3 #include <vector>

Solution 1

Run Code

Our Solution(s)

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.

```
Run Code
```

Your Solutions

Solution 1 Solution 2 Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<int> subarraySort(vector<int> array) {
5   // Write your code here.
6   return {};
7 }
```

```
4 #include <algorithm>
 5 #include <climits>
 6 using namespace std;
 8 bool isOutOfOrder(int i, int num, vector<int> array);
10 // O(n) time | O(1) space
11 vector<int> subarraySort(vector<int> array) {
12
      int minOutOfOrder = INT_MAX;
      int maxOutOfOrder = INT_MIN;
13
14
      for (int i = 0; i < array.size(); i++) {</pre>
        int num = array[i];
16
        if (isOutOfOrder(i, num, array)) {
17
          minOutOfOrder = min(minOutOfOrder, num);
18
          maxOutOfOrder = max(maxOutOfOrder, num);
19
20
21
      if (minOutOfOrder == INT_MAX) {
22
        return vector<int>{-1, -1};
23
24
      int subarrayLeftIdx = 0;
25
      \label{eq:while} \textbf{while} \; (\texttt{minOutOfOrder} >= \texttt{array}[\texttt{subarrayLeftIdx}]) \; \{
26
        subarrayLeftIdx++;
27
28
      int subarrayRightIdx = array.size() - 1;
29
      while (maxOutOfOrder <= array[subarrayRightIdx]) {</pre>
30
        subarrayRightIdx--;
31
32
      return vector<int>{subarrayLeftIdx, subarrayRightIdx};
33 }
```

Run or submit code when you're ready.

----