Solution 1

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
#include <algorithm>
#include <climits>
using namespace std;

bool isOutOfOrder(int i, int num, vector<int> array);

// O(n) time | O(1) space
vector<int> subarraySort(vector<int> array) {
  int minOutOfOrder = INT_MAX;
  int maxOutOfOrder = INT_MIN;
  for (int i = 0; i < array.size(); i++) {
    int num = array[i];
    if (isOutOfOrder(i, num, array)) {
      minOutOfOrder = min(minOutOfOrder, num);
      maxOutOfOrder = max(maxOutOfOrder, num);
    }
  }
  if (minOutOfOrder == INT_MAX) {
    return vector<int>{-1, -1};
  }
  int subarrayLeftIdx = 0;
  while (minOutOfOrder >= array[subarrayLeftIdx]) {
    subarrayLeftIdx++;
  }
  int subarrayRightIdx = array.size() - 1;
  while (maxOutOfOrder <= array[subarrayRightIdx]) {
    subarrayRightIdx--;
  }
  return vector<int>{subarrayLeftIdx, subarrayRightIdx};
}

bool isOutOfOrder(int i, int num, vector<int> array) {
  if (i == 0) {
    return num > array[i + 1];
  }
  if (i == array.size() - 1) {
    return num < array[i - 1];
  }
  return num > array[i + 1] || num < array[i - 1];
}

```