Solution 1

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

package main

import "math"

// O(n) time | O(1) space
func SubarraySort(array []int) []int {
  minOutOfOrder, maxOutOfOrder := math.MaxInt32, math.MinInt32
  for i, num := range array {
    if isOutOfOrder(i, num, array) {
      minOutOfOrder = min(minOutOfOrder, num)
      maxOutOfOrder = max(maxOutOfOrder, num)
    }
  }
  if minOutOfOrder == math.MaxInt32 {
    return []int{-1, -1}
  }
  subarrayLeft := 0
  for minOutOfOrder >= array[subarrayLeft] {
    subarrayLeft += 1
  }
  subarrayRight := len(array) - 1
  for maxOutOfOrder <= array[subarrayRight] {
    subarrayRight -= 1
  }
  return []int{subarrayLeft, subarrayRight}
}

func isOutOfOrder(i int, num int, array []int) bool {
  if i == 0 {
    return num > array[i+1]
  }
  if i == len(array)-1 {
    return num < array[i-1]
  }
  return num > array[i+1] || num < array[i-1]
}

func min(a, b int) int {
  if a < b {
    return a
  }
  return b
}

func max(a, b int) int {
  if a < b {
    return b
  }
  return a
}

```