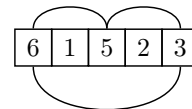


4 Number of Inversions

Problem Introduction

An inversion of a sequence a_0, a_1, \dots, a_{n-1} is a pair of indices $0 \leq i < j < n$ such that $a_i > a_j$. The number of inversions of a sequence in some sense measures how close the sequence is to being sorted. For example, a sorted (in non-descending order) sequence contains no inversions at all, while in a sequence sorted in descending order any two elements constitute an inversion (for a total of $n(n-1)/2$ inversions).



Problem Description

Task. The goal in this problem is to count the number of inversions of a given sequence.

Input Format. The first line contains an integer n , the next one contains a sequence of integers a_0, a_1, \dots, a_{n-1} .

Constraints. $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$ for all $0 \leq i < n$.

Output Format. Output the number of inversions in the sequence.

Sample 1.

Input:

```
5
2 3 9 2 9
```

Output:

```
2
```

The two inversions here are $(1, 3)$ ($a_1 = 3 > 2 = a_3$) and $(2, 3)$ ($a_2 = 9 > 2 = a_3$).

What To Do

This problem can be solved by modifying the merge sort algorithm. For this, we change both the **Merge** and **MergeSort** procedures as follows:

- **Merge**(B, C) returns the resulting sorted array and the number of pairs (b, c) such that $b \in B$, $c \in C$, and $b > c$;
- **MergeSort**(A) returns a sorted array A and the number of inversions in A .

Need Help?

Ask a question or see the questions asked by other learners at [this forum thread](#).