Invert a binary tree.

4

/ \

2 7

/ \ / \

1 3 6 9

to

4

/ \

7 2

/ \ / \

9 6 3 1

**Trivia:**  
This problem was inspired by [this original tweet](https://twitter.com/mxcl/status/608682016205344768) by [Max Howell](https://twitter.com/mxcl):

Google: 90% of our engineers use the software you wrote (Homebrew), but you can’t invert a binary tree on a whiteboard so fuck off.

提交版本

/\*\*

\* Definition for a binary tree node.

\* public class TreeNode {

\* int val;

\* TreeNode left;

\* TreeNode right;

\* TreeNode(int x) { val = x; }

\* }

\*/

public class Solution {

public TreeNode invertTree(TreeNode root) {

if (root == null) return root;

else if (root.left == null && root.right == null) return root;

else {

TreeNode temp = invertTree(root.left);

root.left = invertTree(root.right);

root.right = temp;

return root;

}

}

public static void main (String ags[]){

}

}

自带测试版本

**public** **class** InvertBinaryTree {

**public** TreeNode invertTree(TreeNode root) {

**if** (root == **null**) **return** root;

**else** **if** (root.left == **null** && root.right == **null**) **return** root;

**else** {

TreeNode temp = invertTree(root.left);

root.left = invertTree(root.right);

root.right = temp;

**return** root;

}

}

**public** **static** **void** main (String ags[]){

InvertBinaryTree test =**new** InvertBinaryTree();

TreeNode a = **new** TreeNode(5);

TreeNode al = **new** TreeNode(2);

TreeNode ar = **new** TreeNode(3);

TreeNode all = **new** TreeNode(4);

TreeNode alr = **new** TreeNode(7);

TreeNode arl = **new** TreeNode(8);

TreeNode arr = **new** TreeNode(9);

a.left = al;

a.right = ar;

al.left = all;

al.right = alr;

ar.left = arl;

ar.right = arr;

System.***out***.print(a.val);

System.***out***.print(a.left.val);

System.***out***.print(a.right.val);

System.***out***.print(al.left.val);

System.***out***.print(al.right.val);

System.***out***.print(ar.left.val);

System.***out***.print(ar.right.val);

test.invertTree(a);

System.***out***.println("");

System.***out***.print(a.val);

System.***out***.print(a.left.val);

System.***out***.print(a.right.val);

System.***out***.print(al.left.val);

System.***out***.print(al.right.val);

System.***out***.print(ar.left.val);

System.***out***.print(ar.right.val);

}

}