

Project Euler: Problem 549

Problem. Let $s(n)$ be the smallest number m such that n divides $m!$. Define $S(n)$ to be

$$S(n) := \sum_{i=2}^n s(i).$$

Compute $S(10^8)$.

Solution. We shall first obtain some intuition for $s(n)$. In particular, one can observe the following:

Observation. Suppose $n = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$, where for convenience $p_1 < p_2 < \cdots < p_n$. Clearly then $p_i^{e_i} \mid m!$ for each i . For each i , we may then use Legendre's formula to linearly search numbers of the form $(kp_i)!$ until the resulting power is high enough, letting the first such number inside the factorial be denoted by $a_i(n)$. This takes $O(\log n)$ time. With this developed, we have that

$$s(n) = \max_{1 \leq i \leq n} a_i(n).$$

Using sieve methods, we may compute the smallest prime factor of each number in $[2, n]$ in $O(\sqrt{n})$ time. This allows us to iterate over all numbers in $[2, n]$ and prime factorize them in $O(n \log n)$ time. Applying the method above for finding $s(n)$ makes the computational complexity $O(\sqrt{n} + n \log^2 n) = O(n \log^2 n)$ in total.

Unfortunately, this runs really slow in implementation and is unsuitable for computing something as large as $S(10^8)$. Are there any optimizations we can make?

Idea. Can we simply take the component $a_i(n)$ for which $p_i^{e_i}$ is maximal?

No, one counterexample would be $40 = 2^3 \cdot 5$.

Perhaps a different approach can be taken where one precomputes the values of $s(n)$ for all pure prime powers n , making the rest of the numbers just lookups and avoiding recomputation. This works in decent time, and gave the correct answer, but it was rather slow on my machine. Perhaps another optimization could be to store the value of $s(n)$ for each n in an array and then update values in this array when computing the sieve and such. This avoids the need to factor each number when iterating.

Question. This is just purely unrelated to the question, but the idea of $s(n)$ is an interesting number theoretic concept, and there's lots of we can potentially play around with:

- Can we derive an asymptotic for $s(n)$?