Remittance (Solution)

Subtask 1

We can solve Subtask 1 if we search all possible ways of remittance. In fact, if $A_i \leq 5$, it can be proved that the amount of money in every house does not exceed 9 yen if we use the remittance service in any way. Since $N \leq 7$, there are at most 10^N possible states of the amount of money in each house. We can search all of them in time.

Subtasks 2 and 3

In order to solve Subtask 2, we shall solve equations. Let us consider the process where the amount of money in a house becomes B yen. Let X_i be the amount of money sent by House i. House i pays $2X_i$ yen, and receives X_{i-1} yen. Here we put $X_0 = X_N$. Then we have the equation

$$B_i - A_i = X_{i-1} - 2X_i.$$

Let us solve it.

Taking the sum of the equation times 2^{i-1} for every House i, we get

$$\sum_{i=1}^{N} 2^{i-1} (B_i - A_i) = \left(1 - 2^N\right) X_N.$$

From this, we get $X_N = X_0$. Then, solving the equation for each House i in order, we get X_i.

Each X_i much be a non-negative integer. If this condition is not satisfied, the answer is `No`. In fact, if every X_i is a non-negative integer and at least one of B_i is non-zero, the answer is `Yes`. We shall prove this.

We consider the operations in the opposite direction. We shall transform B into A.

If some entry of the sequence B is non-zero and the sequence A is different from B, we have $B_{i+1} > 0$ and $X_i > 0$ for some i. In fact, if this condition is not satisfied, we have $B_{i+1} = 0$ for some i with $X_i > 0$. Since $A_{i+1} \geq 0$, we have $X_{i+1} > 0$ by the above equation. Hence we have $X_i > 0$ for every i, and $B_i = 0$, which is absurd.

For such i, we apply the operation which is opposite to sending 1 yen from House i to House $i + 1$ to the sequence B. Then B_{i+1} decreases by 1, B_i increases by 2, and X_i decreases by 1. Repeating this operation, we can transform B into A.

Therefore, if all entries of B are 0, the answer is `Yes` if all entries of A are 0. Otherwise, the answer is `No`.

In Subtask 2, the value of

$$\sum_{i=1}^{N} 2^{i-1}(B_i - A_i)$$

can be calculated as a 64 bit integer. In Subtask 3, this value can be very large. Even in such cases, we can solve the task in $O(N)$ time using multiple precision integers.

The sample source file `2019-open-remittance-sol1.cpp` implements this solution.

Another solution

In fact, we can solve this task by the following simulations.

For each House i, if it has a yen, send $\left\lceil \frac{a - B_i}{2} \right\rceil$ yen if $a > B_i$. Repeat this for each i. We increase i by 1 in each step. When i becomes N, we return to $i = 1$. Note that the amount of money in the House can be -1 in this process.

If the answer is `Yes`, after repeating this process in $N + O\left(\log \sum_{i=1}^{N} A_i\right)$ steps, the amount of money in every House becomes B. Let us prove this fact. During this simulation, the amount of money sent from House i does not exceed X_i yen. Thus, from the intermediate status, there is a way to make the amount of money in every House be B. After one cycle of i, the amount of money in every House does not exceed B except possibly for one House. Moreover, the excess for that House becomes half after each step. Therefore, after $N + O\left(\log \sum_{i=1}^{N} A_i\right)$ steps, the amount of money in every House becomes B.

By the above argument, we can also show that the answer is `Yes` if we can make the amount of money in every House be B and the sequence B contains a non-zero value.

The sample source file `2019-open-remittance-sol2.cpp` implements this solution.