Print Postorder traversal from given Inorder and Preorder traversals

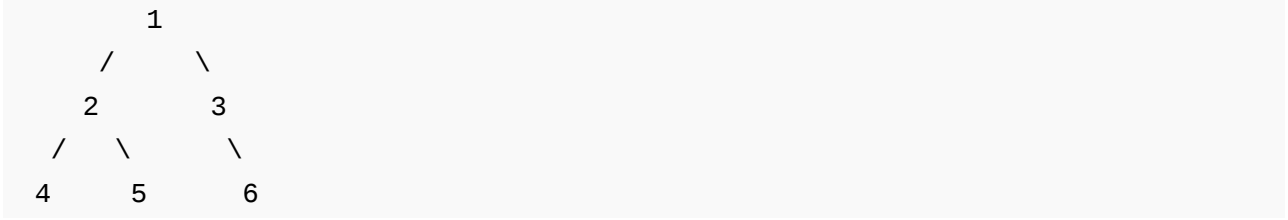
Given Inorder and Preorder traversals of a binary tree, print Postorder traversal.

Example:

```
Input:
Inorder traversal in[] = {4, 2, 5, 1, 3, 6}
Preorder traversal pre[] = {1, 2, 4, 5, 3, 6}

Output:
Postorder traversal is {4, 5, 2, 6, 3, 1}
```

Trversals in the above example represents following tree



A **naive method** is to first construct the tree, then use simple recursive method to print postorder traversal of the constructed tree.

We can print postorder traversal without constructing the tree. The idea is, root is always the first item in preorder traversal and it must be the last item in postorder traversal. We first recursively print left subtree, then recursively print right subtree. Finally, print root. To find boundaries of left and right subtrees in pre[] and in[], we search root in in[], all elements before

root in in[] are elements of left subtree and all elements after root are elements of right subtree. In pre[], all elements after index of root in in[] are elements of right subtree. And elements before index (including the element at index and excluding the first element) are elements of left subtree.

```
// C++ program to print postorder traversal from preorder and inorder
#include <iostream>
using namespace std;

// A utility function to search x in arr[] of size n
int search(int arr[], int x, int n)
{
    for (int i = 0; i < n; i++)
      if (arr[i] == x)
         return i;
    return -1;
}

// Prints postorder traversal from given inorder and preorder traversa
void printPostOrder(int in[], int pre[], int n)
{
   // The first element in pre[] is always root, search it
   // in in[] to find left and right subtrees
   int root = search(in, pre[0], n);

   // If left subtree is not empty, print left subtree
   if (root != 0)
      printPostOrder(in, pre+1, root);

   // If right subtree is not empty, print right subtree
   if (root != n-1)
      printPostOrder(in+root+1, pre+root+1, n-root-1);

   // Print root
   cout << pre[0] << " ";
}

// Driver program to test above functions
int main()
{
   int in[] = {4, 2, 5, 1, 3, 6};
   int pre[] =  {1, 2, 4, 5, 3, 6};
   int n = sizeof(in)/sizeof(in[0]);
   cout << "Postorder traversal " << endl;
   printPostOrder(in, pre, n);
   return 0;
```


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

```
}
```

Output

```
Postorder traversal
4 5 2 6 3 1
```

Time Complexity: The above function visits every node in array. For every visit, it calls search which takes O(n) time. Therefore, overall time complexity of the function is $O(n^2)$

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Tpoics:

- 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

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

18 Comments GeeksforGeeks

Sort by Newest

prashant jha • 3 days ago

```
#include<iostream>
using namespace std;
struct tnode
{
tnode* lchild;
int data;
tnode* rchild;
tnode(int d)
{
lchild=NULL;
data=d;
rchild=NULL;
}
```


Recent Comments

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

Backtracking | Set 7 (Sudoku) · 29 minutes ago

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

Flipkart Interview | Set 6 · 49 minutes ago

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

Software Engineering Lab, Samsung Interview | Set 2 · 49 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

```
};
int search(int n,int in[],int low1,int high1)
{
for(int i=low1;i<=high1;i++)
{
```

see more

• Reply • Share ›

arun kumar • 7 months ago

I have an O(n) solution for the same problem

```
#include <iostream>
#include <map>
#include <assert.h>
using namespace std;

int IN[] = {8,9,2,1,6,7,3,5,10,4};
int PRE[] = {1,2,8,9,5,3,6,7,4,10};
map<int, int=""> pos;

void printPostorder(int s, int e, int count)

{

if(s==e)
{
cout<<in[s]<<endl; return;="" }="" int="" root="PRE[count];" int="" rpos="pos[r
printpostorder(s,="" rpos-1,="" count);="" if(rpos="" !="e)" printpostorder(rpos+
cout<<root<<endl;="" }="" int="" main()="" {="" for(int="" i="0;" i<10;="" i++)="
}="">
```

2 • Reply • Share ›

bhavneet • 8 months ago

```
#include<stdio.h>

int search( int arr[], int start , int end , int se)
{
int index;
for(index=start ; index<= end ; index++)
if(arr[index]==se)
return index;
}

void inprePost(int inorder[], int start , int end , int preorder[], int *index)
{
if( start > end )
return ;
int in_index = search ( inorder, start , end , preorder[*index]);
(*index)++;
inprePost( inorder, start, in_index-1, preorder, index);
inprePost( inorder, in_index+1, end , preorder, index);
printf("%d", inorder[in_index]);
```

see more

7 ^ | v • Reply • Share ›

GeeksforGeeks Mod • 8 months ago

@All: Thanks for your inputs. We have updated the post and added time comp

1 ^ | v • Reply • Share ›

Guest • 8 months ago

@GeeksforGeeks

Why won't the complexity be O(n^2)
In case of a skewed tree for each node in pre we will have to search till the en

If there is some ammortized analysis then please add it to the post

If there is some ammortized analysis then please add it to the post

Reply • Share ›

Ravi chandra • 8 months ago

What would be time complexity of the above algorithm?
With some preliminary analysis I am assuming it is O(n^2) and space comple
if I am wrong.

Reply • Share ›

GeeksforGeeks Mod ➦ Ravi chandra • 8 months ago

Ravi Chandra: Time complexity seems to be O(n). The recurrence for

$T(n) = T(k) + T(n-k-1) + O(1)$

The above recurrence has solution as O(n), it is same as recurrence c

Reply • Share ›

Ronny ➦ GeeksforGeeks • 8 months ago

@GeeksforGeeks

How is it

$T(n)=T(k)+T(n-k-1)+O(1)$??

Shouldn't it be
$T(n)=T(k)+T(n-k-1)+O(n)$

because for each recursion we are doing O(n) search.

Kindly let us know your thoughts.

Reply • Share ›

GeeksforGeeks Mod ➦ Ronny • 8 months ago

Ronny:

Thanks for pointing this out. It should be T(n)=T(k)+T(n-

Apologies for the confusion.

^ | v • Reply • Share ›

Ronny ➚ GeeksforGeeks • 8 months ago

So the complexity of the code is O(n^2).

^ | v • Reply • Share ›

GeeksforGeeks Mod ➚ Ronny • 8 months ago

Yes, it is O(n^2).

^ | v • Reply • Share ›

rahul23 ➚ GeeksforGeeks • 8 months ago

@GeeksforGeeks

Wat about Searching in inorder arrary?In normal inorder we are
nodes.But in this we need to search in inorder array also.In cas
n^2?

1 ^ | v • Reply • Share ›

GeeksforGeeks Mod ➚ rahul23 • 8 months ago

Please tale a closer look, the code searches in inorder a
recurrence "T(n) = T(k) + T(n-k-1) + O(1)" is always O(

^ | v • Reply • Share ›

rahul23 ➚ GeeksforGeeks • 8 months ago

@GeeksforGeeks

Please clear following doubt:-

take a left skew tree with 6 nodes
1
Inorder is :- 6 5 4 3 2 1

preorder:- 1 2 3 4 5 6

For 1 you search n elements
For 2 it will search n-1 elements
For 3 it will search n-2 and so on

so isnt it n^2
For normal inorder it wiuld acces it node twice so its 0(r

Reply • Share ›

rahul23 rahul23 • 8 months ago

In this k turns out to be n so wouldnt it be n^2

Reply • Share ›

GeeksforGeeks • 8 months ago

Updated the example.

Reply • Share ›

GeeksforGeeks • 8 months ago

Thanks for pointing this out. We will update the example soon.

Reply • Share ›

Sanjay Agarwal • 8 months ago

In the example given , output is wrong. It should be:
Output:
Postorder traversal is {4, 5, 2, 6, 3, 1}.

Please correct it.

1 Reply • Share ›