Code Jam 2021 - Round 2

Analysis: Minimum Sort

In this problem, we are given only one tool to get information about our input: the minimum query. Moreover, the query becomes cheaper when done over long distances. This means a sorting algorithm that normally spends running time finding minimums would align well with our needs. One well known such algorithm is Selection sort.

In Selection sort, we search for the minimum of the full list, put that in its correct position (the beginning), and continue solving recursively searching for the minimum of a suffix of the list, and moving that minimum to the beginning. This can be implemented with $N-1$ pairs of the minimum query and (possibly) a swap operation as follows:

```
for i := 1 to N-1
  j = minimum_index_of(i, i+1, ..., N)
  if i != j:
    swap(i, j)
```

The minimum queries above are over ranges of sizes $N, N-1, \ldots, 2$. This means the total cost is exactly

$$\sum_{i=2}^{\mathbf{N}} \left\lceil \frac{10^8}{i} \right\rceil.$$

For $\mathbf{N} = 100$, this is 418737795, which is less than the limit of 6×10^8.