Code Jam 2018 - Round 1B

Analysis: Rounding Error

Test set 1

This test set can be solved using a complete search. We can try every possible partition of **N** voters among **N** languages. If two partitions differ only in the order of their languages, then we consider those partitions equivalent.

Therefore, we can consider a partition as an **N**-tuple $(x_1, x_2, ..., x_{\mathbf{N}})$, where $x_i \geq x_{i+1}$ and $\Sigma\, x_i = \mathbf{N}$.

Even with **N** = 25, there are no more than 2,000 different partitions. For each partition, we can use the following greedy algorithm to check whether the partition can be achieved by only adding voters: let us sort the $\mathbf{C}_i$ values in non-increasing order — that is, such that $\mathbf{C}_i \geq \mathbf{C}_{i+1}$. Then the partition can be achieved by only adding voters if and only if $\mathbf{x}_i \geq \mathbf{C}_i$ for all $1 \leq i \leq \mathbf{L}$. Among all such partitions, we can find the largest percentage sum, which is our answer.

Test set 2

For this test set, we can remove our assumption that a partition and **C** must be sorted non-increasingly. Therefore, we consider a partition of **N** voters to **N** languages as $(x_1, x_2, ..., x_{\mathbf{N}})$, where $\Sigma\, x_i = \mathbf{N}$.

To solve this test set, we can use dynamic programming (DP). We define a function f(a, b) as the following:

Among all partitions $(x_1, x_2, ..., x_a)$ such that $\Sigma\ (1 \leq i \leq a)\ x_i = b$ and $x_i \geq \mathbf{C}_i$ for all $1 \leq i \leq a$, what is the maximum $\Sigma\ (1 \leq i \leq a)\ \mathrm{round}(x_i / \mathbf{N} \times 100)$ possible? If there is no satisfying partition, then $f(a, b) = -\infty$. We can assume $\mathbf{C}_i = 0$ for $i > \mathbf{L}$.

We can first handle the base case of the function. We can easily compute f(1, b) since there is only at most one satisfying partition. Therefore, $f(1, b) = \mathrm{round}(b / \mathbf{N} \times 100)$ if $b \geq \mathbf{C}_1$, or $-\infty$ otherwise.

The recurrence f(a, b) of this function can be computed by considering all possible values of x_a. Let i be the value of x_a. Therefore, x_a contributed $\mathrm{round}(i / \mathbf{N} \times 100)$ to the total percentage, and there are (b - i) votes left to be distributed among $x_1, x_2, ..., x_{a-1}$. Therefore, for a > 1, $f(a, b) = \max\ (\mathbf{C}_a \leq i \leq b)\ (\mathrm{round}(i / \mathbf{N} \times 100) + f(a - 1, b - i))$.

Since we want to distribute **N** voters to $x_1, x_2, ..., x_{\mathbf{N}}$, the answer for the problem is f(**N**, **N**).

Function f has $O(\mathbf{N}^2)$ possible states and each state takes $O(\mathbf{N})$ time to compute. Therefore, this solution runs in $O(\mathbf{N}^3)$ time.

Test set 3

We can solve this test set with a greedy strategy. For each language, we will either be rounding the percentage up or down. We get the maximum answer when as many of these as possible are rounded up.

Therefore, we can ignore any languages that are already being rounded up. Since there can be arbitrarily many languages, nothing ever forces us to disturb these languages by adding another vote—it's no worse to add that vote to some new language instead. We figure out how many more votes each language (including languages nobody has even mentioned yet) would need in order for it to be rounded up.

We greedily satisfy as many of these as possible, starting with the ones that take the fewest additional votes. This solution runs in O(**N** × log(**N**)) time.