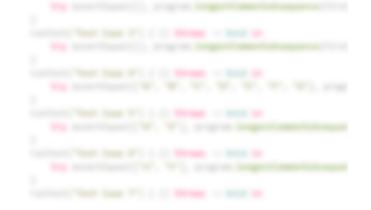**Our Solution(s)**   Run Code

Solution 1   Solution 2   Solution 3   Solution 4

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

class Program {
    // O(nm * min(n, m)) time | O(nm * min(n, m)) space
    func longestCommonSubsequence(firstString: String, secondString:
        var lcs = [[[String]]]()

        for _ in 0 ..< firstString.count + 1 {
            let row = Array(repeating: [String](), count: secondStrin
            lcs.append(row)
        }

        for i in stride(from: 1, to: firstString.count + 1, by: 1) {
            for j in stride(from: 1, to: secondString.count + 1, by:
                let firstIndex = firstString.index(firstString.startI
                let secondIndex = secondString.index(secondString.star

                if firstString[firstIndex] == secondString[secondIndex
                    var diagonal = lcs[i - 1][j - 1]
                    let char = String(firstString[firstIndex])
                    diagonal.append(char)

                    lcs[i][j] = diagonal
                } else {
                    let left = lcs[i][j - 1]
                    let top = lcs[i - 1][j]

                    lcs[i][j] = left.count > top.count ? left : top
                }
            }
        }

        return lcs[firstString.count][secondString.count]
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```swift
class Program {
    func longestCommonSubsequence(firstString: String, secondString: S
        // Write your code here.
        return []
    }
}
```

Our Tests

Custom Output   Submit Code

Run or submit code when you're ready.