Factoring on Microcomputers

It is now possible to factor large numbers—and factor them fast—with a series of microcomputers hooked up to run in parallel

MANY results in mathematics are slow to be distributed and even slower to be published. But factoring—breaking numbers down into primes, their smallest component parts—is an exception. The work is a sort of mathematical game that also has national security implications because factoring is necessary for a type of code-breaking. The field of factorers is wide, and the race to see who can factor the largest and most difficult numbers has become so competitive that a monthly factoring newsletter, edited by Samuel Wagstaff of Purdue University, announces the latest results.

This year, faithful readers of the newsletter have noticed something new. All of a sudden, the newsletter is dominated by two dark horses in the factoring race, Robert Silverman of the Mitre Corporation in Bedford, Massachusetts, and Peter Montgomery of System Development Corporation in Santa Monica, California. Nearly all of the 50 or so new factorings announced each month in the past year have been done by these two. Silverman and Montgomery, says Gus Simmons of Sandia National Laboratory, "are carving up the world of factoring."

What Silverman and Montgomery have is a new way of factoring that relies on microcomputers hooked up so that they work independently on different parts of the same problem. In contrast, other factoring methods rely on ultrafast computers or special-purpose computers designed only for factoring. The method developed by Simmons and his colleagues, for example, relies on the world's fastest computer, the Cray XMP. The advantage of the microcomputer method is that it is very inexpensive but still fast. In general, says Simmons, "factoring is rarely a race against time. What really concerns people is cost." So even if the microcomputers take a little longer than a Cray XMP, there is no question that they are much more cost-effective.

Silverman and Montgomery recently factored an 81-digit number that had never been factored before, using eight microcomputers, each of which ran for 150 hours. The Sandia group believes it could have factored a number of that size in 12 hours on the Cray, but, says Simmons, the cost would have been substantially greater.

Factoring is one of the oldest mathematical problems—it dates back to the time of the ancient Greeks. But it is only comparatively recently that mathematicians made significant progress. They developed clever factoring algorithms, for example, and learned to exploit computer designs in implementing the algorithms.

Silverman says he has been advised, for national security reasons, not to speculate on the size of the numbers that can be attacked by his method.

To factor a number, it must be broken into the smaller prime numbers that make it up. This is simple for small numbers. They can be factored by trying all primes smaller than their square root to see which are divisors. For example, 30 is $2 \times 3 \times 5$. But large numbers require special tricks. Hugh Williams of the University of Manitoba points out that a computer performing a billion operations per second would take more than 35,000 years to factor the 58-digit number $2^{193} - 1$ with this brute force method.

The number of computations involved in factoring a number grows exponentially as numbers get larger. For every three digits added to a number, the factoring time doubles, Simmons says. But it is difficult to predict just how big a number must be for it to be beyond mathematicians' reach.

The question of what size numbers can be factored is of practical importance because a code, called RSA, relies for its security on the difficulty of factoring. The RSA code is at least under consideration if not in actual use by the National Security Agency and the Defense Department; its security is of great interest. Numbers of any size can be chosen as the basis of the code, so it might seem that, to play it safe, enormous numbers should be selected. But the larger the numbers, the slower the encoding. The goal in using the RSA code is to choose numbers that are beyond the reach of factorers but not unnecessarily large.

When the RSA code was proposed in 1977, its developers suggested using 80-digit numbers, reasoning that they could not be factored. Now they suggest using 200-digit numbers which, for the time being, appear safe. Silverman says he has been advised, for national security reasons, not to speculate on the size of the numbers that can be attacked by his method.

The new method had its origins 2 years ago when Montgomery suggested a way of improving a popular factoring scheme, called the quadratic sieve. Silverman took Montgomery's mathematical ideas and implemented them first on a VAX and now on a microcomputer.

The idea behind the quadratic sieve is to concentrate on factoring a very large collection of numbers, each of which is much smaller than the number you are interested in and then to use the information from those smaller problems to factor the large number. It sounds like the kind of problem that could easily be tackled with a collection of computers running all at once and independently—just give each computer its own collection of smaller problems. But the way the method was originally used, the smaller problems were interdependent. They could not be broken up and solved in isolation from each other. What Montgomery did was to find a way to make the smaller problems independent.

The 81-digit number that Silverman and Montgomery factored is on the mathematicians' "most wanted list," a list of numbers compiled by John Selfridge of the University of Northern Illinois, that are known to have factors but whose factorizations have eluded everyone's best attempts. Selfridge intentionally designs the numbers on the most-wanted list so that they are challenging but not ridiculously hard. The list contains no 200-digit numbers, for example. Yet factoring progress has been so rapid that the list has been revised twice in the past few years and now the factorers would like to see it revised again.

What is ironic about the new result is that it reverses the factorers' trend toward using larger and larger or more and more specialized computers. It shows once again that in factoring, as elsewhere in science, it pays to expect the unexpected. ■ GINA KOLATA