Probabilistic Factorisation of large integers

Project LIMA

INTRODUCTION

> "*The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. It has engaged the industry and wisdom of ancient and modern geometers to such an extent ... [that] the dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated.*"
>
> *Carl Friedrich Gauss*

A brief history of prime

Prime numbers and their properties have been the source of fascination since early times. The concept of prime probably existed around the Babylonian time. In Euclid's Elements (300 BC), several important results about primes had been proved. For example, there are infinitely many prime numbers and the *Fundamental Theorem of Arithmetic* every integer can be uniquely represented as a product of primes.

The Greek cryptographer Eratosthenes (200 BC) also studied primes in depth and eventually devised an algorithm for calculating them *The Sieve of Eratosthenes.*

The next important developments were made by Pierre de Fermat at the beginning of the 17th Century. Amongst those he proved a result known as the *Fermat's Little Theorem*, which states that if p is a prime then for any integer a we have $a^p = a \mod p$. Fermat's Little Theorem gave light to many other results in Number Theory and is the basis for some methods of primality checking which are still in use on today's computers.

Father Marin Mersenne, a contemporary and friend of Fermat, stated that $2^p - 1$ is prime for certain primes p. As a result we now call any prime of the form $2^p - 1$ a *Mersenne prime* in his honour.

Fermat's Little Theorem was subsequently extended by Euler, who introduced the ϕ-function. Later, Gauss and Legendre investigated the density of primes and came to similar conclusion that for large n the density of primes near n is roughly $1/log(n)$.

Prime and the Computer Revolution

For a long time, the study of prime numbers was perceived as a 'pure' mathematical pursuit with little practicality. However this was to change with the rise of computational technology. With the power of computers, we can now find exceedingly large primes. On January 30, 1952, the SWAC computer proved that $2^{521} - 1$ is prime, then had time left over to prove that $2^{607} - 1$ is also prime. Since then it has almost became a ritual to give the latest supercomputer a record high prime to prove. The picture changed again when in 1996, the first of 5 Mersenne primes were discovered using desktop computers.

Modern cryptography is heavily based on large composite numbers, more specifically, exceedingly large numbers consisting of two very large prime factors. The RSA cryptographic system is based on the assumption that the product of two very large prime numbers can not be easily factored, whereas to check if a number is prime can be done relatively quickly, thus very large primes can be generated easily.

Complexity of Primality

While the *Sieve of Eratosthenes* works fine for small prime numbers, its complexity increases exponentially as the numbers get larger to the extent that it would require a computer the size of the entire universe to determine if $2^{257} - 1$ is prime or not using this method.

Many other factorisation methods exist, but they vary in performance and sophistication. Generally speaking, factoring algorithms come in two broad categories: special purpose and general purpose. The efficiency of the former depends on the unknown factors, whereas the efficiency of the latter depends on the number to be factored. Special-purpose algorithms are best for factoring numbers with small factors, while general-purpose factoring algorithms cope better with larger factors and are more important in the context of cryptographic systems and their security. These are often termed as probabilistic algorithms in the sense that they use pseudo-random sequences to guide them.

Currently, some of the best factoring algorithms include:

- Number field sieve the general number field sieve is the fastest known factoring algorithm for numbers larger than 110 digits with running time of approximately $O(e^{1.9(logn)1/3(loglogn2/3)})$.
- Quadratic sieve the fastest known algorithm for numbers less than 110 decimal digits long, with running time $O(e^{1+o(1)\sqrt{logn(loglogn)}})$.
- Elliptic curve method this method has been used to find 43 digit factors, but not larger. Its asymptotic running time is $O(e^{\sqrt{2logp(loglogp)}})$.
- Pollard Rho special purpose algorithm with expected running time $O(\sqrt{p})$.

The search for a better algorithm continues...