

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n) time | O(min(n, a)) space
5     func longestSubstringWithoutDuplication(_ string: String) -> String {
6         var startingPointer = 0
7         var indicesOfLongestSubstring = [0, 1]
8         var indicesDictionary = [Character: Int]()
9
10        for (index, character) in string.enumerated() {
11            if let index = indicesDictionary[character] {
12                startingPointer = max(startingPointer, index + 1)
13            }
14
15            if indicesOfLongestSubstring[1] - indicesOfLongestSubstring[0] < index + 1 - startingPointer {
16                indicesOfLongestSubstring = [startingPointer, index + 1]
17            }
18
19            indicesDictionary[character] = index
20        }
21
22        let start = indicesOfLongestSubstring[0]
23        let startingIndex = string.index(string.startIndex, offsetBy: start)
24
25        let end = indicesOfLongestSubstring[1]
26        let endingIndex = string.index(string.startIndex, offsetBy: end)
27
28        return String(string[startingIndex ..< endingIndex])
29    }
30 }
31
```

