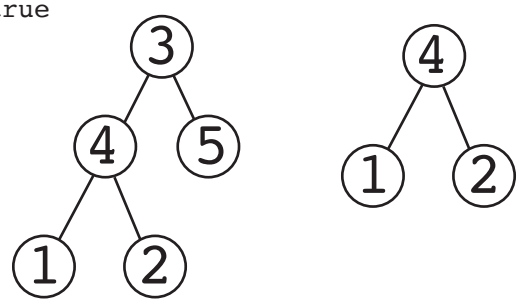
572. Subtree of Another Tree
Easy

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.

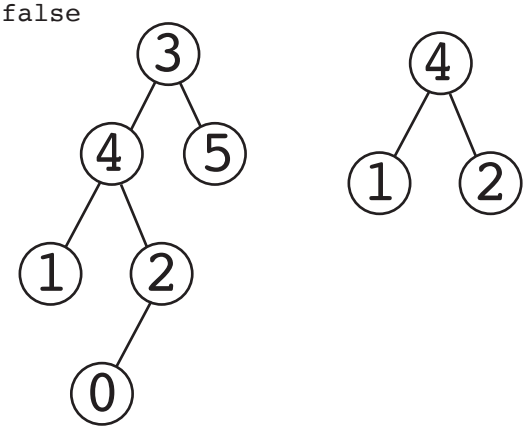
Example 1

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



True: a subtree exists
False: no subtree

```python
class Solution:
    def isSubtree(self, root: Optional[TreeNode], subRoot: Optional[TreeNode]) -> bool:
        if not subRoot:
            return False
        if not root:
            return False

        if self.sametree(root, subRoot):
            return True
        return self.isSubtree(root.left, subRoot) or self.isSubtree(root.right, subRoot)


    def sametree(self, root, subRoot):
        if not root and not subRoot:
            return True
        if root and subRoot and root.val == subRoot.val:
            return self.sametree(root.left, subRoot.left) and self.sametree(root.right, subRoot.right)

        return False
```