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

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2

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

class Program {
    // O(n^3) time | O(n^2) space
    func palindromePartitioingMinCuts(_ string: String) -> Int {
        var palindromes = string.map { _ in Array(repeating: false, count: string.count) }

        for i in 0 ..< string.count {
            for j in i ..< string.count {
                let leftIndex = string.index(string.startIndex, offsetBy: i)
                let rightIndex = string.index(string.startIndex, offsetBy: j)
                let subString = String(string[leftIndex ... rightIndex])

                palindromes[i][j] = isPalindrome(subString)
            }
        }

        var cuts = Array(repeating: Int.max, count: string.count)

        for i in 0 ..< string.count {
            if palindromes[0][i] {
                cuts[i] = 0
            } else {
                cuts[i] = cuts[i - 1] + 1

                for j in 1 ..< i {
                    if palindromes[j][i], cuts[j - 1] + 1 < cuts[i] {
                        cuts[i] = cuts[j - 1] + 1
                    }
                }
            }
        }

        return cuts[string.count - 1]
    }

    func isPalindrome(_ string: String) -> Bool {
        var leftIndex = 0
        var rightIndex = string.count - 1

        while leftIndex < rightIndex {
            let leftStringIndex = string.index(string.startIndex, offsetBy: leftIndex)
            let rightStringIndex = string.index(string.startIndex, offsetBy: rightIndex)

            if string[leftStringIndex] != string[rightStringIndex] {
                return false
            }

            leftIndex += 1
            rightIndex -= 1
        }

        return true
    }
}

```