***Print Nodes at K Distance***

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.

1

/ \

2 3

/ \ /

4 5 8

The problem can be solved using recursion. Thanks to eldho for suggesting the solution.

**Implementation:**

C++Java

// Java program to print nodes at k distance from root

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

and a pointer to right child \*/

class Node

{

int data;

Node left, right;

Node(int item)

{

data = item;

left = right = null;

}

}

class BinaryTree

{

Node root;

void printKDistant(Node node, int k)

{

if (node == null|| k < 0 )

//Base case

return;

if (k == 0)

{

System.out.print(node.data + " ");

return;

}

//recursively traversing

printKDistant(node.left, k - 1);

printKDistant(node.right, k - 1);

}

/\* Driver program to test above functions \*/

public static void main(String args[]) {

BinaryTree tree = new BinaryTree();

/\* Constructed binary tree is

1

/ \

2 3

/ \ /

4 5 8

\*/

tree.root = new Node(1);

tree.root.left = new Node(2);

tree.root.right = new Node(3);

tree.root.left.left = new Node(4);

tree.root.left.right = new Node(5);

tree.root.right.left = new Node(8);

tree.printKDistant(tree.root, 2);

}

}

// This code has been contributed by Mayank Jaiswal

**Output**

**4 5 8**

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

**Space Complexity :**O(height of the binary tree).

**Note-**

* **If it’s true print the node –** Always check the K distance == 0 at every node
* **the left or right subtree –**Decrement the distance by 1 when you are passing to its subtree