

Peak Reduction

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

You are given a permutation* p of length n . In one operation, you may remove an element p_i if the following conditions are satisfied:

- $1 < i < |p|$ (where $|p|$ denotes the current length of p),
- $p_{i-1} < p_i > p_{i+1}$ (i.e., p_i is a local maximum).

Determine whether it is possible to reduce the permutation length to exactly 2 by applying the operation any number of times (possibly zero).

* A permutation of length n is a sequence of n integers where each integer from 1 to n appears exactly once. For example, $[1, 3, 2]$ is a permutation, but $[2, 3, 2]$ and $[4, 1, 2]$ are not.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains an integer n ($3 \leq n \leq 3 \cdot 10^5$) — the length of the permutation.

The second line contains n distinct integers p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — the elements of the permutation.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, print YES if it is possible to reduce the permutation to length 2, otherwise print NO.

Example

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

Note

In the first test case, $p = [1, 3, 2]$. Since $p_1 < p_2 > p_3$, we can remove $p_2 = 3$. The resulting array $[1, 2]$ has length 2.

In the second test case, no element satisfies the local maximum condition, so it is impossible to reduce the length.