

# GeeksforGeeks

A computer science portal for geeks

Placements Practice GATE CS IDE Q&A  
GeeksQuiz

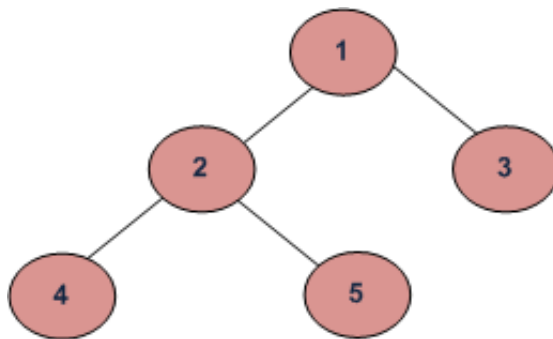
## Program to count leaf nodes in a binary tree

A node is a leaf node if both left and right child nodes of it are NULL.

Here is an algorithm to get the leaf node count.

```
getLeafCount(node)
```

- 1) If node is NULL then return 0.
- 2) Else If left and right child nodes are NULL return 1.
- 3) Else recursively calculate leaf count of the tree using below formula.  
Leaf count of a tree = Leaf count of left subtree +  
Leaf count of right subtree



*Example Tree*

Leaf count for the above tree is 3.

### Implementation:

C

```
#include <stdio.h>
#include <stdlib.h>

/* A binary tree node has data, pointer to left child
```

```
    and a pointer to right child */
struct node
{
    int data;
    struct node* left;
    struct node* right;
};

/* Function to get the count of leaf nodes in a binary tree*/
unsigned int getLeafCount(struct node* node)
{
    if(node == NULL)
        return 0;
    if(node->left == NULL && node->right==NULL)
        return 1;
    else
        return getLeafCount(node->left)+
               getLeafCount(node->right);
}

/* Helper function that allocates a new node with the
   given data and NULL left and right pointers. */
struct node* newNode(int data)
{
    struct node* node = (struct node*)
                        malloc(sizeof(struct node));

    node->data = data;
    node->left = NULL;
    node->right = NULL;

    return(node);
}

/*Driver program to test above functions*/
int main()
{
    /*create a tree*/
    struct node *root = newNode(1);
    root->left      = newNode(2);
    root->right     = newNode(3);
    root->left->left = newNode(4);
    root->left->right = newNode(5);

    /*get leaf count of the above created tree*/
    printf("Leaf count of the tree is %d", getLeafCount(root));

    getchar();
    return 0;
}
```

# Java



```
//Java implementation to find leaf count of a given Binary tree

/* Class containing left and right child of current
node and key value*/
class Node {

    int data;
    Node left, right;

    public Node(int item) {
        data = item;
        left = right = null;
    }
}

public class BinaryTree {

    //Root of the Binary Tree
    Node root;

    /* Function to get the count of leaf nodes in a binary tree*/
    int getLeafCount() {
        return getLeafCount(root);
    }

    int getLeafCount(Node node) {
        if (node == null) {
            return 0;
        }
        if (node.left == null && node.right == null) {
            return 1;
        } else {
            return getLeafCount(node.left) + getLeafCount(node.right);
        }
    }

    /* Driver program to test above functions */
    public static void main(String args[]) {
        BinaryTree tree = new BinaryTree();
        tree.root = new Node(1);
        tree.root.left = new Node(2);
        tree.root.right = new Node(3);
        tree.root.left.left = new Node(4);
        tree.root.left.right = new Node(5);

        /* get leaf count of the above tree */
        System.out.println("The leaf count of binary tree is : " + tree.getLeafCount());
    }
}
```

```
}  
}
```

**Time & Space Complexities:** Since this program is similar to traversal of tree, time and space complexities will be same as Tree traversal (Please see our [Tree Traversal](#) post for details)

Please write comments if you find any bug in the above programs/algorithms or other ways to solve the same problem.



36 Comments Category: Trees

## Related Posts:

- [Check if removing an edge can divide a Binary Tree in two halves](#)
- [Check sum of Covered and Uncovered nodes of Binary Tree](#)
- [Lowest Common Ancestor in a Binary Tree | Set 2 \(Using Parent Pointer\)](#)
- [Construct a Binary Search Tree from given postorder](#)
- [BFS vs DFS for Binary Tree](#)
- [Maximum difference between node and its ancestor in Binary Tree](#)
- [Inorder Non-threaded Binary Tree Traversal without Recursion or Stack](#)
- [Check if leaf traversal of two Binary Trees is same?](#)

(Login to Rate and Mark)

**1.6** Average Difficulty : **1.6/5.0**  
Based on **16** vote(s)

☐  
☐

Add to TODO List

Mark as DONE

Like Share 10 people like this.

Writing code in comment? Please use [code.geeksforgeeks.org](https://code.geeksforgeeks.org), generate link and share the link here.

@geeksforgeeks, Some rights reserved

[Contact Us!](#)

[About Us!](#)

[Advertise with us!](#)