100. Same Tree

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

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

Example 1:

```
Input:     1         1
          / \       / \
         2   3     2   3

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

Output: true
```

Example 2:

```
Input:     1         1
          /           \
         2             2

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

Output: false
```

Example 3:

```
Input:     1         1
          / \       / \
         2   1     1   2

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

Output: false
```

【思路】

递归。两棵树按相同的方式遍历，遍历过程中比较树节点，不相同就返回false。

```
/**
 * 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 isSameTree(TreeNode* p, TreeNode* q) {
       if(p==NULL && q==NULL) return true;
       if(p==NULL && q!=NULL || p!=NULL && q==NULL) return false;
       if(p->val != q->val) return false;
       if(!isSameTree(p->left, q->left)) return false;
       if(!isSameTree(p->right, q->right)) return false;
       return true;
    }
};
```