Vicious Labyrinth

June 13, 2025

1 Problem

There are n cells in a labyrinth, and cell i ($1 \leq i \leq n$) is $n - i$ kilometers away from the exit. In particular, cell n is the exit. Note also that each cell is connected to the exit but is not accessible from any other cell in any way.

In each cell, there is initially exactly one person stuck in it. You want to help everyone get as close to the exit as possible by installing a teleporter in each cell i ($1 \leq i \leq n$), which translocates the person in that cell to another cell a_i.

The labyrinth owner caught you in the act. Amused, she let you continue, but under some conditions:

- Everyone must use the teleporter exactly k times.
- No teleporter in any cell can lead to the same cell it is in. Formally, $i \neq a_i$ for all $1 \leq i \leq n$.

You must find a teleporter configuration that minimizes the sum of distances of all individuals from the exit after using the teleporter exactly k times while still satisfying the restrictions of the labyrinth owner.

If there are many possible configurations, you can output any of them.

2 Solution

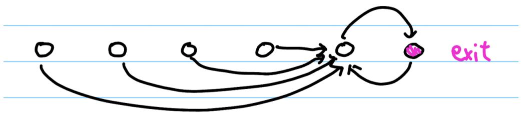
We present an $O(n)$ time solution.

```
f(n,k){
        A = array of size n
        if(k is even){
                A[n] = n-1
                A[n-1] = n
                for(i = 1; i < n - 1; ++i)        A[i] = n-1;
        }else{
                A[n] = n-1
                A[n-1] = n
                for(i = 1; i < n - 1; ++i)        A[i] = n;
        }
        return A
}
```

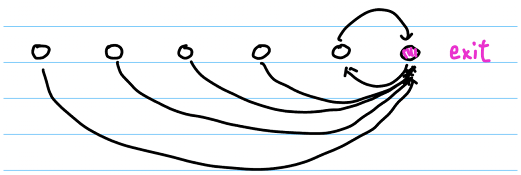
For better visualization, given a teleporter configuration $A = \{a_1, ..., a_n\}$ we can form a directed graph $G_A = (V, E_A)$ where there's a vertex for each cell and $E_A = \{(i, a_i) : i = 1, ..., n\}$.

Let A be the teleporter configuration that our algorithm found and d be the sum of distances of all individuals from the exit after using the teleporter exactly k times under configuration A.

When k is even, the configuration A that our algorithm finds will be of the form



when k is odd it'll be of form



Lemma 1. $d = 1$.

proof. When k is even. For a person initially in cell i, after using the teleporter exactly k times under configuration A :

- if $i < n-1$, he'll first go to cell $n-1$, then cycle between cell $n-1$ and n. It's not hard to see he'll be at cell n at the end when k is even.

- if $i = n-1$, he'll cycle between cell $n-1$ and n. It's not hard to see he'll be at cell $n-1$ at the end when k is even.

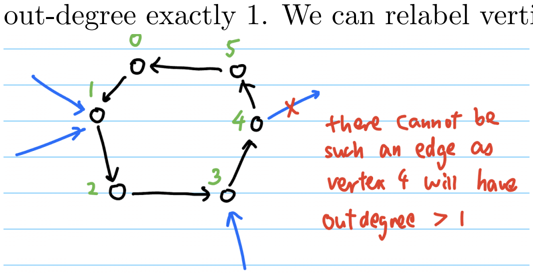
- if $i = n$, he'll cycle between cell $n-1$ and n. It's not hard to see he'll be at cell n at the end when k is even.

Thus at the end, $n-1$ person will be at cell n and 1 person at cell $n-1$, yielding $d = 1$.

Similar argument for odd k. ∎

Lemma 2. Let d^* be the sum of distances of all individuals from the exit after using the teleporter exactly k times under the optimal configuration A^*. Then $d^* \geq 1$.

proof. It's not hard to see there must be a cycle C in G_Q for any configuration Q because each node has out-degree 1. Also notice there can be edges going into C (ie. from a vertex in $G_Q - C$ to a vertex in C) but cannot be edges leaving the cycle (ie. from a vertex in C to a vertex in $G_Q - C$) again because each node has out-degree exactly 1. We can relabel vertices in the the cycle $0, ..., |C| - 1$ following the direction of the edges.



Notice the person initially in vertex $(i-k) \bmod |C|$ will be at vertex i after using the teleporter exactly k times. This holds for any vertex i of C, thus after everybody used the teleporter exactly k times, for each vertex of C, there'll be at least a person there. Thus it's impossible for everybody to gather at the exit, which means $d^* \geq 1$. ∎

Lemma 1 and 2 along with the fact that A is a valid teleporter configuration (not hard to see) implies our algorithm finds an optimal solution.