

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n) time | O(min(n, a)) space
7     public static String longestSubstringWithoutDuplication(String str) {
8         Map<Character, Integer> lastSeen = new HashMap<Character, Integer>();
9         int[] longest = {0, 1};
10        int startIdx = 0;
11        for (int i = 0; i < str.length(); i++) {
12            char c = str.charAt(i);
13            if (lastSeen.containsKey(c)) {
14                startIdx = Math.max(startIdx, lastSeen.get(c) + 1);
15            }
16            if (longest[1] - longest[0] < i + 1 - startIdx) {
17                longest = new int[] {startIdx, i + 1};
18            }
19            lastSeen.put(c, i);
20        }
21        String result = str.substring(longest[0], longest[1]);
22        return result;
23    }
24 }
25
```