

Insertion Sort Advanced Analysis

Extracted from: HackerRank

Source file name: analysis.py

Time limit: 3 seconds

Insertion Sort is a simple sorting technique which was covered in previous challenges. Sometimes, arrays may be too large for us to wait around for insertion sort to finish. Is there some other way we can calculate the number of times Insertion Sort shifts each elements when sorting an array?

If K_i is the number of elements over which the i^{th} element of the array has to shift, then the total number of shifts will be $k_1 + k_2 + \dots + k_N$.

Input

The first line contains a single integer, T ($1 \leq T \leq 15$), denoting the number of queries to perform. The subsequent lines describe each query over two lines:

1. The first line contains an integer, N ($1 \leq N \leq 100000$), denoting the number of elements to be sorted.
2. The second line contains N space-separated integers describing the respective values of $a[1] + a[2] + \dots + a[N]$ ($1 \leq a[i] \leq 10000000$).

The input must be read from standard input.

Output

Print T lines containing the required answer for each query.

The output must be written to standard output.

Sample Input 1	Sample Output 1
2	0
5	4
1 1 1 2 2	
5	
2 1 3 1 2	

Explanation

The first query is already sorted, therefore there's no need to shift any element. In the second case, it will proceed in the following way.

Array: 2 1 3 1 2 -> 1 2 3 1 2 -> 1 1 2 3 2 -> 1 1 2 2 3
Moves: - 1 - 2 - 1 = 4

This statement is based on a HackerRank Problem.