Solution 1

```go
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
```