Inorder Tree Traversal without recursion and without stack!

Using Morris Traversal, we can traverse the tree without using stack and recursion. The idea of Morris Traversal is based on Threaded Binary Tree. In this traversal, we first create links to Inorder successor and print the data using these links, and finally revert the changes to restore original tree.

```
1. Initialize current as root
2. While current is not NULL
   If current does not have left child
      a) Print current’s data
      b) Go to the right, i.e., current = current->right
   Else
      a) Make current as right child of the rightmost node in current's left subtree
      b) Go to this left child, i.e., current = current->left
```

Although the tree is modified through the traversal, it is reverted back to its original shape after the completion. Unlike Stack based traversal, no extra space is required for this traversal.

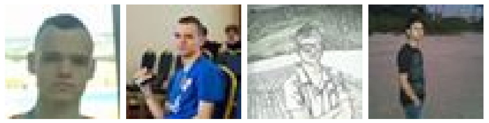
```
#include<stdio.h>
#include<stdlib.h>

/* A binary tree tNode has data, pointer to left child
   and a pointer to right child */
struct tNode
{
   int data;
   struct tNode* left;
   struct tNode* right;
```

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

```
};

/* Function to traverse binary tree without recursion and
   without stack */
void MorrisTraversal(struct tNode *root)
{
  struct tNode *current,*pre;

  if(root == NULL)
     return;

  current = root;
  while(current != NULL)
  {
    if(current->left == NULL)
    {
      printf(" %d ", current->data);
      current = current->right;
    }
    else
    {
      /* Find the inorder predecessor of current */
      pre = current->left;
      while(pre->right != NULL && pre->right != current)
        pre = pre->right;

      /* Make current as right child of its inorder predecessor */
      if(pre->right == NULL)
      {
        pre->right = current;
        current = current->left;
      }

      /* Revert the changes made in if part to restore the original
        tree i.e., fix the right child of predecssor */
      else
      {
        pre->right = NULL;
        printf(" %d ",current->data);
        current = current->right;
      } /* End of if condition pre->right == NULL */
    } /* End of if condition current->left == NULL*/
  } /* End of while */
}

/* UTILITY FUNCTIONS */
/* Helper function that allocates a new tNode with the
```

Geometric Algorithms

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

```
   given data and NULL left and right pointers. */
struct tNode* newtNode(int data)
{
  struct tNode* tNode = (struct tNode*)
                       malloc(sizeof(struct tNode));
  tNode->data = data;
  tNode->left = NULL;
  tNode->right = NULL;

  return(tNode);
}

/* Driver program to test above functions*/
int main()
{

  /* Constructed binary tree is
            1
          /   \
        2      3
      /  \
    4     5
  */
  struct tNode *root = newtNode(1);
  root->left        = newtNode(2);
  root->right       = newtNode(3);
  root->left->left  = newtNode(4);
  root->left->right = newtNode(5);

  MorrisTraversal(root);

  getchar();
  return 0;
}
```

References:

www.liacs.nl/~deutz/DS/september28.pdf

http://comsci.liu.edu/~murali/algo/Morris.htm

www.scss.tcd.ie/disciplines/software_systems/…/HughGibbonsSlides.pdf

Please write comments if you find any bug in above code/algorithm, or want to share more information about stack Morris Inorder Tree Traversal.

Better Than Hadoop.
HPCC Systems is Big Data Processing and Analytics
Open Source. Proven. Trusted.
LexisNexis
Learn More

695 Subscribe

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

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

24 Tweet 0 9

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

62 Comments GeeksforGeeks

Sort by Newest ▾

Join the discussion…

alien • 20 days ago
what is the complexity of this algorithm?
Reply • Share ›

Himanshu Dagar ➦ alien • 11 days ago
alien it will be O(n^2)
b/c every time its going down for setting or erasing the links of inorder s
Reply • Share ›

AlienOnEarth ➦ Himanshu Dagar • 4 days ago
@Himanshu Dagar: http://stackoverflow.com/quest...
this link claims it to be o(n). Even its hard for me to agree with t
Reply • Share ›

Gaurav Gupta • 21 days ago
could any one pls explain what is use for threaded binary tree..???
Reply • Share ›

Vulkum • 22 days ago
What is "pre" used for? It has no use whatsoever in this algorithm. It only appe
operations
Reply • Share ›

Vulkum ➦ Vulkum • 22 days ago
oh, nvm, missed the assignment line
Reply • Share ›

Ashutosh Litelo · 2 months ago

Dint understand the restoration part.can u explain wid a dry run

```
/* Revert the changes made in if part to restore the original
tree i.e., fix the right child of predecssor */
else
{
pre->right = NULL;
printf(" %d ",current->data);
current = current->right;
} /* End of if
```

· Reply · Share ›

Vinod · 3 months ago

Iterative inorder traversal of a BST without stack - http://ideone.com/qqd2PJ

Idea:

```
curr = minimumBST(root);
while(curr) {
print curr->data;
curr = successorBST(curr);
}
```

2 · Reply · Share ›

Ram Sinha · 8 months ago

This will cause the infinite loop.. better use some flag as pointed out by some

5 · Reply · Share ›

ibn · 9 months ago

1. Initialize current as root
2. While current is not NULL
If current does not have left child
a) Print current's data

b) Go to the right, i.e., current = current->right
Else
a) Make current as right child of the rightmost node in current's left subtree
b) Go to this left child, i.e., current = current->left

The pseudo code has infinite loop due to missing the undo step.

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

Reply • Share ›

Asap • 10 months ago
How Can we use this approach for postorder traversal?
Reply • Share ›

shivi • 10 months ago

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

Reply • Share ›

Coder • 10 months ago
What is the complexity of the coding O(n) or O(nlogn) ??
Reply • Share ›

abhishek08aug • a year ago
Intelligent :D
1 • Reply • Share ›

Vinod Panchal • a year ago
Else

a) Make current as left child of the leftmost node in current&#039s subtree.

b) Go to this left child, i.e., current = current->left.

· Reply · Share ›

Psycho · a year ago

Is it possible to acheive if the tree is readonly ?

· Reply · Share ›

denial Psycho · 10 months ago

@Psycho : this method only applicable when writes are allowed in the stack in-order traversal.

· Reply · Share ›

Anirvana Mishra · a year ago

Sandeep - My earlier comment was based on your algorithm. When I read you mechanism to prevent infinite looping. You should mention it in the algorithm a ELSE condition saying "If current already is Right child of rightmost child in the the threading link, print current&#039s data and current=current->right.

· Reply · Share ›

Sandeep Jain · a year ago

Anirvana, Thanks for inputs. Could you please provide an example tree for whi

· Reply · Share ›

Anirvana Mishra · a year ago

There is a bug in the algorithm that will cause infinite loop in many test cases. between current and current->left in the ELSE condition of the algorithm.

If you don&#039t want to destroy the left link you&#039ll either need to use sor keep a flag in each node that will tell you if the node has already been visited.

1 · Reply · Share ›

yelnatz • a year ago

Does this destroy the tree?

Reply • Share ›

bunnylol ➦ yelnatz • a year ago

To answer my question from 3 months ago.

No.

Reply • Share ›

googlybhai • a year ago

```
Good Code
```

Reply • Share ›

Ramakrishna • a year ago

```
if(current->left == NULL)
   {
     printf(" %d ", current->data);
     current = current->right;
   }
```

For the given example tree where '4' is the left most leaf node, the above code
current to null and hence the while loop breaks, after printing '4'. The tree is nc
something here? Please help me.

Reply • Share ›

aygul ➦ Ramakrishna • a year ago

Yes you are mssing the point that in the previous iteration right pt of 4 v
current->right will be 2 and it will be printed in the inner else part just af
put the tree back to original. Is it clear now ?

Reply • Share ›

Reply · Share ›

Ramakrishna aygul · a year ago

Thanks very much aygul ... I got that now..

Reply · Share ›

nikhil Ramakrishna · a year ago

I have noticed the same thing.
Something is missing in the algo.

Reply · Share ›

Harish Ladhani · 2 years ago

```
Void inorder(struct stack btree *root)
{
struct stack s;
struct btree*p;
if(root==NULL)
{Print msg empty tree and return
}
p=root;
s.tos=-1;
push(&s,root);
while(s.tos!=-1)
{
a: do
{
if(p->left!=null)
push(&s,p->left);
```

see more

Reply · Share ›

Reply Share ›

Siva Harish Ladhani • a year ago

The rule here clearly says don't use stack

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

Reply • Share ›

arpit • 2 years ago

converting tree to threaded tree uses recursion. so how can we consider this

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

Reply • Share ›

arpit arpit • 2 years ago

sorry, i didn't go through references.

Reply • Share ›

atul • 2 years ago

optimized code :-

```
 void MorrisTraversal(struct tNode *root)
{
  struct tNode *current,*pre;

  if(root == NULL)
     return;

  current = root;
  while(current != NULL)
  {
    if(current->left == NULL)
    {
      printf(" %d ", current->data);
      current = current->right;
    }
    else
```

see more

Reply • Share ›

atul atul • 2 years ago

typo error :-
pre->right=NULL instead of pre->next=NULL;

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

Reply • Share ›

atul atul • 2 years ago

little typo error in above code :-
pre->right=NULL instead of pre->next=NULL;

Reply • Share ›

Ranga • 2 years ago

This code doesnt seem to work for the below tree
A
/
B
\
C
right?
Infinite loop isnt it?

^ | v • Reply • Share ›

GeeksforGeeks → Ranga • a year ago

It works for this tree also. Please see following code. The code prints 2

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

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

/* Function to traverse binary tree without recursion and
   without stack */
void MorrisTraversal(struct tNode *root)
{
```

see more

^ | v • Reply • Share ›

Aniruddha • 2 years ago

Morris Traversal might HANG in infinite loop for certain TREE structure.

Try running given code for below tree:
[You will see it printing wrong data and infinitely looping]

```
        A
    B       C
```

```
D   E         F
  G       K
```

Morris traversal gets stuck , after visiting left subtree of A and then while traver
It will try to reLink E->A continously and gets shifted to left subtree of A (which

Reply • Share ›

akshat gupta Aniruddha • 11 months ago

Agreed..
One Solution: If we modify the Node structure,to conatain an extra "lvis
if(lvisited==1)
{
=>Left Subtree has been Visited
=>so Print Node Value
=>Move to Right Child
}
else
{
=>Left Tree has been Unvisited
=>Link Rightmost node ,of leftmost tree with Node
=>Mark lvisited=1
=>Move to Left Child
}

Reply • Share ›

Steven Aniruddha • a year ago

Why this will be infinite?

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

Reply • Share ›

srvkumar ➦ Aniruddha • 2 years ago

May be the method works fine only for 'complete trees'. But I am not su

^ | v • Reply • Share ›

srvkumar ➦ Aniruddha • 2 years ago

May be the method works fine for 'complete trees'. But I am not sure, h

^ | v • Reply • Share ›

Jack • 2 years ago

Here's a challenge: How can you do that with O(1) space complexity without c
in addition to the regular operations you can do with trees, you can also go to a
Is it possible?

^ | v • Reply • Share ›

Dr.Sai • 2 years ago

See page 298 Data structures using C and C++ Tanenbaum Threaded binary

^ | v • Reply • Share ›

abhk • 2 years ago

@karthik whats d time complexcity of morris algo , pls explain in detail ?

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

^ | v • Reply • Share ›

foram • 3 years ago

Are there similar methods for preorder and postorder traversals?

^ | v • Reply • Share ›

Anand • 3 years ago

In order, post order and pre order traversal without recursion.
http://anandtechblog.blogspot....

• Reply • Share ›

rahul ➦ Anand • 4 months ago

Those are using stack in their iterative method. I think with Morris meth
complexity for postorder traversal

• Reply • Share ›

raju • 3 years ago

I don't understand how you can revert the changes !!!
Where do you put that code to change the pointers back to NULL.

• Reply • Share ›

intel2390 ➦ raju • 3 years ago

one more thing .. why we have changed pre->right=NULL before currre
explain ??

• Reply • Share ›

Pranay Choudhary ➦ intel2390 • 3 years ago

You have answered your own question. pre->right=NULL is dor
tree.
Draw a tree. Trace the code. Its really simple.

• Reply • Share ›

Load more comments

Subscribe   Add Disqus to your site