00. Same Tree

Question

Total Accepted: **101592** Total Submissions: **239636** Difficulty: **Easy**

Given two binary trees, write a function to check if they are equal or not.

Two binary trees are considered equal if they are structurally identical and the nodes have the same value.

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 boolean isSameTree(TreeNode p, TreeNode q) {

if(p == null&& q == null) return true;

else if(p == null|| q == null) return false;

else if(p.val == q.val){

if(p.left == null&&p.right == null&&q.left == null && q.right == null) return true;

else {

if(isSameTree(p.left,q.left)&&isSameTree(p.right,q.right)) return true;

else return false;

}

}

return false;

}

}

带测试版本

**class** TreeNode {

**int** val;

TreeNode left;

TreeNode right;

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

}

**public** **class** SameTree {

**public** **boolean** isSameTree(TreeNode p, TreeNode q) {

**if**(p == **null**&& q == **null**) **return** **true**;

**else** **if**(p == **null**|| q == **null**) **return** **false**;

**else** **if**(p.val == q.val){

**if**(p.left == **null**&&p.right == **null**&&q.left == **null** && q.right == **null**) **return** **true**;

**else** {

**if**(isSameTree(p.left,q.left)&&isSameTree(p.right,q.right)) **return** **true**;

**else** **return** **false**;

}

}

**return** **false**;

}

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

SameTree st = **new** SameTree();

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

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

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

a.left = al;

a.right = ar;

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

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

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

b.left = bl;

b.right = br;

System.***out***.println(st.isSameTree(a,b));

System.***out***.println(st.isSameTree(al,b));

System.***out***.println(st.isSameTree(a,bl));

System.***out***.println(st.isSameTree(al,bl));

}}