Problem B2
Sum of Pairs (Hard Version)

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

Given an array of n integers $a_1, a_2, \ldots, a_n$. Find the sum of $a_j - a_i$ over all integer pairs (i, j) satisfying $1 \leq i < j \leq n$ and $a_i < a_j$.

In other words, you are required to find the following sum:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \max(0, a_j - a_i)$$

Input Format

The first line of the input contains an integer n denoting the length of the array. The second line of the input contains n space-separated integers $a_1, a_2, \ldots, a_n$.

Output Format

Output the desired sum in one line.

Technical Specification

- $2 \leq n \leq 3 \times 10^5$
- $1 \leq a_i \leq 10^7$ for $i = 1, 2, \ldots, n$

Sample Input 1

```
5
4 10 3 8 2
```

Sample Output 1

```
15
```

Sample Input 2

```
7
82 283 194 30 201 30 217
```

Sample Output 2

```
1158
```

Hint

This problem is an application of the divide-and-conquer algorithm.

In each iteration of the algorithm, we divide the array into two parts and calculate the sum of the following three categories:

- Differences $a_j - a_i$ where both i and j are on the left side of the division.
- Differences $a_j - a_i$ where both i and j are on the right side of the division.
- Differences $a_j - a_i$ where i is on the left side, and j is on the right side of the division.

The first two sums can be found using recursion. The third sum can be calculated using a process similar to merge-sort: We simultaneously perform a merge sort on the array while computing the sum. For each element j on the right side, we calculate the sum of differences $a_j - a_i$ by determining the sum and count of elements a_i that satisfy the condition $a_i < a_j$. The whole algorithm runs in $\mathcal{O}(n \log n)$ time.

The same technique can also be applied to solve problems such as counting inversions in an array.