

Coding competition

Daya Singh

December 2023

Summary

These questions are designed to be balanced amongst most, if not all coding languages. To be safe, these are restricted to Python, Javascript, C++, C, C#, R, Matlab, and Wolfram Mathematica. Tread caution with the last due to the number of in-built functions. The 'Checks' section will be available for the first 2 questions (for checking purposes only). Furthermore, the internet is only to be used to check syntax for your coding language (e.g documentation).

To motivate people to, as opposed to running their code for longer, make small optimisations (optimisations that speed up code but not increase time complexity), students are restricted to use their code (for non-checking purposes) in the last 15 minutes. Furthermore, to ensure students have no hardware advantages, all code must be run locally on the University terminals. The software allowed to be used:

- Python: Jupyter or Python
- R: RStudio
- Javascript: Visual Studio Code
- MatLab: MatLab
- C++, C: Visual Studio Code (With C/C++ MS extension)
- Wolfram Mathematica: Mathematica
- C#: Visual Studio Code (With C# Dev Kit MS extension)

Challenge 1: Landau's Function

A partition of n is n represented as a sum of naturals less than or equal to n . 3 can be expressed as $1 + 1 + 1$, $2 + 1$ and 3. Define Landau's function $g(n)$ to be the largest LCM of the addends of the partitions of n . $g(3) = 3$ as the LCM of 3 is 3 which is larger than the LCM of $\{2, 1\}$ and $\{1, 1, 1\}$. Your task is to generate as many of the first values of the sequence $g(n)$ in 45 minutes.

Challenge 2: Feigenbaum's Constant

Consider the logistic map, $f(x) = ax(1 - x)$, $x \in (0, 1)$. Let g_n be defined as f compounded n times. For $a = a_0$, $g_n(x)$ is a convergent sequence. Define a_n as the infimum of possible a such that $g_k(x)$, for some x , is a periodic sequence of period 2^n . The first Feigenbaum constant, δ , is the limit of the ratios of these values:

$$\delta := \lim_{n \rightarrow \infty} \frac{a_{n-1} - a_{n-2}}{a_n - a_{n-1}}$$

Your task is to approximate this constant as best as you can in 45 minutes.

Challenge 3: Superpermutations

A superpermutation of order n is a string of n characters that contain every possible arrangement of those n characters. For $n = 3$, such a string would be 123121321, as it contains 123, 213, 312, 132, 321, and 231. Finding the smallest superpermutation of order $n > 5$ is still an open problem in mathematics. Your task is to generate the smallest superpermutation of order 8 that you can in 45 minutes.

Checks

Feigenbaum's Constant

Feigenbaum's constant to 100 decimal places:

$$\delta = 4.6692016091029906718532038204662016172581855774757686327456513430041343302113147371386897440239480138$$

Landau's function

This can be found on the OEIS here: <https://oeis.org/A000793>.

Superpermutations

The following python script can check a superpermutation (Credit: Kevin):

```
1 from itertools import permutations
2
3 def is_superpermutation(n, string):
4     all_permutations = set([
5         permutations(
6             .join(str(i) for i in range(1,
7                 n+1)), n)])
8     return all(perm in string for perm in
9         all_permutations)
```