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

class Program {
    // O(n^2) 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 {
            palindromes[i][i] = true
        }

        for length in stride(from: 2, through: string.count, by: 1) {
            for i in stride(from: 0, through: string.count - length, by: 1) {
                let j = i + length - 1

                let iStringIndex = string.index(string.startIndex, offsetBy: i)
                let jStringIndex = string.index(string.startIndex, offsetBy: j)

                if length == 2, string[iStringIndex] == string[jStringIndex] {
                    palindromes[i][j] = true
                } else if palindromes[i + 1][j - 1], string[iStringIndex] == string[jStringIndex] {
                    palindromes[i][j] = true
                }
            }
        }

        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]
    }
}

```