

给定两个二叉树，编写一个函数来检验它们是否相同。

如果两个树在结构上相同，并且节点具有相同的值，则认为它们是相同的。

示例 1:

输入:

```
      1      1
     / \    / \
    2   3   2   3

    [1,2,3], [1,2,3]
```

输出: true

示例 2:

输入:

```
      1      1
     /      \
    2         2

    [1,2], [1,null,2]
```

输出: false

示例 3:

输入:

```
      1      1
     / \    / \
    2   1   1   2

    [1,2,1], [1,1,2]
```

输出: false

左树右树同时遍历，值相等，则返回true，值不等，则返回false，需注意处理的情况为左右树都为NULL，左右树有一个为NULL的情况。

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    bool dfs(TreeNode* p, TreeNode* q){
        if(p->val!=q->val){
            return false;
```

```

    }
    if(p->left!=NULL&&q->left!=NULL) {
        if(!dfs(p->left,q->left)){
            return false;
        }
    }else if(!(p->left==NULL&&q->left==NULL)){
        return false;
    }
    if(p->right!=NULL&&q->right!=NULL) {
        if(!dfs(p->right,q->right)){
            return false;
        }
    }else if(!(p->right==NULL&&q->right==NULL)){
        return false;
    }
    return true;
}

bool isSameTree(TreeNode* p, TreeNode* q) {
    if(p!=NULL&&q!=NULL) {
        return dfs(p,q);
    }else if(p==NULL&&q==NULL) {
        return true;
    }else{
        return false;
    }
}

};

```