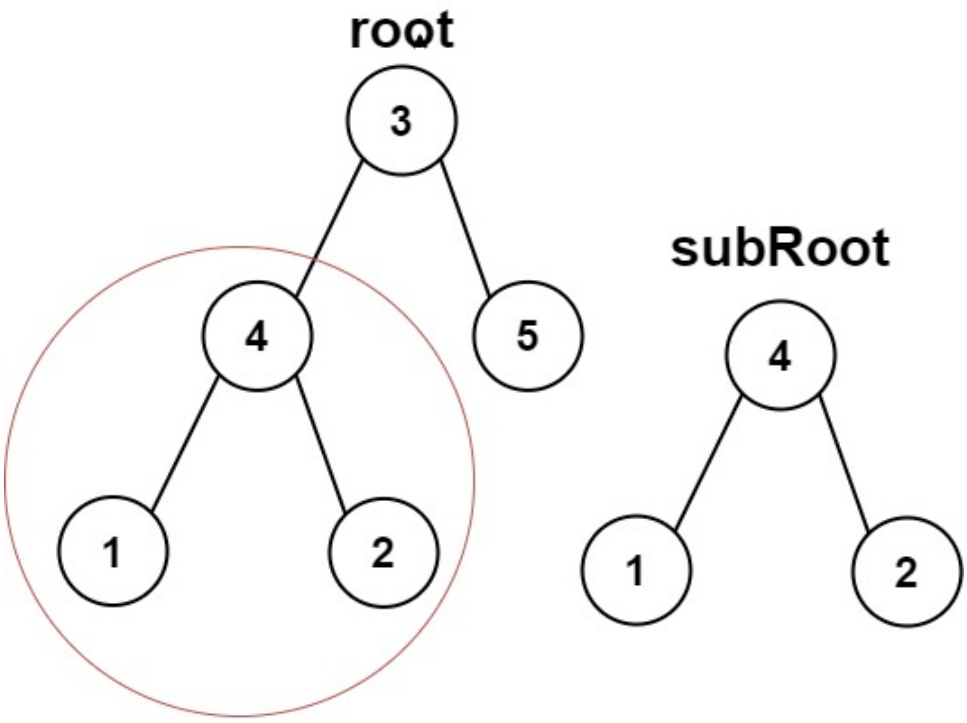
572. Subtree of Another Tree

Created	@November 26, 2022 1:22 PM
Difficulty	Easy
LC Url	https://leetcode.com/problems/subtree-of-another-tree/
Importance	
Tag	NEET Tree
Video	

Given the roots of two binary trees `root` and `subRoot`, return `true` if there is a subtree of `root` with the same structure and node values of `subRoot` and `false` otherwise.

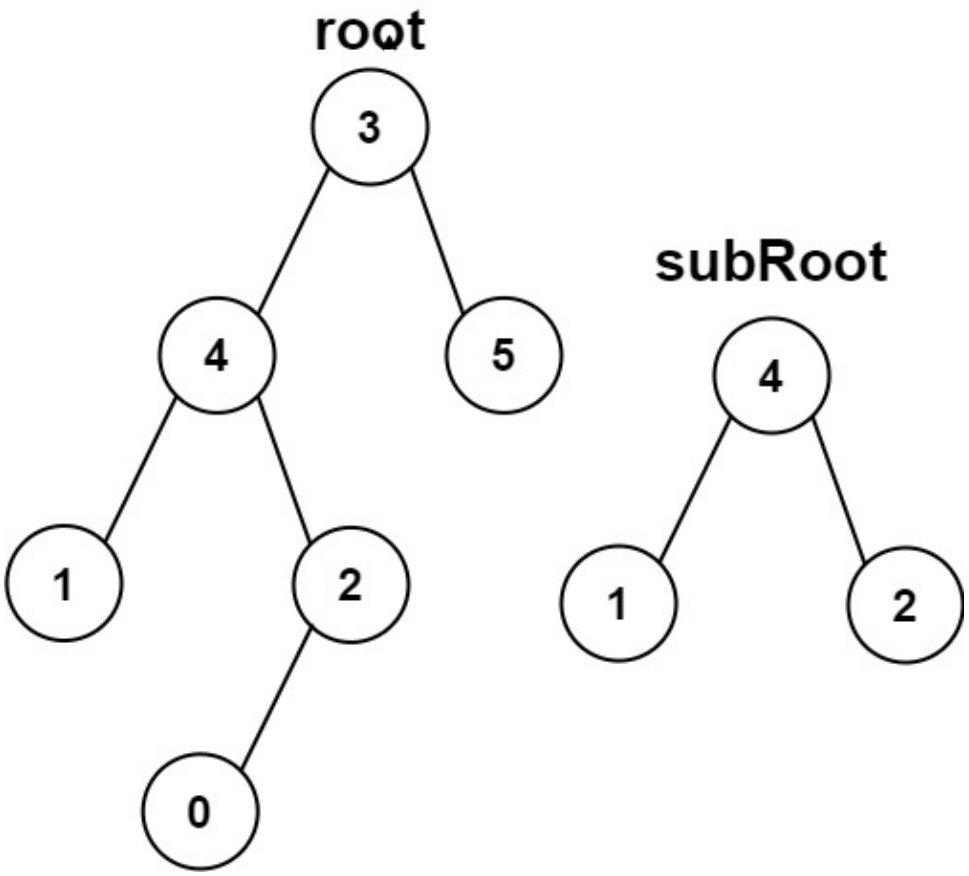
A subtree of a binary tree `tree` is a tree that consists of a node in `tree` and all of this node's descendants. The tree `tree` could also be considered as a subtree of itself.

Example 1:



```
Input: root = [3,4,5,1,2], subRoot = [4,1,2]
Output: true
```

Example 2:




```
Input: root = [3,4,5,1,2,null,null,null,null,0], subRoot = [4,1,2]
Output: false
```

Constraints:

- The number of nodes in the `root` tree is in the range `[1, 2000]`.
- The number of nodes in the `subRoot` tree is in the range `[1, 1000]`.
- `10 4 <= root.val <= 10 4`
- `10 4 <= subRoot.val <= 10 4`

Solution

前文 手把手刷二叉树总结篇 说过二叉树的递归分为「遍历」和「分解问题」两种思维模式，这道题需要用到「遍历」的思维。

遍历以 `root` 为根的这棵二叉树的所有节点，用 100. 相同的树 中的 `isSame` 函数判断以该节点为根的子树是否和以 `subRoot` 为根的那棵树相同。

```
# 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 isSubtree(self, root: Optional[TreeNode], subRoot: Optional[TreeNode]) -> bool:
        if not root:
            return subRoot == None

        # 判断以root为根的二叉树是否与subRoot相同
        if self.isSameTree(root, subRoot):
            return True

        # 去左右子树中判断是否有和subRoot相同的子树
        return self.isSubtree(root.left, subRoot) or self.isSubtree(root.right, subRoot)

    def isSameTree(self, p, q):
        if not p and not q:
            return True

        if not p or not q:
            return False

        if p.val != q.val:
            return False

        return self.isSameTree(p.left, q.left) and self.isSameTree(p.right, q.right)
```