Problem G. Quick Sort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

When Prof. Pang was young, he wrote the following code for quick sort. Please calculate how many swaps are performed when calling QUICKSORT$(A, 1, n)$. A is a given permutation with length n.

Algorithm 2 An implementation of quick sort

```
1: procedure QUICKSORT(A, lo, hi)
2:     if lo ≥ 0 and hi ≥ 0 and lo < hi then
3:         p ← PARTITION(A, lo, hi)
4:         QUICKSORT(A, lo, p)
5:         QUICKSORT(A, p + 1, hi)
6:     end if
7: end procedure
8: procedure PARTITION(A, lo, hi)
9:     pivot ← A[floor((hi + lo)/2)]
10:    i ← lo − 1
11:    j ← hi + 1
12:    while True do
13:        repeat
14:            i ← i + 1
15:        until A[i] ≥ pivot
16:        repeat
17:            j ← j − 1
18:        until A[j] ≤ pivot
19:        if i ≥ j then
20:            return j
21:        end if
22:        Swap A[i] with A[j]
23:    end while
24: end procedure
```

Input

The first line contains one integer T ($1 \le T \le 10^5$), the number of test cases.

For each test case, the first line contains one positive integer n ($1 \le n \le 5 \times 10^5$). The next line contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le n$) denoting the permutation A. It is guaranteed that $a_1, \ldots, a_n$ form a permutation, i.e. $a_i \ne a_j$ for $i \ne j$.

It is guaranteed that the sum of n over all test cases is no more than 5×10^5.

Output

For each test case, output one line containing the number of swaps performed when calling `quicksort`$(A, 1, n)$.

Example

standard input	standard output
3 3 3 2 1 5 2 4 5 3 1 10 7 2 4 6 1 9 10 8 5 3	1 4 7