

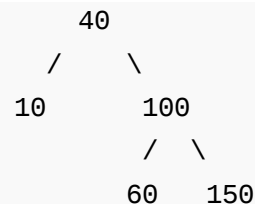
## Check if a given Binary Tree is height balanced like a Red-Black Tree

In a **Red-Black Tree**, the maximum height of a node is at most twice the minimum height (The four Red-Black tree properties make sure this is always followed). Given a Binary Search Tree, we need to check for following property.

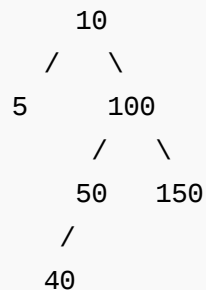
*For every node, length of the longest leaf to node path has not more than twice the nodes on shortest path from node to leaf.*



Cannot be a Red-Black Tree  
with any color assignment  
Max height of 12 is 1  
Min height of 12 is 3



It can be Red-Black Tree



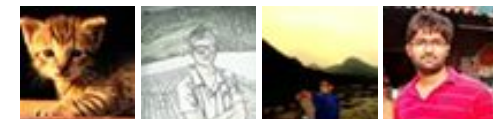
Google™ Custom Search



GeeksforGeeks



52,731 people like GeeksforGeeks.



Interview Experiences

Advanced Data Structures

Dynamic Programming

Greedy Algorithms

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

Geometric Algorithms

It can also be Red-Black Tree

Expected time complexity is  $O(n)$ . The tree should be traversed at-most once in the solution.

***We strongly recommend to minimize the browser and try this yourself first.***

For every node, we need to get the maximum and minimum heights and compare them. The idea is to traverse the tree and for every node check if it's balanced. We need to write a recursive function that returns three things, a boolean value to indicate the tree is balanced or not, minimum height and maximum height. To return multiple values, we can either use a structure or pass variables by reference. We have passed maxh and minh by reference so that the values can be used in parent calls.

```
/* Program to check if a given Binary Tree is balanced like a Red-Black
#include <iostream>
using namespace std;

struct Node
{
    int key;
    Node *left, *right;
};

/* utility that allocates a new Node with the given key */
Node* newNode(int key)
{
    Node* node = new Node;
    node->key = key;
    node->left = node->right = NULL;
    return (node);
}

// Returns returns tree if the Binary tree is balanced like a Red-Black
// tree. This function also sets value in maxh and minh (passed by
// reference). maxh and minh are set as maximum and minimum heights of
bool isBalancedUtil(Node *root, int &maxh, int &minh)
{
    // Base case
    if (root == NULL)
    {
        maxh = minh = 0;
        return true;
    }

    int lmhx, lmn; // To store max and min heights of left subtree
```

# ITT Tech - Official Site

[itt-tech.edu](http://itt-tech.edu)

Tech-Oriented Degree Programs.  
Education for the Future.



## Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

```

    int rmhx, rmnh; // To store max and min heights of right subtree

    // Check if left subtree is balanced, also set lmxh and lmnh
    if (isBalancedUtil(root->left, lmxh, lmnh) == false)
        return false;

    // Check if right subtree is balanced, also set rmhx and rmnh
    if (isBalancedUtil(root->right, rmhx, rmnh) == false)
        return false;

    // Set the max and min heights of this node for the parent call
    maxh = max(lmxh, rmhx) + 1;
    minh = min(lmnh, rmnh) + 1;

    // See if this node is balanced
    if (maxh <= 2*minh)
        return true;

    return false;
}

// A wrapper over isBalancedUtil()
bool isBalanced(Node *root)
{
    int maxh, minh;
    return isBalancedUtil(root, maxh, minh);
}

/* Driver program to test above functions*/
int main()
{
    Node * root = newNode(10);
    root->left = newNode(5);
    root->right = newNode(100);
    root->right->left = newNode(50);
    root->right->right = newNode(150);
    root->right->left->left = newNode(40);
    isBalanced(root)? cout << "Balanced" : cout << "Not Balanced";

    return 0;
}

```

Output:

Balanced



Time Complexity: Time Complexity of above code is  $O(n)$  as the code does a simple tree traversal.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



## Related Tpoics:

- [Print a Binary Tree in Vertical Order | Set 2 \(HashMap based Method\)](#)
- [Print Right View of a Binary Tree](#)
- [Red-Black Tree | Set 3 \(Delete\)](#)
- [Construct a tree from Inorder and Level order traversals](#)
- [Print all nodes at distance k from a given node](#)
- [Print a Binary Tree in Vertical Order | Set 1](#)
- [Interval Tree](#)

695



## Recent Comments

[affizerv](#) Your example has two 4s on row 3, that's why it...

[Backtracking | Set 7 \(Sudoku\)](#) · 25 minutes ago

[RVM](#) Can someone please elaborate this Qs from above...

[Flipkart Interview | Set 6](#) · 45 minutes ago

[Vishal Gupta](#) I talked about as an Interviewer in general,...

[Software Engineering Lab, Samsung Interview | Set 2](#) · 45 minutes ago

[@meya](#) Working solution for question 2 of 4f2f round....

[Amazon Interview | Set 53 \(For SDE-1\)](#) · 1 hour ago

[sandeep void](#) rearrange(struct node \*head) {...

[Given a linked list, reverse alternate nodes and append at the end](#) · 2 hours ago

[Neha](#) I think that is what it should return as, in...

[Find depth of the deepest odd level leaf node](#) · 2 hours ago

- Print all nodes that are at distance k from a leaf node



8



Tweet

3



1

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

6 Comments

GeeksforGeeks

Sort by Newest ▾



Join the discussion...



**Suman** · 2 months ago

Is the third tree height balanced?

max = root (10) -> leaf (40) = 3

min = root(10) -> leaf (5) = 1

max <= 1 \* min not true

Please let me know if I am missing something?

^ | ▾ · Reply · Share ›



**Coder011** → Suman · 2 months ago

Actually, i was also thinking along the same lines, but acc. to me they l well, so it actually becomes, max= 4 and min=2, which is balanced.

^ | ▾ · Reply · Share ›



**Coder011** · 2 months ago

Simple dfs (level order) also works, <http://ideone.com/dw5k1A>

^ | ▾ · Reply · Share ›



**Sourabh Upadhyay** → Coder011 · 2 months ago

True it will work.

AdChoices ▸

▶ [Tree Root](#)

▶ [Node](#)

▶ [Java Source Code](#)

AdChoices ▸

▶ [Program Java](#)

▶ [C++ Program](#)

▶ [Binary Tree](#)

AdChoices ▸

▶ [Trees Tree](#)

▶ [Java Me](#)

▶ [Red Black Tree](#)

^ | v • Reply • Share ›



**learner** • 2 months ago

thanks for sharing this, could someone please explain how red black propertie  
twice min height

1 ^ | v • Reply • Share ›



**Kartik** → learner • 2 months ago

Tha max height and min height paths should have same number of bla  
consecutive red nodes.

2 ^ | v • Reply • Share ›



Subscribe



Add Disqus to your site

@geeksforgeeks, **Some rights reserved**

**Contact Us!**

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team