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

class Program {
    // O(bns) time | O(n) space
    func multiStringSearch(_ bigString: String, _ smallStrings: [String]) -> [Bool] {
        return smallStrings.map { isInBigString($0, bigString) }
    }

    func isInBigString(_ smallString: String, _ bigString: String) -> Bool {
        for i in 0 ..< bigString.count {
            if i + smallString.count > bigString.count {
                break
            }

            if isInBigStringHelper(i, smallString, bigString) {
                return true
            }
        }

        return false
    }

    func isInBigStringHelper(_ startIndex: Int, _ smallString: String, _ bigString: String) -> Bool {
        var leftSmallIndex = 0
        var rightSmallIndex = smallString.count - 1

        var leftBigIndex = startIndex
        var rightBigIndex = startIndex + smallString.count - 1

        while leftBigIndex < rightBigIndex {
            let leftSmallStringIndex = smallString.index(smallString.startIndex, offsetBy: leftSmallIndex)
            let rightSmallStringIndex = smallString.index(smallString.startIndex, offsetBy: rightSmallIndex)

            let leftBigStringIndex = bigString.index(bigString.startIndex, offsetBy: leftBigIndex)
            let rightBigStringIndex = bigString.index(bigString.startIndex, offsetBy: rightBigIndex)

            if smallString[leftSmallStringIndex] != bigString[leftBigStringIndex] || smallString[rightSmallStringIndex] != bigString[rightBigStringIndex] {
                return false
            }

            leftSmallIndex += 1
            rightSmallIndex -= 1
            leftBigIndex += 1
            rightBigIndex -= 1
        }

        return true
    }
}
```