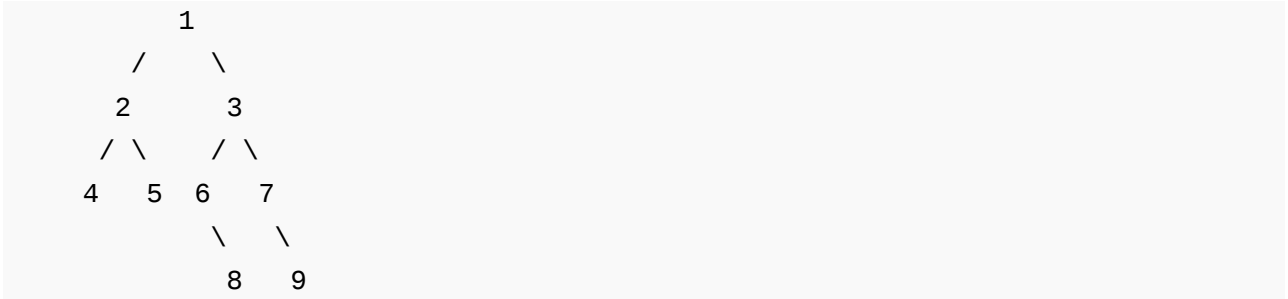
Print a Binary Tree in Vertical Order | Set 1

Given a binary tree, print it vertically. The following example illustrates vertical order traversal.



```
The output of print this tree vertically will be:
4
2
1 5 6
3 8
7
9
```

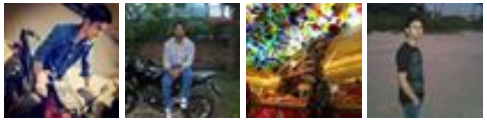
We strongly recommend to minimize the browser and try this yourself first.

The idea is to traverse the tree once and get the minimum and maximum horizontal distance with respect to root. For the tree shown above, minimum distance is -2 (for node with value 4) and maximum distance is 3 (For node with value 9).
Once we have maximum and minimum distances from root, we iterate for each vertical line at distance minimum to maximum from root, and for each vertical line traverse the tree and print the

Google™ Custom Search

GeeksforGeeks

52,731 people like GeeksforGeeks.



open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

nodes which lie on that vertical line.

Algorithm:

```
// min --> Minimum horizontal distance from root
// max --> Maximum horizontal distance from root
// hd  --> Horizontal distance of current node from root
findMinMax(tree, min, max, hd)
     if tree is NULL then return;

     if hd is less than min then
          min = hd;
     else if hd is greater than max then
          *max = hd;

     findMinMax(tree->left, min, max, hd-1);
     findMinMax(tree->right, min, max, hd+1);


printVerticalLine(tree, line_no, hd)
     if tree is NULL then return;

     if hd is equal to line_no, then
          print(tree->data);
     printVerticalLine(tree->left, line_no, hd-1);
     printVerticalLine(tree->right, line_no, hd+1);
```

Implementation:
Following is C++ implementation of above algorithm.

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

// A node of binary tree
struct Node
{
    int data;
    struct Node *left, *right;
```



Integrated
Desktop & Mobile Device
Management
ManageEngine
Desktop Central
Download

Popular Posts

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
};

// A utility function to create a new Binary Tree node
Node* newNode(int data)
{
    Node *temp = new Node;
    temp->data = data;
    temp->left = temp->right = NULL;
    return temp;
}

// A utility function to find min and max distances with respect
// to root.
void findMinMax(Node *node, int *min, int *max, int hd)
{
    // Base case
    if (node == NULL) return;

    // Update min and max
    if (hd < *min)  *min = hd;
    else if (hd > *max) *max = hd;

    // Recur for left and right subtrees
    findMinMax(node->left, min, max, hd-1);
    findMinMax(node->right, min, max, hd+1);
}

// A utility function to print all nodes on a given line_no.
// hd is horizontal distance of current node with respect to root.
void printVerticalLine(Node *node, int line_no, int hd)
{
    // Base case
    if (node == NULL) return;

    // If this node is on the given line number
    if (hd == line_no)
        cout << node->data << " ";

    // Recur for left and right subtrees
    printVerticalLine(node->left, line_no, hd-1);
    printVerticalLine(node->right, line_no, hd+1);
}

// The main function that prints a given binary tree in
// vertical order
void verticalOrder(Node *root)
{
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    // Find min and max distances with resepect to root
    int min = 0, max = 0;
    findMinMax(root, &min, &max, 0);

    // Iterate through all possible vertical lines starting
    // from the leftmost line and print nodes line by line
    for (int line_no = min; line_no <= max; line_no++)
    {
        printVerticalLine(root, line_no, 0);
        cout << endl;
    }
}

// Driver program to test above functions
int main()
{
    // Create binary tree shown in above figure
    Node *root = newNode(1);
    root->left = newNode(2);
    root->right = newNode(3);
    root->left->left = newNode(4);
    root->left->right = newNode(5);
    root->right->left = newNode(6);
    root->right->right = newNode(7);
    root->right->left->right = newNode(8);
    root->right->right->right = newNode(9);

    cout << "Vertical order traversal is \n";
    verticalOrder(root);

    return 0;
}
```

Output:

```
Vertical order traversal is
4
2
1 5 6
3 8
7
9
```

Time Complexity: Time complexity of above algorithm is O(w*n) where w is width of Binary Tree

695 Subscribe

Recent Comments

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

Backtracking | Set 7 (Sudoku) · 25 minutes ago

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

Flipkart Interview | Set 6 · 45 minutes ago

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

Software Engineering Lab, Samsung Interview | Set 2 · 45 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

AdChoices

▶ Tree Diagram

▶ Binary Tree

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

and n is number of nodes in Binary Tree. In worst case, the value of w can be O(n) (consider a complete tree for example) and time complexity can become $O(n^2)$.

This problem can be solved more efficiently using the technique discussed in this post. We will soon be discussing complete algorithm and implementation of more efficient method.

This article is contributed by **Shalki Agarwal**. 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
- Interval Tree
- Check if a given Binary Tree is height balanced like a Red-Black Tree
- Print all nodes that are at distance k from a leaf node

AdChoices
▶ Vertical Line
▶ 4 Vertical
▶ Java Tree
AdChoices
▶ Graph Java
▶ Java Print
▶ Print Vertical

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

13 Tweet 3 1

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

9 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Kirtika Salhotra • 15 days ago

Java Implementation for the same can be find below :

```
public void findMinimumDistance(Node root, int minDistance,HashMap<intege

LinkedList<integer> temp = null;

if(root != null)
{

findMinimumDistance(root.leftChild,minDistance-1,hm);
//root.minDistance = minDistance++;

findMinimumDistance(root.rightChild,minDistance+1,hm);

if(hm.get(minDistance) != null)
{
temp = hm.get(minDistance);
}
else
{
```

see more

• Reply • Share ›

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

newbie • 16 days ago

add root->left->right->right= newNode(11);

it will produce wrong results as the case when there are more elements in rigl
right sub tree is not taken care of here.

• Reply • Share ›

Guest • 23 days ago

An O(n) solution [Space complexity = Time complexity = O(n)] :
(Language: C++)
Complete Program: http://ideone.com/sGMGGk

```
// Global variables "mapped", "mini" and "maxi"
map<int, int=""> mapped;
int mini, maxi;

void vertical(struct node *root, int dist, vector<int> ans[]) {
        if(root == NULL)
                return;

        mini = min(mini, dist);
        maxi = max(maxi, dist);

        int index = mapped[dist];
```

see more

• Reply • Share ›

Zheng Luo • a month ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Reply • Share ›

Suman • a month ago

java code that uses hashmap

http://ideone.com/J2lnia

Reply • Share ›

Suman • a month ago

```
import java.util.HashMap;

import java.util.LinkedList;

import java.util.List;

import java.util.Map;

import java.util.TreeMap;

//http://www.geeksforgeeks.org/p...

public class PrintBinaryTreeInVerticalOrder

{

static class Node

{

int data;
```

see more

Reply • Share ›

Chirag • a month ago

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

C++ code using map and vector
Time Complexity : O(n) , Space Complexity : O(n)
http://ideone.com/wlZrLt

2 ^ | v • Reply • Share ›

Chirag • a month ago

C++ code using map and vector
http://ideone.com/wlZrLt

^ | v • Reply • Share ›

dhruv • a month ago

if we use a hashmap with key as distance and value as linked list having node restrict it to O(n) time complexity but with space complexity increased.

1 ^ | v • Reply • Share ›

Subscribe Add Disqus to your site

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com