Almost Sorted

by PRASHANTB1984

Problem | Submissions | Leaderboard | Discussions | Editorial

Given an array with n elements, can you sort this array in *ascending order* using only one of the following operations?

1. Swap two elements.
2. Reverse one sub-segment.

Input Format

The first line contains a single integer, n, which indicates the size of the array.
The next line contains n integers separated by spaces.

```
n
d1 d2 ... dn
```

Constraints

$2 \leq n \leq 100000$
$0 \leq d_i \leq 1000000$
All d_i are distinct.

Output Format

1. If the array is already sorted, output *yes* on the first line. You do not need to output anything else.

1. If you can sort this array using one single operation (from the two permitted operations) then output *yes* on the first line and then:

   a. If you can sort the array by swapping d_l and d_r, output *swap l r* in the second line. l and r are the indices of the elements to be swapped, assuming that the array is indexed from 1 to n.

   b. Else if it is possible to sort the array by reversing the segment $d[l \ldots r]$, output *reverse l r* in the second line. l and r are the indices of the first and last elements of the subsequence to be reversed, assuming that the array is indexed from 1 to n.

   $d[l \ldots r]$ represents the sub-sequence of the array, beginning at index l and ending at index r, both inclusive.

   If an array can be sorted by either swapping or reversing, stick to the swap-based method.

2. If you cannot sort the array in either of the above ways, output *no* in the first line.

Sample Input #1

```
2
4 2
```

Sample Output #1

```
yes
swap 1 2
```

Sample Input #2

```
3
3 1 2
```

Sample Output #2

```
no
```

Sample Input #3

```
6
1 5 4 3 2 6
```

Sample Output #3

```
yes
reverse 2 5
```

Explanation

For #1, you can both *swap(1, 2)* and *reverse(1, 2)*, but if you can sort the array using swap, output swap only.
For #2, it is impossible to sort by one single operation (among those permitted).
For #3, you can reverse the sub-array *d[2...5] = "5 4 3 2"*, then the array becomes sorted.


```
using System;
using System.Collections.Generic;
using System.IO;
class Solution {
    static void Main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution */


         int n = int.Parse(Console.ReadLine());
            int[] arr = Array.ConvertAll(Console.ReadLine().Split(' '), e => int.Parse(e));

            //int[] arr = { 1, 5, 4, 3, 2, 6 };
            int[] original = new int[arr.Length];
            Array.Copy(arr, original, arr.Length);

            Array.Sort(arr);

            int i = 0;
            int distintos = 0;
            int iprim = -1, iult = 0;
            for ( i = 0; i < arr.Length; i++)
            {
                if (arr[i] != original[i])
                {
                    distintos++;
                }
                if (iprim ==-1 && arr[i] != original[i])
                {
                    iprim = i;
```

```
30                  }
31                  if (iprim != -1 && arr[i] != original[i])
32                  {
33                      iult = i;
34                  }
35              }
36
37              if (distintos == 2)
38              {
39                  Console.WriteLine("yes");
40                  Console.WriteLine("swap {0} {1}", iprim+1, iult+1);
41              }
42              else if (distintos > 2)
43              {
44                  int[] rev = new int[iult - iprim + 1];
45                  Array.Copy(original, iprim, rev, 0, iult - iprim + 1);
46                  Array.Reverse(rev);
47                  //foreach (int elem in rev)
48                  //{
49                  //    Console.Write(elem + " ");
50                  //}
51                  int indice_rev =0;
52                  int[] nuevo = new int[arr.Length];
53                  Array.Copy(original, nuevo, arr.Length);
54                  for ( i = iprim; i <= iult; i++)
55                  {
56                      nuevo[i] = rev[indice_rev++];
57                  }
58                  //me fijo si esta ordenado
59                  i = 0;
60                  for ( i = 0; i < arr.Length; i++)
61                  {
62                      if (nuevo[i] != arr[i])
63                      {
64                          Console.WriteLine("no");
65                          break;
66                      }
67                  }
68                  if (i == arr.Length)
69                  {
70                      Console.WriteLine("yes");
71                      Console.WriteLine("reverse {0} {1}", iprim + 1, iult + 1);
72                  }
73
74                 // Console.WriteLine();
75              }
76
77
78      }
79  }
```

Line: 66 Col: 22

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14
- ✔ Test Case #15
- ✔ Test Case #16
- ✔ Test Case #17

- ✔ Test Case #18
- ✔ Test Case #19
- ✔ Test Case #20
- ✔ Test Case #21
- ✔ Test Case #22
- ✔ Test Case #23
- ✔ Test Case #24

Next Challenge