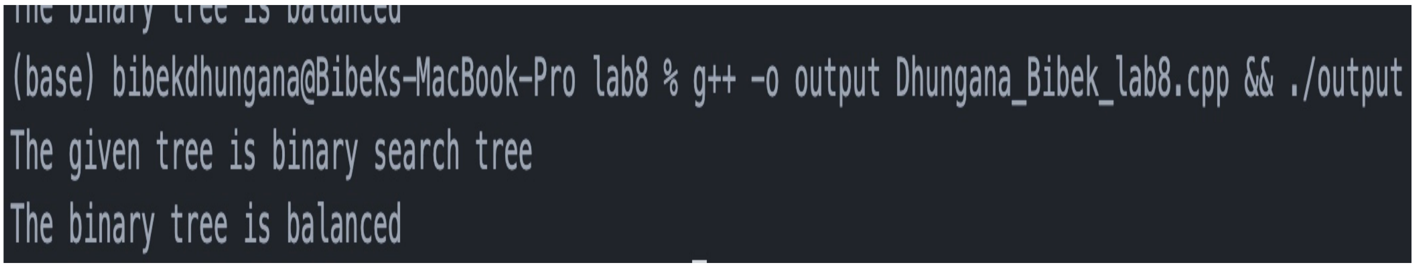
Output Lab 8
Bibek Dhungana

Case 1: with input given in the lab.

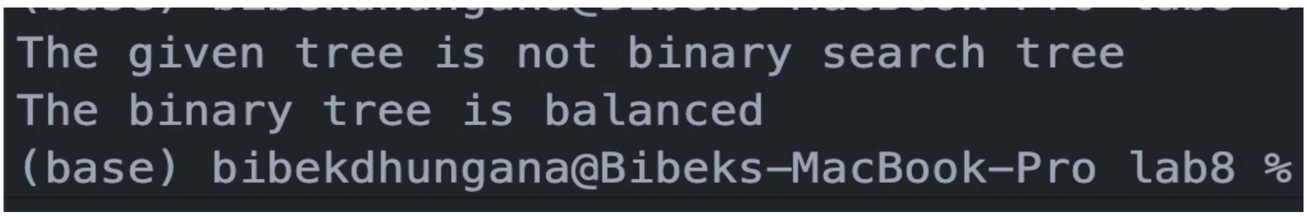
BinaryTreeNode* root = new BinaryTreeNode(4);
root->left = new BinaryTreeNode(2);
root->right = new BinaryTreeNode(5);
root->left->left = new BinaryTreeNode(1);
root->left->right = new BinaryTreeNode(3)
The tree is tree is balanced and is also binary search tree.



```
The binary tree is balanced
(base) bibekdhungana@Bibeks-MacBook-Pro lab8 % g++ -o output Dhungana_Bibek_lab8.cpp && ./output
The given tree is binary search tree
The binary tree is balanced
```

Case 2: Balanced but not Binary search tree

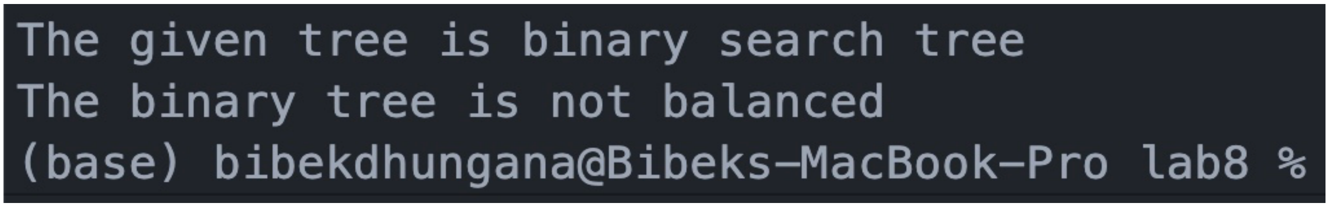
BinaryTreeNode* root = new BinaryTreeNode(20);
root->left = new BinaryTreeNode(2);
root->right = new BinaryTreeNode(5);
root->left->left = new BinaryTreeNode(1);
root->left->right = new BinaryTreeNode(3)



```
(base) bibekdhungana@Bibeks-MacBook-Pro lab8 %
The given tree is not binary search tree
The binary tree is balanced
(base) bibekdhungana@Bibeks-MacBook-Pro lab8 %
```

Case 3: Binary search tree but not balanced

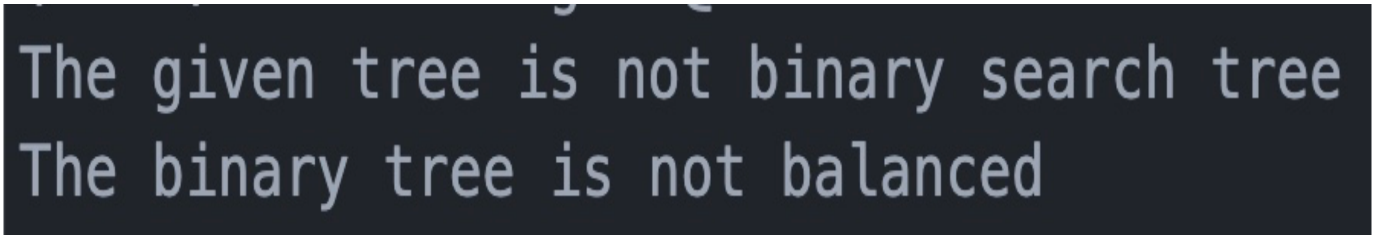
```
BinaryTreeNode* root = new BinaryTreeNode(4);
root->left = new BinaryTreeNode(2);
root->right = new BinaryTreeNode(5);
root->left->left = new BinaryTreeNode(1);
root->left->right = new BinaryTreeNode(3)
root->left->left->left = new BinaryTreeNode(0);
```



```
The given tree is binary search tree
The binary tree is not balanced
(base) bibekdhungana@Bibeks-MacBook-Pro lab8 %
```

Case 4: Not Binary search tree and not balanced.

```
BinaryTreeNode* root = new BinaryTreeNode(20);
root->left = new BinaryTreeNode(2);
root->right = new BinaryTreeNode(5);
root->left->left = new BinaryTreeNode(1);
root->left->right = new BinaryTreeNode(3)
root->left->left->left = new BinaryTreeNode(0);
```



```
The given tree is not binary search tree
The binary tree is not balanced
```