



# Accurate Sorting

by zemen

Problem

Submissions

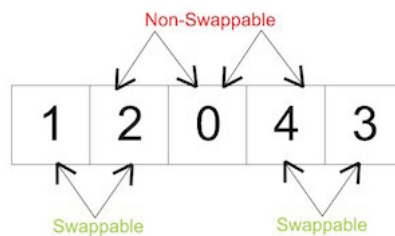
Leaderboard

Discussions

Your submission will run against only preliminary test cases. Full test cases will run at the end of the day.

Consider an unsorted array,  $A = a_0, a_1, \dots, a_{n-1}$ , of distinct integers from  $0$  to  $n - 1$ . We can *swap* two adjacent elements in  $A$  any number of times as long as the absolute difference between these elements is  $1$ .

For example, the diagram below depicts an array where we can swap adjacent elements  $1$  and  $2$  or  $4$  and  $3$ , but we cannot swap adjacent elements  $2$  and  $0$  or  $0$  and  $4$ :



Answer  $q$  queries, where each query consists of some array  $A$ . For each query, print **Yes** on a new line if it's possible to sort the array in ascending order by performing the swap operation defined above; otherwise, print **No** instead.

## Input Format

The first line contains a single integer denoting  $q$ . The subsequent lines describe each of the  $q$  queries in the following format:

1. The first line contains an integer denoting  $n$ .
2. The second line contains  $n$  space-separated integers describing the respective values of  $a_0, a_1, \dots, a_{n-1}$ .

## Constraints

- $1 \leq q \leq 10$
- $1 \leq n \leq 10^5$
- The sum of  $n$  over all queries doesn't exceed  $10^5$ .

## Output Format

For each query, print **Yes** on a new line if it's possible to sort the array; otherwise, print **No** instead.

## Sample Input 0

```
2
4
1 0 3 2
3
2 1 0
```

## Sample Output 0

Yes  
No

## Explanation 0

We perform the following  $q = 2$  queries:

1. The following sequence of swaps will sort the array in ascending order:

$A = [1, 0, 3, 2] \rightarrow [0, 1, 3, 2] \rightarrow [0, 1, 2, 3]$

Because  $A$  is now sorted, we print **Yes** on a new line.

2. Initially, we can perform two possible swaps:

1.  $A = [2, 1, 0] \rightarrow [1, 2, 0]$

After performing this swap, no number of additional swaps can move **0** to the front of the array.

2.  $A = [2, 1, 0] \rightarrow [2, 0, 1]$

After performing this swap, no number of additional swaps can move **2** to the back of the array.

Because there's no way for us to sort the array, we print **No** on a new line.

f t in

Contest ends in 6 days

Submissions: 1365



Max Score: 25



Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)  

C#  

```

1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 class Solution {
6
7     static void Main(String[] args) {
8         int q = Convert.ToInt32(Console.ReadLine());
9         for (int a0 = 0; a0 < q; a0++)
10         {
11             int n = Convert.ToInt32(Console.ReadLine());
12             string[] a_temp = Console.ReadLine().Split(' ');
13             int[] a = Array.ConvertAll(a_temp, e => int.Parse(e));
14             // Write Your Code Here
15
16             //int[] a = { 1, 0, 3, 2 };
17             //sort = { 0, 1, 2, 3}
18             //int[] a = { 2, 1, 0 };
19
20             int[] sort = a.ToList().ToArray();
21             Array.Sort(sort);
22
23             Dictionary<int, int> indices_a = new Dictionary<int, int>();
24             for (int i = 0; i < a.Length; i++)
25             {
26                 indices_a[a[i]] = i;
27             }
28             Dictionary<int, int> indices_sort = new Dictionary<int, int>();
29             for (int i = 0; i < sort.Length; i++)
30             {

```

```
31         indices_sort[sort[i]] = i;
32     }
33
34     string ans = "Yes";
35     for (int i = 0; i < a.Length; i++)
36     {
37         int i_copia = indices_a[a[i]]; // Array.IndexOf(a, a[i]);
38         int i_sort = indices_sort[a[i_copia]]; // Array.IndexOf(sort, a[i_copia]);
39
40         while (i_copia < i_sort)
41         {
42             if (i_copia + 1 < a.Length && Math.Abs(a[i_copia] - a[i_copia + 1]) <= 1)
43             {
44                 int temp = a[i_copia];
45                 a[i_copia] = a[i_copia + 1];
46                 a[i_copia + 1] = temp;
47
48                 i_copia++;
49             }
50             else
51             {
52                 break;
53             }
54         }
55         while (i_copia > i_sort)
56         {
57             if (i_copia - 1 >= 0 && Math.Abs(a[i_copia] - a[i_copia - 1]) <= 1)
58             {
59                 int temp = a[i_copia];
60                 a[i_copia] = a[i_copia - 1];
61                 a[i_copia - 1] = temp;
62
63                 i_copia--;
64             }
65             else
66             {
67                 break;
68             }
69         }
70
71         if (i_copia != i_sort)
72         {
73             ans = "No";
74             break;
75         }
76     }
77
78     Console.WriteLine(ans);
79 }
80
81 }
82
83 }
84
```

Line: 81 Col: 1

 Upload Code as File☐ Test against custom input

Run Code

Submit Code

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)