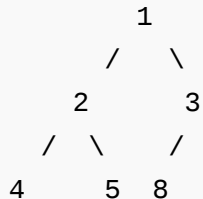


## Print nodes at k distance from root

Given a root of a tree, and an integer k. Print all the nodes which are at k distance from root.

For example, in the below tree, 4, 5 & 8 are at distance 2 from root.



The problem can be solved using recursion. Thanks to [eldho](#) for suggesting the solution.

```
#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;
};

void printKDistant(struct node *root , int k)
{
    if(root == NULL)
        return;
    if( k == 0 )
    {
        printf( "%d ", root->data );
    }
}
```

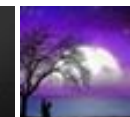
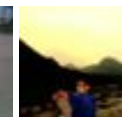
Google™ Custom Search



GeeksforGeeks



52,731 people like [GeeksforGeeks](#).



Interview Experiences

Advanced Data Structures

Dynamic Programming

Greedy Algorithms

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

```

        return ;
    }
    else
    {
        printKDistant( root->left, k-1 ) ;
        printKDistant( root->right, k-1 ) ;
    }
}

/* 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);
}

/* Driver program to test above functions*/
int main()
{
    /* Constructed binary tree is
        1
       / \
      2   3
     / \ / \
    4  5 8
    */
    struct 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(8);

    printKDistant(root, 2);

    getchar();
    return 0;
}

```

The above program prints 4, 5 and 8.



**Visual Studio Extension**  
For Faster, Smarter Coding  
Telerik JustCode™

System.Drawing.Color

```

public float GetSaturation()
{
    float r = (float)this.R / 255f;
    float g = (float)this.G / 255f;
    float b = (float)this.B / 255f;
    float single = 0f;
    float single1 = r;
    float single2 = r;
}

```

Autos

Name	Value	Type

**Download free trial**

Telerik®

## 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

Time Complexity:  $O(n)$  where  $n$  is number of nodes in the given binary tree.

Please write comments if you find the above code/algorithm incorrect, or find better ways to solve the same problem.



Better Than Hadoop.

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

LexisNexis® Learn More ▶

## 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](#)



1



Tweet 0



1

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

Shouldn't you  
expect a  
cloud with:

# SYSTEM MONITORING

Plus the experts  
to help run it?

TRY MANAGED CLOUD ▶

 **rackspace**  
the open cloud company



Sort by Newest ▼



Join the discussion...



alien · 20 days ago

We can use Level order traversal for this. It is the implied traversal for this prob

^ | v · Reply · Share ›



Gandalf · 2 months ago

```
public boolean printNodesAtKDistanceFromLeaf(Node node, int k) {
```

```
    if (node == null)
```

```
        return false;
```

```
    if (node.left == null & node.right == null) {
```

```
        kDistance[0] = 1;
```

```
        return true;
```

```
    }
```

```
    if (printNodesAtKDistanceFromLeaf(node.left, k) && kDistance[0] <= k) {
```

```
        System.out.println(node.left.data);
```

## Recent Comments

affizerv 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

```

}

if (printNodesAtKDistanceFromLeaf(node.right, k) && kDistance[0] <= k) {

System.out.println(node.right.data);

}

kDistance[0]++;

return true;

}

```

^ | v • Reply • Share ›



**MacM** • 2 months ago

If we want to get all nodes which are at "K" from any given node  
Here in this example I used value of node (int required node) .. from this value track all nodes which are at "K" distance.

```

public int PrintNodesAtKDistance(Node root, int requiredNode, int iDistance)
{
if ((root == null) || (iDistance < 0))
return -1;

int lPath = -1, rPath = -1;

if(root.value == requiredNode)
{
PrintChildNodes(root, iDistance);
return iDistance - 1;
}

lPath = PrintNodesAtKDistance(root.left, requiredNode, iDistance);

```

AdChoices ▶

▶ [Java Tree](#)

▶ [Java to C++](#)

▶ [Nodes](#)

AdChoices ▶

▶ [Root Size](#)

▶ [Java Tree](#)

▶ [Java to C++](#)

AdChoices ▶

▶ [Root Block](#)

▶ [Root Development](#)

▶ [Tree Block](#)

Print - PrintNodesAtDistance(root, right, requiredNode, idistance),

[see more](#)

^ | v • Reply • Share ›



**Guest** • 2 months ago

// Print\_K\_Distance\_Nodes.cpp : Defines the entry point for the console applic  
//

/\*

Problem: Given a Source Node in a binary tree, Print all nodes that are "K" ho  
can be the children / grand - children, parent / grand - parent or siblings / gran  
Note : -You cannot use a parent pointer to solve this problem.

Example1:

Input : k = 2, Node = 70

45

/\

/\

/\

/\

22 70

/\ /\

/\ /\

[see more](#)

^ | v • Reply • Share ›



**Guest** • 2 months ago

test

^ | v • Reply • Share ›





How can we do it when we have to print the nodes at distance K from leaves a

^ | v • Reply • Share ›



Nitesh • 10 months ago

```
/*Elements with K distance*/  
void kDistance(node *root, int dist)  
{  
    if(root == NULL)  
        return;  
    if(dist == 0)  
        cout<<root->data<<endl;  
    kDistance(root->left, dist-1);  
    kDistance(root->right, dist-1);  
}
```

^ | v • Reply • Share ›



abhishek08aug • a year ago

C++ code:

```
#include <iostream>  
#include <stdlib.h>  
using namespace std;  
  
class tree_node {  
    private:  
        int data;  
        tree_node * left;  
        tree_node * right;  
    public:
```

```

        left=NULL;
        right=NULL;
    }
    void set_data(int data) {
        this->data=data;
    }

```

[see more](#)

^ | v • Reply • Share ›



**Anonymous** • a year ago

How is the complexity  $O(n)$ .

I don't understand...I think it should be  $O(2^k)$ .

Please explain.

^ | v • Reply • Share ›



**Nirdesh Mani Sharma.** • a year ago

Above solution can be modified in order to work more efficiently by incorporating

```

if(root == NULL || k<0)
    return;

```

This is because, when  $k<0$ ; that means the  $k$ th level is already crossed. So we

^ | v • Reply • Share ›



**Ameya** • a year ago

How about this modified level order traversal as a iterative solution for the above

```

/* Modified level order traversal to print all nodes at distance k from root */

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

```



```
#define MAX_Q_SIZE 50

/* 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;
};
```

[see more](#)

^ | v • Reply • Share ›



**ABHINAV** • a year ago  
**JUST A PROTOTYPE**

```
function tree(depth,node)
{
    if(depth==k)
    {count++;return;}

    if(node->left==null && node->right==null)
    return;
    tree(depth+1,node->left);
    tree(depth+1,node->right);
}

function(0,root);
```

^ | v • Reply • Share ›



**ABHINAV** → ABHINAV • a year ago

i mean to say tree(0,root);

^ | v • Reply • Share ›



**PsychoCoder** • 2 years ago

BFS Implementation to print nodes at K-distance from the root.

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

typedef struct node {
    int data ;
    struct node *left ;
    struct node *right ;
}node;

typedef struct list {
    node *data ;
    struct list *next;
}list;
```

see more

^ | v • Reply • Share ›



**Bugaboo** • 2 years ago

You can extend the problem to find 'k' distance nodes from any node in either t

- DO a level order traversal of the binary tree and store traversal in a queue (or
- But after each level, insert a special value (could be any character) to indicate
- Traverse the queue (or stack) to first find the required element and display all current level of the required node.

^ | v • Reply • Share ›



**ashish** • 3 years ago

We can print all nodes at level k by level order traversal

^ | v • Reply • Share ›



**shek8034** → ashish • 11 months ago

Hmmmm... That's what I was thinking. Print all the nodes at k level (level)

^ | v • Reply • Share ›



**Saravanan Mani** • 3 years ago

```
void printKDistant(node *root , int k)
{
    if(root == NULL)
        return;
    if( k == 0 )
    {
        printf( "%d ", root->data );
        return ;
    }
    else if(k>0) /* No need to visit childs below Kth level*/
    {
        printKDistant( root->left, k-1 ) ;
        printKDistant( root->right, k-1 ) ;
    }
}
```

^ | v • Reply • Share ›



**anusha** → Saravanan Mani • 9 months ago

can i get c program for finding rank of a node in BST ??

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

^ | v • Reply • Share ›



**spandan** • 4 years ago

```
void PrintKNodes(struct node* root,int k){

int ht=height(root);
if(k>ht) return;

if(k==0) {
printf("%d",root->data);
return;
}

if(root->left!=NULL){
    printKnodes(root->left,k-1);
    printKnodes(root->right,k-1);
}

}
```

this checks  $k > ht$  and terminates intially.

^ | v • Reply • Share ›



**spandan** → spandan · a year ago

if(root->left!=NULL) should be replaced by else.

^ | v · Reply · Share ›



**wannabecoder** → spandan · 3 years ago

its  $O(n^2)$  in worst case...no need of calculating height

^ | v · Reply · Share ›



**geekva** → wannabecoder · 2 years ago

Put these lines at the starting of the function.

```
if(t==NULL) return 0;
if(t->left==NULL && t->right==NULL && k!=0) return;
```

^ | v · Reply · Share ›



**geekva** → geekva · 2 years ago

A minor correction : first line is

```
if(t==NULL) return;
```

^ | v · Reply · Share ›



**dev** · 4 years ago

Is it possible to print all nodes (any direction) k distance from a given node, sh

^ | v · Reply · Share ›



**Venki** → dev · 4 years ago

I think a wrapper that traces the given node and calling printKDistant() | can serve the purpose.

^ | v · Reply · Share ›



**wgpshashank** → Venki · 2 years ago

@Venki..Can You Write the code for the same ?

^ | v · Reply · Share ›



**Ashish** → Venki · 3 years ago

@venky no i think it will not work. You can give child at distance

^ | v · Reply · Share ›



**Mohit Ahuja** → Ashish · 3 years ago

but k level order traversal works :)

^ | v · Reply · Share ›



Subscribe



Add Disqus to your site

