

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(n) time | O(min(n, a)) space
4 function longestSubstringWithoutDuplication(string) {
5   const lastSeen = {};
6   let longest = [0, 1];
7   let startIdx = 0;
8   for (let i = 0; i < string.length; i++) {
9     const char = string[i];
10    if (char in lastSeen) {
11      startIdx = Math.max(startIdx, lastSeen[char] + 1);
12    }
13    if (longest[1] - longest[0] < i + 1 - startIdx) {
14      longest = [startIdx, i + 1];
15    }
16    lastSeen[char] = i;
17  }
18  return string.slice(longest[0], longest[1]);
19 }
20
21 exports.longestSubstringWithoutDuplication = longestSubstringWithoutDu
22
```

Solution 1

Solution 2

Solution 3

```
1 function longestSubstringWithoutDuplication(string) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.longestSubstringWithoutDuplication = longestSubstringWithoutDu
7
```

Our Tests

```
1 const longest = longestSubstringWithoutDuplication('abcabcbb');
2 const test = longest === 'abc';
3
4 // Test Case 2: 'bbbbb' -> 'b'
5 const test2 = longestSubstringWithoutDuplication('bbbbb') === 'b';
6
```

Custom Output

Submit Code

```
100 Test Case 001 - Pass (1/1)
101 Run: assert program: IntegerTest.IntegerTest() == 0 (1/1)
102
103
104 Test Case 001 - Pass (1/1)
105 Run: assert program: IntegerTest.IntegerTest() == 0 (1/1)
106
107
108 Test Case 001 - Pass (1/1)
109 Run: assert program: IntegerTest.IntegerTest() == 0 (1/1)
110
111
112 Test Case 001 - Pass (1/1)
113 Run: assert program: IntegerTest.IntegerTest() == 0 (1/1)
114
115
```

Run or submit code when you're ready.