

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(n) time | O(1) space
4 function subarraySort(array) {
5   let minOutOfOrder = Infinity;
6   let maxOutOfOrder = -Infinity;
7   for (let i = 0; i < array.length; i++) {
8     const num = array[i];
9     if (isOutOfOrder(i, num, array)) {
10       minOutOfOrder = Math.min(minOutOfOrder, num);
11       maxOutOfOrder = Math.max(maxOutOfOrder, num);
12     }
13   }
14   if (minOutOfOrder === Infinity) {
15     return [-1, -1];
16   }
17   let subarrayLeftIdx = 0;
18   while (minOutOfOrder >= array[subarrayLeftIdx]) {
19     subarrayLeftIdx++;
20   }
21   let subarrayRightIdx = array.length - 1;
22   while (maxOutOfOrder <= array[subarrayRightIdx]) {
23     subarrayRightIdx--;
24   }
25   return [subarrayLeftIdx, subarrayRightIdx];
26 }
27
28 function isOutOfOrder(i, num, array) {
29   if (i === 0) return num > array[i + 1];
30   if (i === array.length - 1) return num < array[i - 1];
31   return num > array[i + 1] || num < array[i - 1];
32 }
33
34 exports.subarraySort = subarraySort;
35
```