AlgoExpert   Quad Layout   Swift   12px   Sublime   Monokai   00:00:00

Prompt   Scratchpad   Our Solution(s)   Video Explanation   Run Code

Solution 1

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    class BinaryTree {
        var value: Int
        var left: BinaryTree?
        var right: BinaryTree?

        init(value: Int) {
            self.value = value
            left = nil
            right = nil
        }
    }

    // O(n) time | O(d) space - where n is the number of nodes in
    // the Binary Tree and d is the depth (height) of the Binary Tree
    func rightSiblingTree(root: BinaryTree) -> BinaryTree {
        mutate(node: root, parent: nil, isLeftChild: false)
        return root
    }

    func mutate(node: BinaryTree?, parent: BinaryTree?, isLeftChild: Bool) {
        if let tree = node {
            var left = tree.left
            var right = tree.right
            mutate(node: left, parent: tree, isLeftChild: true)
            if let p = parent {
                if isLeftChild {
                    tree.right = p.right
                } else {
                    if let right = p.right {
                        tree.right = right.left
                    } else {
                        tree.right = nil
                    }
                }
            } else {
                tree.right = nil
            }

            mutate(node: right, parent: tree, isLeftChild: false)
        }
    }
}
```