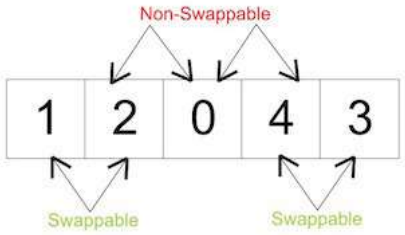
Accurate Sorting

by zemen

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

Consider an unsorted array, $A = a_0, a_1, \cdots, a_{n-1}$, 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 $a_0, a_1, \cdots, a_{n-1}$.

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.


```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static boolean checkSorted(int[] a) {

        for (int i=1; i<a.length; i++) {
            if (a[i] < a[i-1])
                return false;
        }
        return true;
    }

    public static void sortMyNos(int[] a) {

        for (int i = 1; i<a.length; i++) {
            if ((Math.abs(a[i]-a[i-1]) == 1) && a[i] < a[i-1]) {
                int temp = a[i];
                a[i] = a[i-1];
                a[i-1] = temp;
            }
        }

        //check if they're sorted
        if (checkSorted(a))
            System.out.println("Yes");
```

```
        else
            System.out.println("No");

    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int q = in.nextInt();
        for(int a0 = 0; a0 < q; a0++){
            int n = in.nextInt();
            int[] a = new int[n];
            for(int a_i=0; a_i < n; a_i++){
                a[a_i] = in.nextInt();
            }
            // Write Your Code Here
            sortMyNos(a);

        }
    }
}

```

Line: 29 Col: 9

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