Experiment No-06: Important Problems on Binary Tree.

Objectives

- Find the height of a binary tree.
- Check whether a tree is balanced or not.
- Check whether a tree is BST or not.

~~**Example 1**~~**:** Find the height of a binary tree.

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


// Function to find the tree height
int maxDepth(Node* root)
{
    if (root == NULL) return 0;

    int lh = 1+ maxDepth(root->left); // calculate height of left
        subtree
    int rh = 1+ maxDepth(root->right); // calculate height of right
        subtree

    return max(lh,rh); // return max between two numbers
}

int main()
{
   Node* root = new Node(1);
   root -> left = new Node(2);
   root -> right = new Node(3);
   root -> left -> left = new Node(4);
   root -> left -> right = new Node(5);
   root -> right -> left = new Node(6);
   root -> right -> right = new Node(7);
   root -> left -> left -> left = new Node(9);

   int h = maxDepth(root);

   cout<<"Height: "<<h<<endl;

}
```

~~Example 2~~: Check whether a tree is balanced or not.

```
// Height calculation function
int maxDepth(Node* root)
{
    if (root == NULL) return 0;

    int lh = 1+ maxDepth(root->left);

    if (lh == -1) return -1;

    int rh = 1+ maxDepth(root->right);

    if (rh == -1) return -1;

    if (abs(lh-rh)>1) // Check for imbalanced condition
        return -1;

    return max(lh,rh);
}


bool isbalanced(Node *root)
{
    // return 1 if true otherwise return 0
    return maxDepth(root)!=-1;
}

int main()
{

   Node* root = new Node(1);
   root -> left = new Node(2);
   root -> left -> left = new Node(4);

   int h = isbalanced(root);

   if(h==0)
   {
    cout<<"Tree in not balanced"<<endl;
   }
    else{
    cout<<"Tree is balanced"<<endl;
    }
}
```

Practice Exercise

1. Write a C++ program to find the height of the following tree (Figure 1).
2. Write a C++ program to check whether the following tree (Figure 1) is balanced.
3. Write a C++ program to check whether a given tree is BST.
4. Write a C++ program to determine whether a given tree is perfect. [**Hint:** height of left subtree and right subtree is equal]
5. Write a C++ program to find the sum of the left child of a given tree. [**Hint:** use level order traversal]

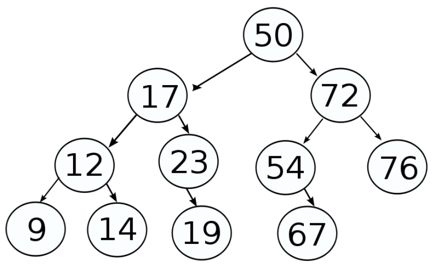

Figure 1

Resources (Link)

Try to solve similar problems at an online Judge.

1. Height of a Binary Tree
2. Balanced Tree