

# Flip Equivalent Binary Trees

🕒 solved by	Senan
🌐 Platform	LeetCode
🔧 difficulty	Medium
# Serial	951
≡ tags	DFS Tree
🗨 language	C++
📅 solved on	@24/10/2024
🔗 link	<a href="https://leetcode.com/problems/flip-equivalent-binary-trees/description/">https://leetcode.com/problems/flip-equivalent-binary-trees/description/</a>
☑ Completion	✓

## Intuition

The problem asks if two binary trees are *flip equivalent*, meaning one tree can be transformed into the other by flipping some of its subtrees. A tree's subtree can be flipped by swapping its left and right children. The intuition is to recursively check if two trees are either structurally identical or identical after flipping their subtrees.

## Approach

1. Recursively compare both trees from the root.
2. If both nodes are `NULL`, they are trivially equivalent, so return `true`.
3. If one node is `NULL` and the other is not, return `false` since they can't be equivalent.
4. If the values of the current nodes differ, return `false` because the trees can't be equivalent.
5. Recursively check if the subtrees are either equivalent without any flip or equivalent after flipping the left and right subtrees.
6. Return `true` if either condition holds.

## Complexity

### Time Complexity:

The time complexity is  $O(n)$ , where `n` is the number of nodes in the tree. Each node is visited once, and we perform constant work at each node.

### Space Complexity:

The space complexity is  $O(h)$ , where `h` is the height of the tree. This is due to the recursion stack, and in the worst case (a completely unbalanced tree), `h = O(n)`.

## Code

```
class Solution {
public:
    bool flipEquiv(TreeNode* root1, TreeNode* root2) {
        if(root1 == NULL && root2 == NULL) return true;
```

```
    if(root1 == NULL || root2 == NULL) return false;
    if(root1->val != root2->val) return false;

    return (flipEquiv(root1->left, root2->left) && flipEquiv(root1->right, root2->
right))
        || (flipEquiv(root1->left, root2->right) && flipEquiv(root1->right, root2-
>left));
    }
};
```