Longest increasing subsequence (LIS)

Given a sequence **A** of size **N**, find the **length** of the **longest increasing subsequence** from a given sequence .
The longest increasing subsequence means to find a subsequence of a given sequence in which the subsequence's elements are in sorted order, lowest to highest, and in which the subsequence is as long as possible. This subsequence is not necessarily contiguous, or unique.

Note: Duplicate numbers are not counted as increasing subsequence.

Input:
The first line contains an integer T, depicting total number of test cases. Then following T lines contains an integer N depicting the size of array and next line followed by the value of array.

Output:
For each testcase, in a new line, print the Max length of the subsequence in a separate line.

Constraints:
1 ≤ T ≤ 100
1 ≤ N ≤ 1000
0 ≤ A[i] ≤ 300

Example:
Input:

```
2
16
0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15
6
5 8 3 7 9 1
```

Output:

```
6
3
```

Explanation:
Testcase 2: Longest increasing subsequence is of size 3 with elements (there are many subsequence, but listing one of them): **5 7 9**.