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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1

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

class Program {
    // O(n^2) time | O(n) space
    func maximumSumIncreasingSubsequence(array: [Int]) -> (Int, [Int]) {
        var maxSumIndex = 0
        var sums = array.map { $0 }
        var previousIndices: [Int?] = Array(repeating: nil, count: array.count)

        for i in 0 ..< array.count {
            let currentNumber = array[i]
            for j in 0 ..< i {
                let previousNumber = array[j]
                if previousNumber < currentNumber, sums[j] + currentNumber > sums[i] {
                    sums[i] = sums[j] + currentNumber
                    previousIndices[i] = j
                }
            }

            if sums[i] > sums[maxSumIndex] {
                maxSumIndex = i
            }
        }

        return (sums[maxSumIndex], buildSequence(array, maxSumIndex, previousIndices))
    }

    func buildSequence(_ array: [Int], _ maxSumIndex: Int, _ previousIndices: [Int?]) -> [Int] {
        var sequence = [Int]()
        var currentIndex: Int? = maxSumIndex

        while currentIndex != nil {
            sequence.insert(array[currentIndex!], at: 0)
            currentIndex = previousIndices[currentIndex!]
        }

        return sequence
    }
}
```