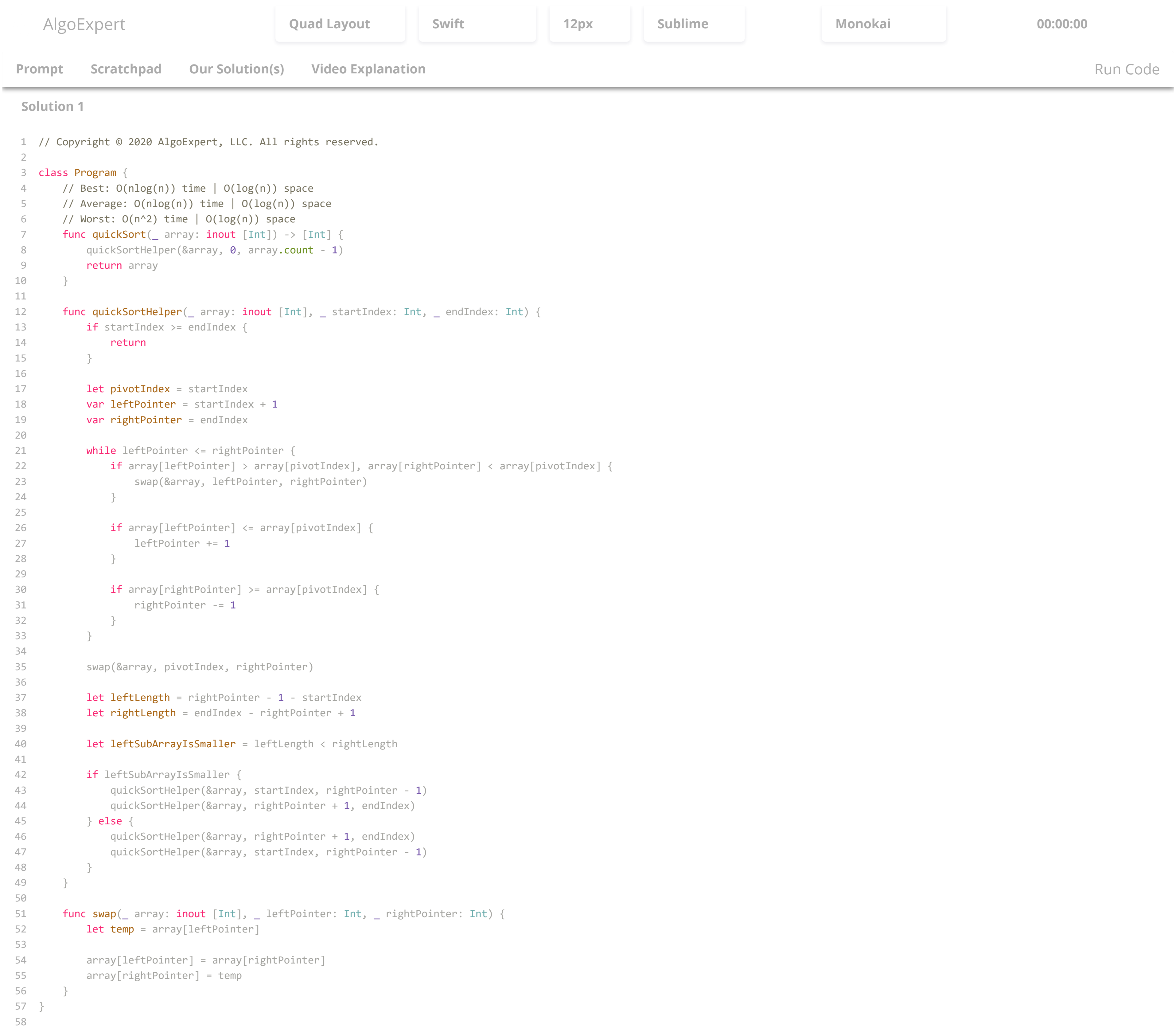

Solution 1

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

class Program {
    // Best: O(nlog(n)) time | O(log(n)) space
    // Average: O(nlog(n)) time | O(log(n)) space
    // Worst: O(n^2) time | O(log(n)) space
    func quickSort(_ array: inout [Int]) -> [Int] {
        quickSortHelper(&array, 0, array.count - 1)
        return array
    }

    func quickSortHelper(_ array: inout [Int], _ startIndex: Int, _ endIndex: Int) {
        if startIndex >= endIndex {
            return
        }

        let pivotIndex = startIndex
        var leftPointer = startIndex + 1
        var rightPointer = endIndex

        while leftPointer <= rightPointer {
            if array[leftPointer] > array[pivotIndex], array[rightPointer] < array[pivotIndex] {
                swap(&array, leftPointer, rightPointer)
            }

            if array[leftPointer] <= array[pivotIndex] {
                leftPointer += 1
            }

            if array[rightPointer] >= array[pivotIndex] {
                rightPointer -= 1
            }
        }

        swap(&array, pivotIndex, rightPointer)

        let leftLength = rightPointer - 1 - startIndex
        let rightLength = endIndex - rightPointer + 1

        let leftSubArrayIsSmaller = leftLength < rightLength

        if leftSubArrayIsSmaller {
            quickSortHelper(&array, startIndex, rightPointer - 1)
            quickSortHelper(&array, rightPointer + 1, endIndex)
        } else {
            quickSortHelper(&array, rightPointer + 1, endIndex)
            quickSortHelper(&array, startIndex, rightPointer - 1)
        }
    }

    func swap(_ array: inout [Int], _ leftPointer: Int, _ rightPointer: Int) {
        let temp = array[leftPointer]

        array[leftPointer] = array[rightPointer]
        array[rightPointer] = temp
    }
}
```