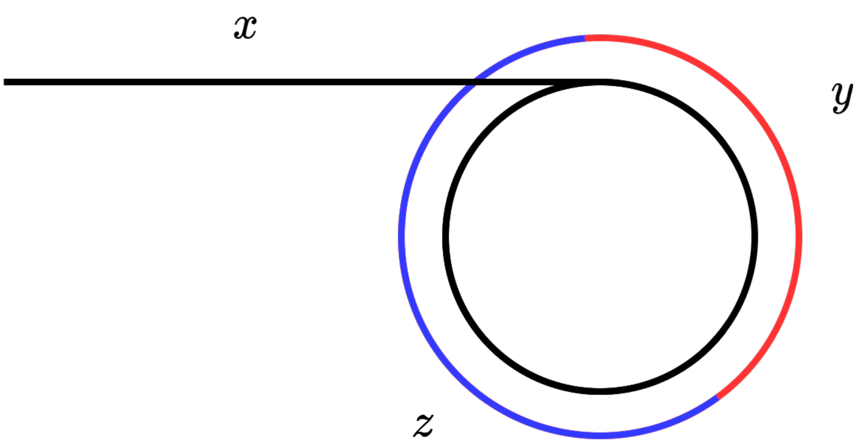
Floyd's cycle detection algorithm

1 Why the tortoise and rabbit meet?

- Imagine a loop (and nothing else) of n nodes so it's a polygon with n vertices.
- Place the tortoise and the rabbit on any two nodes of your choice.
- The key is to not really number any of the nodes so they are indistinguishable.
- Now, let the distance between the rabbit and the tortoise be d hops.
- At each step, the tortoise moves to the adjacent node, while the rabbit hops two nodes. Now since the nodes are indistinguishable, we can place ourselves in the frame of reference of the tortoise.
- Then, from the perspective of the tortoise, after each step the tortoise remains where it was while the rabbit moves one hop closer to the tortoise.
- Thus the sequence of distances eventually approaches 0 ($\{d, d-1, d-2, \ldots, 1, 0\}$.
- So, whenever tortoise and rabbit enter the same loop, they'll definitely meet at some point.
- QED $\blacksquare$

2 Where do they meet?



- Begin by looking at the above diagram. x is the non cycle distance, y is where in the cycle rabbit and tortoise meet, z is the rest (distance left to start of the cycle).
- When the two meet, tortoise covers $x+\lambda_1 n+y$ distance for some λ_1 (where distance is number of nodes they hop on). λ_1 represents the number of times the two go around the cycle from the start to meet at y.
  Note: $\lambda_1 \leq 1$. Why?
- Also, rabbit covers $x + \lambda_2 n + y$.
- Since rabbit moves at twice the speed of tortoise:

$$\begin{aligned} 2(x+\lambda_1 n+y) &= x+\lambda_2 n+y \\ \Rightarrow 2x+2\lambda_1 n+2y &= x+\lambda_2 n+y \\ \Rightarrow x &= \lambda n - y, \text{ where } \lambda = \lambda_2 - \lambda_1 \text{ and since } \lambda_2 \geq \lambda_1,\ \lambda \geq 0. \end{aligned}$$

- Now, let $x = \alpha n + \beta$, then:

$$\begin{aligned} & x \bmod n = -y \bmod n = z \bmod n \\ \Rightarrow\ & \beta \bmod n = \quad z \bmod n \\ & \text{and since both } \beta < n \text{ and } z < n \text{ the equality holds without "mod"} \\ \Rightarrow\ & \beta = z \end{aligned}$$

- What this means is we can move one of the pointers to the head of the LL while leave the other at the meeting point (and we no longer distinguish between tortoise or rabbit, they are both tortoise now.) Then we can start both of these moving one hop at a time. The one at the head will move αn times $\beta = z$ will be left in the non cycle segment, while the one inside the cycle will move in cycles *alpha* times and stay at the location y in the loop with z left. After next $\beta = z$ moves, the two will meet again at the beginning of the loop. ■