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

Prompt　Scratchpad　Our Solution(s)　Video Explanation　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: String) -> [String] {
        var lcs = [[[String]]]()

        for _ in 0 ..< firstString.count + 1 {
            let row = Array(repeating: [String](), count: secondString.count + 1)
            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: 1) {
                let firstIndex = firstString.index(firstString.startIndex, offsetBy: i - 1)
                let secondIndex = secondString.index(secondString.startIndex, offsetBy: j - 1)

                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]
    }
}
```