March of the Sheep

Problem author: Alexander Babin

Problem developer: Yuri Fedorov

1 $T = 2$

Notice that every second the parity of each sheep's position changes. Therefore, if there exist two sheep with different position parity, then the answer is obviously `No`, since there will always be a sheep on both an even and an odd position. Otherwise, it is claimed that the answer always exists. For example, the following algorithm works:

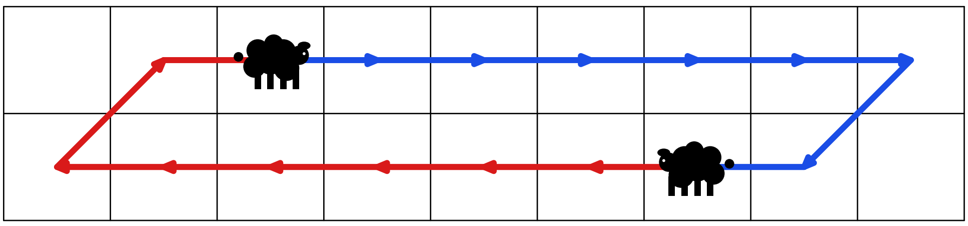
If the parity of the current last column does not match the parity of the sheep positions, then it is always possible to cut off a segment of length 1 (since there is guaranteed to be no sheep in the last column, due to parity). Thus, we can reduce the number of columns to 1, where all sheep are guaranteed to move in the same way.

2 $n \le 3, m \le 4$

If we solve $T = 1$, then there are very few remaining cases, and they can be checked by hand, or we can do brute force. As the state, we can take the positions of each sheep, as well as the length of the segment. Then there are two kinds of transitions: we simulate one second, or we decrease the segment. This can be implemented using any graph algorithm. There are also other brute-force approaches.

3 $n = 2$

In the case $a_1 = 1, a_2 = m$, we can draw it and see that the answer is $\frac{m+1}{2}$, and it is enough to cut down to this size after this same amount of time.



Now let us look at the two distances marked in blue and red. This is the distance from the first sheep to the second, and from the second to the first in the order of their path. Then notice two facts:

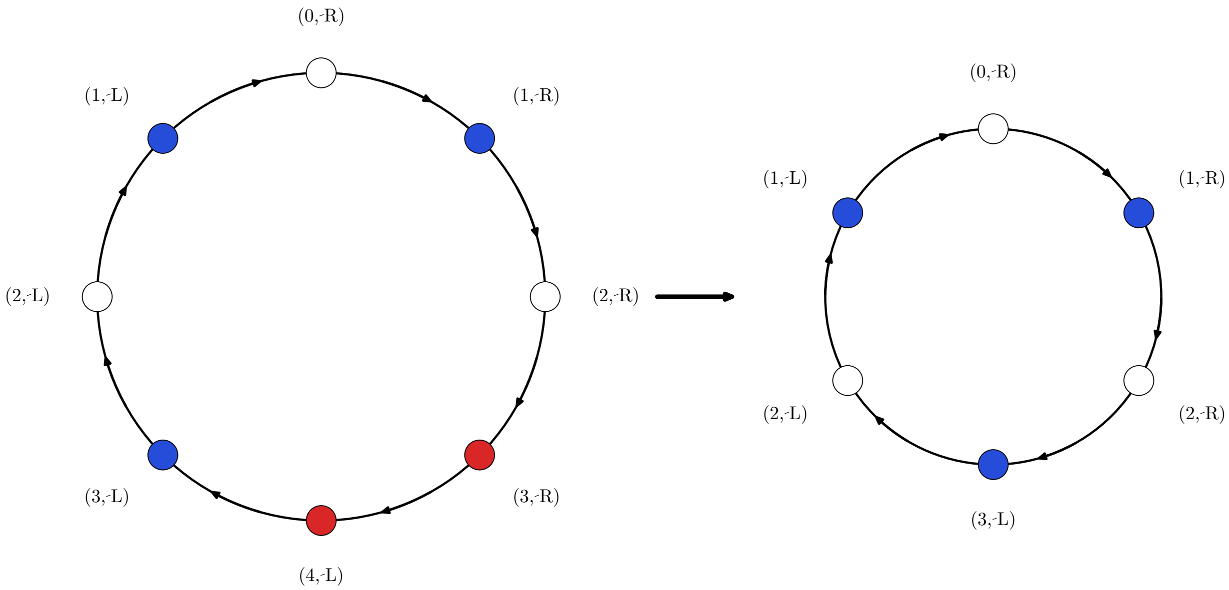
1. The sheep will move in the same way if and only if the distance from one to the other is exactly zero.
2. After decreasing the field size by 1, one of these distances decreases by 2.

From this, an $O(m)$ solution follows immediately: we can simulate their movement, and if cutting decreases the smaller of the distances, then we cut the segment by 1. In $4 \cdot m$ operations, this process is guaranteed to finish.

Now let us learn how to solve it in $O(1)$: notice that if we keep the larger of the distances, then if this length is x, the answer will be $\frac{x}{2} + 1$, since if we kept length x, then the distance from one sheep to the other would be $2 \cdot x - 2$. The details of reconstructing the answer will be considered in the full solution.

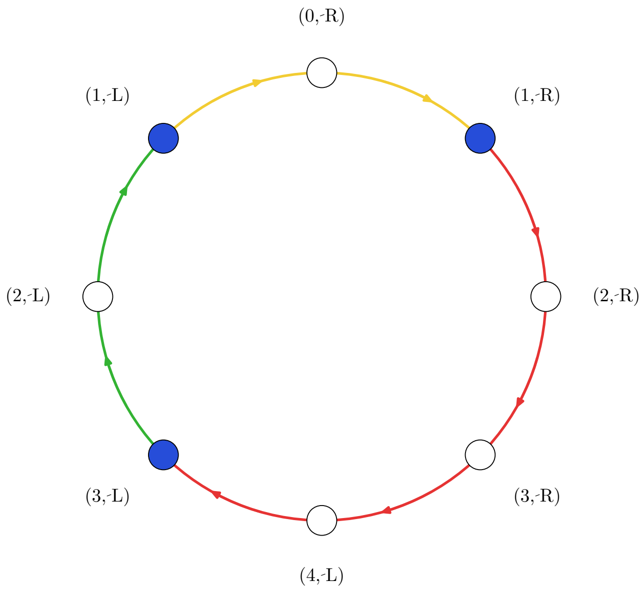
4 $T = 2$

Let us slightly develop the ideas from $n = 2$. Notice that the segment can be represented as a cycle of length $2 \cdot m - 2$. And decreasing the segment length by 1 is equivalent to decreasing the cycle length by 2:



This picture shows an example where 3 sheep are in positions $(2, 4, 2)$, and their movement directions are `RLL` (that is, the blue points are the sheep). For convenience of the further explanation, the positions are shifted to 0-indexing. The red vertices mark those that are removed after decreasing the cycle.

Then the distances to neighboring sheep can be represented as the number of arcs from a sheep to the next sheep along the cycle (here different colors denote different arcs, which represent distances to neighboring sheep):



Then notice that this construction is not fundamentally different from the $n = 2$ situation. We are still only interested in the maximum distance between neighboring sheep on the cycle. And, as in the $n = 2, m \leq 2 \cdot 10^5$ version, we can simulate the movement and cut at moments when the current length is not the maximum; in this case the complexity is $O(n \cdot m)$. Or we can say that the answer is $\frac{x}{2} + 1$, where x is the maximum distance between neighboring sheep.

5 Full solution

Now we only need to learn how to reconstruct. In the case $n, m \leq 1000$ we already know how to solve it, so it remains to learn how to reconstruct under the full constraints.

Let us sort all sheep by their position on the cycle. And let i be the position of a sheep such that the arc from sheep i to the next sheep on the cycle is the largest. Then reconstruction can be done as follows:

- Renumber the array so that sheep number i becomes the last sheep.
- Let us wait long enough so that the last sheep is at position $m-1$ on the cycle in 0-indexing (in other words, so that it stands in the last column).
- Then, to cut the arc from sheep $n-1$ to sheep n, it is enough to "rotate"the cycle so that the sheep is at position $m-1+\frac{d}{2}$, where d is the length of the arc from sheep $n-1$ to sheep n. Then decrease the segment length by $\frac{d}{2}$, after which sheep number n will stand at position $m-1$ on the cycle.
- After that, we can ignore sheep number n and process sheep number $n-1$, and so on.

And it is easy to notice that this indeed removes all arcs except the maximum one.