Solution 1

Run Code

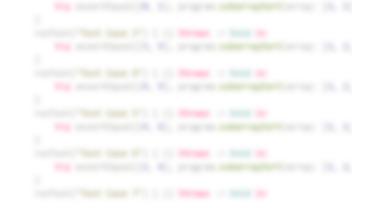
Our Solution(s)

Run Code

```
Your Solutions
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   class Program {
        // O(n) time | O(1) space
        func subarraySort(array: [Int]) -> [Int] {
            var minimumOutOfOrder = Int(Int16.max)
            var maximumOutOfOrder = -Int(Int16.max)
 9
            for i in 0 ..< array.count {</pre>
                let currentNumber = array[i]
10
11
                 \textbf{if} \  \, \textbf{isOutOfOrder(i, array, currentNumber)} \ \{ \\
12
13
                     minimumOutOfOrder = min(currentNumber, minimumOutOfOrd
14
                     maximumOutOfOrder = max(currentNumber, maximumOutOfOrd
16
17
            if minimumOutOfOrder == Int(Int16.max) {
18
19
                return [-1, -1]
20
21
22
            var subarrayLeftIndex = 0
23
            while minimumOutOfOrder >= array[subarrayLeftIndex] {
                subarrayLeftIndex += 1
25
26
27
            var subarrayRightIndex = array.count - 1
            while maximumOutOfOrder <= array[subarrayRightIndex] {</pre>
28
29
                subarrayRightIndex -= 1
30
31
32
            return [subarrayLeftIndex, subarrayRightIndex]
33
```

\_\_\_\_



Run or submit code when you're ready.