Competitive Programming and Mathematics Society

Programming Workshop #4

The Problem Solving Process

Angus Ritossa

Table of contents

Problem Statement

CPMSOC

You are given an array $A = \{A_1, ..., A_N\}$ of integers.

Problem Statement

You are given an array $A = \{A_1, ..., A_N\}$ of integers.

You must count the number of pairs (i, j) where $i < j$ and $A_i \neq A_j$.

Problem Statement

CPMSOC

You are given an array $A = \{A_1, ..., A_N\}$ of integers.

You must count the number of pairs (i, j) where $i < j$ and $A_i \neq A_j$.

Constraints:

- $2 \leq N \leq 300\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000$.

Problem Statement

CPMSOC

You are given an array $A = \{A_1, ..., A_N\}$ of integers.

You must count the number of pairs (i, j) where $i < j$ and $A_i \neq A_j$.

Constraints:

- $2 \leq N \leq 300\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000$.

Sample Input:

```
3
1 7 1
```

Problem Statement

CPMSOC

Arc CLUBS SUPPORTED BY Arc INDEPENDENTLY RUN

You are given an array $A = \{A_1, ..., A_N\}$ of integers.

You must count the number of pairs (i, j) where $i < j$ and $A_i \neq A_j$.

Constraints:

- $2 \leq N \leq 300\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000$.

Sample Input:

```
3
1 7 1
```

Sample Output:

```
2
```

$O(N^2)$ Solution

- We have two nested for loops, and consider each pair (i, j) individually.

$O(N^2)$ Solution

- We have two nested for loops, and consider each pair (i, j) individually.
- This is too slow to solve the problem, as $N^2 = 300\,000^2 = 90\,000\,000\,000$.

$O(N \log N)$ Solution

- We will instead count the number of pairs (i, j) where $A_i = A_j$ and $i < j$.

$O(N \log N)$ Solution

- We will instead count the number of pairs (i, j) where $A_i = A_j$ and $i < j$.
- If this count is x, then the answer to the original problem is $\frac{N(N-1)}{2} - x$.

$O(N \log N)$ Solution

- We will instead count the number of pairs (i, j) where $A_i = A_j$ and $i < j$.
- If this count is x, then the answer to the original problem is $\frac{N(N-1)}{2} - x$.
- Define c_j to be the number of $A_i = j$. We have that $x = \sum_j \frac{c_j(c_j-1)}{2}$.

$O(N \log N)$ Solution

- We will instead count the number of pairs (i, j) where $A_i = A_j$ and $i < j$.
- If this count is x, then the answer to the original problem is $\frac{N(N-1)}{2} - x$.
- Define c_j to be the number of $A_i = j$. We have that $x = \sum_j \frac{c_j(c_j-1)}{2}$.
- To find c_j for each j, we sort the array and count the ranges of values which are the same.

$O(N \log N)$ Solution

CPMSOC

- We will instead count the number of pairs (i, j) where $A_i = A_j$ and $i < j$.
- If this count is x, then the answer to the original problem is $\frac{N(N-1)}{2} - x$.
- Define c_j to be the number of $A_i = j$. We have that $x = \sum_j \frac{c_j(c_j-1)}{2}$.
- To find c_j for each j, we sort the array and count the ranges of values which are the same.
- The complexity of sort is $O(N \log N)$, and the complexity of the rest of the procedure is $O(N)$. This gives an overall complexity of $O(N \log N)$, which is fast enough.