Problem 100: Same Tree

Problem Information

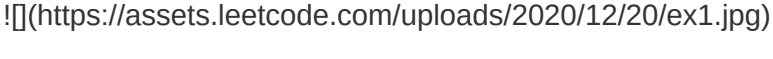
Difficulty: Easy
Acceptance Rate: 66.13%
Paid Only: No
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

Given the roots of two binary trees `p` and `q`, write a function to check if they are the same or not.

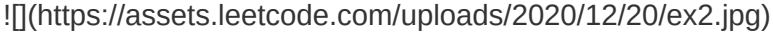
Two binary trees are considered the same if they are structurally identical, and the nodes have the same value.

Example 1:



Input: p = [1,2,3], q = [1,2,3] **Output:** true

Example 2:



Input: p = [1,2], q = [1,null,2] **Output:** false

Example 3:

Input: p = [1,2,1], q = [1,1,2] **Output:** false

Constraints:

* The number of nodes in both trees is in the range `[0, 100]`. * `-104 <= Node.val <= 104`

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
bool isSameTree(TreeNode* p, TreeNode* q) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
* this.left = left;
* this.right = right;
* }
* }
```

```
*/
class Solution {
public boolean isSameTree(TreeNode p, TreeNode q) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def isSameTree(self, p: Optional[TreeNode], q: Optional[TreeNode]) -> bool:
```