Problem 7 - 10001st Prime

Gautam Manohar

10 June 2018

This document originally appeared as a blog post on my website. Find it at gautammanohar.com/euler/7.

1 Problem Statement

By listing the first six prime numbers: 2, 3, 5, 7, 11, and 13, we can see that the 6th prime is 13.

What is the N-th prime number?

2 My Algorithm

We use the following identity, which is a consequence of the Prime Number Theorem:

$$n(\log n + \log\log n - 1) < p_n < n(\log n + \log\log n), n > 5. \tag{1}$$

If $n \leq 5$, we use a look-up table. Using a Sieve of Eratosthenes, we generate a list of the primes up to $n(\log n + \log\log n)$, which is guaranteed to contain the n-th prime. Our answer is the n-th element of this array. We do this with time complexity $O(n \log n \log\log(n \log n))$, using $p_n \approx n \log n$. Because $O(\log\log(n \log n)) \in O(\log n)$, this can be simpified to $O(n \log^2 n)$.

To optimize the constant factor, we compute the first $N_{\max} = 10^4$ primes and store them in memory, rather than computing the list for each query, which takes too long.