

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <unordered_map>
4 #include <vector>
5 using namespace std;
6
7 // O(n) time | O(min(n, a)) space
8 string longestSubstringWithoutDuplication(string str) {
9     unordered_map<char, int> lastSeen;
10    vector<int> longest{0, 1};
11    int startIdx = 0;
12    for (int i = 0; i < str.length(); i++) {
13        char character = str[i];
14        if (lastSeen.find(character) != lastSeen.end()) {
15            startIdx = max(startIdx, lastSeen[character] + 1);
16        }
17        if (longest[1] - longest[0] < i + 1 - startIdx) {
18            longest = {startIdx, i + 1};
19        }
20        lastSeen[character] = i;
21    }
22    string result = str.substr(longest[0], longest[1] - longest[0]);
23    return result;
24 }
25
```

Solution 1 Solution 2 Solution 3

```
1 using namespace std;
2
3 string longestSubstringWithoutDuplication(string str) {
4     // Write your code here.
5     return "";
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <unordered_map>
4 #include <vector>
5 using namespace std;
6
7 // O(n) time | O(min(n, a)) space
8 string longestSubstringWithoutDuplication(string str) {
9     unordered_map<char, int> lastSeen;
10    vector<int> longest{0, 1};
11    int startIdx = 0;
12    for (int i = 0; i < str.length(); i++) {
13        char character = str[i];
14        if (lastSeen.find(character) != lastSeen.end()) {
15            startIdx = max(startIdx, lastSeen[character] + 1);
16        }
17        if (longest[1] - longest[0] < i + 1 - startIdx) {
18            longest = {startIdx, i + 1};
19        }
20        lastSeen[character] = i;
21    }
22    string result = str.substr(longest[0], longest[1] - longest[0]);
23    return result;
24 }
```

```
1 using namespace std;
2
3 string longestSubstringWithoutDuplication(string str) {
4     // Write your code here.
5     return "";
6 }
7
```

