Solution 1

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

class Program {
    // O(n + m) time | O(n) space
    func underscorifySubstring(_ string: String, _ substring: String) -> String {
        let locations = getLocations(string, substring)
        let collapsedLocations = collapse(locations)

        return underscorify(string, collapsedLocations)
    }

    func getLocations(_ string: String, _ substring: String) -> [[String.Index]] {
        var locations = [[String.Index]]()

        var start = 0
        var startIndex = string.index(string.startIndex, offsetBy: start)

        while start < string.count {
            if let rangeOfSubstring = string.range(of: substring, options: [], range: startIndex ..< string.endIndex, locale: nil) {
                locations.append([rangeOfSubstring.lowerBound, rangeOfSubstring.upperBound])

                let startPos = string.distance(from: string.startIndex, to: rangeOfSubstring.lowerBound)
                start = startPos + 1
                startIndex = string.index(string.startIndex, offsetBy: start)
            } else {
                break
            }
        }

        return locations
    }

    func collapse(_ locations: [[String.Index]]) -> [[String.Index]] {
        if locations.count == 0 {
            return locations
        }

        var newLocations = [locations[0]]
        var previousLocationIndex = 0
        for i in 1 ..< locations.count {
            let currentLocation = locations[i]

            if currentLocation[0] <= newLocations[previousLocationIndex][1] {
                newLocations[previousLocationIndex][1] = currentLocation[1]
            } else {
                newLocations.append(currentLocation)
                previousLocationIndex += 1
            }
        }

        return newLocations
    }

    func underscorify(_ string: String, _ locations: [[String.Index]]) -> String {
        var subIndex = 0
        var stringIndex = 0
        var locationIndex = 0
        var isInBetweenUnderscres = false
        var currentIndex = string.index(string.startIndex, offsetBy: stringIndex)

        var finalCharacters = [Character]()

        while stringIndex < string.count, locationIndex < locations.count {
            currentIndex = string.index(string.startIndex, offsetBy: stringIndex)

            if currentIndex == locations[locationIndex][subIndex] {
                finalCharacters.append("_")

                isInBetweenUnderscres = !isInBetweenUnderscres

                if !isInBetweenUnderscres {
                    locationIndex += 1
                }

                if subIndex == 0 {
                    subIndex = 1
                } else {
                    subIndex = 0
                }
            }

            finalCharacters.append(string[currentIndex])
            stringIndex += 1
        }

        if locationIndex < locations.count {
            finalCharacters.append("_")
        } else if stringIndex < string.count {
            currentIndex = string.index(string.startIndex, offsetBy: stringIndex)
            let restOfCharacters = Array(string[currentIndex ..< string.endIndex])
            finalCharacters.append(contentsOf: restOfCharacters)
        }

        let result = finalCharacters.compactMap { String($0) }.joined()
        return result
    }
}

```