Problem 30 - Digit Fifth Powers

Gautam Manohar

23 June 2018

This document originally appeared as a blog post on my website. Find it at gautammanohar.com/euler/30.

1 Problem Statement

Surprisingly, there are only three numbers that can be written as the sum of fourth powers of their digits:

$$\begin{aligned} 1634 &= 1^4 + 6^4 + 3^4 + 4^4 \\ 8208 &= 8^4 + 2^4 + 0^4 + 8^4 \\ 9474 &= 9^4 + 4^4 + 7^4 + 4^4 \end{aligned}$$

Note that because $1 = 1^4$ is not a sum, it is not included. The sum of these numbers is $11634 + 8208 + 9474 = 9316$.

Find the sum of all numbers that can be written as the sum of N-th powers of their digits.

2 My Algorithm

The sum of the N-th powers of the digits of a number with d digits is at most $d \cdot 9^N$. And so a hard upper bound on a number equal to the sum of the N-th powers of its digits is $N \cdot 9^N$. This number has $\lfloor \log_{10} N + N \log_{10} 9 \rfloor + 1$ digits. Because the argument of the floor is never a power of 10, this is equal to $\lceil \log_{10} N + N \log_{10} 9 \rceil$. So we could check all numbers $11 \leq n < 10^{\lceil \log_{10} N + N \log_{10} 9 \rceil}$. However, this is much too slow to pass $N = 6$.

We know that any number that fits the criteria is the sum of some combination (with replacement) of integers in $0^N, 1^N, \ldots, 9^N$. And so we can check each combination with replacement of $2 \leq k \leq \lceil \log_{10} N + N \log_{10} 9 \rceil$ elements (this represents the number of digits) from this set of 10 N-th powers. We can find these combinations using `combinations_with_replacement` from the `itertools`↩ package. We skip the case where $k = 1$ because, as stated in the problem, the sum of the digits in a one-digit number is a degenerate sum.

To avoid counting duplicate combinations with extra `0`s, we use multisets, implemented with `Counter` from `collections`. A multiset is like a set, but elements can appear more than once. Two multisets are equal if and only if they contain the same elements the same number of times.

We get the multiset of the N-th roots of each combination. Then, we sum the elements of the combination and find the multiset of digits of the sum. If these two multisets are equal, we have found a valid solution.

There are $\binom{n+k-1}{k}$ ways to choose k elements with replacement from a pool of n elements. In our case, $n = 10$. We therefore perform

$$\sum_{k=2}^{\lceil \log_{10} N9^N \rceil} \binom{9+k}{k} \tag{1}$$

operations. For $N = 6$, this turns out to be 19437—much less than the 10^7 required by brute force.

Just for fun, let's do some upper bound analysis on this expression. Using Stirling's approximation for $n!$, we find that $\binom{n}{k} \leq \frac{n^k}{k!}$. In our case, we have $\frac{(9+k)^k}{k!} \approx \frac{k^k}{k!}$. The upper limit of our sum is $O(\log_{10} N + N \log_{10} 9) \in O(N + \log_{10} N) \in O(N)$. And so our solution is $O(\frac{N^N}{N!})$. This is not a tight upper bound; it grows much faster than the exact number of operations given in (1).

2.1 Other Solutions

The brute force approach described above is sufficient to solve the original Project Euler problem. It has complexity $O(N10^N)$.