NUMBER THEORY AND THE DESIGN OF FAST COMPUTER ALGORITHMS

C. T. Chong
National University of Singapore

The theory of numbers has long been considered to be among the purest of pure mathematics. Gauss (1777- 1855) called it the queen of mathematics. Hardy (1877 --1947) took pride in the belief that his best contributions in mathematics, namely analytic number theory, had no practical applications whatsoever. In the history of mathematics, number theory may justifiably claim to have as its devotees many of the best mathematicians. Fermat (1632 – 1690), Euler (1670 – 1743), Gauss, Riemann (1826 – 1866), Eisenstein (1823 – 1852), Dirichlet (1805 – 1859), Hilbert (1862 – 1943) were men who shaped modern mathematics and who could count number theory among their most impressive achievements. Closer to our times, we have Artin (1898 – 1962), Weil (1906 –) and Serre (1926 –) on the list of leading mathematicians who have spent a great proportion of their energy in the investigations of number theoretic questions.

The basic building blocks of numbers (i.e. integers) are the prime numbers. A prime number is a positive integer greater than 1 which is divisible only by 1 and itself. For example, 2, 3, 5, 7, 11, 13 are prime numbers. So is the number $2^{44497}-1$. This number has 13,395 digits and was proved to be a prime only in the year 1979 using a CRAY-1 supercomputer. It has been known for a few thousand years that every number can be factorized into a product of powers of primes, the so-called prime decomposition. Such basic facts are now taught to school children at an early age. Euclid (end of third century B.C.?) already showed that there are infinitely many primes (i.e. for every positive number there is a larger number which is a prime). Still,questions about prime numbers abound. For example, how are the prime numbers distributed? In other words, for a given positive number x, how many primes are there which are less than x? The Prime Number Theorem answers this by giving an asymtotic estimation. Yet attempts by Riemann to answer this question raised the intriguing problem of studying the distribution of the zeroes of the Riemann zeta function. This leads to the Riemann hypothesis, considered today to be perhaps the most important unsolved problem in mathematics. There is another example: Is every even number (numbers divisible by 2) greater than 2 the sum of two primes? This was conjectured by Goldbach (1690–1764) and, although considerable progress has been made, no solution of the problem is in sight. One more example: For a prime p greater than 2, how many triples (x,y,z) are there which satisfy the equation $x^p + y^p = z^p$? Fermat believed that they were none, and thought he had a proof. Today many think that his proof was wrong. It was not until 1983 that a proof was given (Faltings (1954–)) showing that there are at most finitely many such triples (for $p = 2$ it is well-known that there are infinitely many such triples).

With the advent of modern ultra high speed digital computers, the usefulness of number theory in practical applications emerged unexpectedly within the last ten

years. The applications center around the subject of cryptography, the study of secure communication. The objective is to encrypt messages to make decoding impossible, using the idea of public-key crypto system. It is based on the fact that given a mathematical function, for example the function $f(x,y) = xy$ (product of the numbers x and y), going from the input to the output is easy (from (x,y) to xy), whereas going from the output to the input is difficult (to 'decode' the pair (x,y) from xy is not always possible). Let us take the example of prime numbers under discussion here. Although mathematically speaking there are algorithms to determine where a number is prime, some of these may in practice be very difficult to carry out. The most simple-minded algorithm is that of factorization. Given a positive number n, to decide whether n is a prime number we simply divide n successively by the numbers less than n and greater than 1. If there is one such number which divides n without leaving a remainder, then n is clearly not a prime. We can add a little bit of sophistication into this algorithm. Namely it is not necessary to test the primality of n by the division of numbers less than n. Indeed it is sufficient to test n by dividing it by numbers not exceeding the square root. The reason is that if n has a factor greater than $\sqrt{n}$ and less than n, then it has a factor less than $\sqrt{n}$. Now until 1979, it would require 10^{6680} years of computer time to decide that $2^{44497} - 1$ is a prime number using this algorithm. Clearly it is a very impractical and primitive method of primality testing. Hence given that large prime numbers are difficult to detect, one may take two 100 digit prime numbers, for example, and multiply them to obtain a 200 digit number n. The process of multiplying two numbers to obtain n is straightforward, whereas to factorize n into its prime decomposition may not be practically feasible. One can then exploit this fact to send messages by encoding them into something associated with n, a number which is made public. Anyone wishing to unscramble the messages would need to know tha prime factors of n, which the receiver keeps to himself. Thus the sender and the receiver never need to exchange the secret key for cipher, and yet messages are transmitted.

It follows that factoring a number is of tremendous practical importance. If one could devise a fast computer algorithm for this, then numbers previously thought to have a large prime factorization, and therefore used for public-key crypto system purposes, may have to be discarded for reason of security. If, on the other hand, there is no fast and efficient method for the factorization of numbers, then government and commercial organisations could safely use numbers with large prime factors for encryption purposes. It is interesting to note how problems of very practical nature rely on very pure and abstract mathematics for answer.

While fast algorithms for prime factorization are at the moment still unknown, the closely-related question of primality testing is not. Let us elaborate on this apparently contradictory remark. It is easy to see that knowledge of the prime factorization of a number implies knowledge of the primality of the number. The converse is not true. It is indeed possible to decide that a number is a composite (i.e. not a prime) without factoring. This fact relies on a result called Fermat's Little Theorem which is taught in every introductory course in group theory and number theory. We begin the discussion with a quick review of modular arithmetic invented by Gauss.

Fix a number n. Two numbers a and b are said to be *congruent modulo n* if their difference is divisible by n. We write this as $a \equiv b \ (mod\ n)$. Congruent numbers can be added and multiplied, as in ordinary arithmetic. Thus we have, given $a \equiv b \ (mod\ n)$ and c ≡ d (mod n),

$$a + c \equiv c + d \ (mod\ n) \text{ and } ac \equiv bd \ (mod\ n).$$

The advantage of considering congruence relation over ordinary equality is that it reduces large numbers to relatively small ones so that calculations within the limits of large computers can be carried out. Suppose that n = 8191. Then for a = 0, 1,2.. 8190, we have $a \equiv a$ (mod 8191). However for a = 8191, we clearly have 8191 ≡ 0 (mod 8191), and also 8192 ≡ 1 (mod 8191),. . . ,16381 ≡ 8190 (mod 8191). This means that there is a cycle of length 8191, so to speak, where the numbers modulo 8191 repeat themselves.

Modular arithmetic is also convenient in another way. This is that one can take powers of numbers with relative ease. In other words, given that $a \equiv b$ (mod 8191), one has $a^2 \equiv b^2$ (mod 8191). Hence given that 16381 ≡ 8190 (mod 8191), we get $16381^2 \equiv 8190^2$ (mod 8191). Now 8190 = 2. 4095 = 2. 3. 1365 = 2. 3^2 . 455 = 2. 3^2 . 5. 91. By quick calculation, we have $2^2 \equiv 4$ (mod 8191), $(3^2)^2 \equiv 81$ (mod 8191), $5^2 \equiv 25$ (mod 8191), $91^2 \equiv 90$ (mod 8191). And so $8190^2 \equiv 4.\ 81.\ 25.\ 90$ (mod 8191). The latter is in turn congruent to 1 modulo 8191 Hence $16381^2 \equiv 1$ (mod 8191). The fairly large number 16381^2 is now simply equal to 1 in arithmetic modulo 8191.

Perhaps the most important elementary fact from the point of view of primality testing is Fermat's Little Theorem This theorem states that for all prime numbers n, $a^{n-1} \equiv 1$ (mod n). Thus for n = 8191 which is a prime number, we have $2^{8190} \equiv 1$ (mod 8191). Also, once $2^{44497} - 1$ is known to be a prime, one may theoretically speaking calculate 71^{n-1}, where $n = 2^{44497} - 1$, and find its value modulo n. By Fermat's Little Theorem, we know that this number is always congruent to 1 modulo n.

Thus a way to test primality would be the following: A number n is a composite if it is found, for example, that 2^{n-1} is not congruent to 1 modulo n. But is this a good test? If $2^{n-1} \equiv 1$ (mod n), does it make n a prime? This is important for ensuring that what is thought to be a prime is indeed a prime. If we set n = 341 (equal to 11. 31) then we see that $2^{340} \equiv 1$ (mod 341), even though 341 is not a prime. Thus Fermat's Little Theorem does not provide a good recipe for primality testing. Despite this, it is known that there are very few composite numbers n which satisfy $2^{n-1} \equiv 1$ (mod n). Up to 20,000,000,000 it is known that there are only 19,865 such composite numbers. If we vary the base number from 2 to 3, or to some other number, composite numbers n that satisfy $a^{n-1} \equiv 1 \ (mod\ n)$ for a = 2 may not do so for a = 3, or 4. There do exist composites n that satisfy $a^{n-1} \equiv 1$ (mod n) for all numbers a. These numbers (call Carmichael primes)bear an even stronger resemblance to primes. They are very much rarer than those mentioned above.

In 1977, R. Solovay and V. Strassen introduced a 'Monte Carlo' algorithm for primality testing. The algorithm is inspired by the fact that composite numbers n that satisfy $a^{n-1} \equiv 1 \pmod{n}$ are rare. It has the feature that if the input n is prime, then it will output 'possibly prime'. If the number n is composite, then at least half of the set of numbers a in (1, ... , n) will show that a^{n-1} is not congruent to 1 modulo n, thus testifying that n is not a prime. In other words, the probability is at least 1/2 that the algorithm will output 'n is a composite'. Thus if n is composite and k successive runs are conducted using the algorithm, with k randomly chosen a's between 1 and n, then the probability that n will pass off as 'possibly prime' in the output is less than $(1/2)^k$. This is an important result because for practical purposes, it may be worthwhile to accept a very small error factor if a considerable saving in computer time is achieved. For example, if 200 trials are conducted on a large composite number n (much larger than 200), then the probability that n is thought to be prime after these trials is less than $(1/2)^{200}$, an extremely small number. Thus it takes very few steps to ensure that the number n chosen is 99.99999999999999 % composite, while it may be an impossible task to proof that n is 100 % composite.

In 1983 Adleman, Rumely and Pomerance introduced an extremely efficient algorithm that removed the randomness in the Solovay-Strassen method. In this method one again begins by subjecting the given number n through tests similar to the test of whether $a^{n-1} \equiv 1 \pmod{n}$. If n does not pass all of these tests, it is a composite. Otherwise there is a small set of numbers containing all divisors of n less than or equal to $\sqrt{n}$. Checking these individually one can decide whether or not n is a prime. Using their method as improved by others, it is possible to use a supercomputer to test the primality of a 1000 digit number in seven days, whereas previously it would take about 10^{44} years using other algorithms. About 10^{486} years would be required if one were to use ordinary division as the method of test, using a computer that does a million divisions per second. This shows the enormous progress made on primality testing in recent years. A very interesting feature of the Adleman, Rumely and Pomerance algorithm is that its verification uses the deep mathematical theory of algebraic numbers. This theory was originally developed by Kummer (1810 - 1893) in his attempt to solve Fermat's conjecture. Over the years it has evolved into one of the most important areas in pure mathematics, pursued by many of the most talented minds. The fact that such a pure area of mathematics is applicable in a very practical situation is one of the most surprising elements of this work.

An algorithm is 'fast' if its running time is polynomially bounded. This means that there is a fixed number k such that for each n it takes less than m^k steps to test whether n is a prime, where m is the number of digits in n. From this point of view, the Solovay-Strassen method is 'fast' although it is not determinate. It is known that the Adleman-Rumely-Pomerance method is not 'fast'. This is however a somewhat misleading statement. For a careful analysis shows that for all n less than 10^b, where b = 999,999,999, it takes less than the number of digits in n raised to a fixed power k to test the primality of n. Hence although this method is not 'fast' in the strict sense, it is fast for most practical purposes. G. Miller has introduced a 'fast' algorithm for primality testing under the assumption of the Generalized Riemann hypothesis, a hypothesis which is

as yet approved. Thus again deep concepts in the theory of numbers enter into the design of fast computer algorithms.

All of the examples discussed above were discovered only within the last ten years. Fifteen years ago it would have been impossible to envisage such a thing happening: purest of pure mathematics applied to very practical problems. The age-old saying that fundamental research is very relevant because one day it could be useful has justification after all. What is in store for the future is very hard to tell. One can in any case be sure that some very pure mathematics will be instrumental in the solutions of some very down-to-earth problems.

References

L. M Adleman, C. Pomerance, and R. S. Rumely, On distinguishing prime numbers from composite numbers, Annals of Mathematics 117 (1983), 173-206

G. Miller, Riemann's hypothesis and tests for primality, Journal of Computers and Systems Science 13 (1976), 300-317

C. Pomerance, The search for prime numbers, Scientific American (1983), 122-130

R. Solovay and V. Strassen, A fast Monte-Carlo test for primality, SIAM Journal for Computing 6 (1977), 84-85