Total number of nodes & leaves in a Tree using Recursion

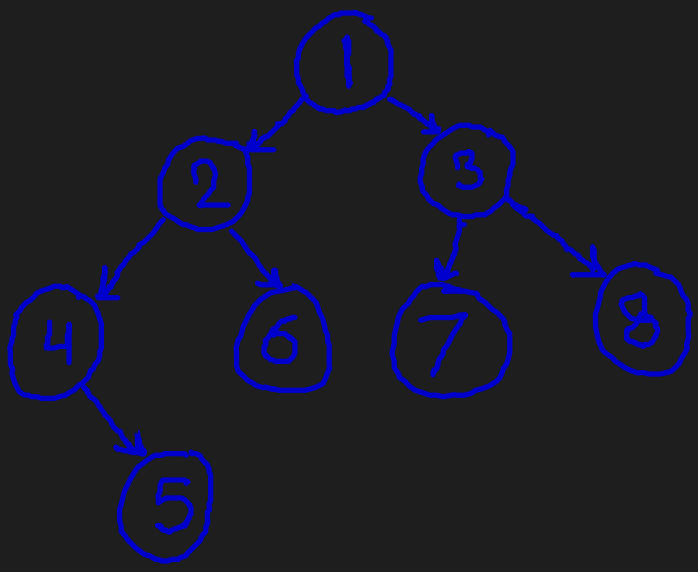
```cpp
// Aneesh Panchal
// 2K20/MC/21

#include<bits/stdc++.h>
using namespace std;

class Node{
public:
    int data;
    Node* left=NULL;
    Node* right=NULL;
};

int nodes(Node *root){
    if(root==NULL)
        return 0;
    return 1 + nodes(root->left) + nodes(root->right);
}

int leaf(Node *root){
    if(root==NULL)
        return 0;
    if(root->left==NULL && root->right==NULL)
        return 1;
    else
        return leaf(root->left) + leaf(root->right);
}

int main(){
    Node *n1 = new Node;
    Node *n2 = new Node;
    Node *n3 = new Node;
    Node *n4 = new Node;
    Node *n5 = new Node;
    Node *n6 = new Node;
    Node *n7 = new Node;
    Node *n8 = new Node;

    n1->data = 1;
    n2->data = 2;
    n3->data = 3;
    n4->data = 4;
    n5->data = 5;
    n6->data = 6;
    n7->data = 7;
    n8->data = 8;

    n1->left = n2; n1->right = n3;
    n2->left = n4; n2->right = n6;
    n4->right = n5;
    n3->left = n7; n3->right = n8;
```

```cpp
    int num_nodes = nodes(n1);
    int num_leaves = leaf(n1);
    cout<<endl;
    cout<<"Number of Nodes: "<<num_nodes<<endl<<"Number of Leaves: "<<num_leaves<<endl;
    cout<<endl;
    return 0;
}
```

File Edit Selection View Go Run Terminal Help — count_tree.cpp - Codes - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Codes> cd "e:\Codes\CS 251 DS\6. Trees\" ; if ($?) { g++ count_tree.cpp -o count_tree } ; if ($?) { .\count_tree }

Number of Nodes: 8
Number of Leaves: 4

PS E:\Codes\CS 251 DS\6. Trees>
```



Recursive Checking Equality of a Tree

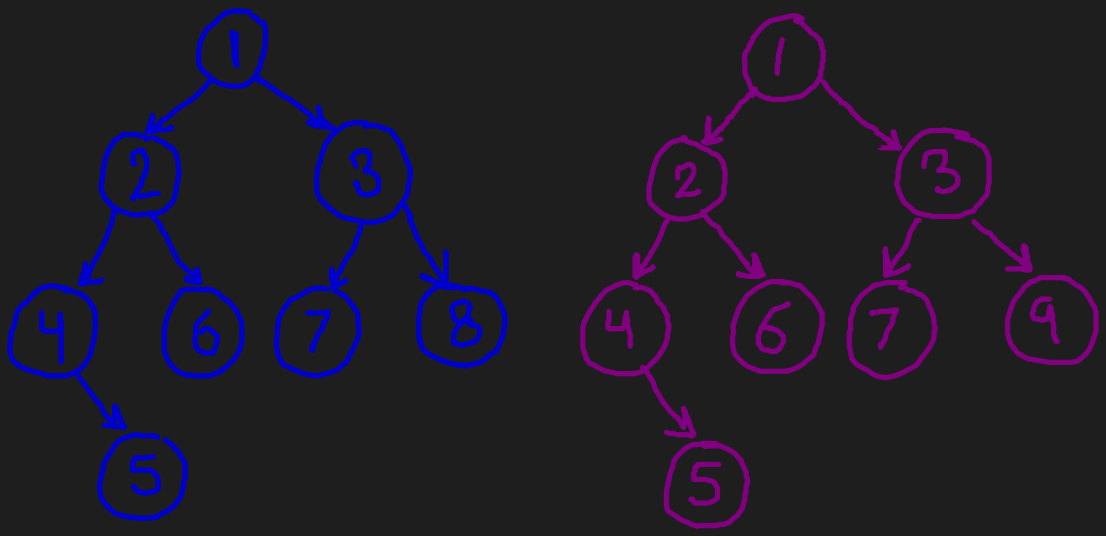
```
// Aneesh Panchal
// 2K20/MC/21

#include<bits/stdc++.h>
using namespace std;

class Node{
public:
    int data;
    Node* left=NULL;
    Node* right=NULL;
};

int check(Node* root1,Node* root2){
    if(root1==NULL && root2==NULL)
        return 1;
    return (root1 && root2) && (root1->data == root2->data) && check(root1->left,root2-
>left) && check(root1->right,root2->right);
}

int identical(Node* root1,Node* root2){
    if(root1==NULL && root2==NULL)
        return 0;
    else{
        int equal = check(root1,root2);
        return equal;
    }
}

int main(){
    Node *n1 = new Node;
    Node *n2 = new Node;
    Node *n3 = new Node;
    Node *n4 = new Node;
    Node *n5 = new Node;
    Node *n6 = new Node;
    Node *n7 = new Node;
    Node *n8 = new Node;

    n1->data = 1;
    n2->data = 2;
    n3->data = 3;
    n4->data = 4;
    n5->data = 5;
    n6->data = 6;
    n7->data = 7;
    n8->data = 8;

    n1->left = n2; n1->right = n3;
    n2->left = n4; n2->right = n6;
    n4->right = n5;
```

```
    n3->left = n7; n3->right = n8;

    Node *m1 = new Node;
    Node *m2 = new Node;
    Node *m3 = new Node;
    Node *m4 = new Node;
    Node *m5 = new Node;
    Node *m6 = new Node;
    Node *m7 = new Node;
    Node *m8 = new Node;

    m1->data = 1;
    m2->data = 2;
    m3->data = 3;
    m4->data = 4;
    m5->data = 5;
    m6->data = 6;
    m7->data = 7;
    m8->data = 8;

    m1->left = m2; m1->right = m3;
    m2->left = m4; m2->right = m6;
    m4->right = m5;
    m3->left = m7; m3->right = m8;

    int same1 = identical(n1,m1);
    cout<<endl;
    if(same1)
        cout<<"Trees are same !!!"<<endl;
    else
        cout<<"Trees are not same !!!"<<endl;

    m8->data = 9;
    cout<<endl<<"After changing data of m8 Node: "<<endl;
    int same2 = identical(n1,m1);
    if(same2)
        cout<<"Trees are same !!!"<<endl;
    else
        cout<<"Trees are not same !!!"<<endl;
    cout<<endl;
    return 0;
}
```

File Edit Selection View Go Run Terminal Help — IdenticalTrees.cpp - Codes - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Codes> cd "e:\Codes\CS 251 DS\6. Trees\" ; if ($?) { g++ IdenticalTrees.cpp -o IdenticalTrees } ; if ($?) { .\IdenticalTrees }

Trees are same !!!

After changing data of m8 Node:
Trees are not same !!!

PS E:\Codes\CS 251 DS\6. Trees>
```



Recursive BST Insertion

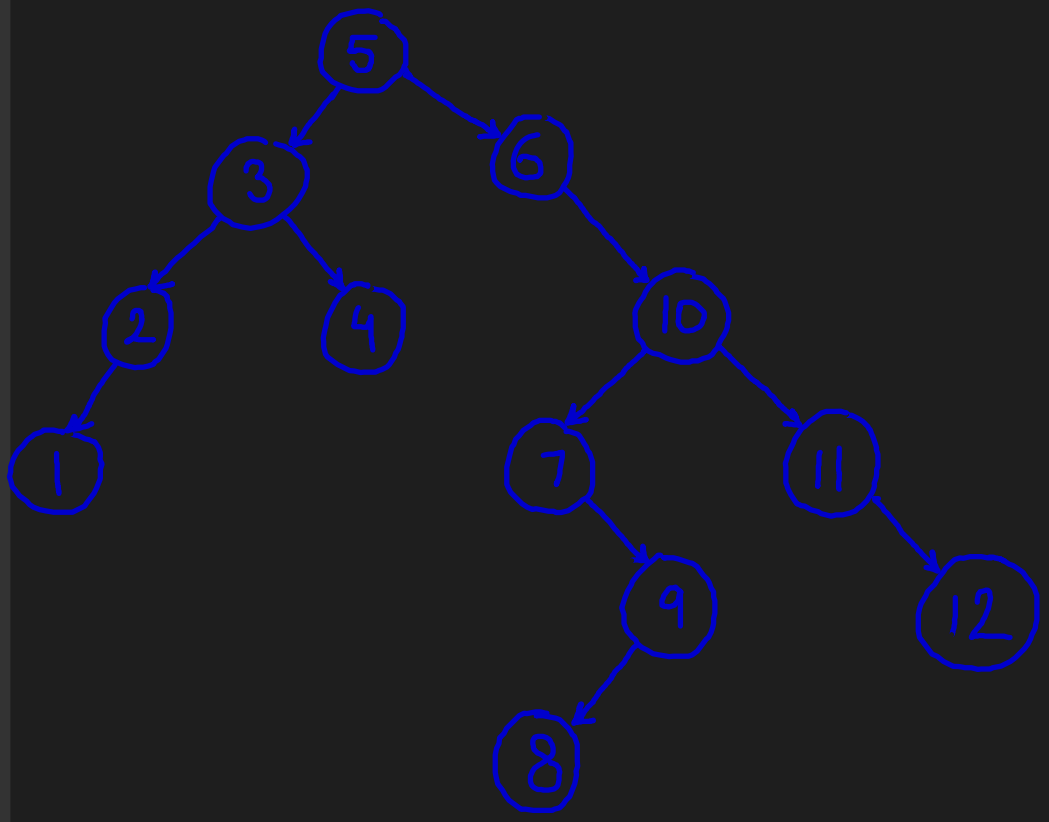
```cpp
// Aneesh Panchal
// 2K20/MC/21

#include<bits/stdc++.h>
using namespace std;

class Node{
public:
    int data;
    Node* left=NULL;
    Node* right=NULL;
};

Node* insert(Node* root,int data_){
    if(root==NULL){
        Node* newNode = new Node;
        newNode->data = data_;
        return newNode;
    }
    if(data_<root->data){
        root->left = insert(root->left,data_);
    }
    else{
        root->right = insert(root->right,data_);
    }
}

void show(Node *Root){
    if(Root==NULL){
        return;
    }
    show(Root->left);
    cout<<Root->data<<" ";
    show(Root->right);
}

int main(){
    Node* root = NULL;
    root = insert(root,5);
    root = insert(root,3);
    root = insert(root,2);
    root = insert(root,6);
    root = insert(root,4);
    root = insert(root,1);
    root = insert(root,10);
    root = insert(root,7);
    root = insert(root,9);
    root = insert(root,8);
    root = insert(root,11);
    root = insert(root,12);
    cout<<endl<<"Inorder Traversal: "<<endl;
```

```
    show(root);
    cout<<endl<<endl;
    return 0;
}
```

File Edit Selection View Go Run Terminal Help — BSTR.cpp - Codes - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Codes> cd "e:\Codes\CS 251 DS\6. Trees\" ; if ($?) { g++ BSTR.cpp -o BSTR } ; if ($?) { .\BSTR }

Inorder Traversal:
1 2 3 4 5 6 7 8 9 10 11 12

PS E:\Codes\CS 251 DS\6. Trees>
```



Non Recursive BST Insertion

```
// Aneesh Panchal
// 2K20/MC/21

#include<bits/stdc++.h>
using namespace std;

class Node {
public:
    int data=-1;
    Node* right=NULL;
    Node* left=NULL;
};

class BinarySearchTree {
    Node* root;

    public:
        BinarySearchTree(){root = NULL;}

    void insert(int data_){
        Node* temp = root;
        Node* newNode = new Node();
        newNode->data = data_;
        if(root==NULL){
            root = newNode;
            return;
        }
        while(1){
            if(data_<temp->data){
                if(temp->left==NULL){
                    temp->left = newNode;
                    return;
                }
                else
                    temp = temp->left;
            }
            else{
                if(temp->right==NULL){
                    temp->right = newNode;
                    return;
                }
                else
                    temp = temp->right;
            }
        }
    }

    void show(){
        stack<Node*> nodestack;
        Node *curr = root;
        if(root==NULL){
```

```
            cout<<"Empty Tree !!!!!"<<endl;
            return;
        }
        while(curr!=NULL || nodestack.empty()==false){
            while(curr!=NULL){
                nodestack.push(curr);
                curr = curr->left;
            }
            curr = nodestack.top();
            nodestack.pop();
            cout<<curr->data<<" ";
            curr = curr->right;
        }
        cout<<endl;
    }
};

int main(){
    BinarySearchTree BST;
    cout<<endl;
    BST.show(); //Error
    BST.insert(5);
    BST.insert(3);
    BST.insert(2);
    BST.insert(6);
    BST.insert(4);
    BST.insert(1);
    BST.insert(10);
    BST.insert(7);
    BST.insert(9);
    BST.insert(8);
    BST.insert(11);
    BST.insert(12);
    cout<<endl<<"After Insertion: "<<endl<<"Inorder Traversal: "<<endl;
    BST.show();
    cout<<endl<<endl;
    return 0;
}
```

File Edit Selection View Go Run Terminal Help BSTNR_insert.cpp - Codes - Visual Studio Code

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Codes> cd "e:\Codes\CS 251 DS\6. Trees\" ; if ($?) { g++ BSTNR_insert.cpp -o BSTNR_insert } ; if ($?) { .\BSTNR_insert }

Empty Tree !!!!!

After Insertion:
Inorder Traversal:
1 2 3 4 5 6 7 8 9 10 11 12


PS E:\Codes\CS 251 DS\6. Trees>
```