Code Jam 2019 - Round 2

Analysis: New Elements: Part 2

Let w_C be the atomic weight of Codium and w_J be the atomic weight of Jamarium according to the rules given in the problem statement. Let ΔC_i equal $\mathbf{C}_{i+1} - \mathbf{C}_i$ and ΔJ_i equal $\mathbf{J}_{i+1} - \mathbf{J}_i$, for all $1 \le i < \mathbf{N}$. As in our analysis for New Elements, Part 1, we have:

- $-\Delta C_i / \Delta J_i < w_J / w_C$, if $\Delta J_i > 0$.
- $w_J / w_C < -\Delta C_i / \Delta J_i$, if $\Delta J_i < 0$.
- $-\Delta C_i \times w_C < 0$, if $\Delta J_i = 0$.

Therefore, we can get the lower bound and upper bound of w_J / w_C just by looking at consecutive indices. We can initially set the lower bound (let us represent it with the reduced fraction L_N / L_D) to be 0 and the upper bound (let us represent it with the reduced fraction U_N / U_D) to be ∞. We update either L_N / L_D or U_N / U_D for each pair of consecutive indexes, just as in our analysis from Part 1.

Once we have L_N / L_D and U_N / U_D, we want to find a rational number w_J / w_C such that $L_N / L_D < w_J / w_C < U_N / U_D$. If $L_N / L_D \ge U_N / U_D$, then there is certainly no solution. Otherwise, there must be at least one solution; for example, the mediant $(L_N + U_N) / (L_D + U_D)$ is certainly between the bounds. However, the problem asks us to minimize w_C and w_J (first w_C, and then w_J).

Test set 1

Since $\Delta J_i \le 99$ in this test set, we get $L_D + U_D \le 198$. Therefore, we know that a solution with $w_C \le 198$ exists. We can try all possible values from 1 to 198 as w_C. For each choice of w_C, we can derive the smallest w_J such that $L_N / L_D < w_J / w_C$, and then we can check whether $w_J / w_C < U_N / U_D$.

Test set 2

For each integer C (from 1 to $L_D + U_D$), we can check whether there is a rational number that is strictly between L_N / L_D and U_N / U_D, and has a denominator that is not more than C. To do that, we can find a rational number with denominator not more than C closest to the average of L_N / L_D and U_N / U_D. This is because all rational numbers that are strictly between L_N / L_D and U_N / U_D are closer to the average of L_N / L_D and U_N / U_D than all rational numbers that are not strictly between L_N / L_D and U_N / U_D. We can do so by using a library function like Python's fractions.limit_denominator, or by implementing our own approximation using continued fractions.

Once we can solve the problem given in the previous paragraph, we can use binary search to find w_C as the smallest C such that a rational number with denominator not more than C, and strictly between L_N / L_D and U_N / U_D exists. Just as we did for the previous test set, we can derive the smallest w_J such that $L_N / L_D < w_J / w_C$.