

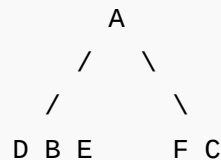
Construct Tree from given Inorder and Preorder traversals

Let us consider the below traversals:

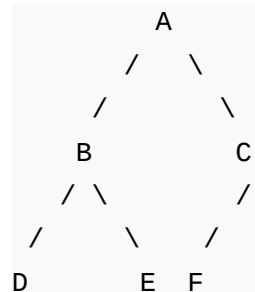
Inorder sequence: D B E A F C

Preorder sequence: A B D E C F

In a Preorder sequence, leftmost element is the root of the tree. So we know 'A' is root for given sequences. By searching 'A' in Inorder sequence, we can find out all elements on left side of 'A' are in left subtree and elements on right are in right subtree. So we know below structure now.



We recursively follow above steps and get the following tree.



Algorithm: buildTree()

1) Pick an element from Preorder. Increment a Preorder Index Variable (preIndex in below code)

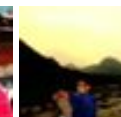
Google™ Custom Search



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](#)

to pick next element in next recursive call.

2) Create a new tree node tNode with the data as picked element.

3) Find the picked element's index in Inorder. Let the index be inIndex.

4) Call buildTree for elements before inIndex and make the built tree as left subtree of tNode.

5) Call buildTree for elements after inIndex and make the built tree as right subtree of tNode.

6) return tNode.

Thanks to Rohini and [Tushar](#) for suggesting the code.

```
/* program to construct tree using inorder and preorder traversals */
#include<stdio.h>
#include<stdlib.h>

/* A binary tree node has data, pointer to left child
   and a pointer to right child */
struct node
{
    char data;
    struct node* left;
    struct node* right;
};

/* Prototypes for utility functions */
int search(char arr[], int strt, int end, char value);
struct node* newNode(char data);

/* Recursive function to construct binary of size len from
   Inorder traversal in[] and Preorder traversal pre[]. Initial value
   of inStrt and inEnd should be 0 and len -1. The function doesn't
   do any error checking for cases where inorder and preorder
   do not form a tree */
struct node* buildTree(char in[], char pre[], int inStrt, int inEnd)
{
    static int preIndex = 0;

    if(inStrt > inEnd)
        return NULL;

    /* Pick current node from Preorder traversal using preIndex
       and increment preIndex */
    struct node *tNode = newNode(pre[preIndex++]);

    /* If this node has no children then return */
    if(inStrt == inEnd)
        return tNode;
```

ITT Tech - Official Site

itt-tech.edu

Associate, Bachelor Degree
Programs Browse Programs Now &
Learn More.



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](#)

```

/* Else find the index of this node in Inorder traversal */
int inIndex = search(in, inStrt, inEnd, tNode->data);

/* Using index in Inorder traversal, construct left and
   right subtress */
tNode->left = buildTree(in, pre, inStrt, inIndex-1);
tNode->right = buildTree(in, pre, inIndex+1, inEnd);

return tNode;
}

/* UTILITY FUNCTIONS */
/* Function to find index of value in arr[start...end]
   The function assumes that value is present in in[] */
int search(char arr[], int strt, int end, char value)
{
    int i;
    for(i = strt; i <= end; i++)
    {
        if(arr[i] == value)
            return i;
    }
}

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

    return (node);
}

/* This funcion is here just to test buildTree() */
void printInorder(struct node* node)
{
    if (node == NULL)
        return;

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

    /* then print the data of node */

```



Recent Comments

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

[Backtracking | Set 7 \(Sudoku\)](#) · 43 minutes ago

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

[Flipkart Interview | Set 6](#) · 1 hour ago

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

[Software Engineering Lab, Samsung Interview | Set 2](#) · 1 hour 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 · 3 hours ago

Neha I think that is what it should return as, in...

Find depth of the deepest odd level leaf node · 3 hours ago

AdChoices

[▶ Binary Tree](#)

[▶ Java Tree](#)

[▶ XML Tree Viewer](#)

```
printf("%c ", node->data);

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

/* Driver program to test above functions */
int main()
{
    char in[] = {'D', 'B', 'E', 'A', 'F', 'C'};
    char pre[] = {'A', 'B', 'D', 'E', 'C', 'F'};
    int len = sizeof(in)/sizeof(in[0]);
    struct node *root = buildTree(in, pre, 0, len - 1);

    /* Let us test the built tree by printing Inorder traversal */
    printf("\n Inorder traversal of the constructed tree is \n");
    printInorder(root);
    getchar();
}
```

Time Complexity: $O(n^2)$. Worst case occurs when tree is left skewed. Example Preorder and Inorder traversals for worst case are {A, B, C, D} and {D, C, B, A}.

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



Better Than Hadoop.

HPCC Systems is Big Data Processing and Analytics
Open Source. Proven. Trusted.

 LexisNexis

Learn More 

[▶ Tree Structure](#)[▶ Tree Root](#)[▶ Tree Trees](#)[▶ In the Tree](#)[▶ Preorder](#)[▶ Element Tree](#)

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](#)



36



Tweet

0



1

Writing code in comment? Please use ideone.com and share the link here.

54 Comments**GeeksforGeeks**

Sort by Newest ▼



Join the discussion...

**Karthikeyan** • 4 days ago

What if elements are repeated ?

^ | v • Reply • Share ›

**Marsha Donna** • 2 months agocan sum1 help me correct this <http://ideone.com/O6x1Dg>

^ | v • Reply • Share ›

**Abhishek Kumar** • 2 months ago

somebody plzz tell me wat is instrt and inend used in d above code ???

^ | v • Reply • Share ›



Ashutosh Litelo · 2 months ago

How is the complexity of the above code $O(n^2)$. Please explain

^ | v · Reply · Share ›



Narene Nagares · 3 months ago

Thanks! However, when I used it in a loop, output didn't work right on the seco

Here is the code:

```
/* program to construct tree using inorder and preorder traversals */
```

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

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

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

```
and a pointer to right child */
```

```
struct node
```

```
{
```

```
char data;
```

```
struct node* left;
```

[see more](#)

^ | v · Reply · Share ›



Feier Chen · 7 months ago

Here is My Code.

$O(n)$ time && $O(1)$ auxiliary space

```
public class CTree {
```

```
public class CTree2 {  
  
    public static class Node{  
  
        int val;  
  
        Node leftChild;  
  
        Node rightChild;  
  
        Node next;  
  
        public Node(int value)  
  
        {  
  
            val = value;  

```

[see more](#)

4 ^ | v • Reply • Share ›



Ajay • 7 months ago

Here is my implementation

```
#include<iostream.h>  
  
typedef struct node  
  
{  
  
    char x;  
  
    struct node* left;  
  
    struct node* right;  

```

```
}Node;
```

```
Node* makeNode(char x)
```

```
{
```

```
Node *n = (Node*)malloc(sizeof(Node));
```

[see more](#)

^ | v • Reply • Share ›



Prakash. • 7 months ago

why can't we do binary search, instead of linear search here, since we are supposed to be sorted.

^ | v • Reply • Share ›



samsammy → Prakash. • 7 months ago

Tree mentioned here is Binary Tree, Not Binary Search Tree.. So Binary search is not applicable...)

^ | v • Reply • Share ›



miandfhy • 8 months ago

Why it is character in search function??

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

^ | v • Reply • Share ›



rahul38 • 9 months ago

```
/**
```

```
* Definition for binary tree
```

```
* struct TreeNode {
```

```
* int val;
```

```
* struct TreeNode *left;
```



```

TreeNode *left,
* TreeNode *right;
* TreeNode(int x) : val(x), left(NULL), right(NULL) {}
* };
*/
class Solution {
public:
TreeNode *convert(vector<int> &v1, vector<int> &v2,int l1,int r1,int l2,int r2)
{
int i;
if(l1>r1)
return NULL;
TreeNode *t;
for( i=l1;i<=r1;i++)

```

[see more](#)

^ | v • Reply • Share ›



rahul38 • 9 months ago

```

/* /**
 * Definition for binary tree
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */

```

this is simple c++ code (recursive ..

```

class Solution {

```

```

public:

```

```
TreeNode *convert(vector<int> &v1, vector<int> &v2,int l1,int r1,
{
    int i;
    if(l1>r1)
```

[see more](#)

^ | v • Reply • Share ›



Hacker • 10 months ago

Please help me! I am unable to find error in this code.

[sourcecode language="C++"]

```
int build_cnt=0;
struct tree* build_tree(char *in,char *pre, int in_strt, int in_end)
{
    if(in_strt>in_end)
        return NULL;

    struct tree *root = new struct tree;

    int in_pos;
    for(int i=in_strt;i<=in_end;i++)
        if(pre[build_cnt]==in[i])
        {
            in_pos=i;
            break;
        }
    root = create_tree(pre[build_cnt]:
```

[see more](#)

^ | v • Reply • Share ›



sakekuma • 11 months ago



//Calling the function using below client code

```
//MyBinaryTree bt = new MyBinarytree();
```

```
//bt.pre_inorder_tree_helper(.....);
```

//Function starts..

```
[sourcecode language="JAVA"]
```

```
public void pre_inorder_tree_helper(int[] preorder, int[]  
inorder,int pre,int min,int max){
```

```
root=pre_inorder_tree(preorder,inorder,pre,min,max);  
}
```

```
private Node pre_inorder_tree(int[] preorder, int[] inorder,int pre,int min,int max
```

```
int left =0;int right=0;
```

```
if(min>max||pre>=preorder.length)  
return null;
```

see more

^ | v • Reply • Share ›



Faris A. Momani • a year ago

can u show how to draw if we have a post order traverse with inorder.

^ | v • Reply • Share ›



Bokang Moqelane • a year ago

this is a programming.

1 ^ | v • Reply • Share ›



Inder • a year ago

I have also given a tutorial here <http://bloggerplugnplay.blogspot...>

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

^ | v • Reply • Share ›



Spock • 2 years ago

Well here static is the keyword. If you won't declare preindex as static then the

^ | v • Reply • Share ›



kk • 2 years ago

How does the worst case complexity turn out to be $O(n^2)$ for left-skewed tree explain it in detail?

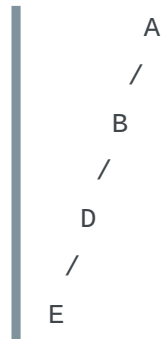
/* Paste your code here (You may **delete** these lines **if not** writing c

^ | v • Reply • Share ›



kartik → **kk** • 2 years ago

Let the skewed tree be



The given inorder traversal would be E D B A, and preorder would be A
In the above code, we one by one take elements from preorder travers
traversal. Searching the first element will take $O(n)$ time, searching the
searching the 3rd element will take $O(n-2)$ time, and so on. So total tim

... + 1. The sum of this series is $O(n*(n-1)/2)$ which is $O(n^2)$

[sourcecode language=""]

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

^ | v • Reply • Share ›



PG → kartik • a year ago

We can reduce this search time by using the binary search. rig
time.

/* Paste your code here (You may **delete** these lines **if**

^ | v • Reply • Share ›



kartik → PG • a year ago

Binary Search can only be applied for sorted arrays.

^ | v • Reply • Share ›



Ameya → kartik • a year ago

Sorry! I was thinking about just the BST.

/* Paste your code here (You may **delete** these li

^ | v • Reply • Share ›



Ameya → kartik • a year ago

Yes, but then inorder array will always be a sorted array

^ | v • Reply • Share ›



tiger • 2 years ago

Hi friends,
Here is my code.

Any suggestions?

Any suggestions?

```
#include<stdio.h>

struct node
{
    char val;
    struct node *left;
    struct node *right;
};

struct node *pre_in(char in[],char pre[],int end)
{

    static pre_ind=0;int pos,i;
```

[see more](#)

^ | v • Reply • Share ›



udaya → tiger • 2 years ago

Good.

But try without static variables.

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

^ | v • Reply • Share ›



tiger → udaya • 2 years ago

But why shouldn't static variables be used?

Please help me.

^ | v • Reply • Share ›



kartik → tiger • 2 years ago

@tiger: When you call the same function for different input, you get unexpected results. Because static variables may be shared across calls for a different input.

^ | v • Reply • Share ›



tiger → kartik • 2 years ago

Ya i get your point. Thank You.

```
/* Paste your code here (You may delete these lines) */
```

^ | v • Reply • Share ›



Braga • 2 years ago

Here is a Java Version of the same problem

<http://www.technicalypto.com/2...>

^ | v • Reply • Share ›



Rahul Menon • 2 years ago

I got a question saying

"Illustrate the construction of tree of a binary tree given its in order and post order"

Inorder: HDIJEKBALFMCNGO

POSTORDER:HIDJKEBLMFNOGCA

i got two different trees in my answer. Is it right? pls help me out

^ | v • Reply • Share ›



kartik → Rahul Menon • 2 years ago

@Rahul: Looks like you missed something. You can't get two trees for traversals.

^ | v • Reply • Share ›



gg · 3 years ago

the preIndex seems doesn't change cuz every time it starts from 0?

^ | v · Reply · Share ›



Mohit Ahuja → gg · 3 years ago

it would change because it is static.

^ | v · Reply · Share ›



mbrao · 3 years ago

is there a need to send preOrder end index to buildTree right, If we just increment trees out of leaf nodes as well ?

^ | v · Reply · Share ›



Anand · 3 years ago

Given a pre - order and post - order traversal construct a tree.

Assuming that the internal nodes has exactly two children we can uniquely identify
<http://anandtechblog.blogspot.com/2011/06/construct-given-tree-from-pre-order>

^ | v · Reply · Share ›



ai · 3 years ago

Construct a binary tree using the following 10
pre-order and in-order sequences :

Pre-order : 35, 31, 15, 7, 33, 32, 43, 38, 40, 49

In-order : 7, 15, 31, 32, 33, 35, 38, 40, 43, 49

how i make it?

thanks in advance for help.

^ | v · Reply · Share ›



Hatlar → ai · 2 years ago



Iterative DFS, looking for an efficient implementation. I know an explicit : explicit stack, and keep pushing all the adjacent nodes while exploring. Following this approach, in the worst case, explicit stack will have all the nodes in graph. Whereas in recursive approach, the system stack will be the depth of the longest node. Any thoughts? I am thinking to keep track of nodes in its adjacent list. In other words, mitigating system stack functionality

^ | v • Reply • Share ›



ankit → ai • 3 years ago

i think u know that very well, pre order is necessary for making a tree. ok now

pre order is like that, root left right.

so 35 is our parent node or root.

then covered it by a circle in both pre and in order.

in pre order the next element is 31, now we search the respective position

31 is in the left side of the root node '35' ok

that means 31 is the left child of 35.

similarly we check for all elements.

if any other query then contact me on ankitkharkwal@gmail.com

^ | v • Reply • Share ›



ankit → ai • 3 years ago

i think u know that very well, pre order is necessary for making a tree. ok now

pre order is like that, root left right.

so 35 is our parent node or root.

then covered it by a circle in both pre and in order.

in pre order the next element is 31, now we search the respective position

31 is in the left side of the root node '35' ok

that means 31 is the left child of 35.

similarly we check for all elements.

if any other query then contact me on ankitkharkwal@gmail.com

^ | v • Reply • Share ›



Abhishek • 3 years ago

Use Binary Search in

```
int search(char arr[], int strt, int end, char value)
```

Since InOrder is Increasing sequence, and that would make it $O(n \log n)$ solution

^ | v • Reply • Share ›



Kk → Abhishek • 3 years ago

It is a binary tree that we are trying to construct, not a binary search tree. The tree may / may not yield a sorted array.

Additionally, if we wanted to construct a binary search tree, pre-order traversal is necessary.

^ | v • Reply • Share ›



aayush kumar → Kk • 2 years ago

yup for binary tree i think inorder traversal is necessary and other but for BST preorder/postorder alone are sufficient.

but using postorder to construct is very difficult. however preorder constructing binary tree by including some symbol in place of null. see leetcode

<http://www.leetcode.com/2010/0/...>

^ | v • Reply • Share ›



xTristan • 3 years ago

(assume no duplicate chars allowed, otherwise there is some chance that the example above)

to optimize, you may pre-process the inorder string, create a hash map between characters and their indices, which eliminates the need for a linear search in your search() method, reducing the time complexity from linear to constant.

^ | v • Reply • Share ›



poonam • 3 years ago

I have done this :-

```
public BinaryNode BuildTree(int[] inOrder,int[] preOrder,int start,int end)
{
    if (start > end)
    {
        return null;
    }
    else
    {
        BinaryNode newNode = new BinaryNode(preOrder[preIndex]);
        if (start == end)
        {
            return newNode;
        }
        int inIndex = SearchInIndex(inOrder,start,end,newNode.val);
        preIndex++;
        newNode.left = BuildTree(inOrder,preOrder,start,inIndex);
        newNode.right = BuildTree(inOrder,preOrder,inIndex+1,end);
    }
}
```

[see more](#)

^ | v • Reply • Share ›



sharat • 3 years ago

As an optimization: We can have the first two arguments in[] and pre[] to be passed as global variables to the recursive function. This will save space in the process stack :)

^ | v • Reply • Share ›



Harsh • 4 years ago

If there are duplicates then i don't think you can construct a tree using preorde

I try this

Inorder : AAA

Preorder : AAA

^ | v • Reply • Share ›



rasmit → Harsh • 3 years ago

While searching in the inorder list, find the right most un-traversed element

^ | v • Reply • Share ›



thecodefreak • 4 years ago

I didn't get the logic of passing start and end and maintaining static variable.

try this

```
node* BuildFromInorderPreOrder(int *Inorder, int *Preorder, int length)
{
    if(length == 0)
        return NULL;
    if(length == 1)
        return new node(*Preorder);

    int RootIndex = 0;
    for(int i = 0; i < length && Inorder[i] != *Preorder; i++, RootIndex++)
        ;

    struct node *CurrentRoot = BuildFromInorderPreOrder(Inorder+RootIndex, Preorder, length-RootIndex-1);
    CurrentRoot->left = BuildFromInorderPreOrder(Inorder, Preorder, RootIndex);
    CurrentRoot->right = BuildFromInorderPreOrder(Inorder+RootIndex+1, Preorder+1, length-RootIndex-1);

    return CurrentRoot;
}
```

^ | v • Reply • Share ›



codegeek → thecodefreak • 4 years ago



codegeek [@theCodeGeek](#) · 4 years ago

in 2nd last line:

```
CurrentRoot->right = BuildFromInorderPreOrder(Inorder + RootIndex  
- RootIndex - 1);
```

Preorder + RootIndex + 1
does not look correct.

^ | v · Reply · Share ›

Load more comments



Subscribe



Add Disqus to your site

@geeksforgeeks, **Some rights reserved**

Contact Us!

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