Facebook Hacker Cup '15 Round 2 P1 - Lazy Sort

Time limit: 25.0s **Memory limit:** 1G

Facebook Hacker Cup 2015 Round 2

Much to Mr. Book's amazement, his entire first-grade math class has completed the enormous amount of homework that he assigned only the day before. Being hyperactive little kids, the grade-schoolers have dumped their homework, one page per student, unceremoniously in a pile on Mr. Book's desk.

There are N students in Mr. Book's class, and they each have a distinct integer student ID from 1 to N.
Mr. Book wants the homework to be sorted by student ID with the page from student #1 at the top, all the way down to the page from student #N at the bottom.
As we know, Mr. Book is a lazy fellow. He's not going to put much effort into sorting the pile himself.

As he considers the pile before him, Mr. Book decides that he'll sort the homework as follows: Every time he moves a page, he'll only move it from the original pile to his finished stack, which initially starts empty. Furthermore, he'll only move the top or bottom page of the remaining original pile, and he'll only place it on the top or bottom of his neat destination stack.

Is it possible for Mr. Book to sort the homework with such a method?

Input

Input begins with an integer T, the number of test cases.
For each case, there are two lines. The first contains the integer N.
The second contains a permutation of the integers from 1 to N.
These are the students' IDs in the order their homework appears in the pile, from bottom to top.

Output

For the i^{th} case, print a line containing `Case #i:` followed by `yes` if Mr. Book can sort the homework by student ID, or `no` if he can't.

Constraints

$1 \leq T \leq 20$
$1 \leq N \leq 50\,000$

Explanation of Sample

In the first case, Mr. Book can just keep moving the top of the pile to the top of his final stack, which will reverse the order of the pages.

In the fourth case, one approach is for Mr. Book to move pages 4, 3, 2, and 1 to the top of his final stack, and then move 5 and 6 to the bottom.

Sample Input

```
5
4
1 2 3 4
4
1 3 2 4
9
2 4 3 1 5 9 6 7 8
6
4 3 2 1 6 5
5
3 4 5 1 2
```

Sample Output

```
Case #1: yes
Case #2: no
Case #3: no
Case #4: yes
Case #5: yes
```