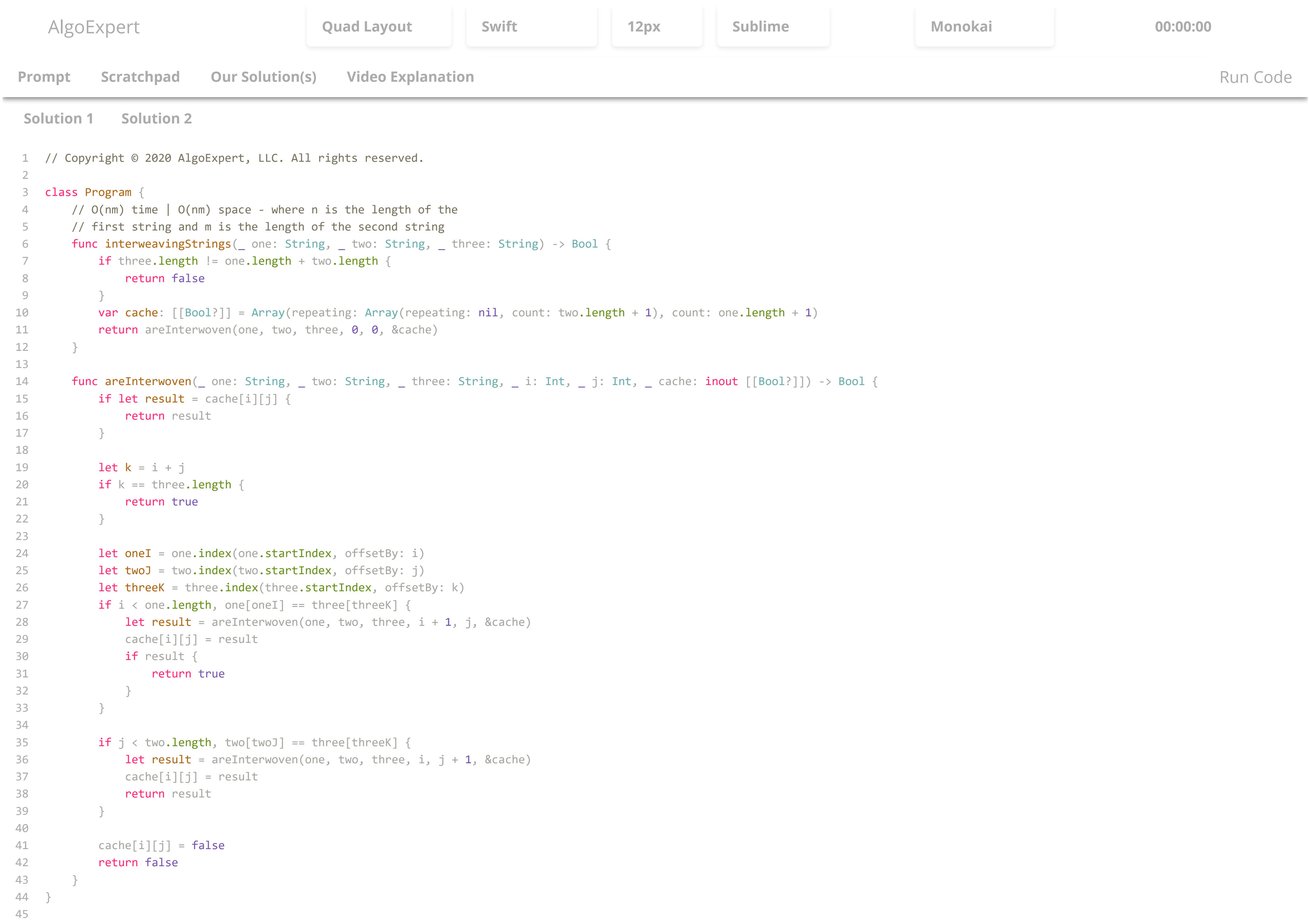

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    // O(nm) time | O(nm) space - where n is the length of the
    // first string and m is the length of the second string
    func interweavingStrings(_ one: String, _ two: String, _ three: String) -> Bool {
        if three.length != one.length + two.length {
            return false
        }
        var cache: [[Bool?]] = Array(repeating: Array(repeating: nil, count: two.length + 1), count: one.length + 1)
        return areInterwoven(one, two, three, 0, 0, &cache)
    }

    func areInterwoven(_ one: String, _ two: String, _ three: String, _ i: Int, _ j: Int, _ cache: inout [[Bool?]]) -> Bool {
        if let result = cache[i][j] {
            return result
        }

        let k = i + j
        if k == three.length {
            return true
        }

        let oneI = one.index(one.startIndex, offsetBy: i)
        let twoJ = two.index(two.startIndex, offsetBy: j)
        let threeK = three.index(three.startIndex, offsetBy: k)
        if i < one.length, one[oneI] == three[threeK] {
            let result = areInterwoven(one, two, three, i + 1, j, &cache)
            cache[i][j] = result
            if result {
                return true
            }
        }

        if j < two.length, two[twoJ] == three[threeK] {
            let result = areInterwoven(one, two, three, i, j + 1, &cache)
            cache[i][j] = result
            return result
        }

        cache[i][j] = false
        return false
    }
}
```