Champernownes constant is a decimal created when you keep appending numbers after the decimal point. Being that it is a series problem (each number depends on the numbers before it) it can be tricky to find out what a number is at a specific point in the constant. For example, lets say you wanted to know what the 542nd digit of Champernownes constant? What about the 110th? How about the 843,323,431st? There’s not an obvious analytical solution, so let’s try finding it computationally. Here’s some source code.

The tempting brute force approach

[insert e40.rb]

Unfortunately after the ten thousandth or so iteration this takes up an absurd amount of memory, so let’s try to be smarter.

Notice how the current index depends on the length of each number (for example, to add ‘104’ the index would change by three places).

[insert approach #2]

Despite that intuition and a bunch of incrementor debugging, I found that it still took up too much memory to work for large cases.

So why not split up the work? Instead of trying to get everything done at once, I decided to divide and conquer to find what we’re looking for. Store it all in a file and then read the relevant digits. It’s not a perfect answer, but it’s pretty good.