Deterministic Trajectory Intersection Analysis

1 Problem Statement

We consider two deterministic agents moving on a finite discrete state space of size L, represented as:

$$\{0, 1, 2, \ldots, L-1\}$$

Each agent follows a deterministic trajectory defined by:

- an initial position
- a fixed step size applied at every discrete time step

Objective:

1. Determine whether the trajectories of the two agents intersect
2. If they intersect, compute the earliest time of intersection

The solution must be derived analytically, without brute-force simulation.

2 Deterministic Trajectory Model

Let:

- Agent 1 start at position a_1 with step size b_1
- Agent 2 start at position a_2 with step size b_2

At discrete time $t \geq 0$, the positions are given by:

$$P_1(t) = (a_1 + b_1 t) \bmod L$$

$$P_2(t) = (a_2 + b_2 t) \bmod L$$

The modulo operation ensures all positions lie within the finite state space.

3 Trajectory Intersection Condition

The trajectories intersect at time t if:

$$P_1(t) = P_2(t)$$

Substituting the position functions:

$$(a_1 + b_1 t) \equiv (a_2 + b_2 t) \pmod{L}$$

Rearranging terms:

$$(b_1 - b_2)t \equiv (a_2 - a_1) \pmod{L}$$

Define:

$$\Delta b = b_1 - b_2, \quad \Delta a = a_2 - a_1$$

The problem reduces to solving the linear congruence:

$$\boxed{\Delta b \cdot t \equiv \Delta a \pmod{L}}$$

4 Solvability Criterion

A linear congruence of the form

$$Ax \equiv C \pmod{M}$$

has a solution if and only if:

$$\gcd(A, M) \mid C$$

Applying this condition:

$$\boxed{\gcd(\Delta b, L) \mid \Delta a}$$

If this condition is not satisfied, the trajectories never intersect.

Let:

$$g = \gcd(\Delta b, L)$$

5 Reduction of the Congruence

Assuming $g \mid \Delta a$, divide the congruence by g:

$$\frac{\Delta b}{g} \cdot t \equiv \frac{\Delta a}{g} \pmod{\frac{L}{g}}$$

Define:

$$B = \frac{\Delta b}{g}, \quad A = \frac{\Delta a}{g}, \quad M = \frac{L}{g}$$

This yields:

$$\boxed{Bt \equiv A \pmod{M}}$$

Since $\gcd(B, M) = 1$, a unique solution exists modulo M.

6 Modular Inverse via Extended Euclidean Algorithm

To solve:

$$Bt \equiv A \pmod{M}$$

we require the modular inverse of B modulo M.

The Extended Euclidean Algorithm computes integers x, y such that:

$$Bx + My = \gcd(B, M)$$

Since $\gcd(B, M) = 1$, this implies:

$$Bx \equiv 1 \pmod{M}$$

Hence:

$$B^{-1} \equiv x \pmod{M}$$

This method is used to compute the modular inverse efficiently in $\mathcal{O}(\log M)$ time.

7 Earliest Intersection Time

Multiplying both sides of the reduced congruence by B^{-1}:

$$\boxed{t \equiv A \cdot B^{-1} \pmod{M}}$$

General Solution

$$t = t_0 + kM, \quad k \in \mathbb{Z}_{\geq 0}$$

Earliest Intersection Time

$$\boxed{t_{\min} = t_0}$$

where:

$$t_0 = (A \cdot B^{-1}) \bmod M$$

8 Special Cases

8.1 Same Initial Position

If:

$$a_1 \equiv a_2 \pmod{L}$$

Then:

$$\Delta b \cdot t \equiv 0 \pmod{L}$$

The earliest positive intersection time is:

$$\boxed{t = \frac{L}{\gcd(\Delta b, L)}}$$

If $\Delta b = 0$, the agents remain synchronized and intersect at $t = 0$.

8.2 Identical Step Sizes

If:

$$b_1 = b_2$$

Then:

- If $a_1 \equiv a_2 \pmod{L}$, intersection occurs at $t = 0$
- Otherwise, the trajectories never intersect

9 Handling Negative Step Sizes

Negative step sizes represent motion in the opposite direction.

Since all motion occurs modulo L, step sizes are normalized as:

$$b \leftarrow (b \bmod L + L) \bmod L$$

This preserves the trajectory while simplifying arithmetic.

—

10 Final Result Summary

The trajectories intersect if and only if:

$$\boxed{\gcd(b_1 - b_2, L) \mid (a_2 - a_1)}$$

The earliest intersection time is given by:

$$\boxed{t = \begin{cases} 0, & a_1 \equiv a_2 \pmod{L},\ b_1 = b_2 \\ \dfrac{L}{\gcd(b_1 - b_2, L)}, & a_1 \equiv a_2 \pmod{L}, b_1 != b_2 \\ (A \cdot B^{-1}) \bmod M, & \text{otherwise} \end{cases}}$$

where:

$$g = \gcd(b_1 - b_2, L), \quad A = \frac{a_2 - a_1}{g}, \quad B = \frac{b_1 - b_2}{g}, \quad M = \frac{L}{g}$$

11 Algorithmic Complexity

- **Time Complexity:** $\mathcal{O}(\log L)$
- **Space Complexity:** $\mathcal{O}(1)$

No simulation over time steps is required.

12 Conclusion

This analysis reduces deterministic trajectory intersection to solving a linear congruence over a finite cyclic group. By applying number-theoretic results and using the Extended Euclidean Algorithm to compute modular inverses, the intersection condition and earliest meeting time are determined efficiently and exactly.