Problem

Chef considers a permutation P of $\{1, 2, 3, \ldots, N\}$ `End Sorted` if and only if $P_1 = 1$ and $P_N = N$.

Chef is given a permutation P.

In one operation Chef can choose any index i $(1 \leq i \leq N - 1)$ and swap P_i and P_{i+1}. Determine the minimum number of operations required by Chef to make the permutation P `End Sorted`.

Note: An array P is said to be a permutation of $\{1, 2, 3, \ldots, N\}$ if P contains each element of $\{1, 2, 3, \ldots, N\}$ exactly once.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains a single integer N, denoting the length of the permutation P.
  - The second line contains N space-separated integers $P_1, P_2, P_3, \ldots, P_N$, denoting the permutation P.

Output Format

For each test case, output minimum number of operations required by Chef to make the permutation P `End Sorted`.

Constraints

- $1 \leq T \leq 1000$
- $2 \leq N \leq 10^5$
- P `is a permutation of` $\{1, 2, 3, \ldots N\}$
- `The sum of` N `over all test cases does not exceed` $3 \cdot 10^5$.

For each test case, output minimum number of operations required by Chef to make the permutation P `End Sorted`.

Constraints

- $1 \leq T \leq 1000$
- $2 \leq N \leq 10^5$
- P `is a permutation of` $\{1, 2, 3, \ldots N\}$
- `The sum of` N `over all test cases does not exceed` $3 \cdot 10^5$.

Sample 1:

Input	Output
4 4 1 3 2 4 3 3 2 1 2 2 1 3 2 1 3	0 3 1 1

Explanation:

Test case 1: P is already `End Sorted`.

Test case 2: P can be made `End Sorted` using 3 operations as follows: $[3, 2, 1] \rightarrow [2, 3, 1] \rightarrow [2, 1, 3] \rightarrow [1, 2, 3]$. It can be shown that achieving this in fewer than 3 moves is impossible.

Test case 3: P can be made `End Sorted` using one operation, by swapping 1 and 2.

Test case 4: P can be made `End Sorted` using one operation, by swapping 1 and 2.