Prompt | Scratchpad | Our Solution(s) | Video Explanation | Run Code

Solution 1

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

class Program {
    class AncestralTree {
        var name = String()
        var ancestor: AncestralTree?

        init(name: String) {
            self.name = name
            ancestor = nil
        }
    }

    // O(d) time | O(1) space
    func getYoungestCommonAncestor(_ topAncestor: AncestralTree?, _ descendantOne: inout An
        var firstDescendant = descendantOne
        var secondDescendant = descendantTwo

        let depthOne = getDescendantDepth(&descendantOne, topAncestor)
        let depthTwo = getDescendantDepth(&descendantTwo, topAncestor)

        if depthOne > depthTwo {
            var difference = depthOne - depthTwo
            return backTrackAncestralTree(&firstDescendant, &secondDescendant, &difference)
        } else {
            var difference = depthTwo - depthOne
            return backTrackAncestralTree(&secondDescendant, &firstDescendant, &difference)
        }
    }

    func getDescendantDepth(_ descendant: inout AncestralTree?, _ topAncestor: AncestralTre
        var depth = 0

        while descendant !== topAncestor {
            depth += 1
            descendant = descendant?.ancestor
        }

        return depth
    }

    func backTrackAncestralTree(_ lowerDescendant: inout AncestralTree?, _ higherDescendant
        while difference > 0 {
            difference -= 1
            lowerDescendant = lowerDescendant?.ancestor
        }

        while lowerDescendant !== higherDescendant {
            lowerDescendant = lowerDescendant?.ancestor
            higherDescendant = higherDescendant?.ancestor
        }

        return lowerDescendant!
    }
}

```

Your Solutions | Run Code

Solution 1 | Solution 2 | Solution 3

```swift
class Program {
    // This is an input class. Do not edit.
    class AncestralTree {
        var name = String()
        var ancestor: AncestralTree?

        init(name: String) {
            self.name = name
            ancestor = nil
        }
    }

    func getYoungestCommonAncestor(_ topAncestor: AncestralTree?, _ descendantOne: inout An
        // Write your code here.
        return AncestralTree(name: "replace me") // replace me
    }
}

```

Custom Output | Raw Output | Submit Code

Run or submit code when you're ready.