

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 using namespace std;
6
7 // O(n) time | O(n) space
8 vector<int> largestRange(vector<int> array) {
9     vector<int> bestRange = {};
10    int longestLength = 0;
11    unordered_map<int, bool> nums = {};
12    for (int num : array) {
13        nums[num] = true;
14    }
15    for (int num : array) {
16        if (!nums[num]) {
17            continue;
18        }
19        nums[num] = false;
20        int currentLength = 1;
21        int left = num - 1;
22        int right = num + 1;
23        while (nums.find(left) != nums.end()) {
24            nums[left] = false;
25            currentLength++;
26            left--;
27        }
28        while (nums.find(right) != nums.end()) {
29            nums[right] = false;
30            currentLength++;
31            right++;
32        }
33        if (currentLength > longestLength) {
```

Solution 1   Solution 2   Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<int> largestRange(vector<int> array) {
5     // Write your code here.
6     return {};
7 }
8
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 using namespace std;
6
7 // O(n) time | O(n) space
8 vector<int> largestRange(vector<int> array) {
9     vector<int> bestRange = {};
10    int longestLength = 0;
11    unordered_map<int, bool> nums = {};
12    for (int num : array) {
13        nums[num] = true;
14    }
15    for (int num : array) {
16        if (!nums[num]) {
17            continue;
18        }
19        nums[num] = false;
20        int currentLength = 1;
21        int left = num - 1;
22        int right = num + 1;
23        while (nums.find(left) != nums.end()) {
24            nums[left] = false;
25            currentLength++;
26            left--;
27        }
28        while (nums.find(right) != nums.end()) {
29            nums[right] = false;
30            currentLength++;
31            right++;
32        }
33        if (currentLength > longestLength) {
```

```
24
25
26 # Python 3
27 # Python 2
28 # Python 3
29 # Python 2
30 # Python 3
31 # Python 2
32 # Python 3
33 # Python 2
34 # Python 3
35 # Python 2
36 # Python 3
37 # Python 2
38 # Python 3
39 # Python 2
40 # Python 3
41 # Python 2
42 # Python 3
43 # Python 2
44 # Python 3
45 # Python 2
46 # Python 3
47 # Python 2
48 # Python 3
49 # Python 2
50 # Python 3
51 # Python 2
52 # Python 3
53 # Python 2
54 # Python 3
55 # Python 2
56 # Python 3
57 # Python 2
58 # Python 3
59 # Python 2
60 # Python 3
61 # Python 2
62 # Python 3
63 # Python 2
64 # Python 3
65 # Python 2
66 # Python 3
67 # Python 2
68 # Python 3
69 # Python 2
70 # Python 3
71 # Python 2
72 # Python 3
73 # Python 2
74 # Python 3
75 # Python 2
76 # Python 3
77 # Python 2
78 # Python 3
79 # Python 2
80 # Python 3
81 # Python 2
82 # Python 3
83 # Python 2
84 # Python 3
85 # Python 2
86 # Python 3
87 # Python 2
88 # Python 3
89 # Python 2
90 # Python 3
91 # Python 2
92 # Python 3
93 # Python 2
94 # Python 3
95 # Python 2
96 # Python 3
97 # Python 2
98 # Python 3
99 # Python 2
100 # Python 3
```

Run or submit code when you're ready.