15 Number Theory

15.2 Computing $\pi(x)$ (7 units)

Background material for this project is contained in the Part II Number Theory course.

1 Introduction

The function $\pi(x)$ is defined as the number of primes $\leqslant x$, so that $\pi(11) = \pi(11.5) = \pi(12) = 5$. A simple way to compute $\pi(x)$ is to form a list of all the primes $\leqslant x$ and count. This could be done by testing all the integers up to x for primality and one way of doing this would be by trial division. A somewhat more efficient method of finding the primes up to N in terms of time, but with greater requirements in terms of storage, is the *sieve of Eratosthenes.* List the numbers from 2 to N. Mark all the multiples of 2 other than 2 itself. The next unmarked number is 3, so mark all the multiples of 3 other than 3 itself. Continue in this way until no more numbers can be marked. The unmarked numbers are now the primes up to N.

> **Question 1** Write programs to list the primes up to, say, $15,000$; one version should use some form of primality test and the other a sieve. Your program should be capable of reading in x and printing out $\pi(x)$ for x in this range. Comment on the time and storage theoretically required by each algorithm for large values of x.

2 Legendre's formula

Listing the primes is not a very efficient way of computing $\pi(x)$ so we should look for indirect methods. One such method is *Legendre's formula* which counts the primes by the inclusion-exclusion principle.

$$\pi(x) + 1 = \pi(\sqrt{x}) + [x] - \sum_{p_i \leqslant \sqrt{x}} \left[\frac{x}{p_i}\right] + \sum_{p_i < p_j \leqslant \sqrt{x}} \left[\frac{x}{p_i p_j}\right] - \sum_{p_i < p_j < p_k \leqslant \sqrt{x}} \left[\frac{x}{p_i p_j p_k}\right] + \cdots$$

where as usual $[x]$ denotes the integer part of x and $p_1 = 2$, $p_2 = 3$, $\ldots$ is the sequence of primes.

> **Question 2** Compute $\pi(132)$ using Legendre's formula **by hand**.

This formula is not suited to practical computation, because of the difficulty of programming the multiple summations, because the number of terms involved increases rapidly with x and because it requires knowledge of the list of primes up to $\sqrt{x}$. We need to organise the terms more efficiently.

The multiple sums on the right-hand side of Legendre's formula can be interpreted as counting the number of integers $\leqslant x$ not divisible by any of the primes $\leqslant \sqrt{x}$. Define $\psi(x, a)$ to be the numbers of integers $\leqslant x$ not divisible by any of the first a primes. Then

$$\psi(x, a) = [x] - \sum_{i \leqslant a} \left[\frac{x}{p_i}\right] + \sum_{i < j \leqslant a} \left[\frac{x}{p_i p_j}\right] - \sum_{i < j < k \leqslant a} \left[\frac{x}{p_i p_j p_k}\right] + \cdots$$

and Legendre's formula can be written as

$$\pi(x) + 1 = \pi(\sqrt{x}) + \psi(x, \pi(\sqrt{x})).$$

We have a recursion relation for $\psi(x, a)$,

$$\psi(x, a) = \psi(x, a-1) - \psi(x/p_a, a-1)$$

and $\psi(x, 0) = [x]$. This relation allows us to overcome the difficulty in programming the multiple summations.

> **Question 3** Write a program to compute $\pi(x)$ using Legendre's formula and the recursion relation for $\psi(x, a)$ which will work for $x \leqslant 10^6$. Use your program to tabulate $\pi(x)$ for x up to 100 in steps of 10, up to 1000 in steps of 100, etc. as far as practical. Show in detail what values of $\psi(x, a)$ your program uses on the way to computing $\pi(132)$.

This algorithm is still inefficient because we compute $\psi(x, a)$ many times over for small values of a. If we write

$$m_k = \prod_{i=1}^{k} p_i$$

then we see that the pattern of multiples of the first k primes repeats in a cycle of length m_k. In fact,

$$\psi(sm_k + t, k) = s\phi(m_k) + \psi(t, k).$$

(Here $\phi(m_k)$ denotes the usual Euler ϕ-function.) If we pick a suitable value for k and store the values $\psi(t, k)$ for $1 \leqslant t \leqslant m_k$ then we can curtail the recursion formula for $\psi(x, a)$ when a is reduced to k rather than 0.

> **Question 4** Compute the first few values of m_k and find a suitable value of k for which you can store the values of $\psi(t, k)$ for t up to m_k. (You should be able to take k at least 4.) Modify your previous program to use these values in the recursion relation for $\psi(x, a)$ and repeat your tabulations as far as practical.

3 Meissel's formula

We modify Legendre's formula to produce *Meissel's formula*. Put $b = \pi(\sqrt{x})$, $c = \pi(x^{1/3})$. Then

$$\pi(x) = \psi(x, c) + \frac{1}{2}(b + c - 2)(b - c + 1) - \sum_{c < i \leqslant b} \pi\left(\frac{x}{p_i}\right).$$

Note that it is now necessary to compute the values of $\pi(y)$ for certain values of y in the range $x^{1/2}$ to $x^{2/3}$, and to have a list of primes up to $\sqrt{x}$. We save by only having to compute $\psi(x, \pi(x^{1/3}))$ rather than $\psi(x, \pi(x^{1/2}))$.

> **Question 5** Modify the program of questions 3 or 4 to use Meissel's formula and repeat your tabulations as far as practical.

4 The Li function

The asymptotic behaviour of $\pi(x)$ is given by the Prime Number Theorem

$$\pi(x) \sim \frac{x}{\log x}$$

and it is also known that a better approximation is

$$\pi(x) \sim \mathrm{Li}(x)$$

where Li is the *logarithmic integral*

$$\mathrm{Li}(x) = \int_0^x \frac{\mathrm{d}t}{\log t}$$

suitably interpreted at the singular point $t = 1$. For computational purposes it is easier to take the lower limit of the integral to be 2 and use the approximation $\mathrm{Li}(2) \approx 1.045$.

Question 6 Use a suitable numerical procedure to approximate $\mathrm{Li}(x)$ and tabulate the values of $\frac{x}{\log x}$ and $\mathrm{Li}(x)$ for the same values of x for which you have tabulated $\pi(x)$. Tabulate the ratios $\frac{\pi(x)}{x/\log x}$, $\frac{\pi(x)}{\mathrm{Li}(x)}$ and the differences $\pi(x) - \frac{x}{\log x}$, $\pi(x) - \mathrm{Li}(x)$. Discuss the accuracy of these approximations to $\pi(x)$.

Question 7 Conjecture a possible order of magnitude for $\pi(x) - \mathrm{Li}(x)$.

[*You may wish to consider a change of variable* $u = \log t$ *in the integral.*]

References

[1] Riesel, H., *Prime numbers and computer methods for factorization.* Birkhauser Verlag 1985.

