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :-10
New tree inorder sequence is :-10 -5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :0
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :200
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 200

Enter your choice :
```

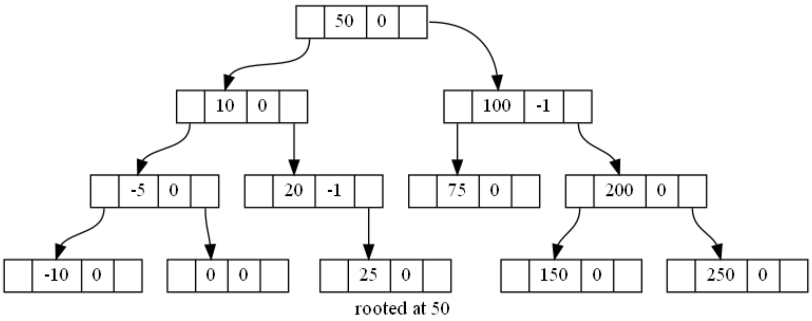


Insert 250
After inserting element node 150 will unbalance



C:\Users\Gaurav\Desktop\avl\avl2\main.exe

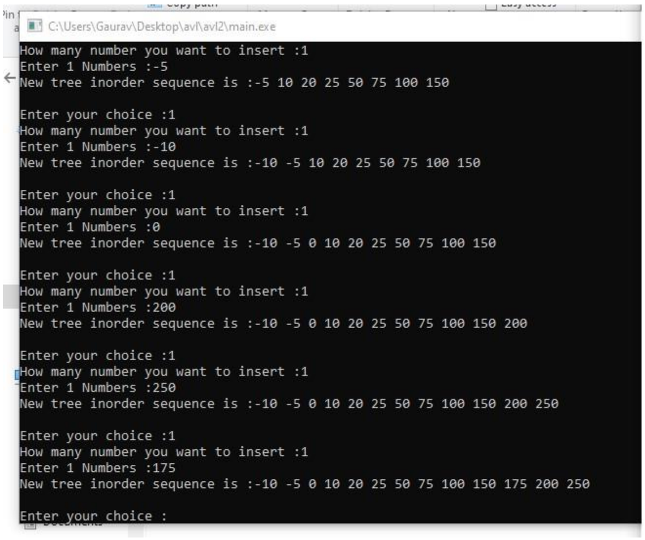
```
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :-5
New tree inorder sequence is :-5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :-10
New tree inorder sequence is :-10 -5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :0
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :200
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 200

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :250
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 200 250

Enter your choice :
```



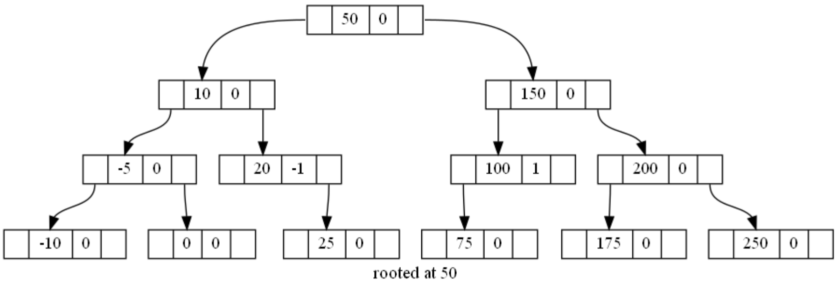
- RL unbalance

  Insert 175

  After inserting 175 node 100 will unbalance



```
How many number you want to insert :1
Enter 1 Numbers :-5
New tree inorder sequence is :-5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :-10
New tree inorder sequence is :-10 -5 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :0
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :200
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 200

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :250
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 200 250

Enter your choice :1
How many number you want to insert :1
Enter 1 Numbers :175
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :
```

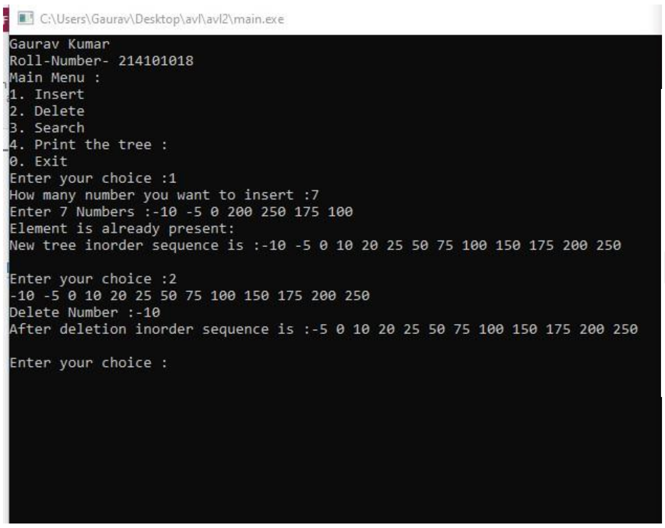


rooted at 50

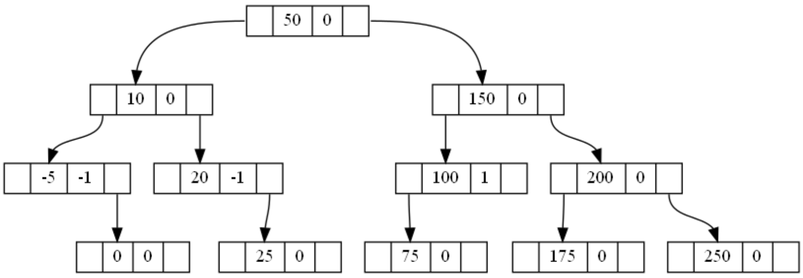
Delete :-

Deleted node have no children

Delete(-10)



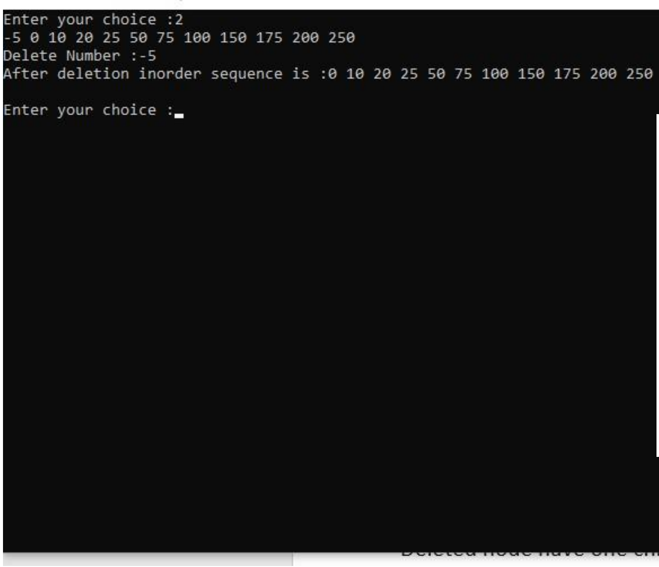
```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :7
Enter 7 Numbers :-10 -5 0 200 250 175 100
Element is already present:
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
-10 -5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-10
After deletion inorder sequence is :-5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :
```



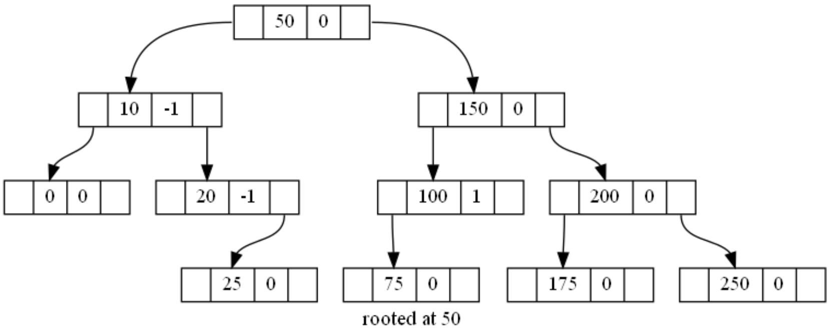
rooted at 50

Deleted node have one child

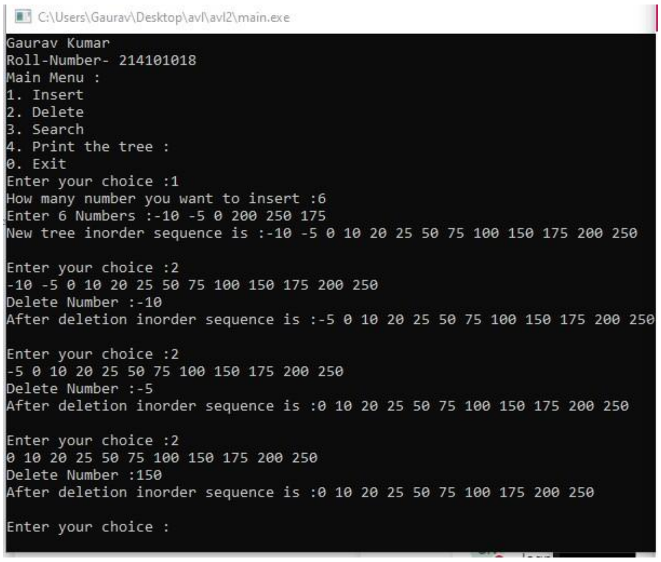
Delete(-5)



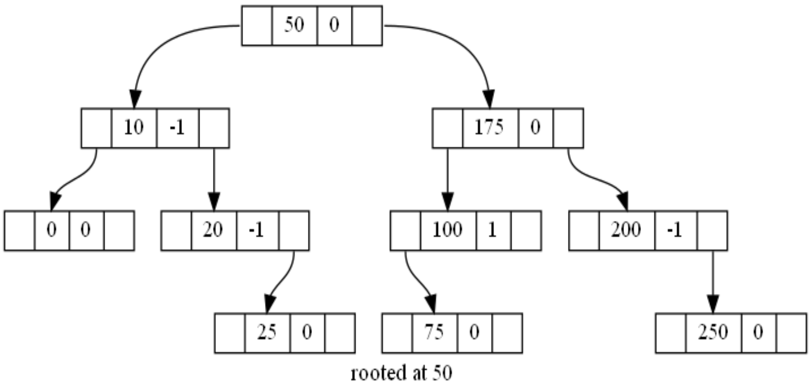
```
Enter your choice :2
-5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-5
After deletion inorder sequence is :0 10 20 25 50 75 100 150 175 200 250

Enter your choice :_
```



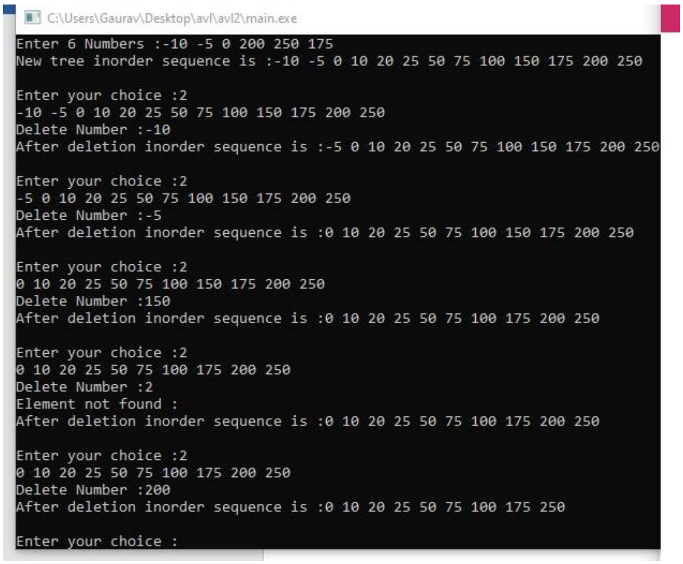
Delete (150)



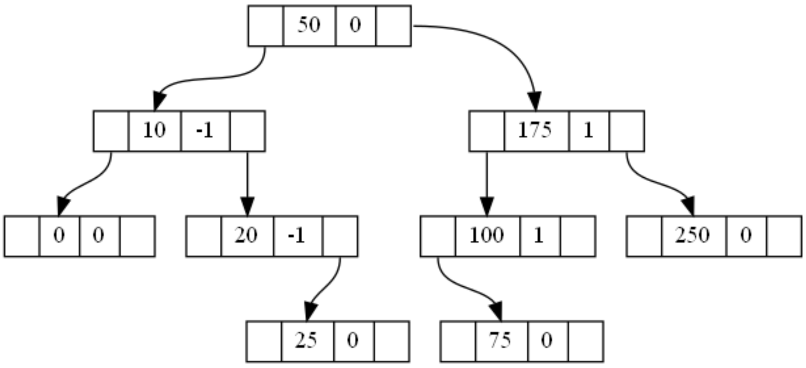
```
Gaurav Kumar
Roll-Number- 214101018
Main Menu :
1. Insert
2. Delete
3. Search
4. Print the tree :
0. Exit
Enter your choice :1
How many number you want to insert :6
Enter 6 Numbers :-10 -5 0 200 250 175
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
-10 -5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-10
After deletion inorder sequence is :-5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
-5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-5
After deletion inorder sequence is :0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 150 175 200 250
Delete Number :150
After deletion inorder sequence is :0 10 20 25 50 75 100 175 200 250

Enter your choice :
```



Delete 200

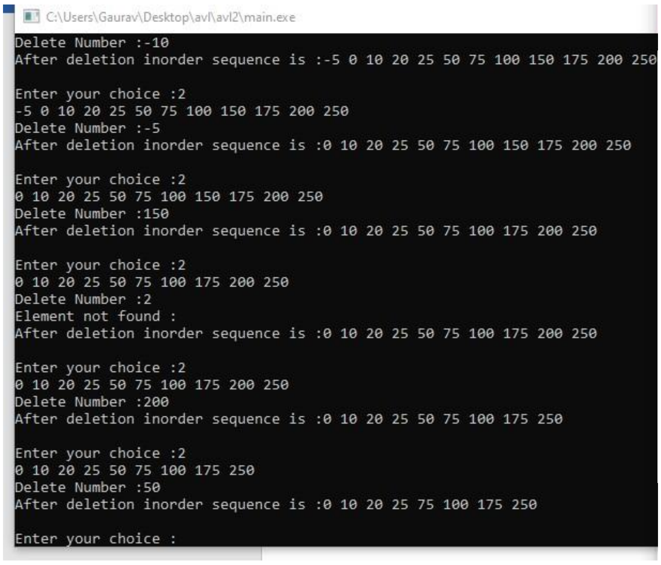


```
Enter 6 Numbers :-10 -5 0 200 250 175
New tree inorder sequence is :-10 -5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
-10 -5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-10
After deletion inorder sequence is :-5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
-5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-5
After deletion inorder sequence is :0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 150 175 200 250
Delete Number :150
After deletion inorder sequence is :0 10 20 25 50 75 100 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 175 200 250
Delete Number :2
Element not found :
After deletion inorder sequence is :0 10 20 25 50 75 100 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 175 200 250
Delete Number :200
After deletion inorder sequence is :0 10 20 25 50 75 100 175 250

Enter your choice :
```

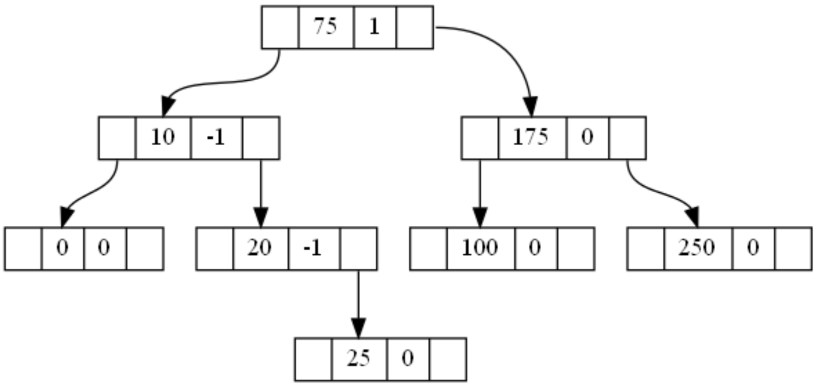


rooted at 50

Delete 50



```
Delete Number :-10
After deletion inorder sequence is :-5 0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
-5 0 10 20 25 50 75 100 150 175 200 250
Delete Number :-5
After deletion inorder sequence is :0 10 20 25 50 75 100 150 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 150 175 200 250
Delete Number :150
After deletion inorder sequence is :0 10 20 25 50 75 100 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 175 200 250
Delete Number :2
Element not found :
After deletion inorder sequence is :0 10 20 25 50 75 100 175 200 250

Enter your choice :2
0 10 20 25 50 75 100 175 200 250
Delete Number :200
After deletion inorder sequence is :0 10 20 25 50 75 100 175 250

Enter your choice :2
0 10 20 25 50 75 100 175 250
Delete Number :50
After deletion inorder sequence is :0 10 20 25 75 100 175 250

Enter your choice :
```



rooted at 75

****