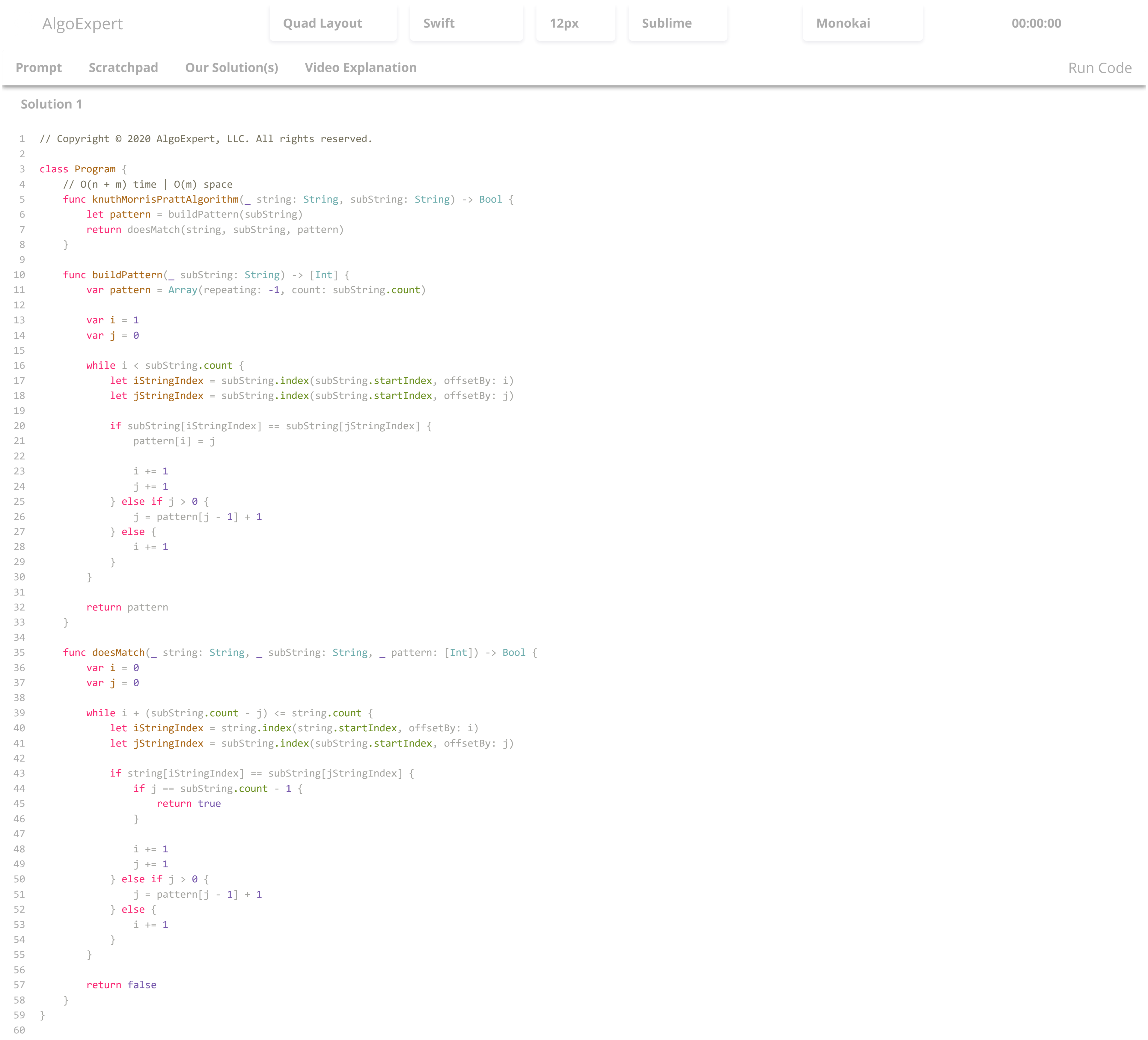

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    // O(n + m) time | O(m) space
    func knuthMorrisPrattAlgorithm(_ string: String, subString: String) -> Bool {
        let pattern = buildPattern(subString)
        return doesMatch(string, subString, pattern)
    }

    func buildPattern(_ subString: String) -> [Int] {
        var pattern = Array(repeating: -1, count: subString.count)

        var i = 1
        var j = 0

        while i < subString.count {
            let iStringIndex = subString.index(subString.startIndex, offsetBy: i)
            let jStringIndex = subString.index(subString.startIndex, offsetBy: j)

            if subString[iStringIndex] == subString[jStringIndex] {
                pattern[i] = j

                i += 1
                j += 1
            } else if j > 0 {
                j = pattern[j - 1] + 1
            } else {
                i += 1
            }
        }

        return pattern
    }

    func doesMatch(_ string: String, _ subString: String, _ pattern: [Int]) -> Bool {
        var i = 0
        var j = 0

        while i + (subString.count - j) <= string.count {
            let iStringIndex = string.index(string.startIndex, offsetBy: i)
            let jStringIndex = subString.index(subString.startIndex, offsetBy: j)

            if string[iStringIndex] == subString[jStringIndex] {
                if j == subString.count - 1 {
                    return true
                }

                i += 1
                j += 1
            } else if j > 0 {
                j = pattern[j - 1] + 1
            } else {
                i += 1
            }
        }

        return false
    }
}
```