PCs Factor a "Most Wanted" Number

Mathematicians at University of Georgia mobilize phalanx of personal computers to factor a 95-digit number

A NETWORK OF COMPUTERS grabbed headlines recently by factoring a 100-digit number, but a group at the University of Georgia at Athens has had significant success of another sort: a 95-digit factorization done on a small army of personal computers. William Alford, Carl Pomerance, and Jeffrey Smith implemented a new variation of a factoring algorithm known as the "quadratic sieve" on 140 Zenith microcomputers to attack the 95-digit number which remains when the "small" factors 17 and 11,953 are divided out of the number of $2^{332} + 1$.

Factoring numbers, such as $105 = 3 \times 5 \times 7$, sounds easy enough—but then most things that computers do are basically easy. It is the size of the problem that causes trouble. For factoring, the obvious approach of trial-and-error division rapidly exceeds the capacity of even imaginary computers, with run times dwarfing the age of the universe.

Nevertheless, factoring numbers in the 90-digit range is becoming routine. Mark Manasse of the Digital Equipment Corporation and Arjen Lenstra of the University of Chicago recently became the first to reach the 100-digit mark, breaking the number $(11^{104} + 1)/11^8 + 1)$ into factors of 41 and 60 digits respectively (*Science*, 21 October, p. 374). They did so by parceling out the work to hundreds of computers in a dozen locations in the United States, Holland, and Australia. The Georgia project now shows that big machines are not crucial—the key is in the algorithm.

Improvements on trial-and-error factoring have been around for ages. The quadratic sieve is based on a method introduced in the 17th century by Pierre Fermat, whose clever ideas in number theory continue to haunt the subject. Fermat's approach to factoring a number N was to find two numbers X and Y such that N equals $X^2 - Y^2$, which factors as $(X - Y)(X + Y)$. A significant modification of Fermat's approach is to require only that N divide, not necessarily equal, $X^2 - Y^2$. The idea is that the distinct prime factors of N will "randomly" choose to divide $(X - Y)$ or $(X + Y)$. Unless all of the primes dividing N accidentally make the same choice, computing the greatest common divisor of $X - Y$ (or $X + Y$) and N will produce a factor of N.

If N is the product of two primes, there is something like a 50:50 chance of succeeding with any given choice of X and Y; and odds go up if N has more than two prime factors. It thus suffices to have a handful of "independent" choices for X and Y for which N divides $X^2 - Y^2$. The odds that, say, ten choices of X and Y would fail to produce a factorization are comparable to the odds of flipping a coin ten times and always getting tails.

The obvious approach of trial-and-error division rapidly exceeds the capacity of even imaginary computers.

The obvious problem, of course, is how to find the numbers X and Y. In the 1920s Maurice Kraitchik suggested piecing together X and Y out of numbers x for which the remainder of x^2 divided by N factors easily into small primes belonging to a preestablished "factor base" for N. Given enough such x, some combination of the remainders can likely be found whose product has only even powers of the small primes, and hence has the form Y^2. (For example, if $N = 22$, then 5^2 has remainder 3, 7^2 has remainder 5, and 9^2 has remainder $15 = 3{\times}5$, and one can form $Y^2 = 3 \times 5 \times 15 = 3^2 5^2$.) Letting X be the product of the corresponding x's, it is an elementary fact in number theory that N divides the difference $X^2 - Y^2$.

The crux of the problem now shifts to finding numbers x for which the remainder of x^2 factors nicely over the factor base. Several methods exist, but currently the most efficient is the quadratic sieve, which Pomerance introduced around 1981. The quadratic sieve operates by repeatedly scanning a range of x's slightly in excess of $\sqrt{N}$. For each prime p in the factor base, the sieve finds the first x for which p divides the remainder $x^2 - N$. This involves only a small amount of trial and error. The sieve then skips to the locations $x + p$, $x + 2p$, and so forth, until it reaches the end of the range; these are the only values for which p divides the remainder. Once this has been done for all primes in the factor base the sieve easily finds the x's whose remainders have been completely factored. (The exact operation of the sieve is slightly different from this schematic description. Also, numerous modifications have been made to speed up the procedure; see accompanying article.)

The quadratic sieve is not the only factoring algorithm currently in use, but it is currently the best for numbers that are the product of two primes of roughly the same size. A randomly produced number will generally have several small prime divisors which can be found by trial-and-error. Two methods introduced by John M. Pollard in the mid-1970s are useful for finding prime factors in the 15-digit range and prime factors with a special property (the "Pollard $p - 1$ test" finds prime factors p for which $p - 1$ factors completely into small primes). A few years ago Hendrik Lenstra, Jr. (a brother of Arjen Lenstra), introduced an "elliptic curve method" which generalizes Pollard's $p - 1$ test. The elliptic curve method has proved highly successful at finding prime factors in the 20- to 30-digit range.

Rigorous estimates are hard to come by, but heuristic arguments indicate that the amount of computation done by the quadratic sieve grows like $N^{\sqrt{(\ln \ln N/\ln \quad N)}}$, where $\ln N$ is the natural logarithm of N. This estimate compares very favorably to $N^{1/2}$, which is the corresponding estimate for trial-and-error factoring. For numbers in the 100-digit range, the quadratic-sieve estimate roughly doubles with each three digits added, whereas the trial-and-error estimate increases by a factor of 10 with each two digits.

Implementing the quadratic sieve on microcomputers was proposed a year ago by Alford, who wrote code in assembly language—which he had learned by programming video games for his children—in order to get the sieve operating in the microcomputers' limited memory. (The sieve used a factor base with more than 32,000 primes.) The machines sieved at night, weekends, and during holidays from June to Labor Day. The factorization was finally completed on 23 October at 8:05 p.m. (factorizations have come to be timed to the minute, somewhat like the birth of a child): In addition to 17 and 11,953, $2^{332} + 1$ contains two factors of 44 and 52 digits.

When Alford began, $2^{332} + 1$ was the largest "hard" number to be attempted on any system, and the 32,000-prime factor base was the largest of its kind. Manasse

Lenstra's factorization of $11^{104} + 1$, which contains a 100-digit "hard" part when the "small" factor $11^8 + 1$ is removed, claimed the record for raw size on 11 October, with a factor base of 50,000 primes.

Alford's result has been registered with Samuel Wagstaff of Purdue University, who maintains the "ten most wanted" list of numbers to be factored. These are mostly numbers remaining from a project begun in 1925 to completely factor numbers of the form $bn \pm 1$.

Alford's number had been third on the list. Currently at the top is the 148-digit number that remains when $2^{512} + 1$ is divided by 2,424,833. Five years ago the most—wanted list consisted of numbers ranging from 53 to 71 digits. Most of these causes were by the elliptic method.

How is it that mathematicians know in advance that the numbers on the most-wanted list actually do have factors and are not simply large primes themselves? The answer to this also originates with Fermat. Fermat proved that if N is a prime, then for any number b, N divides the number $bN - b$. For instance, 5 divides $2^5 - 2 = 30$, whereas 6 does not divide $2^6 - 2 = 62$. Fermat's test will not prove that a number is prime—more elaborate tests are required to certify primality—but failing it even once immediately proves nonprimality. In general a nonprime number will fail Fermat's test for almost all b's.

Pomerance and Smith are working on another implementation of the quadratic sieve that promises to eclipse all others: a special-purpose, breadbasket-sized computer that does nothing but sieve. When completed and hooked up with a Sun workstation to handle other parts of the algorithm, the sieving device should be able to factor a 100-digit number in 2 to 3 weeks. Smith, who is actually building the machine, says that the device will be "most efficient" for numbers up to 115 digits.

Pomerance emphasizes cost-effectiveness as a criterion for evaluating factoring projects. Given $10 million (roughly the cost of a supercomputer), what sort of factoring system should you invest in and how large a number could such a system factor in one year? The sieving device should cost about $25,000 to duplicate, Pomerance says. He estimates that $10 million would buy the factorization of a 144-digit number in 1 year. The extrapolated cost of factoring a 200-digit number in a year would be about $100 billion—or, Pomerance points out, "only" about 6 months' interest on the national debt. ■ **BARRY A. CIPRA**

Barry A. Cipra is a mathematician and writer based in Northfield, Minnesota.

Souping up the Sieve

Hot-rodders call it boring and stroking: making small changes in the diameter (bore) of an engine's cylinders and the length of the piston stroke in order to gain power to make the car run faster. Factoring enthusiasts have proved equally resourceful in modifying their implementations of the quadratic sieve.

The most radical change, equivalent to adding more cylinders, is the use of more than one polynomial to produce the square numbers which are sieved over the factor base. Pomerance's "Model T" sieve used the single expression $P(x) = (x + [\sqrt{N}])^2$ and sieved through "small" values of x. ($[\sqrt{N}]$ denotes the integer part of $\sqrt{N}$.) The problem is that the "small" values of x actually get quite large, resulting in large values of $P(x)$, which are less likely to factor completely over the factor base.

Around 1983, James A. Davis at Sandia National Laboratories and Peter Montgomery at the System Development Corporation in Los Angeles independently discovered that it was helpful to change polynomials. The basic idea is to use polynomials of the form $(ax + b)^2$, where b is chosen so that a divides $b^2 - N$. The polynomial $ax^2 + 2bx + (b^2 - N)/a$ is then sieved over a much shorter range. The use of many polynomials makes possible the distributed processing employed by Manasse and Lenstra and the Georgia project—each processor is given a separate batch of polynomials to sieve through.

There is a price to pay, however: each polynomial requires an "initialization" which involves a computation with the coefficient a that must be done for each prime p in the factor base. With factor bases having tens of thousands of primes, sieve initialization becomes a substantial fraction of the overall effort.

Pomerance's group has found a way to get more than one polynomial out of a single initialization, and employed it in the factorization of $2^{332} + 1$. The idea is to use a's which are the product of several smaller primes. It turns out that each additional prime factor in a doubles the number of b's (less than a) for which $b^2 - N$ is divisible by a, thus doubling the number of polynomials without affecting the initialization step. For $2^{332} + 1$, the a's were taken to have three prime factors (giving four polynomials per a).

Sieve initialization could be virtually eliminated by forming all a's as products of, say, 10 primes taken from a fixed set of 20 primes. There is a huge number of such combinations, and each choice yields over 500 polynomials. Doing so would use a lot of memory, however: it would require storing the results of computations pairing each of the 20 fixed primes with each prime in the factor base.

Another modification that most implementations use is known as the large prime variation. The problem with using a factor base is that a lot of numbers do not quite factor completely over it. The large prime variation keeps track of numbers that leave one extra ("large") prime factor when they are raked over the factor base. If enough such numbers are accumulated, examples can usually be found of numbers having the same extra prime, in the same way that a group of 25 or so people usually has at least one pair of matching birthdays. Alford found 25,000 matches out of roughly 900,000 numbers having an extra prime. (He also found 14,000 numbers that factored completely over the factor base.)

The latest modification coming out of Georgia is to the part of the algorithm that finds combinations of remainders whose product is a square. That step is best cast as a matrix reduction problem. Each row of the matrix corresponds to a number x, and each column to a prime p in the factor base. The entry in "row x" and "column p" is a 1 if the exponent of p in the remainder of x^2 is odd, and a 0 if the exponent is even. The matrix turns out to be rather sparse: there are many more 0's than 1's. This much is old hat. The modification is to exploit the fact that the matrix is not uniformly sparse. Andrew Odlyzko of Bell Labs introduced what Pomerance calls an "intelligent Gaussian elimination" scheme in 1984 for reducing sparse matrices with a small number of "heavy" columns. The Georgia group has found a variant that works for matrices that are not quite as sparse, as typically occur in factorization problems.

Researchers will continue tinkering with the quadratic sieve. But just as race cars do not go appreciably faster than 200 miles per hour, it is unlikely that the quadratic sieve will factor numbers much in excess of 200 digits, if that many. What is needed, researchers agree, is an entirely new mode of transportation.

Beam me up, Scotty. ■ **B.A.C.**