

## Codeforces Problem 1909D

**Problem.** Given a list of  $n$  numbers  $a_1, a_2, \dots, a_n$  and an integer  $k$ , one can perform the following operation any number of times:

- Choose (and then delete) some number  $a_i$ , and
- Partition the sum of  $a_i + k$  into two numbers and then add these into the list.

Determine whether it is possible to make all elements in  $a$  equal using this operation, and if so, the minimum number of operations required to do so.

**Solution.** Our primary approach shall be to look at the sums after resulting operations done.

**Observation.** Suppose we perform  $q_i$  operations on some element  $a_i$  in order to split it into  $q_i + 1$  equal elements of some value  $d$ . This tells us that

$$d = \frac{a_i + kq_i}{q_i + 1} = \frac{a_i - k}{q_i + 1} + k.$$

Note that since we're trying to make all elements equal, we can chain this condition to see that

$$\begin{aligned} d &= \frac{a_1 - k}{q_1 + 1} + k = \frac{a_2 - k}{q_2 + 1} + k = \dots = \frac{a_n - k}{q_n + 1} + k \\ \implies \frac{a_1 - k}{q_1 + 1} &= \frac{a_2 - k}{q_2 + 1} = \dots = \frac{a_n - k}{q_n + 1}. \end{aligned}$$

Equivalently, for all pairs of elements  $(a_i, a_j)$ , the following must be satisfied (although note that we only ever need to check  $n - 1$  consecutive pairs)

$$(a_i - k)(q_j + 1) = (a_j - k)(q_i + 1).$$

One may solve this equation for  $q_i, q_j$  as normal, and to do this, the first order of business is to divide out the common factors from both sides. This strategy leads to the following claim.

**Claim.** We have that for each index  $i$ ,

$$q_i + 1 = \frac{a_i - k}{\gcd(a_1 - k, a_2 - k, \dots, a_n - k)}.$$

One should note that there are a couple edge cases to keep in mind when using the following:

- Case  $a_1 - k = a_2 - k = \dots = a_n - k = 0$ : This should be a valid arrangement, but it encounters a division by 0. This is easily circumvented, however, by initially checking if all elements are equal (or simply checking if the greatest common factor is 0, as any equal element sequence should still give the correct answer with this algorithm).
- Case  $q_i < 0$ : This can happen a few ways, but luckily this is a bug, not a feature. Whenever this case is encountered, it is impossible to make all elements equal.

Once each  $q_i$  is determined, we can sum them to find our resulting answer.

**Remark.** While the following was not used in the final solution, it came up while brainstorming some possible methods, and it does in theory provide an alternate method for obtaining  $q$  in terms of  $d$ .

Let the minimum number of operations be  $q$ . Then the equal element should be

$$d = \frac{1}{n + q} \left( qk + \sum_{i=1}^n a_i \right).$$

In particular, this is equivalent to

$$q = \frac{1}{d - k} \left( \sum_{i=1}^n a_i - dn \right).$$