Array Balancing

Walid Krichene

1 Problem

Let A_0 be an array of N non-negative integers, indexed from 0 to $N-1$. We assume that the sum of its elements is a multiple of N

$$\sum_{n=0}^{N-1} A_0(n) = N.a$$

where a is a positive integer. We would like to balance the array in a minimum number of iterations, where each iteration transforms the array using a number of allowed transfers described below. Balancing the array means transforming it into the uniform target array $T = [a, \dots, a]$. To describe the allowed transfers, let A_i denote the state of the array at the beginning of iteration i. During iteration i, each postive cell $A_i(n) > 0$ can transfer one unit to its left neighbor (left transfer), or transfer one unit to its right neighbor (right transfer). A left transfer is described by the operations

$$\begin{aligned} A_{i+1}(n-1) &= A_i(n-1) + 1 \\ A_{i+1}(n) &= A_i(n) - 1 \geq 0 \end{aligned}$$

Note that a left transfer is not possible for the left-most cell $A_i(0)$ (similarly, a right transfer is not possible for the right-most cell). Also note that all transfers occur simultaneously, meaning that in the example array $A_0 = [3, 0, 0]$ we cannot make a right transfer from $A(0)$ to $A(1)$ *then* a right transfer from $A(1)$ to $A(2)$ in a single iteration, obtaining $A_1 = [2, 0, 1]$. This transformation will require two iterations.

Example In the following example we start from array $A_0 = [0, 3, 0]$. Balancing this array requires a minimum of 2 iterations. Here is one example of an optimal sequence of transformations

$$\begin{aligned} A_0 &= [0, 3, 0] \\ A_1 &= [1, 2, 0] \\ A_2 &= [1, 1, 1] \end{aligned}$$

We would like to find the minimum number of iterations required to balance any given array A_0.

2 Notations

We start by defining, $\forall n$, the left sum $A_i^-(n)$, the target left sum $T^-(n)$, the right sum $A_i^+(n)$ and the target right sum $T^+(n)$ as follows

$$A_i^-(n) = \sum_{k=0}^{n-1} A_i(k) \qquad\qquad A_i^+(n) = \sum_{k=n+1}^{N-1} A_i(k)$$

$$T^-(n) = \sum_{k=0}^{n-1} T(k) = n.a \qquad\qquad T^+(n) = \sum_{k=n+1}^{N-1} T(k) = (N-n-1)a$$

$A_i^-(n)$ is the sum of elements that are (strictly) to the left of cell n at the beginning of iteration i. $T^-(n)$ is the target left sum.

$$A_i = \overset{A_i^-(n)}{[\dots]}\ A_i(n)\ \overset{A_i^+(n)}{[\dots]}$$

We will say that the left sub-array $A_i[0:n-1]$ has a deficit if $A_i^-(n) < T^-(n)$, and that the right sub-array $A_i[n+1:N-1]$ has a deficit if $A_i^+(n) < T^+(n)$. One important observation is that in order to balance array A_i, if sub-array $A_i[0:n-1]$ has a deficit, we will need to make $T^-(n) - A_i^-(n)$ left transfers from cell n, and similarly, if sub-array $A_0[n+1:N-1]$ has a deficit, we will need to make $T^+(n) - A_i^+(n)$ right transfers from same cell n. This observation allows us to have a lower bound on the number of iterations.

3 A lower bound

Let

$$\begin{aligned} f_i(n) &= f_i^-(n) + f_i^+(n) \\ &= \max(T^-(n) - A_i^-(n), 0) + \max(T^+(n) - A_i^+(n), 0) \end{aligned}$$

$f_i^-(n) = \max(T^-(n) - A_i^-(n), 0)$ is the exact number of left transfers from cell n required to balance $A_i[0:n-1]$. Similarly, $f_i^+(n)$ is the exact number of right transfers required to balance sub array $A_i[n+1:N-1]$. $f_i(n) = f_i^-(n) + f_i^+(n)$ can be described as the outgoing flow from cell n required to balance left and right sub arrays. Since we cannot make more than one transfer at each iteration, we need at least $f_i(n)$ iterations starting from array A_i. Since this is true $\forall n$, we need at least

$$I_i = \max_n f_i(n)$$

iterations. Also note that the problem is solved (i.e. $A_i = T$) *iff* $I_i = 0$.

Therefore, we have a lower bound on the number of required iterations. We need at least I_0 iterations starting from array A_0. We will now show that this lower bound is in fact the optimal number of iterations, by designing a strategy that strictly decreases I_i at each iteration (such a strategy is guaranteed to balance the array in at most I_0 iterations, which proves the optimality)

4 Algorithm

We propose the following algorithm

During iteration i
- If $A_i = T$ the problem is solved
- Else $\forall n$, if $A_i(n) > 0$
  - If $f_i^-(n) > 0$ (the left sub array has a deficit), we make a left transfer from $A_i(n)$
  - Else if $f_i^+(n) > 0$ (the right sub array has a deficit), we make a right transfer from $A_i(n)$

Claim the algorithm strictly decreases I_i at each iteration $\forall i$, $I_{i+1} < I_i$.

5 Proof

We consider an iteration i such that $I_i > 0$ (otherwise the array is balanced and the problem is solved). Let n_0 be the index of a cell where the maximum outgoing flow is reached $f_i(n_0) = I_i = \max_n f_i(n)$. (note that n_0 is not necessarily unique, but the following argument holds independently for any such n_0)

5.1 $A_i(n_0) > 0$

Let us first show that $A_i(n_0) > 0$. We will reason by contradiction and assume $A_i(n_0) = 0$.

$$A_i = \overset{A_i^-(n_0)}{[\ldots]}\ 0\ \overset{A_i^+(n_0)}{[\ldots]}$$

Since $f^-(n_0) + f^+(n_0) = I_i > 0$ we have

- either $f^-(n_0) > 0$ and $f^+(n_0) = 0$. This results in [1]

$$f(n_0+1) \geq f^-(n_0+1) = f^-(n_0) + a - 0 > f^-(n_0) = f(n_0)$$

  thus $f(n_0+1) > f(n_0)$ (this translates to "if $A_i[0 : n_0 - 1]$ has a deficit and $A_i(n_0) = 0$, then $A_i[0 : n_0]$ has an even bigger deficit") which contradicts hypothesis $f(n_0)$ is maximal.

- either $f^-(n_0) = 0$ and $f^+(n_0) > 0$ which similarly results in $f(n_0-1) > f(n_0)$ which in turn contradicts the hypothesis $f(n_0)$ is maximal.

- either $f^-(n_0) > 0$ and $f^+(n_0) > 0$ which is not possible (since it would mean $A_i[0 : n_0 - 1]$ and $A_i[n_0 + 1 : N - 1]$ both have a deficit, and this is not possible when $A(n_0) = 0$).

$$t = \overset{A^-(n_0)<T^-(n_0)}{[\ldots]}\quad 0 \quad \overset{A^+(n_0)<T^+(n_0)}{[\ldots]}$$

5.2 $I_{i+1} < I_i$

So far we have shown that $A_i(n_0) > 0$, and according to our initial hypothesis, $f_i(n_0) = I_i > 0$. Thus, the algorithm will make a transfer from $A_i(n_0)$ to one of its neighbors. We assume that there is a left transfer (the right transfer case is similar)

So we assume that $f_i^-(n_0) > 0$ (left sub array has a deficit) and that we make a left transfer from $A_i(n_0)$ to $A_i(n_0 - 1)$. If we can prove that there is no right transfer from $A_i(n_0 - 1)$ to $A_i(n_0)$, we would have

$$f_{i+1}^-(n_0) = f_i^-(n_0) - 1 \text{ the left deficit is decreased by 1}$$
$$f_{i+1}^+(n_0) \leq f_i^+(n_0) \text{ the right deficit is not increased (this never happens)}$$

which results in $f_{i+1}(n_0) \leq f_i(n_0) - 1$. Since this is true $\forall n_0$ such that $f_i(n_0) = I_i$, we will have at the end of the iteration $I_{i+1} \leq I_i - 1$

So all we need to do is show that we cannot have a left transfer from $A_i(n_0)$ to $A_i(n_0 - 1)$ and a right transfer from $A_i(n_0 - 1)$ to $A_i(n_0)$ at the same time. If we assume this happens, we would have

- the right transfer $A(n_0 - 1) \rightarrow A(n_0)$ implies $f_i^+(n_0 - 1) > 0$ (i.e. $A_i[n_0 : N - 1]$ has a deficit)
- the left transfer $A(n_0 - 1) \leftarrow A(n_0)$ implies $f_i^-(n_0) > 0$ (i.e. $A_i[0 : n_0 - 1]$ has a deficit)

which is impossible (sub arrays $A_i[n_0 : N - 1]$ and $A_i[0 : n_0 - 1]$ form a partition of array A_i and cannot both have a deficit).

1.

$$f(n_0+1) \geq f^-(n_0+1) \text{ since } f^+(n_0+1) \geq 0 \text{ by definition}$$
$$f^-(n_0+1) = f^-(n_0) + a - 0 \text{ since } f^-(n_0+1) = T^-(n_0+1) - A^-(n_0+1) = \left(T^-(n_0) + a\right) - \left(A^-(n_0+1) + A(n_0)\right) = f^-(n_0) + a - 0$$
$$f^-(n_0) = f(n_0) \text{ since } f^+(n_0) = 0 \text{ according to the hypothesis}$$

6 Notes

- Note that it is possible to drop the constraint $A_i(n) > 0$ (i.e. allow transfers regardless of whether the cell has elements) without changing the optimal number of iterations (but the actual iterations will be different). This is due to the fact that 'critical' transfers (those that decrement I_i) necessarily have $A_i(n) > 0$.

- This solution works for any target array T of size N that satisfies $\sum_n T(n) = \sum_n A(n)$. All we have to do is work with the general definition of T^- and T^+

$$T^-(n) = \sum_{k=0}^{n-1} T(k) \qquad\qquad T^+(n) = \sum_{k=n+1}^{N-1} T(k)$$