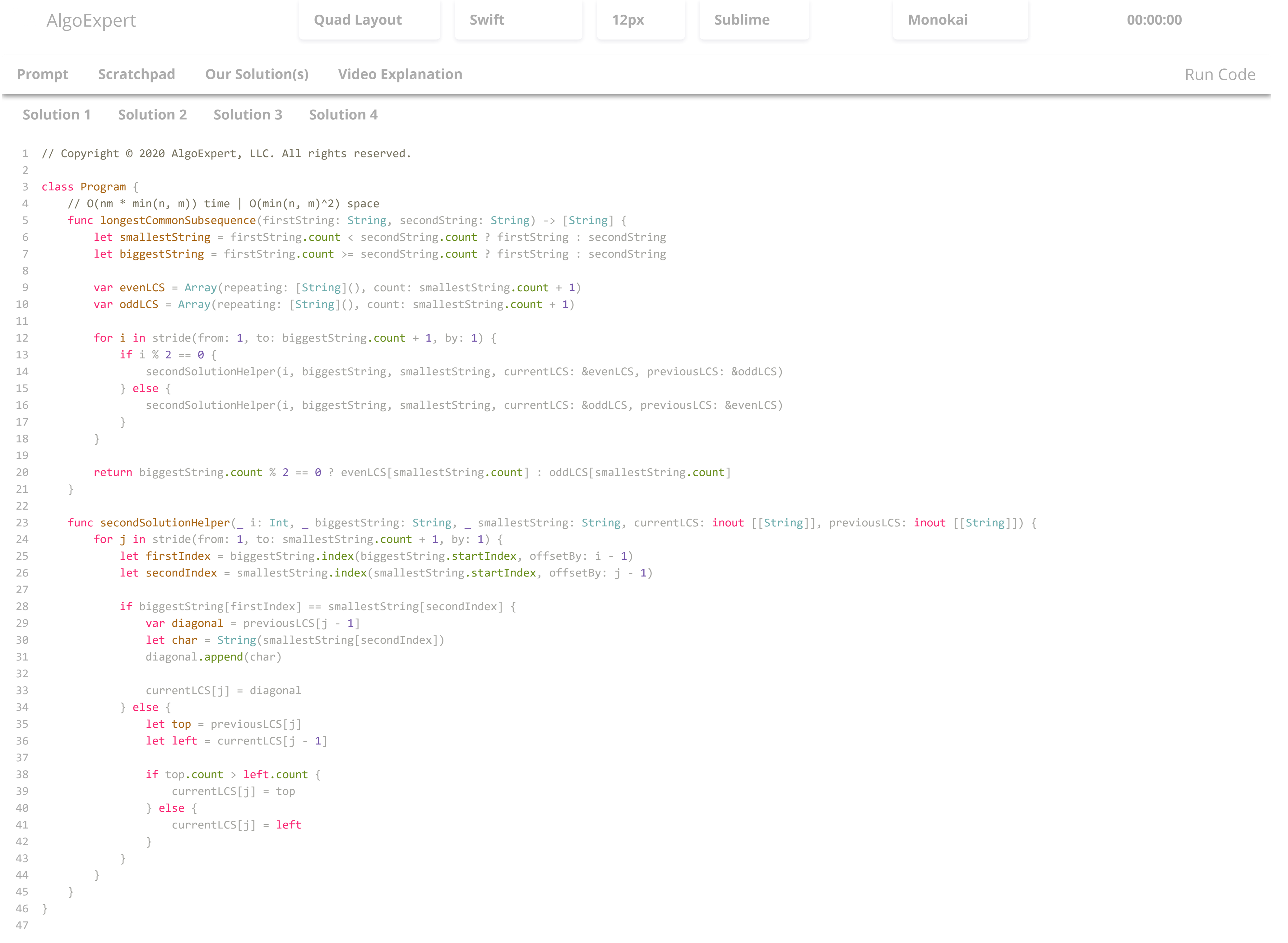

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    // O(nm * min(n, m)) time | O(min(n, m)^2) space
    func longestCommonSubsequence(firstString: String, secondString: String) -> [String] {
        let smallestString = firstString.count < secondString.count ? firstString : secondString
        let biggestString = firstString.count >= secondString.count ? firstString : secondString

        var evenLCS = Array(repeating: [String](), count: smallestString.count + 1)
        var oddLCS = Array(repeating: [String](), count: smallestString.count + 1)

        for i in stride(from: 1, to: biggestString.count + 1, by: 1) {
            if i % 2 == 0 {
                secondSolutionHelper(i, biggestString, smallestString, currentLCS: &evenLCS, previousLCS: &oddLCS)
            } else {
                secondSolutionHelper(i, biggestString, smallestString, currentLCS: &oddLCS, previousLCS: &evenLCS)
            }
        }

        return biggestString.count % 2 == 0 ? evenLCS[smallestString.count] : oddLCS[smallestString.count]
    }

    func secondSolutionHelper(_ i: Int, _ biggestString: String, _ smallestString: String, currentLCS: inout [[String]], previousLCS: inout [[String]]) {
        for j in stride(from: 1, to: smallestString.count + 1, by: 1) {
            let firstIndex = biggestString.index(biggestString.startIndex, offsetBy: i - 1)
            let secondIndex = smallestString.index(smallestString.startIndex, offsetBy: j - 1)

            if biggestString[firstIndex] == smallestString[secondIndex] {
                var diagonal = previousLCS[j - 1]
                let char = String(smallestString[secondIndex])
                diagonal.append(char)

                currentLCS[j] = diagonal
            } else {
                let top = previousLCS[j]
                let left = currentLCS[j - 1]

                if top.count > left.count {
                    currentLCS[j] = top
                } else {
                    currentLCS[j] = left
                }
            }
        }
    }
}

```