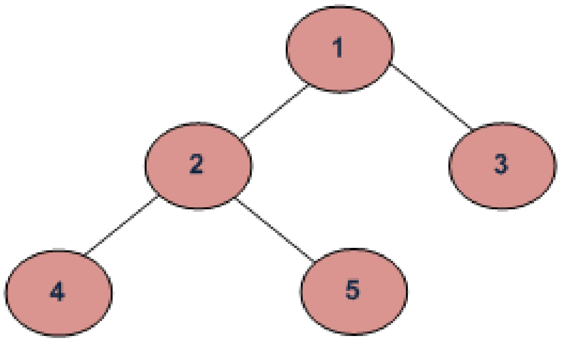
GeeksforGeeks
A computer science portal for geeks

IDE Q&A GeeksQuiz

Tree Traversals

Unlike linear data structures (Array, Linked List, Queues, Stacks, etc) which have only one logical way to traverse them, trees can be traversed in different ways. Following are the generally used ways for traversing trees.



Example Tree

Depth First Traversals:
(a) Inorder
(b) Preorder
(c) Postorder

Breadth First or Level Order Traversal
Please see this post for Breadth First Traversal.

Inorder Traversal:

```
Algorithm Inorder(tree)
   1. Traverse the left subtree, i.e., call Inorder(left-subtree)
   2. Visit the root.
   3. Traverse the right subtree, i.e., call Inorder(right-subtree)
```

Uses of Inorder
In case of binary search trees (BST), Inorder traversal gives nodes in non-decreasing order. To get nodes of BST in non-increasing order, a variation of Inorder traversal where Inorder itraversal s reversed, can be used.
Example: Inorder traversal for the above given figure is 4 2 5 1 3.

Preorder Traversal:

```
Algorithm Preorder(tree)
   1. Visit the root.
   2. Traverse the left subtree, i.e., call Preorder(left-subtree)
   3. Traverse the right subtree, i.e., call Preorder(right-subtree)
```

Uses of Preorder

Preorder traversal is used to create a copy of the tree. Preorder traversal is also used to get prefix expression on of an expression tree. Please see http://en.wikipedia.org/wiki/Polish_notation to know why prefix expressions are useful.

Example: Preorder traversal for the above given figure is 1 2 4 5 3.

Postorder Traversal:

```
Algorithm Postorder(tree)
   1. Traverse the left subtree, i.e., call Postorder(left-subtree)
   2. Traverse the right subtree, i.e., call Postorder(right-subtree)
   3. Visit the root.
```

Uses of Postorder

Postorder traversal is used to delete the tree. Please see the question for deletion of tree for details. Postorder traversal is also useful to get the postfix expression of an expression tree. Please see http://en.wikipedia.org/wiki/Reverse_Polish_notation to for the usage of postfix expression.

Example: Postorder traversal for the above given figure is 4 5 2 3 1.

```
#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);
}

/* Given a binary tree, print its nodes according to the
  "bottom-up" postorder traversal. */
void printPostorder(struct node* node)
{
     if (node == NULL)
        return;

     // first recur on left subtree
     printPostorder(node->left);

     // then recur on right subtree
     printPostorder(node->right);

     // now deal with the node
     printf("%d ", node->data);
}

/* Given a binary tree, print its nodes in inorder*/
void printInorder(struct node* node)
{
     if (node == NULL)
          return;

     /* first recur on left child */
     printInorder(node->left);

     /* then print the data of node */
     printf("%d ", node->data);

     /* now recur on right child */
     printInorder(node->right);
}

/* Given a binary tree, print its nodes in preorder*/
void printPreorder(struct node* node)
{
     if (node == NULL)
          return;

     /* first print data of node */
     printf("%d ", node->data);

     /* then recur on left sutree */
     printPreorder(node->left);

     /* now recur on right subtree */
     printPreorder(node->right);
}

/* Driver program to test above functions*/
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("\n Preorder traversal of binary tree is \n");
     printPreorder(root);

     printf("\n Inorder traversal of binary tree is \n");
     printInorder(root);

     printf("\n Postorder traversal of binary tree is \n");
     printPostorder(root);

     getchar();
     return 0;
```

```
}
```

Run on IDE

Time Complexity: O(n)
Let us prove it:

Complexity function T(n) — for all problem where tree traversal is involved — can be defined as:

$T(n) = T(k) + T(n – k – 1) + c$

Where k is the number of nodes on one side of root and n-k-1 on the other side.

Let’s do analysis of boundary conditions

Case 1: Skewed tree (One of the subtrees is empty and other subtree is non-empty)

k is 0 in this case.
$T(n) = T(0) + T(n-1) + c$
$T(n) = 2T(0) + T(n-2) + 2c$
$T(n) = 3T(0) + T(n-3) + 3c$
$T(n) = 4T(0) + T(n-4) + 4c$

…………………………………………
………………………………………….
$T(n) = (n-1)T(0) + T(1) + (n-1)c$
$T(n) = nT(0) + (n)c$

Value of T(0) will be some constant say d. (traversing a empty tree will take some constants time)

$T(n) = n(c+d)$
$T(n) = (-)(n)$ (Theta of n)

Case 2: Both left and right subtrees have equal number of nodes.

$T(n) = 2T(|\_n/2\_|) + c$

This recursive function is in the standard form ($T(n) = aT(n/b) + (-)(n)$) for master method http://en.wikipedia.org/wiki/Master_theorem. If we solve it by master method we get (-)(n)

Auxiliary Space : If we don’t consider size of stack for function calls then O(1) otherwise O(n).

60 Comments Category: Trees Tags: Inorder Traversal, PostOrder Traversal, Preorder Traversal, Tree Traveral, Trees, Tutorial

Related Posts:

- Find Count of Single Valued Subtrees
- 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

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

(Login to Rate)

Like Share 10 people like this. Be the first of your friends.

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

60 Comments **GeeksforGeeks** 1 **Login**

Recommend 5 **Share** **Sort by Newest**

Join the discussion…

Jex • 16 days ago

Animation of tree traversal
http://animatedarena.com/Jex/f...

• Reply • Share ›

SlickHackz • 24 days ago

Do we have Iterative solution which further improves the Space Complexity?

• Reply • Share ›

amit • 2 months ago

why it is written as struct node * node cant we write something other??thnx

• Reply • Share ›

Himanshu Mishra amit • 2 months ago

You can write. Like : struct node *variable_name;
It means that variable_name is pointer of type of struct node.

• Reply • Share ›

Jayesh • 3 months ago

Jayesh • 3 months ago

Java Implementation

http://javabypatel.blogspot.in...

⌃ | ⌄ • Reply • Share ›

Rohil Bansal • 4 months ago

Can somebody explain how recursion is taking place? and what function does 'return' play in the three traversals when the node is null?

⌃ | ⌄ • Reply • Share ›

ravi ➦ Rohil Bansal • 4 months ago

you can refer to this link to understand this problems

https://goo.gl/ARRJHW

⌃ | ⌄ • Reply • Share ›

Guest • 7 months ago

Could anyone please explain how are left and right subtrees are being printed bcoz there is only one printf for the root node??

⌃ | ⌄ • Reply • Share ›

Aditi Dubey ➦ Guest • 13 days ago

for more clarity i would say that each node gets printed just by one print command...if we write node->left or node->right then it behaves as data at one or other time.In this example 1 is root while 2 and 3 are left and right but in further traversal 2 behaves as data(root node) for left subtree similarly we define for every node

⌃ | ⌄ • Reply • Share ›

Abhivendra Singh ➦ Guest • 4 months ago

It's printing the root in which your pointer is right now so it will go to every node and print every node. You only need to print node once.

2 ⌃ | ⌄ • Reply • Share ›

Mission Peace • 7 months ago

Check out my youtube channel on binary tree questions
https://www.youtube.com/channe...

1 ⌃ | ⌄ • Reply • Share ›

BIPLABA PANDA • 8 months ago

i do not know c programming,data structure and algorithm

⌃ | ⌄ • Reply • Share ›

Kumar Som • 8 months ago

can anybody tell me what's a method to implement binary tree of any kind. it might be skewed......

^ | v • Reply • Share ›

Rahul Jha • a year ago

when i am typing this program same as written above then why it is not excuting and giving me the output on dev c++

^ | v • Reply • Share ›

Vikas ➦ Rahul Jha • a year ago

save file as *.c not *.cpp

1 ^ | v • Reply • Share ›

Saviour ➦ Rahul Jha • a year ago

This is a C code. Syntax differs.

^ | v • Reply • Share ›

Sameer Sah • a year ago

Typing error : Before code for printing in Preorder, in comments you wrote- " inorder". It should be "preorder". Just saying.

2 ^ | v • Reply • Share ›

GeeksforGeeks Mod ➦ Sameer Sah • 7 months ago

Thanks for pointing this out. We have updated the comment.

^ | v • Reply • Share ›

thevagabond85 ➦ Sameer Sah • 10 months ago

that's a copy-paste problem :P :P

3 ^ | v • Reply • Share ›

harveen • a year ago

Done- 14-10-14

^ | v • Reply • Share ›

Divya • a year ago

why should the "Tree traversals" always talks about BST and not the general trees?

6 ^ | v • Reply • Share ›

Aditya Goel ➦ Divya • 8 months ago

You can perform level Order Travesal on simple trees. There is no notion of inorder for tree that can have more than 2 children. So here inorder tree traversal uses BST.

You can modify preorder travesal to visit a node first and then call for all its childrens one by one.

Similarly for doing postorder traversal on a general tree, you can print a node after visit all childrens of a node.

1 ^ | ˅ • Reply • Share ›

RK ➦ Divya • a year ago

1) Any general tree can be represented as a binary tree.
2) Binary trees are used more than other types.
3) It is more important from an interview point of view.

^ | ˅ • Reply • Share ›

Holden ➦ RK • 6 months ago

how a n-try tree can be presented as a binary tree?
"1) Any general tree can be represented as a binary tree."

1 ^ | ˅ • Reply • Share ›

raju • a year ago

@Admin
please illustrate time complexity in simple way.

^ | ˅ • Reply • Share ›

GRV ➦ raju • a year ago

this time complexity for 2nd case

^ | ˅ • Reply • Share ›

GRV ➦ raju • a year ago

According to me time complexity is theta(n) by solving master method according to formula given in cormen

^ | ˅ • Reply • Share ›

CodeMe • a year ago

@Admin : Correct this line :

T(n) = (n-1)T(0) + T(1) + (n-1)c to T(n) = (n-1)T(0) + T(0) + (n-1)c

Only then T(n) = nT(0) + nc will be done.

^ | ˅ • Reply • Share ›

Amit Baghel • a year ago

visited !

^ | ˅ • Reply • Share ›

Deepesh Panjabi • a year ago

http://ideone.com/Tb9b43

˄ | ˅ • Reply • Share ›

CodeMe ➦ Deepesh Panjabi • a year ago

I know c ,c#,java.I have never got a chance to work on pointers directly.Could you please tell me why are we using struct node * instead of struct node ? I get confused at times.

˄ | ˅ • Reply • Share ›

Depen M ➦ CodeMe • a year ago

asterisk(*) before a variable name implies that the variable is one which will hold an address, and that address would be of data of the type T (here struct node which is an ABSTRACT DATA TYPE)

1 ˄ | ˅ • Reply • Share ›

Deepesh Panjabi ➦ CodeMe • a year ago

are you asking why we are using struct node * instead of struct node as the returned value type in newNode() function? or something else

˄ | ˅ • Reply • Share ›

CodeMe ➦ Deepesh Panjabi • a year ago

I get confused whenever i code for linked list related questions or anything that involves pointers.
How do i overcome this problem in me ?
Although i understand pointers but had this solution not given to me ,i wouldn't have coded it even though i know this pretty simple agorithm

˄ | ˅ • Reply • Share ›

Deepesh Panjabi ➦ CodeMe • a year ago

no worries :) just read about pointers and their uses again (under geeksforgeeks c language tab).
and from now on make extensive use of pointers in each and every code you write .. trust me, within 2 days you will get over all your confusion. all the best!

3 ˄ | ˅ • Reply • Share ›

DarkProtocol • 2 years ago

Two Types:

BFS-

1)Inorder - LRoot R --T(n) O(n) , Space used by stack in recursion S(n)=O(n) 2)Preorder

- Root L R --T(n) O(n) , Space used by stack in recursion S(n)=O(n) 3)Postorder - L R Root--T(n) O(n) , Space used by stack in recursion S(n)=O(n)

DFS -

1) Level Order - With Recursion T(n) - O(n2)
- With Queue's - T(n) - O(n) and S(n) - O(n)

4 ⌃ | ⌄ • Reply • Share ›

Holden ➦ DarkProtocol • 6 months ago

I mean "level order" is not DFS
and Inorder, preorder, postorder are not BFS

⌃ | ⌄ • Reply • Share ›

Holden ➦ DarkProtocol • 6 months ago

I think you should change "BFS" and "DFS" together.

⌃ | ⌄ • Reply • Share ›

dipak ➦ DarkProtocol • 2 years ago

preorder space used is O(n)...I am a bit confused can someone please explain with an example ?

⌃ | ⌄ • Reply • Share ›

initialcoder • 2 years ago

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

typedef struct NodeTag{
        char SYMBOL;
        struct NodeTag * LLINK;
        struct NodeTag * RLINK;
} TreeNode;

typedef enum{PreOrder, InOrder, PostOrder} OrderOfTraverse;

void visitNode(TreeNode * node){
        if(node == NULL)
                return;
        printf("%c ", node->SYMBOL);
}

void treeTraverse(TreeNode * treeRoot, OrderOfTraverse TraverseOrder){
```

see more

• Reply • Share ›

abhishek08aug • 3 years ago

Here is the C++ design/code for BST traversals. Recursive insert is bit tricky. I have solved the issue at below URL:

http://stackoverflow.com/quest...

```
 #include<iostream>
using namespace std;

class tree_node {
  private:
    int data;
    tree_node * left;
    tree_node * right;
  public:
    tree_node() {
      left=NULL;
      right=NULL;
    }
```

see more

2 • Reply • Share ›

Anvesh Kumar Somesula abhishek08aug • a year ago

can you please explain insert method

• Reply • Share ›

Anvesh Kumar Somesula abhishek08aug • a year ago

awesome...thank you..

• Reply • Share ›

abhishek08aug • 3 years ago

Here is the C++ design/code for BST traversals. Recursive insert is bit tricky. I have solved the issue at below URL:

http://stackoverflow.com/quest...

[sourcecode language="C++"]
#include<iostream>
using namespace std;

class tree_node {
private:

int data;
tree_node * left;
tree_node * right;
public:
tree_node() {
left=NULL;
right=NULL;
}

see more

^ | ⌄ • Reply • Share ›

Shikha Gupta ↪ abhishek08aug • 2 years ago

i hav written a simpler code to recursively insert in bst..but it gives error as
Runtime error time: 0 memory: 2288 signal:11
can some1 help me out the link is
http://ideone.com/HVLJ1q

^ | ⌄ • Reply • Share ›

Nikin • 3 years ago

```
 void inorder(node *sr)
{
if(sr == NULL)
return;
inorder(sr->left); cout<<sr->data<<" "; inorder(sr->right);
}
```

^ | ⌄ • Reply • Share ›

Deepak • 3 years ago

When inorder traversing a tree resulted E A C K F H D B G; the preorder traversal would return (b. FAEKCDHGB) … but how does it so……. can u explain it

^ | ⌄ • Reply • Share ›

Avinash • 4 years ago

Post order is the toughest out of all the tree traversals when solved iteratively. I am pasting the code in C#.

```
 //Post Order Traversal
        public void printpostorder()
        {
            Node current = Root;
            Node temp;
            if (current == null)
            {
```

```
{
    Console.WriteLine("Empty Tree");
    return;
}
else
{
    Stack<Node> myStack = new Stack<Node>();
    while (true)
    {
```

see more

• Reply • Share ›

Jacopo • 4 years ago

Your analysis is correct but you wronged the math. The last step is: T(n) = nT(0) + T(0) + nc => n(d+1 + c) => O(n). Right?

• Reply • Share ›

raa • 5 years ago

but what about complexity in the following case:
if number of nodes in left and right sub trees are not zero and not equal.

• Reply • Share ›

Load more comments

Subscribe Add Disqus to your site Privacy

@geeksforgeeks, Some rights reserved Contact Us! About Us! Advertise with us!