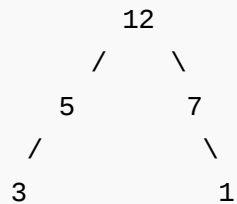
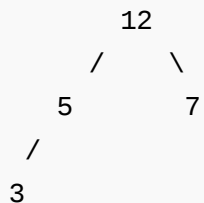


Check if all leaves are at same level

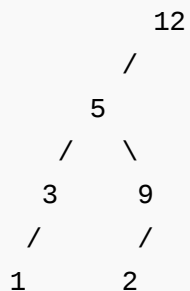
Given a Binary Tree, check if all leaves are at same level or not.



Leaves are at same level



Leaves are Not at same level



Leaves are at same level

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

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

The idea is to first find level of the leftmost leaf and store it in a variable leafLevel. Then compare level of all other leaves with leafLevel, if same, return true, else return false. We traverse the given Binary Tree in Preorder fashion. An argument leaflevel is passed to all calls. The value of leafLevel is initialized as 0 to indicate that the first leaf is not yet seen yet. The value is updated when we find first leaf. Level of subsequent leaves (in preorder) is compared with leafLevel.

```
// C program to check if all leaves are at same level
#include <stdio.h>
#include <stdlib.h>

// A binary tree node
struct Node
{
    int data;
    struct Node *left, *right;
};

// A utility function to allocate a new tree node
struct Node* newNode(int data)
{
    struct Node* node = (struct Node*) malloc(sizeof(struct Node));
    node->data = data;
    node->left = node->right = NULL;
    return node;
}

/* Recursive function which checks whether all leaves are at same level */
bool checkUtil(struct Node *root, int level, int *leafLevel)
{
    // Base case
    if (root == NULL) return true;

    // If a leaf node is encountered
    if (root->left == NULL && root->right == NULL)
    {
        // When a leaf node is found first time
        if (*leafLevel == 0)
        {
            *leafLevel = level; // Set first found leaf's level
            return true;
        }

        // If this is not first leaf node, compare its level with
```

ITT Tech - Official Site

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

```

    // first leaf's level
    return (level == *leafLevel);
}

// If this node is not leaf, recursively check left and right subtrees
return checkUtil(root->left, level+1, leafLevel) &&
        checkUtil(root->right, level+1, leafLevel);
}

/* The main function to check if all leaves are at same level.
   It mainly uses checkUtil() */
bool check(struct Node *root)
{
    int level = 0, leafLevel = 0;
    return checkUtil(root, level, &leafLevel);
}

```

```

// Driver program to test above function
int main()
{
    // Let us create tree shown in third example
    struct Node *root = newNode(12);
    root->left = newNode(5);
    root->left->left = newNode(3);
    root->left->right = newNode(9);
    root->left->left->left = newNode(1);
    root->left->right->left = newNode(1);
    if (check(root))
        printf("Leaves are at same level\n");
    else
        printf("Leaves are not at same level\n");
    getchar();
    return 0;
}

```

Output:

Leaves are at same level

Time Complexity: The function does a simple traversal of the tree, so the complexity is $O(n)$.

This article is contributed by **Chandra Prakash**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



BETTER THAN
**RAINBOWS IN
YOUR PANTS**

yah we said it

START YOUR FREE TRIAL

JRebel

**NO CREDIT CARD REQUIRED
TAKES LESS THAN 1 MIN.**



695



Recent Comments

affiszerv Your example has two 4s on row 3, that's why it...

[Backtracking | Set 7 \(Sudoku\) · 27 minutes ago](#)

RVM Can someone please elaborate this Qs from above...

[Flipkart Interview | Set 6 · 47 minutes ago](#)

Vishal Gupta I talked about as an Interviewer in general,...

[Software Engineering Lab, Samsung Interview | Set 2 · 47 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](#)

Related Topics:

- [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](#)
- [Check if a given Binary Tree is height balanced like a Red-Black Tree](#)



13

Tweet

3



1


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

AdChoices 

▶ [Binary Tree](#)

▶ [Java Tree](#)


▶ [Java to C++](#)

AdChoices 

▶ [Tree Leaves](#)

▶ [Leaf and Leaves](#)

▶ [Java Array](#)

AdChoices 

▶ [Red Leaf Tree](#)

▶ [Tree Root](#)

▶ [Tree Full](#)

@geeksforgeeks, **Some rights reserved**

Contact Us!

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