Codeforces Problem 1902C

> **Problem.** Given an array of not necessarily positive integers $a_1, a_2, \ldots, a_n$, choose values a_{n+1}, x such that for each i, $M - a_i \geq 0$ and $x | M - a_i$ such that
>
> $$\text{cost} := \sum_{i=1}^{n+1} \frac{M - a_i}{x}$$
>
> is minimized, where M denotes the maximum of the array including a_{n+1}.

Solution. Observe that we can first simplify the form of cost:

$$\begin{aligned} \text{cost} &= \frac{1}{x} \sum_{i=1}^{n+1} M - a_i \\ &= \frac{1}{x}\Big((n+1)M - S\Big), \end{aligned}$$

where S is the sum of all the elements in the updated list. Observe that, if we were to know our maximum value, calculating the optimal value for x is easy:

$$x = \gcd(M - a_1, M - a_2, \ldots, M - a_{n+1}).$$

Thus, we must search for the best maximum value. There are two cases to oconsider:

1. The case where $M = a_{n+1}$, and
2. The case where $M \neq a_{n+1}$.

The latter case is far easier to handle since we don't have to choose the value of M, as we can calculate cost directly after finding x. The former case, on the other hand, is not so simple. This motivates the following claim:

Claim. The value of cost is optimal when $M = \max\{a_1, a_2, \ldots, a_n\}$. That is, M is strictly not equal to a_{n+1}.

The proof of this is omitted for time reasons, but it intuitively makes sense as increasing the maximum by x always increases the cost by n; whereas, keeping the maximum constant and adding a smaller element only increases the cost by n in the worst case.

With this, we can then simply choose a_{n+1} to be the greatest value of the form $M - kx$ which isn't already contained in the array.