Quickening the Pursuit of Primes

Proving that a large number is a prime can be very time-consuming, but a new computer algorithm speeds up the process significantly

BY IVARS PETERSON

> ***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.... the dignity of 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,**
> ***Disquisitiones Arithmeticae,* 1801**

A year ago, a fast computer would have taken more time than the age of the universe to prove that a large number is a prime number; that is, evenly divisible only by 1 and itself. Now, two European mathematicians have devised a computer program that does the same job in less than an hour. They based their work on a new mathematical strategy that tests for primes, announced late in 1980.

Leonard M. Adleman of the Massachusetts Institute of Technology and Robert S. Rumely of the University of Georgia invented the testing algorithm. Hendrik W. Lenstra of the University of Amsterdam in the Netherlands then discovered several variations of the new algorithm, which made the procedure easier to run on a computer. He and Henri Cohen of the University of Bordeaux in France worked out the details of the computer program and recently were able to test 100-digit numbers in seconds.

Carl Pomerance of the University of Georgia, who was also involved, writes in THE MATHEMATICAL INTELLIGENCER (Vol. 3, No. 3), "This basic and centuries-old problem has seen a great profusion of activity within the past five or six years." Research in cryptology (SN: 10/17/81, p. 252) and computer protocols (SN: 9/26/81, p. 205) that depends on the ability to find large primes and the inability to factor quickly is one source of this interest.

For a small number like 323, the simplest method of telling whether it is a prime is to try all the divisors from 2 up to the square root of 323. In this case, 17 divides into it 19 times. Therefore, 323 is not a prime but a composite number. The trial division method works, but it takes too long if the number is large and has no small factors. Pomerance says, "If n is a prime near 10^{40}, the running time will be about one million years."

The sieve of Eratosthenes (named for a Greek who lived 2,300 years ago) generates a list of prime numbers. The method is employed by listing all the numbers from 2 to some higher number. Then circle 2, and strike out all multiples of 2. The next unmarked number is 3. Circle it, and strike out all multiples of 3. The procedure continues until only prime numbers, now circled, are left on the list. However, this method, equally effective for proving whether a number is prime, requires considerable computer storage space and a lengthy printout for any reasonably large numbers.

Both the trial division method and the sieve of Eratosthenes not only can prove primality but also can factor. Because factoring is hard, a speedier method would gain time by asking not for factors if the number is composite but simply for a determination of whether a number is prime.

In the 17th century, Pierre de Fermat provided the basis for today's faster primality-testing algorithms. He discovered that if p is a prime number and a is a number between 1 and p-1, then the remainder when a^{p-1} is calculated and divided by p will be 1. Thus, for $p = 11$ and $a = 2$, $2^{10} =$ 1,024, and 1,024 divided by 11 has a remainder of 1. This works for all prime numbers. If the remainder is not 1, then the number is composite. However, the difficulty is that a few composite numbers also give a remainder of 1. These composites are called pseudoprime numbers.

Although calculating 2 (or some other number) to a high power seems even more formidable than trial division, mathematical shortcuts are possible because only the remainder is of interest. All that is needed is some further test to weed out the pseudoprimes, which are rare. For example, below 10^{10} there are 455,052,512 primes, but only 14,884 pseudoprimes when a is 2. The composite number 561 is the smallest pseudoprime for all allowed choices of a.

Adleman says, "Since the basic Fermat test does an excellent job, people have tried to generalize Fermat's test so as to exclude those numbers that are faking being primes with this test." Adleman used the idea that more subtle, discriminating tests not only say "pass" or "fail" but also provide information about the numbers.

Pomerance says, "You do many different kinds of tests, and you learn more and more about this composite number that is masquerading as a prime number. After you've done the last of these tests, you know so much about the number that you know any divisor of this composite number must be among a very small set of numbers." If none of the divisors works, the number is prime.

Adleman describes the results of the tests as a mosaic. "Information about the number is caught up in the mosaic," he says. Looking at a sufficiently large piece of the mosaic will give enough information to decide whether a number is a prime or a composite. "It's like a hologram in that the nature of the number you are dealing with can be found in any small section of the mosaic," Adleman says. This is what helps shorten the running time of the algorithm.

The original paper by Adleman, Rumely and Pomerance presented the theory behind the new primality testing algorithm. Lenstra and Cohen made the algorithm more compatible with real computer applications, implementing it in an elegant way without changing the fundamental ideas. "While theoretically the computer program that was written doesn't change the algorithm, it is the proof of the pudding to a lot of people," says Adleman.

"If you wanted to check a 100-digit number previous to this algorithm, then it would probably have taken in excess of 100 years," says Adleman. "With the new algorithm, a 100-digit number can be done in 15 seconds, and from what I hear from the Europeans, they may get a three-fold improvement in that by the time they fine-tune things."

Adleman, who has worked on public-key cryptologic systems, believes the prime-testing algorithm represents no threat to computer security codes that depend on factoring. In fact, the algorithm may be useful in providing large primes for use in the codes.

"My own intuition is that factoring is hard, and primality testing is easy," Adleman says. "From a mathematical point of view, they are closely related. Nonetheless, it's not unusual to run across problems that are extremely closely related where one is intractable and the other is easy."

Pomerance is a little more hesitant. He writes, "Is factoring inherently more difficult than distinguishing between primes and composites? Most people feel that this is so, but perhaps this problem too will soon yield." □