Problem 40 - Champernowne's Constant

Gautam Manohar

24 June 2018

This document originally appeared as a blog post on my website. Find it at gautammanohar.com/euler/40.

1 Problem Statement

An irrational decimal is created by concatenating the positive integers:

$$0.12345678910\mathbf{1}112131415161718192021\ldots$$

It can be seen that the 12th digit of the fractional part is `1`. If d_n represents the n-th digit of the fractional part, then given integers i_j, find the product of d_{i_j}.

2 My Algorithm

We can split up the fractional part into ‘blocks''where each positive↩ integer used has the same number of digits. For example, ↩ starting at zero, the zeroth block is123456789, the first↩ 101112...9899‘, and so on. The numbers used to make up the n-th block (where the first block is $n = 0$) range from 10^n to $10^{n+1} - 1$, inclusive. There are therefore $10^{n+1} - 10^n = 9 \cdot 10^n$ numbers in the n-th block. Because each number in the n-th block has $n + 1$ digits, the n-th block is $9(n + 1)10^n$ digits long.

We initialize `block` at 1. This means `block` is equal to $n + 1$. It represents the number of digits in each integer in the block that d_i is part of. We initialize `fact` at 9. It represents the number of positive integers that make up `block`. This means `fact*block` represents the length of `block`. Then, as long as the result is positive and `fact * block` is less than $9M$, where M is the maximum possible value of i in d_i (a hard upper bound on the length of the largest block, given the input constraints), we subtract `fact*block` from the given d_i. Then, we multiply `fact` by 10, representing the ten-fold increase in the number of elements of the next block, and increment `block` by 1, representing the fact that we are moving to the next block. The resulting value of `block` is the block in which d_i is located. The resulting value of d_i is `pos+1`, where `pos` is the zero-indexed ‘`position''of $d_i$ in its block; that is, $d_i$ is the`pos‘-th digit in its block.

Because d_i is in the block given by `block`, the first number in its block is $10 * *(block - 1)$. Each number in this block has `block` digits. And so `pos//block` gives the number of positive integers that precede the integer d_i is part of in its block. This means d_i is part of the positive integer `10**(block-1)+ pos//block` The leftover after this division (that is, `pos % block`) is the index (starting at 0) of d_i in the number it is part of. Because we know the number that d_i is part of, we have solved the problem. Using our general procedure, we find d_{i_j} and calculate their product.

The most intensive part of our solution is finding the block that d_i is part of; finding the value of d_i after this is just an $O(1)$ string operation. The lengths of the blocks form an arithmetogeometric sequence, so there are $O(\log n)$ blocks (not a tight upper bound). And so our solution has time complexity $O(D \log i_{j_{\max}})$, where D is the number of digits to find.