← Qualification Round 2019 - Google Code Jam 2019 2

SWITCH TO EDITOR

Dat Bae (14pts, 20pts)

Practice Submissions

Attempt 1 ✓ ✓ Apr 19 2020, 22:36

Last updated: Apr 19 2020, 22:35

PROBLEM ANALYSIS

Problem

A research consortium has built a new database system for their new data center. The database is made up of one master computer and **N** worker computers, which are given IDs from 0 to **N**-1. Each worker stores exactly one bit of information... which seems rather wasteful, but this is very important data!

You have been hired to evaluate the following instruction for the database:

- `TEST_STORE` <bits>: The master reads in <bits>, which is a string of **N** bits, and sends the i-th bit to the i-th worker for storage. The master will then read the bits back from the workers and return them to the user, in the same order in which they were read in.

During normal operation, `TEST_STORE` should return the same string of bits that it read in, but unfortunately, **B** of the workers are broken!

The broken workers are correctly able to store the bits given to them, but whenever the master tries to read from a broken worker, no bit is returned. This causes the `TEST_STORE` operation to return only **N**-**B** bits, which are the bits stored on the non-broken workers (in ascending order of their IDs). For example, suppose **N** = 5 and the 0th and 3rd workers are broken (so **B** = 2). Then:

- `TEST_STORE 01101` returns `111`.
- `TEST_STORE 00110` returns `010`.
- `TEST_STORE 01010` returns `100`.
- `TEST_STORE 11010` also returns `100`.

For security reasons, the database is hidden in an underground mountain vault, so calls to `TEST_STORE` take a very long time. You have been tasked with working out which workers are broken using at most **F** calls to `TEST_STORE`.