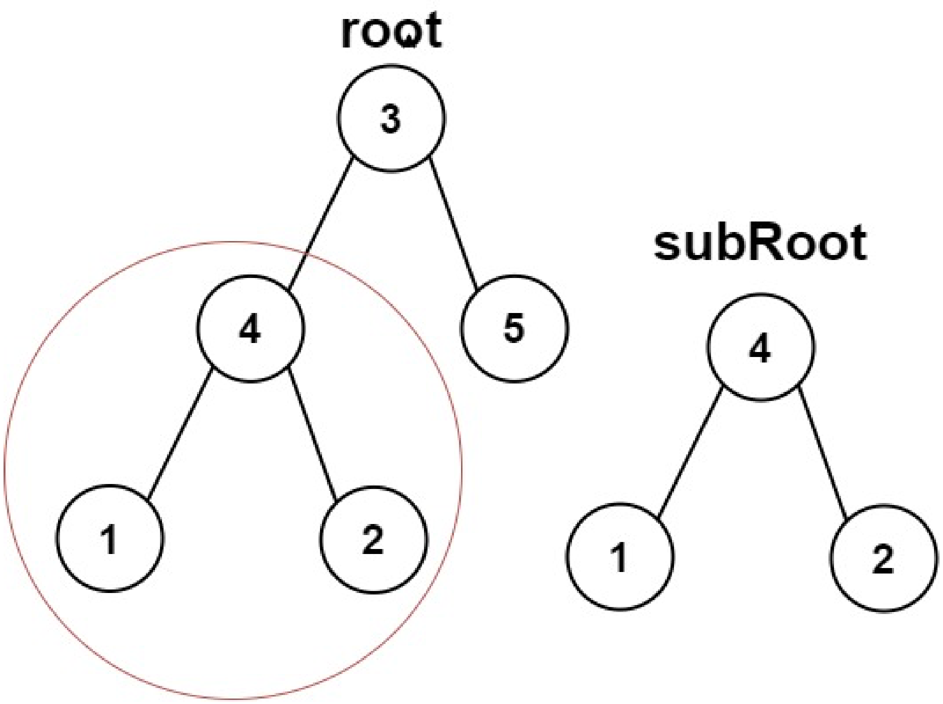
Subtree of Another Tree

Difficulty	Easy
Category	Tree
Question	https://leetcode.com/problems/subtree-of-another-tree/
Solution	https://www.youtube.com/watch?v=E36O5SWp-LE
Status	Done

Question

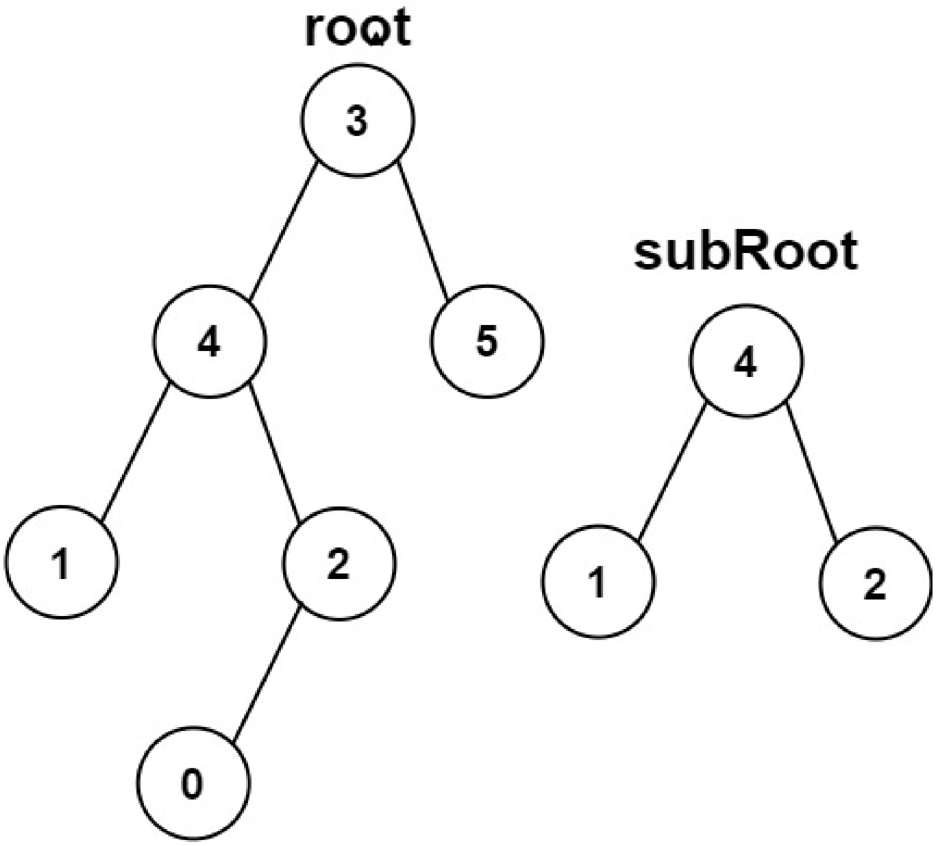
> Given the roots of two binary trees `root` and `subRoot`, return `true` if there is a subtree of `root`with the same structure and node values of `subRoot` and `false` otherwise.
>
> A subtree of a binary tree `tree` is a tree that consists of a node in `tree` and all of this node's descendants. The tree `tree` could also be considered as a subtree of itself.

Example



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
```

Idea

💡 To check if tree `t` is a subtree of tree `s`, recursively compare each subtree of `s` with `t`, and use a helper function to check if two trees are identical.

Solution

```
# 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, s: TreeNode, t: TreeNode) -> bool:
        # If tree t is empty, it's considered a subtree of any tree.
        if not t:
            return True
        # If tree s is empty but t is not, t cannot be a subtree.
        if not s:
            return False

        # Check if the current subtree in s is the same as t.
        if self.sameTree(s, t):
            return True

        # Recursively check the left and right subtrees of s.
        return self.isSubtree(s.left, t) or self.isSubtree(s.right, t)

    def sameTree(self, s, t):
        # Base case: If both trees are empty, they are the same.
        if not s and not t:
            return True
        # If both trees are not empty and have the same value, recursively check left and right subtrees.
        if s and t and s.val == t.val:
            return self.sameTree(s.left, t.left) and self.sameTree(s.right, t.right)
        # If any of the above conditions is not met, the trees are not the same.
        return False
```

Explanation