

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n) time | O(n) space
7     public static int[] largestRange(int[] array) {
8         int[] bestRange = new int[2];
9         int longestLength = 0;
10        Map<Integer, Boolean> nums = new HashMap<Integer, Boolean>();
11        for (int num : array) {
12            nums.put(num, true);
13        }
14        for (int num : array) {
15            if (!nums.get(num)) {
16                continue;
17            }
18            nums.put(num, false);
19            int currentLength = 1;
20            int left = num - 1;
21            int right = num + 1;
22            while (nums.containsKey(left)) {
23                nums.put(left, false);
24                currentLength++;
25                left--;
26            }
27            while (nums.containsKey(right)) {
28                nums.put(right, false);
29                currentLength++;
30                right++;
31            }
32            if (currentLength > longestLength) {
33                longestLength = currentLength;
34                bestRange = new int[] {left + 1, right - 1};
35            }
36        }
37        return bestRange;
38    }
39 }
40
```