Code Jam 2011 - Round 3

Analysis: Perpetual Motion

The first thing we need to notice is that if two lemmings do not end up in the same square after one second, they will never end up in the same square. That is because after one second there will be exactly one lemming in each square, and the state is exactly the same as the second before.

The number of combinations for the conveyor belt directions is $2^{R \cdot C}$, so we can afford to try all of them to solve the easy input and count how many combinations lead to all lemmings being in different squares after one second.

To solve the hard input, we'll have to take a look at the problem from a graph theory perspective. Suppose we created a bipartite graph like this:

- For each cell (r, c), create two nodes $start_{r,\ c}$ and $end_{r,\ c}$.
- Create an edge between $start_{r_1,\ c_1}$ to $end_{r_2,\ c_2}$ if there is a way to choose the direction of the conveyor belt in cell (r_1, c_1) such that the lemming ends up in (r_2, c_2) after one second.

All start nodes will end up being incident to exactly two edges because there are exactly two different cells where a lemming can end up in one second starting from any given cell. End nodes, on the other hand, can be incident to 0-8 edges, depending on the conveyor belt orientations of the neighbouring cells.

Let's observe the graph some more. There are two rules that apply:

1. If there are no edges incident to a node $end_{r,\ c}$ then no lemming can end up in cell (r, c) in one second, so there will be two lemmings in some other cell no matter how we direct the conveyor belts. In that case the answer is simply 0, so we can proceed to the next test case.
2. If there is a node $end_{r_2,\ c_2}$ incident to only one edge leading to $start_{r_1,\ c_1}$ then we have no other choice but to direct the conveyor belt on the cell (r_1, c_1) to lead to the cell (r_2, c_2). Then we can simply remove both $end_{r_2,\ c_2}$ and $start_{r_1,\ c_1}$ from the graph along with all the incident edges.

We can apply these rules iteratively until they can not be applied anymore.

Let N be the number of start nodes left after the above process is done. The number of end nodes is also equal to N, because we removed them in pairs.

The number of edges incident to each start node is still equal to two, so total the number of edges is equal to 2·N because the graph is bipartite. We also know that each end node is now incident to at least two edges, because the above rules do not apply anymore.

But, if any of the end nodes had more than two incident edges then the number of edges incident to all the end nodes combined would be greater than 2·N. This would contradict the fact that the number of edges is equal to 2·N. Therefore, the number of edges incident to each end node is also equal to two.

Any graph having all node degrees equal to two is in fact a set of cycles. For bipartite graphs, the length of each cycle is even. So, we are left with K cycles of even length which we can solve independently and multiply the individual numbers to get the final number.

To solve the circle, select any cell (r_1, c_1) and pick a conveyor belt direction. The lemming ends up in (r_2, c_2). Now remove $start_{r_1, c_1}$ and end_{r_2, c_2} from the graph along with incident edges. This will break up the cycle, and we can proceed to apply rule #2 until we decide the conveyor belt for all the remaining cells. Because the cycle has even length and we always remove nodes in pairs, the rule #1 will never apply.

We can pick the direction of the cell (r_1, c_1) in two ways. So the answer for any cycle is always 2. So the final answer is equal to 2^K (modulo 1000003).