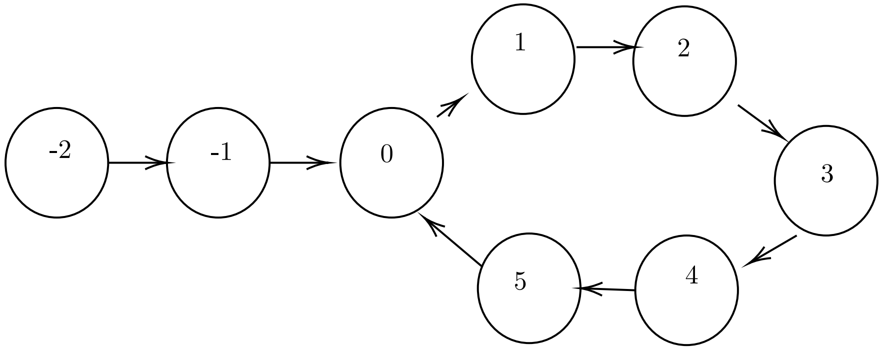
Find Loop LeetCode:



Given a path T from $-m$ to 0 and cycle C with nodes $0,\dots,n-1$, where T is a path of size m and cycle C is of size n. By the division algorithm, we have $m = kn + r$ where $0 \leq r < n$ and where $k \geq 0$.

Suppose we have a slow pointer (moves 1 step at a time) and a fast pointer (moves 2 step at a time) that both start at node $-m$. When both of these moves m times, we have the slow pointer is at node 0 and fast pointer is at node r.

Remark: Observe that if the slow and fast pointer are in the cycle and moves once; it means that the fast pointer is 1 step closer to the slow pointer (to see this, put the slow at 0 and fast at 4. Then when they both move once, the fast is 1 step behind the slow).

Going back to the discussion, we need to take $C - r$ steps so that both slow and fast pointer are on the same node. More specifically, they are on node $C - r$.

Now let first pointer to be on node $-m$ and second pointer to be on $C - r$. Observe that $m = kn + r$. If $k = 0$, both pointers are r distance from node 0; we're done. If $k = 1$, we move both pointers n distance. We see that the second node is at the same spot, but the first pointer is r distance from node 0. Also the second node is r distance from node 0 counterclockwise. If we assume for all $k \geq 0$, this holds, we have the inductive step.