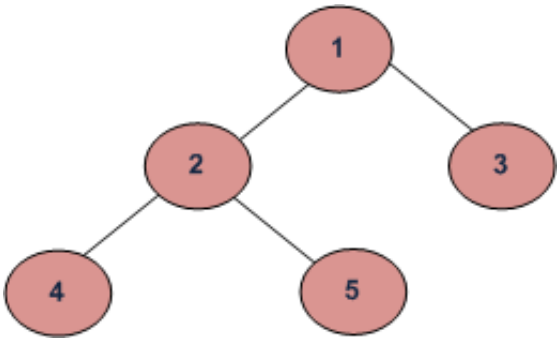
# Write a C program to Calculate Size of a tree

Size of a tree is the number of elements present in the tree. Size of the below tree is 5.




*Example Tree*

Size() function recursively calculates the size of a tree. It works as follows:

Size of a tree = Size of left subtree + 1 + Size of right subtree

**Algorithm:**

```
size(tree)
1. If tree is empty then return 0
2. Else
     (a) Get the size of left subtree recursively  i.e., call
          size( tree->left-subtree)
     (a) Get the size of right subtree recursively  i.e., call
          size( tree->right-subtree)
     (c) Calculate size of the tree as following:
            tree_size  =  size(left-subtree) + size(right-
                                subtree) + 1
     (d) Return tree_size
```

```
#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;
};

/* 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);
}

/* Computes the number of nodes in a tree. */
int size(struct node* node)
{
  if (node==NULL)
    return 0;
  else
    return(size(node->left) + 1 + size(node->right));
}

/* Driver program to test size function*/
int main()
{
  struct node *root = newNode(1);
  root->left        = newNode(2);
  root->right       = newNode(3);
  root->left->left  = newNode(4);
  root->left->right = newNode(5);

  printf("Size of the tree is %d", size(root));
  getchar();
  return 0;
}
```

Run on IDE

**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)

68 Comments Category: Trees Tags: Size of a Tree, Tree Size, TreeSize

## Related Posts:

- Check if a given array can represent Preorder Traversal of Binary Search Tree
- Mirror of n-ary Tree
- Succinct Encoding of Binary Tree
- Construct Binary Tree from given Parent Array representation
- Symmetric Tree (Mirror Image of itself)
- Find Minimum Depth of a Binary Tree

- Maximum Path Sum in a Binary Tree
- Expression Tree

0 Average Difficulty : **0/5.0**
No votes yet.

(Login to Rate)



Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.



**PrateekSingh Chauhan** • 2 months ago
how the above code return size of tree as 5 though it didnt use counter variable??
• Reply • Share ›

**Aditi Dubey** ➚ PrateekSingh Chauhan • 12 days ago
There is no need of counter variable because we do not want a loop to be executed that is to be calculated.And further it is very clear that recursion is used only to call the function again and again until node is equal to NULL.
• Reply • Share ›

**RailTracer** ➚ PrateekSingh Chauhan • a month ago
You dont need a counter variable because of the recursion.
the "return" is all you need..
• Reply • Share ›

**sundar** • 2 months ago
friend int size(tree *t)

{

if(!t)

return 0;

int count =0;

stack <tree *="">s;

s.push(t);

while(!s.empty())

{

++count;

t=t->l;

see more

• Reply • Share ›

**sundar** • 2 months ago

Iterative solution : if iam wrong plz correct me
friend int size(tree *t)

{

if(!t)

return 0;

int count =0;

stack <tree *="">s;

s.push(t);

while(!s.empty())

{

++count;

see more

• Reply • Share ›

**dhiru** • 4 months ago

we can also find the size of the tree while doing any traversal count the number of nodes printed (iteratively )

• Reply • Share ›

**Anup Rai** • 6 months ago

I have implemented this in iterative manner.
I have used vector just for practice.

http://ideone.com/AZ52RI

• Reply • Share ›

**Shantanu** ➚ Anup Rai • 4 months ago

plz post at lest a error free code

• Reply • Share ›

**Anup Rai** ➚ Shantanu • 3 months ago

The code is error free the problem is there is no main. I have just written the function write your main build your tree send the root as argument and the sun will rise for u...

• Reply • Share ›

**Bewkoof_coder** ➚ Anup Rai • 4 months ago

does above algo means,find the left most node(i.e its size from root) and same for right most node...i.e(its size from root)??

• Reply • Share ›

**Anup Rai** ➚ Bewkoof_coder • 4 months ago

Nope this algo means you manage to push all the element into the vector. i.e for each vector u push both of its children. And So on so basically each node will be pushed once onto the vector and will be poped once. Now each time u pop a node u increment ur count and u get the number of nodes in the tree.

• Reply • Share ›

**Komal Singh** • 6 months ago

Why can't we just increment count value each time we insert a node?
Accept the address of the count variable in a pointer in the insertnode function and increment each time a new node is added

• Reply • Share ›

**LNR** ➚ Komal Singh • 6 months ago

Because we may need to find the size of a tree already defined in some other program. Here the tree is defined by us, to show the correct working of the logic.

• Reply • Share ›

**NITIN PANCHAL** • 7 months ago

Yes we can use q queue to traverse the tree levelwise andcount the number of times pop function is called .Following is the code

//to fin d the size of a tree by using a queue

```
#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<queue>

using namespace std ;

struct tree

{

int data ;
```

see more

• Reply • Share ›

**Aryan Parker** • 8 months ago

Is there a way to do this non recursively?

• Reply • Share ›

**Satty** ➚ Aryan Parker • 6 months ago

use level order traversal and keep counting.

• Reply • Share ›

**Prashant Gupta** ➚ Aryan Parker • 8 months ago

Yes, it is possible, you will need to implement a functionality to store the parent of the node. Example: Implement your own stack or Use threaded tree etc.

• Reply • Share ›

**Shikha Gupta** • 9 months ago

The tree traversals always vist each node once hence instead of printing the node data ,a global count variable can be incremented in any of the traversal techniques to find size of the tree

• Reply • Share ›

**debashis_deb** ➚ Shikha Gupta • 8 months ago

No , it wont work . Consider the Inorder traversal , where the Left most node is visted first , then the parent and then the right node . After visiting the right node ,the pointer again comes back to the parent node , and as such count willl get incremented , which shoudn't , happen !! Code your algo and see the diffference !

• Reply • Share ›

**GOPI GOPINATH** ➚ debashis_deb • 8 months ago

size of a tree is just the number of elements or nodes in the tree. We can find it in any traversal methods. Thats what @shikha Guptha meant.

• Reply • Share ›

**Shikha Gupta** ➚ debashis_deb • 8 months ago

It will work for any traversal including inorder...it will work because based on what u r saying if After visiting the right node ,the pointer again comes back to the parent node

it doesn't display the parent node in normal inorder traversal does it...it just teminates..u can see the working code as follows:

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<math.h>

struct node{

int data;

struct node *llink;

see more

• Reply • Share ›

**Neha** • 9 months ago

When i run the code, why am I getting error

error: 'NULL' undeclared (first use in this function)
if (node==NULL)

• Reply • Share ›

**Aditya Goel** ➚ Neha • 9 months ago

Change your compiler. For the time-being add this line in your program
#define NULL 0

1 • Reply • Share ›

**backstabber** • 9 months ago

in the book by goodrich and tamassia it is mentioned that the size() takes O(1) time? can someone explain? shouldn't it be O(h)?

• Reply • Share ›

**Aditya Goel** ➚ backstabber • 9 months ago

No, size will take O(h) time unless we are also storing information about number of nodes in the tree itself.

• Reply • Share ›

**Kriti** • a year ago

Can anyone tell me what will be the output of this program?

1 • Reply • Share ›

**Goku** ➚ Kriti • 10 months ago

output will be 5 which is equal to the number of nodes in the tree.
size = no. of nodes in a tree.

1 • Reply • Share ›

**neeraj kumar** ➚ Kriti • a year ago

which program?

• Reply • Share ›

**Jerry Goyal** • a year ago

*using static variable*

```
int size(struct node* root){
        static int s=0;
        if(root==NULL){
                return;
        }
        s+=1;
        size(root->left);
        size(root->right);
        return s;
}
```

6 • Reply • Share ›

**Praveen Boodagoli** • a year ago

here is the java code , both space and time complexity O(n).

```
public int size()
{
return size(root);
}

private int size(TreeNode root) {
if(root==null) return 0;
return 1+size(root.left)+size(root.right);
```

}

1 • Reply • Share ›

**sexforyou** • a year ago

done

• Reply • Share ›

**Jasleen** • a year ago

Could you please explain the working of size function?

• Reply • Share ›

**CodeMe** ➚ Jasleen • a year ago

It's pretty simple.As it is a binary tree(tree which will have max 2 childrens).So,in order to know the size of a tree calculate the size of the left subtree and the subtree recursively and add 1 to it(one for the parent node).

1 • Reply • Share ›

**Guest** ➚ CodeMe • a year ago

Actually I think that +1 is for every time a node is reached as well.Correct me if I'm wrong.

1 • Reply • Share ›

**code_dweller** ➚ Guest • a year ago

Correct

• Reply • Share ›

**DS+Algo** ➚ Guest • a year ago

yeah, +1 is for the node that is being visited

1 • Reply • Share ›

**CodeMe** ➚ Guest • a year ago

I didn't get you what you are trying to ask.

• Reply • Share ›

**pvp** ➚ CodeMe • 8 months ago

you said that we add one for the parent node this is not true '+1' if for every time when the function calls itself , by the way because of this "+1" we are calculating this whole thing up , if it is not there the compiler will give us zero

• Reply • Share ›

**Ashish Jaiswal** • a year ago

here is simple code using counter variable while unwrapping of function of size function:

```
#include<stdio.h>

#include<stdlib.h>

struct node;

struct node* newnode(int);

int size(struct node*);

int c=0;

typedef struct node

{

int data;

struct node* left;
```

see more

1 • Reply • Share ›

**CodeMe** ➚ Ashish Jaiswal • a year ago
This code/approach is nothing different from what the trivial solution is.In fact your variable has increased the space by one variable.
• Reply • Share ›

**Ashish Jaiswal** ➚ CodeMe • 9 months ago
ok
• Reply • Share ›

**Amit Baghel** • a year ago
visited!
• Reply • Share ›

**Manraj Singh** ➚ Amit Baghel • a year ago
You can use this chrome extension.It helps you keep track of the problems solved as well as doubts.Also you can track your performance using inbuilt chart.

https://chrome.google.com/webs...
• Reply • Share ›

**CodeMe** ➚ Amit Baghel • a year ago
Are you trying to bookmark visited page by commenting here ?
You can ask for personalization to the admin rather,so that others could also get

benefited.

7 • Reply • Share ›

**Holden** ➚ CodeMe • 6 months ago

How we can bookmark problems as read or visited?

• Reply • Share ›

**Deepesh Panjabi** • a year ago

http://ideone.com/flGjiB

• Reply • Share ›

**Suraj Singh** • a year ago

check this out it is in c++ language do comment if you find any bug

```
#include<stdio.h>

#include<iostream>

using namespace std;

struct node

{

int a;

node *left;

node *right;

}*root = NULL;

void new_node(node **temp,int c)
```

see more

• Reply • Share ›

**san** • 2 years ago

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication5

{
```

class Program

{

static void Main(string[] args)

{

Node<int> root = new Node<int>(1);

see more

• Reply • Share ›

**rikitic** • 2 years ago

inserting count variable in pre-order traversal will do it..i suppose . any one contradicting ??

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

**Vibhu Tiwari** rikitic • 2 years ago

I think any order traversal will do it because in the traversal of the trees with any of the mechanism just increment the count variable as in these traversals every time we are encountering a new element. Thus the number of these elements will hold in the variable count.

3 • Reply • Share ›

Load more comments