

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 import "math"
6
7 // O(n) time | O(1) space
8 func SubarraySort(array []int) []int {
9     minOutOfOrder, maxOutOfOrder := math.MaxInt32, math.MinInt32
10    for i, num := range array {
11        if isOutOfOrder(i, num, array) {
12            minOutOfOrder = min(minOutOfOrder, num)
13            maxOutOfOrder = max(maxOutOfOrder, num)
14        }
15    }
16    if minOutOfOrder == math.MaxInt32 {
17        return []int{-1, -1}
18    }
19    subarrayLeft := 0
20    for minOutOfOrder >= array[subarrayLeft] {
21        subarrayLeft += 1
22    }
23    subarrayRight := len(array) - 1
24    for maxOutOfOrder <= array[subarrayRight] {
25        subarrayRight -= 1
26    }
27    return []int{subarrayLeft, subarrayRight}
28 }
29
30 func isOutOfOrder(i int, num int, array []int) bool {
31     if i == 0 {
32         return num > array[i+1]
33     }
```

Solution 1 Solution 2 Solution 3

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

Our Tests

Custom Output

Submit Code

```
18 Run on ThreadLocalTestThreadTest (2)
19 expected = (2000, 5, 5)
20 actual = testRunner().run(2000, 5)
21 assertEquals(expected, actual)
22 }
23
24 Run on ThreadLocalTestThreadTest (2)
25 expected = (2000, 5, 5)
26 actual = testRunner().run(2000, 5)
27 assertEquals(expected, actual)
28 }
29
30 Run on ThreadLocalTestThreadTest (2)
31 expected = (2000, 5, 5)
```

Run or submit code when you're ready.