## B. CopyCopyCopyCopyCopy

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ehab has an array a of length n. He has just enough free time to make a new array consisting of n copies of the old array, written back-to-back. What will be the length of the new array's longest increasing subsequence?

A sequence a is a subsequence of an array b if a can be obtained from b by deletion of several (possibly, zero or all) elements. The longest increasing subsequence of an array is the longest subsequence such that its elements are ordered in strictly increasing order.

### Input
The first line contains an integer t the number of test cases you need to solve. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \leq n \leq 10^5$) – the number of elements in the array a.

The second line contains n space-separated integers $a_1$, $a_2$, ..., $a_n$ ($1 \leq a_i \leq 10^9$) – the elements of the array a.

**The sum of n across the test cases doesn't exceed $10^5$.**

### Output
For each testcase, output the length of the longest increasing subsequence of a if you concatenate it to itself n times.

### Example

| Input | Output |
|---|---|
| 2<br>3<br>3 2 1<br>6<br>3 1 4 1 5 9 | 3<br>5 |

### Note
In the first sample, the new array is [3, 2, **1**, 3, **2**, 1, **3**, 2, 1]. The longest increasing subsequence is marked in bold.

In the second sample, the longest increasing subsequence will be [1, 3, 4, 5, 9].