Our Solution(s) Run Code

Solution 1

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

class Program {
    // O(b + s) time | O(b + s) space - where b is the length of the
    // input string and s is the length of the small input string
    func smallestSubstringContaining(_ bigString: String, _ smallStri
        let targetCharCounts = getCharCounts(smallString)
        let substringBounds = getSubstringBounds(bigString, targetChar
        return getStringFromBounds(bigString, substringBounds)
    }

    func getCharCounts(_ str: String) -> [Character: Int] {
        var charCounts = [Character: Int]()
        for char in str {
            changeCharCount(char, &charCounts, 1)
        }
        return charCounts
    }

    func changeCharCount(_ char: Character, _ charCounts: inout [Chara
        if let count = charCounts[char] {
            charCounts.updateValue(count + change, forKey: char)
            return
        }
        charCounts[char] = change
    }

    func getSubstringBounds(_ str: String, _ targetCharCounts: [Charac
        var substringBounds = [0, Int.max]
        var substringCharCounts = [Character: Int]()
        var numUniqueChars = targetCharCounts.count
        var numUniqueCharsDone = 0
        var leftIdx = 0
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```swift
class Program {
    func smallestSubstringContaining(_ bigString: String, _ smallStrin
        // Write your code here.
        return ""
    }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.