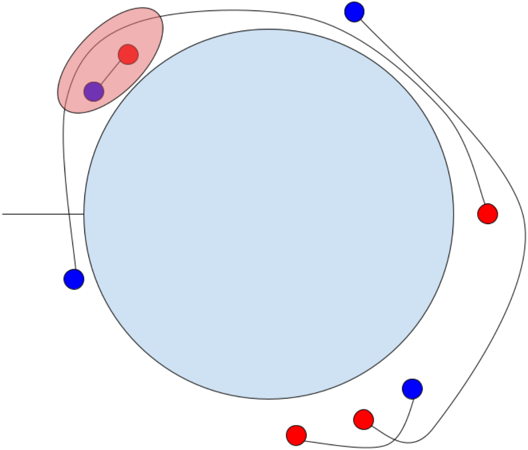
Loop Town – Editorial

Robert Cummings / Eliden

May 14, 2021

Sample Input



Input Processing

There are only two relevant pieces of information from the input:

- The relative cyclic order of starting positions
- The relative cyclic order of final positions

So, we may assume the starting and final positions are in $\{0, 1, \ldots, n-1\}$ (call this set $[n]$). We can also "cycle" them without changing the answer.

Then we have a permutation p such that $p(i)$ is the final position of the person with starting position i.

Choice of f

A single person has the choice of several paths that do not easily reduce to each other (e.g. counter-clockwise, clockwise, clockwise plus one extra rotation). These correspond to the winding number of the path.
We can represent this with an integer f_i such that $f_i \mod n = p_i$. That is, we can add an integer multiple of n to p_i to get f_i.

Define $x_i = f_i - i$. Intuitively, x_i is the displacement travelled by person i.

Determining Crossings from f

There are many choices of x, f, but once we fix it, the minimum number of crossings can be determined as follows.

Consider all pairs i, j of distinct people. Consider the relative position of j relative to i, It changes from $j - i$ to $f_j - f_i$.

The number of crossings of i and j is the number of times the interval from $j - i$ to $f_j - f_i$ contains a multiple of n.

Choosing x

Currently there are too many choices of x, but there is a way to leave only $O(n)$ possibilities!

There is an optimal choice of x for which $|x_i - x_j| < n$ for all i, j.

One such choice is $x_i = (p_i - i) \mod n$. We can add or subtract n from elements x_i, so that they fit in a different interval of n elements (say, $[k, k + n)$).

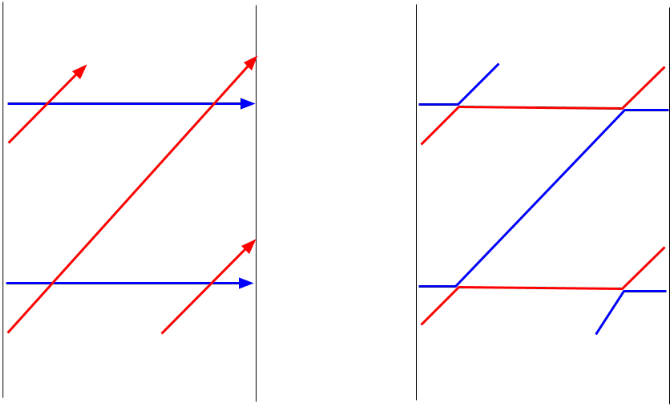
Proving the Theorem

The theorem $|x_i - x_j| < n$ is really hard to prove! The proof has three main steps:

1. Two people cross at most once
2. If $x_i - x_j > n$, we can subtract n from x_i and add n to x_j
3. Go from $|x_i - x_j| \leq n$ to $|x_i - x_j| < n$

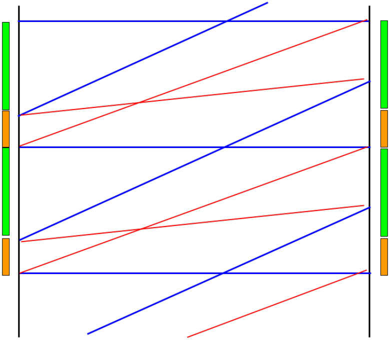
Theorem Part 1

We may assume two people cross at most once. Otherwise, suppose person i crosses person j twice. Then we can avoid these crossings by having person j walk the path of person i. This doesn't affect crossings with other people.



Theorem Part 2

We claim $x_i - x_j > n$, we can subtract n from x_i and add n to x_j.
After repeating this, we will have $|x_i - x_j| \leq n$ for all i, j.
We may look at relative positions and assume $x_j = 0$.



Theorem Part 3

Assume all x_i are in $[0, n]$. Are there optimal solutions with both $x_i = 0$ and $x_i = n$?

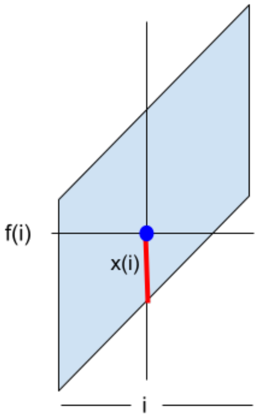
For $x_i = 0$, the number of crossings with a permutation cycle just depends on the cycle, not i. The same holds for $x_i = n$.

So, whether 0 or n is optimal will be the same for all i with $x_i \mod n = 0$.

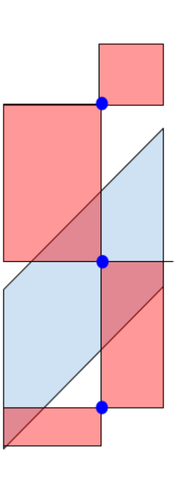
Recap

We may assume $x_i \in [k, k+n)$ for some integer $k \in [0, n)$. There is at most one crossing between any two people.

If $i < j$, there is a crossing iff $f_i > f_j$ or $f_i + n < f_j$.



f(i)
x(i)
i



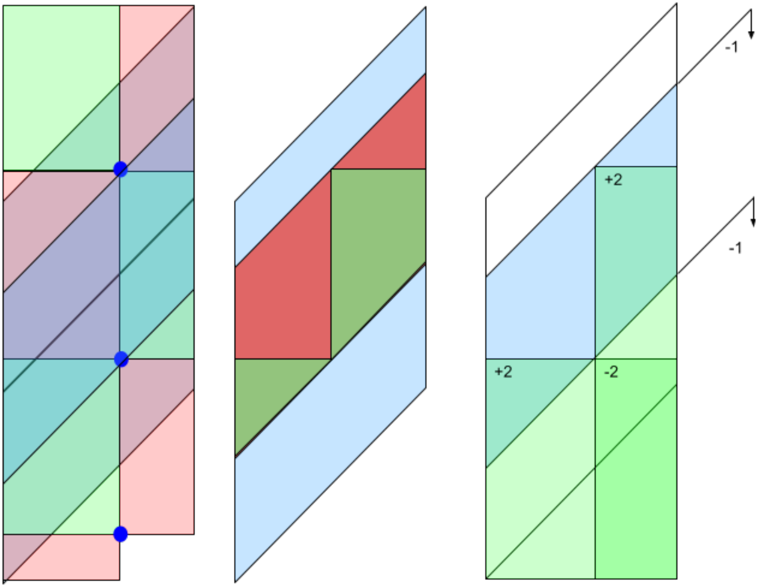
Subtasks

For each person, we can count crossings in $O(n)$ time. There are n intervals $[k, k+n)$ to consider. Let us go through them in order of increasing k.

Then we can find the initial crossings in $O(n^2)$. We change each x_i to $x_i + n$ exactly once. When this happens, it costs $O(n)$ to recompute the answer.

There is also an $O(n^2 \log n)$ solution that computes crossings from scratch each time, but uses a BIT.

There is also an $O(n \log^2 n)$ solution that optimizes the $O(n^2)$ with an online 2D data structure.



-1
+2
-1
+2
-2

Full Solution

The full solution has both x_i and $x_i + n$ in the set of points, and uses an offline linesweep to calculate the answers to $O(n)$ rectangle queries. This runs in $O(n \log n)$.

Problem Inspiration

The problem is equivalent to sorting a permutation with swaps, where you can also swap the first and last elements (and also cycle the permutation for free). However, there is no direct reduction to counting inversions.

There is a 1985 paper by Mark Jerrum, "The complexity of finding minimum-length generator sequences" that deals with this problem, without the cyclic shifts. It gives a rigorous proof of the mathematical part of the problem. However, the time complexity is only $O(n^2)$.