Problem F. Almost Sorted Array

Time limit: 2s
Color of balloons: orange

We are all familiar with sorting algorithms: quick sort, merge sort, heap sort, insertion sort, selection sort, bubble sort, etc. But sometimes it is an overkill to use these algorithms for an almost sorted array.

We say an array is sorted if its elements are in non-decreasing order or non-increasing order. We say an array is almost sorted if we can remove exactly one element from it, and the remaining array is sorted. Now you are given an array $a_1, a_2, \ldots, a_n$, is it almost sorted?

Input

The first line contains an integer T indicating the total number of test cases. Each test case starts with an integer n in one line, then one line with n integers $a_1, a_2, \ldots, a_n$.

- $1 \leq T \leq 2000$
- $2 \leq n \leq 10^5$
- $1 \leq a_i \leq 10^5$
- There are at most 20 test cases with $n > 1000$.

Output

For each test case, please output "'YES'" if it is almost sorted. Otherwise, output "'NO'" (both without quotes).

Sample

standard input	standard output
3 3 2 1 7 3 3 2 1 5 3 1 4 1 5	YES YES NO

This page intentionally left blank.