CS0389

K. Nitheesh
192385090

Assignment-4

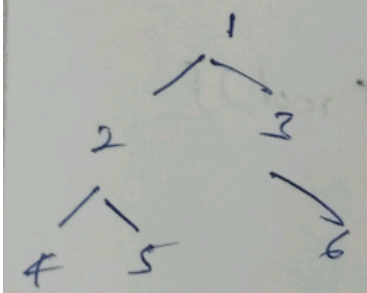
1. Develop a C program to implement the tree Traversal (inorder, preorder, postorder)

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node* left;
    struct node* right;
};
struct node* create node (int data) {
    struct node* new node = (struct node*) malloc (size of(
                                struct node));
new node -> data = data;
new node -> left = Null;
new node -> right = Null;
return new node;
}
void inorder traversal (struct node* root) {
    if (root == Null)
        return;
    inorder traversal (root -> left);
    printf ("%d", root -> data);
    inorder traversal (root -> right);
}
void preorder traversal (struct node* root) {
    if (root == Null)
        return;
    printf ("%d", root -> data);
```

```
    preorder traversal (root -> left);
    preorder traversal (root -> right);
}
int main() {
    struct node* root = create node (1);
    root -> left = create node (2);
    root -> right = create node (3);
    root -> left -> right = create node (5);
printf ("Inorder traversal");
inorder traversal (root);
printf ("\n");
printf ("preorder traversal");
preorder traversal (root);
printf ("\n");
printf ("postorder traversal");
postorder traversal (root);
printf ("\n");
    return 0;
}
```

Input :-

Creating the tree



output :-

Inorder :- 4 2 5 1 3 6

Preorder :- 1 2 4 5 3 6

postorder : 4 5 2 6 3 1

2. construct AVL tree for the following elements 3, 2, 1, 4, 5, 6, 7 followed by 10 to 16 in reverse order.

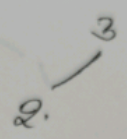
Sol. To construct an AVL tree for the given elements.

Elements :- 3, 2, 1, 4, 5, 6, 7

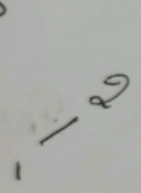
AVL TREE

Insert 3

Insert 2. Balance factor for node 3 is 1, so no rotation needed.

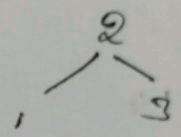


Insert 1

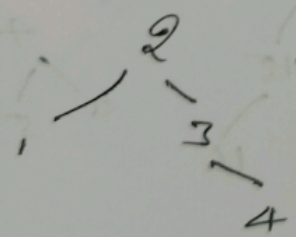


Balance factor for node 3 is 2 and node 2 is 1 so right rotation at node 3

right rotation

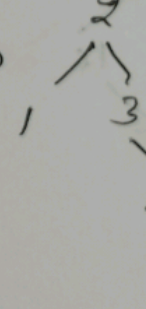


Insert 4



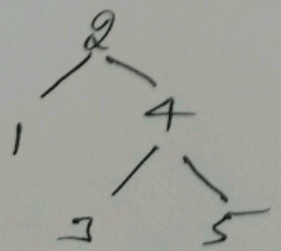
Balance factor for node 2 is 0, no rotation needed.

Insert 5



Balance factor for node 3 is -2, and node 4 is -1, left rotation at node 3.

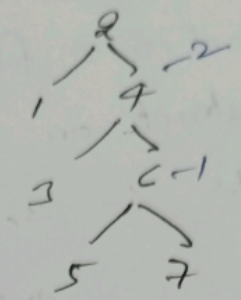
left rotation



Insert 6

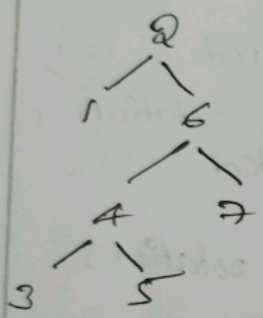
Balance factor for node 4 −1, so no rotation needed

Insert 7

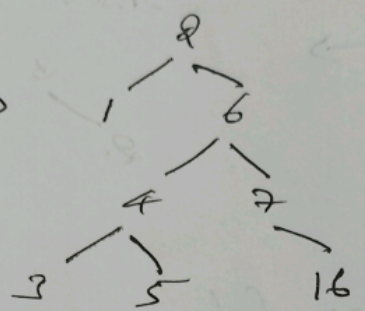


left rotation at node 4

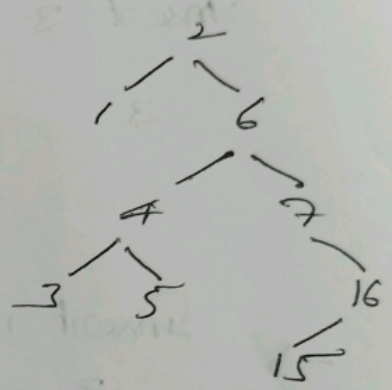
left rotation.



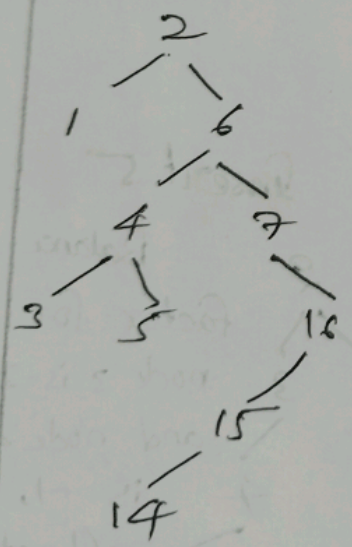
Insert 16



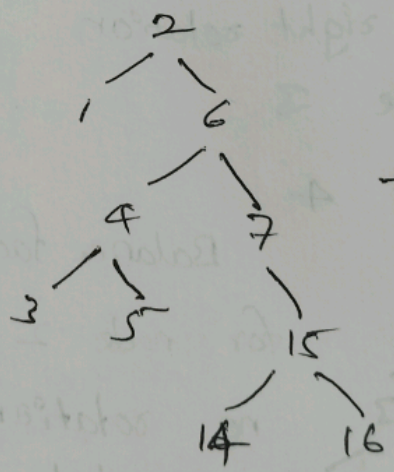
Insert 15



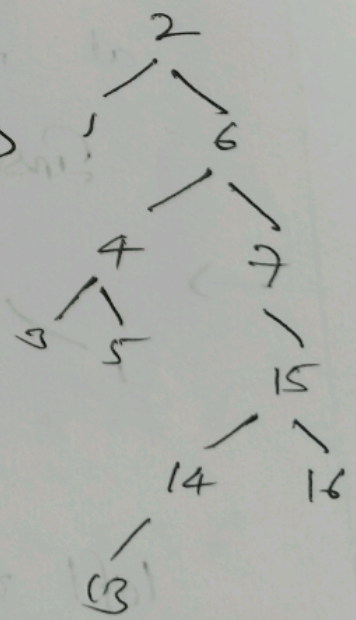
Insert 14



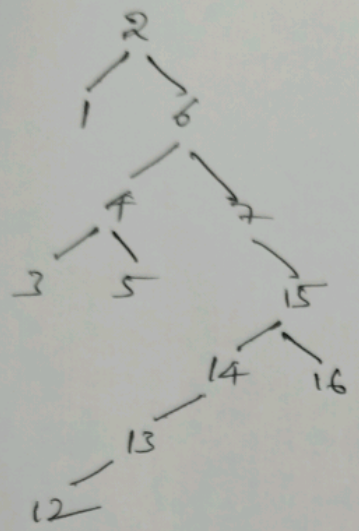
right rotation



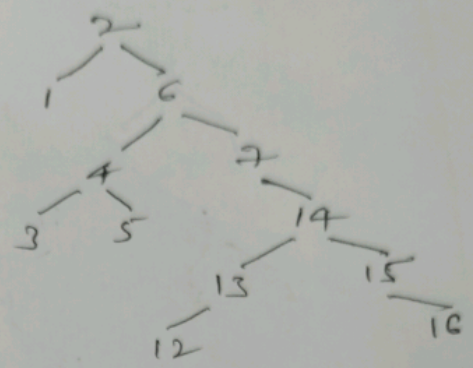
Insert 13



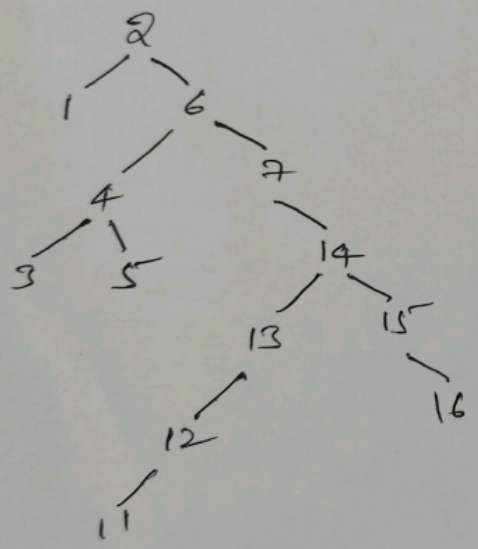
Insert 12



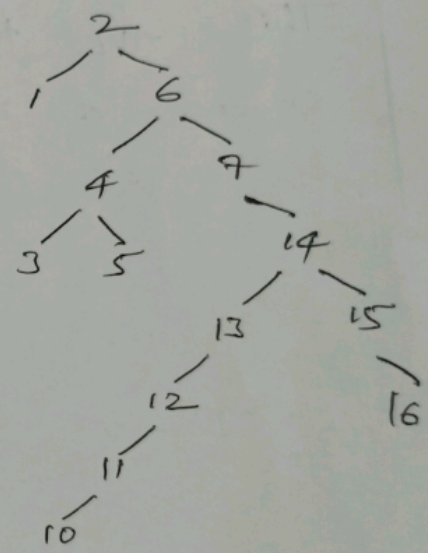
right rotation



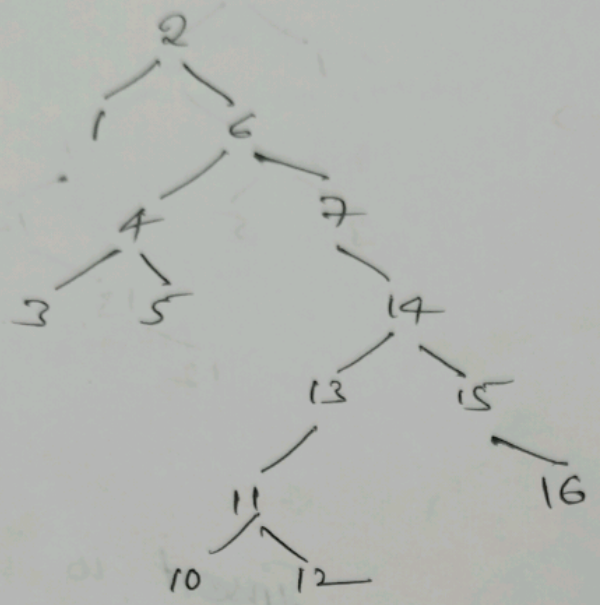
Insert 11



Insert 10



right rotation



This AVL tree is now balanced with given sequence of insertions.