Subtree of Another Tree

```
/**
 * 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 isSubtree(TreeNode* root, TreeNode* subRoot, int check = 0) {

      if (root == nullptr && subRoot == nullptr) return true;
      if (root == nullptr || subRoot == nullptr) return false;

      if (root->right == nullptr && root->left == nullptr){
        if (subRoot->right == nullptr && subRoot->left == nullptr){
          printf("reached\n");
          return root->val == subRoot->val;
        }
      }

      bool awn = false;
      bool awn1 = false;
      bool awn2 = false;
      bool awn3 = false;

      if (root->val == subRoot->val){
        awn = isSubtree(root->left,subRoot->left,1);
        awn1 = isSubtree(root->right,subRoot->right,1);
      }

      if (!awn && check == 0) awn2 = isSubtree(root->left,subRoot);
      if (!awn1 && check == 0) awn3 = isSubtree(root->right,subRoot);

      return (awn && awn1) || (awn2 || awn3);

    }
};
```