

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 = longestSubstringWithoutDuplication;
22
```