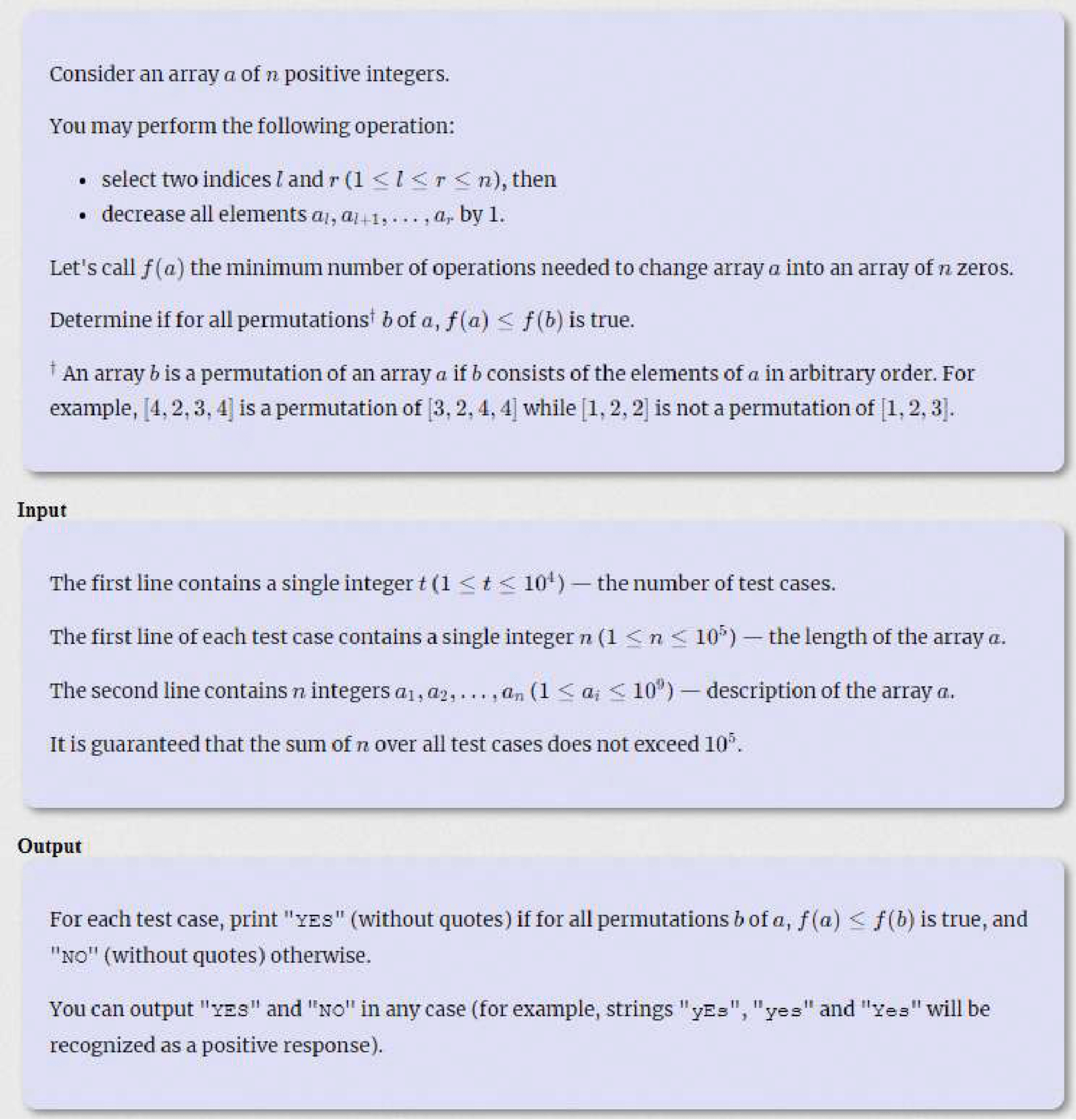


Consider an array a of n positive integers.

You may perform the following operation:

- select two indices l and r ($1 \le l \le r \le n$), then
- decrease all elements $a_l, a_{l+1}, \ldots, a_r$ by 1.

Let's call $f(a)$ the minimum number of operations needed to change array a into an array of n zeros.

Determine if for all permutations† b of a, $f(a) \le f(b)$ is true.

† An array b is a permutation of an array a if b consists of the elements of a in arbitrary order. For example, $[4, 2, 3, 4]$ is a permutation of $[3, 2, 4, 4]$ while $[1, 2, 2]$ is not a permutation of $[1, 2, 3]$.

Input

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

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the length of the array a.

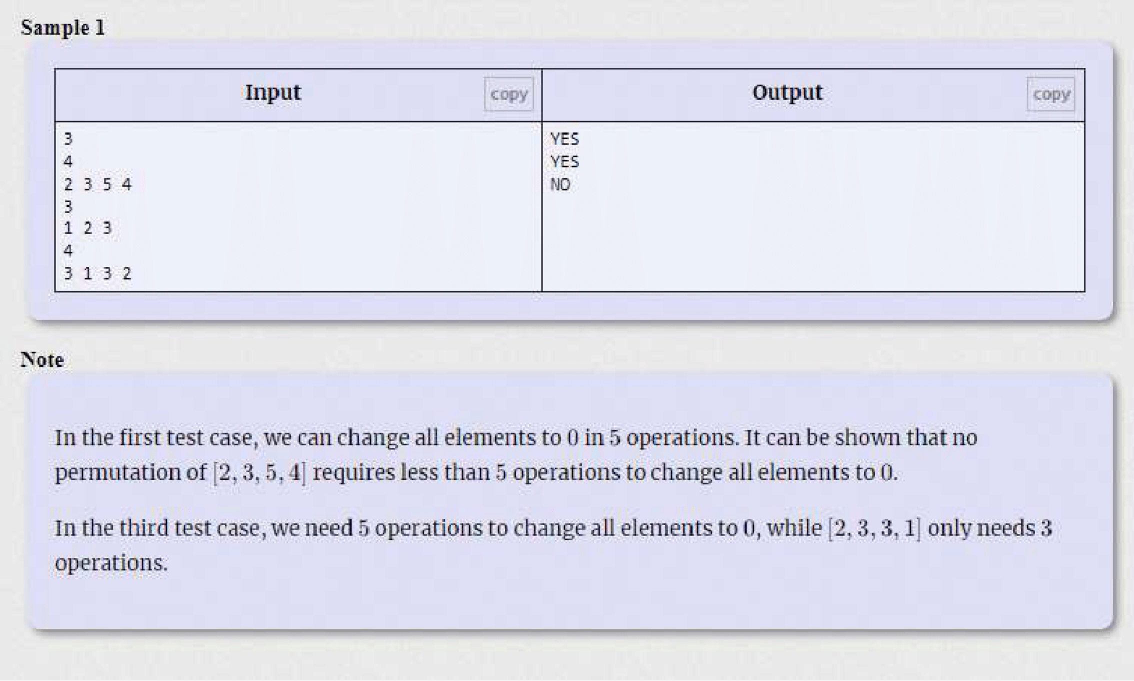
The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — description of the array a.

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

Output

For each test case, print "YES" (without quotes) if for all permutations b of a, $f(a) \le f(b)$ is true, and "NO" (without quotes) otherwise.

You can output "YES" and "NO" in any case (for example, strings "yEs", "yes" and "Yes" will be recognized as a positive response).



Sample 1

Input	Output

3
4
2 3 5 4
3
1 2 3
4
3 1 3 2
	YES
	YES
	NO

Note

In the first test case, we can change all elements to 0 in 5 operations. It can be shown that no permutation of $[2, 3, 5, 4]$ requires less than 5 operations to change all elements to 0.

In the third test case, we need 5 operations to change all elements to 0, while $[2, 3, 3, 1]$ only needs 3 operations.