

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(n) time | O(n) space
6 func LargestRange(array []int) []int {
7     best := []int{}
8     longestLength := 0
9     nums := map[int]bool{}
10    for _, num := range array {
11        nums[num] = true
12    }
13    for _, num := range array {
14        if !nums[num] {
15            continue
16        }
17        nums[num] = false
18        currentLength, left, right := 1, num-1, num+1
19        for nums[left] {
20            nums[left] = false
21            currentLength += 1
22            left -= 1
23        }
24        for nums[right] {
25            nums[right] = false
26            currentLength += 1
27            right += 1
28        }
29        if currentLength > longestLength {
30            longestLength = currentLength
31            best = []int{left + 1, right - 1}
32        }
33    }
```

Our Tests

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(n) time | O(n) space
6 func LargestRange(array []int) []int {
7     best := []int{}
8     longestLength := 0
9     nums := map[int]bool{}
10    for _, num := range array {
11        nums[num] = true
12    }
13    for _, num := range array {
14        if !nums[num] {
15            continue
16        }
17        nums[num] = false
18        currentLength, left, right := 1, num-1, num+1
19        for nums[left] {
20            nums[left] = false
21            currentLength += 1
22            left -= 1
23        }
24        for nums[right] {
25            nums[right] = false
26            currentLength += 1
27            right += 1
28        }
29        if currentLength > longestLength {
30            longestLength = currentLength
31            best = []int{left + 1, right - 1}
32        }
33    }
```

Solution 1 Solution 2 Solution 3

```
1 package main
2
3 func LargestRange(array []int) []int {
4     // Write your code here.
5     return nil
6 }
7
```

Custom Output

Submit Code

```
1 package main
2
3 func LargestRange(array []int) []int {
4     // Write your code here.
5     return nil
6 }
```

```
11 # Run in ThreadLocal ThreadLocal ThreadLocal
12 expected = (20000, 5)
13 output = LongestBipartite(20000, 5)
14 return output, expected, output
15 }
16
17 # Run in ThreadLocal ThreadLocal ThreadLocal
18 expected = (20000, 5)
19 output = LongestBipartite(20000, 5)
20 return output, expected, output
21 }
22
23 # Run in ThreadLocal ThreadLocal ThreadLocal
24 expected = (20000, 5)
```

Run or submit code when you're ready.