

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type substring struct {
6     left int
7     right int
8 }
9
10 func (ss substring) length() int { return ss.right - ss.left }
11
12 // O(n) time | O(min(n, a)) space
13 func LongestSubstringWithoutDuplication(str string) string {
14     lastSeen := map[rune]int{}
15     longest := substring{0, 1}
16     startIndex := 0
17     for i, char := range str {
18         if seenIndex, found := lastSeen[char]; found &&
19             startIndex < seenIndex+1 {
20             startIndex = seenIndex + 1
21         }
22         if longest.length() < i+1-startIndex {
23             longest = substring{startIndex, i + 1}
24         }
25         lastSeen[char] = i
26     }
27     return str[longest.left:longest.right]
28 }
29
```

Solution 1 Solution 2 Solution 3

```
1 package main
2
3 func LongestSubstringWithoutDuplication(str string) string {
4     // Write your code here.
5     return ""
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 # Run in Jupyter Notebook Notebook 2
2 import os
3 output = os.path.dirname(os.path.dirname(__file__))
4 import sys, os, re, json, math
5
6
7
8
9 # Run in Jupyter Notebook Notebook 2
10 import os
11 output = os.path.dirname(os.path.dirname(__file__))
12 import sys, os, re, json, math
13
14
15
16
17 # Run in Jupyter Notebook Notebook 2
18 import os
19 output = os.path.dirname(os.path.dirname(__file__))
```

Run or submit code when you're ready.