Our Solution(s) Run Code

Solution 1

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

class Program {
    // O(n) time | O(min(n, a)) space
    func longestSubstringWithoutDuplication(_ string: String) -> Stri
        var startingPointer = 0
        var indicesOfLongestSubstring = [0, 1]
        var indicesDictionary = [Character: Int]()

        for (index, character) in string.enumerated() {
            if let index = indicesDictionary[character] {
                startingPointer = max(startingPointer, index + 1)
            }

            if indicesOfLongestSubstring[1] - indicesOfLongestSubstri
                indicesOfLongestSubstring = [startingPointer, index +
            }

            indicesDictionary[character] = index
        }

        let start = indicesOfLongestSubstring[0]
        let startingIndex = string.index(string.startIndex, offsetBy:

        let end = indicesOfLongestSubstring[1]
        let endingIndex = string.index(string.startIndex, offsetBy: en

        return String(string[startingIndex ..< endingIndex])
    }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```swift
class Program {
    func longestSubstringWithoutDuplication(_ string: String) -> Strin
        // Write your code here.
        return ""
    }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.