Problem I. Dolls 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Alice has just obtained n Russian dolls of distinct sizes. She arranges these dolls in a row from left to right, with the i-th position containing a doll of size a_i.

Let l_i and r_i denote the size of the smallest and largest doll in the i-th cell, respectively. Two adjacent cells i and $i+1$ can be merged if and only if $r_i < l_{i+1}$ or $r_{i+1} < l_i$. The new doll will contain all dolls from the original cells i and $i+1$, and will be placed in cell i. All dolls in cells with indices greater than $i+1$ will shift one position to the left to fill the gap.

For example, when $n = 5, a = [1, 2, 3, 5, 4]$, Alice can:

1. Merge the dolls in cells 1 and 2. The remaining dolls in the four cells have sizes $[(1, 2), (3), (5), (4)]$.
2. Merge the dolls in cells 2 and 3. The remaining dolls in the three cells have sizes $[(1, 2), (3, 5), (4)]$.
3. Merge the dolls in cells 1 and 2. The remaining dolls in the two cells have sizes $[(1, 2, 3, 5), (4)]$. According to the rules, the dolls in the remaining two cells can no longer be merged.

Alice will continue performing merge operations until no further merges are possible. What is the minimum number of merge operations Alice has to perform?

Input

Each test file contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^4$). The description of the test cases follows.

The first line of each test case contains an integer n ($2 \le n \le 10^6$), denoting the number of dolls.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$, $\forall i \ne j, a_i \ne a_j$), representing the sizes of the dolls at each position initially.

For each test file, it is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single integer on one line, representing the minimum number of merge operations required.

Example

standard input	standard output
5 5 1 2 3 5 4 5 1 2 3 4 5 5 1 3 5 4 2 5 1 5 2 4 3 5 1 2 4 3 5	3 4 3 2 3