ON THE DISCOVERY OF THE 38th KNOWN MERSENNE PRIME

George Woltman
8817 Lake Sheen Ct., Orlando, FL 32836
e-mail: woltman@magicnet.net

(*Submitted July 1999-Final Revision September 1999*)

1. INTRODUCTION

There is a long, rich history in the search for primes of the form $M_p = 2^p - 1$, named *Mersenne primes*. In the early years the study of Mersenne primes led to several important advances in number theory. In recent decades the computer has become instrumental in the search for large primes leading to several algorithmic advances. This paper provides an overview of the history (see [6]) and techniques used in finding the currently largest known explicit Mersenne prime.

It is helpful to tour briefly some known properties of Mersenne primes. The possible exponents p are restricted by an elementary theorem that says: if M_p is prime, then p is prime. Furthermore, known factors of M_p must of necessity be of the form $q = 2kp+1$; also, when $p > 2$ we must have $q = 1$ or 7 (mod 8). These properties can be used effectively to quickly sieve out many composite M_p, thus eliminating the need for the comparatively expensive Lucas-Lehmer test, a definitive and rigorous primality test described below. There is also the ancient connection with perfect numbers. A perfect number equals the sum of its positive divisors excluding itself. The first two examples are $6 = 1+2+3$ and $28 = 1+2+4+7+14$. The Euclid-Euler theorem states that an even number n is perfect if and only if it is of the form $n = 2^{p-1}M_p$, where M_p is prime. Therefore, one can say that every new Mersenne prime discovery immediately begets a new perfect number.

Beyond these proven facts there has been a great deal of conjecture and mystery pertaining to the Mersenne primes. Two widely-believed heuristic arguments (see [3] and [8]) say that the probability that M_p be prime is

$$(e^{\gamma} \log ap) / p \log 2), \text{ where } a = 2 \text{ if } p \equiv 3 \pmod 4 \text{ and } a = 6 \text{ if } p \equiv 1 \pmod 4$$

and the number of Mersenne primes less than or equal to x is about

$$(e^{\gamma} / \log 2) * \log\log x,$$

where $\gamma \approx 0.577$ is the Euler constant. It is interesting to compare this heuristic with the occurrence of the currently known Mersenne primes. Chris Caldwell has done just that, providing several graphs examining these conjectures at

http://www.utm.edu/research/primes/notes/faq/NextMersenne.html.

Since 1952, Robinson, Riesel, Hurwitz, Gillies, Tuckerman, and Noll & Nickel all used the most powerful computers of their day to find new Mersenne primes. In 1979, Slowinski, sometimes partnering with Nelson or Gage, began a 17-year reign finding seven ever larger Mersenne primes with Cray supercomputers. Colquitt and Welsh found an overlooked Mersenne prime in 1988. In 1996, Woltman and Kurowski developed a kind of "Internet supercomputer" that has found the last 4 Mersenne primes. Once again, Caldwell's http://www.utm.edu/research/primes/mersenne.shtml

web page provides a superb and more complete history.

2. RESULTS

On June 1, 1999, the Mersenne prime $2^{6972593}-1$ was discovered by Nayan Hajratwala on his personal computer using software developed by the author and Scott Kurowski. Hajratwala is one of over 12,000 participants in the Great Internet Mersenne Prime Search (GIMPS). As of this writing, the prime is the largest known explicit prime of any type.

The new prime number is 2098960 decimal digits long. It took 111 days running part-time on a 350 MHz Pentium II computer to prove this number prime. Running non-stop the test would have taken just over three weeks.

The number was subsequently verified as prime by three independent parties, each party employing different hardware *and* software. Gerardo Cisneros used David Slowinski and Paul Gage's program on the 4-CPU CRAY Y-MP at the National Autonomous University of Mexico's General Directorate of Academic Computing Services (DGSCA/UNAM). David Willimore used a program by Ernst Mayer on an Aerial Communications 500 MHz Alpha workstation. Cornelius Caesar used a program by John Sweeney on an IBM RS/6000.

There are now 38 known Mersenne primes. M_p is prime for $p=2$, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937, 21701, 23209, 44497, 86243, 110503, 132049, 216091, 756839, 859433, 1257787, 1398269, 2976221, 3021377, 6972593. As we will see later, the new prime number may not be the 38th Mersenne prime by size. In fact, there could be exponents below 2976221 that yield a new Mersenne prime. The current status of the search is frequently updated at

http://www.mersenne.org/status.htm.

3. IMPLEMENTATION OF THE LUCAS-LEHMER TEST

One requirement for finding such gargantuan Mersenne primes is a fast and efficient implementation. The search program uses the celebrated Lucas-Lehmer primality test (see [4] and [5]), a convenient variant of which is to define the sequence starting with $x_1 := 4$, and iterate

$$x_{n+1} := x_n^2 - 2 \pmod{M_p}$$

through the index $n = p-1$. M_p is prime if and only if $x_{p-1} = 0 \pmod{M_p}$. It is clear that the main operation in this rigorous test is that of squaring of numbers of size p bits. Schönhage and Strassen [7] showed that a Fast Fourier Transform (FFT) can be used to square a p-bit number in $O(p \log p \log\log p)$ bit operations. A very rough description of the procedure is as follows. Because multiplication is essentially what is called "acyclic" convolution, and since acyclic convolution is a cyclic convolution of zero-padded sequences, and since the cyclic case can be handled by Fourier transforms, one may zero-pad a p-bit number x to be squared, resulting in a number with about 2p bits, followed by FFT-based convolution to get the desired integer product. In the early 1990s, Richard Crandall [2] observed that, since squaring modulo M_p is a length-p cyclic convolution of the bits of x, it is possible to expand x into approximate digits in an irrational base 2^r, where $r = q/2^k$, and thereby, via cyclic convolution, accomplish two things: eliminate the zero-padding entirely, and allow power-of-two run lengths in the Fourier transforms. In this fashion, the search time for large Mersenne primes was effectively halved. The present author implemented this "irrational base discrete weighted transform" (IBDWT) algorithm in assembly language to take advantage of the pipelining capabilities of the Intel architecture. One extension of the algorithm was to allow convenient, but non-power-of-two run lengths, which further increased search efficiency. The author also used an in-place variant of a David Bailey [1] idea on avoiding power-of-two memory strides that have a devastating impact on memory caches.

The second ingredient for a successful search is a great deal of computing power. Traditionally this involved the use of supercomputers. Today, there is a better alternative—distributed computing. Distributed computing uses the idle cycles on thousands of ordinary computers connected to the Internet. Scott Kurowski, founder of Entropia.com, developed software that makes this easy. An Internet user runs the prime search program which automatically contacts a central server to get a work assignment. The computer works on this assignment off-line at low priority. When the assignment completes the program contacts the server to report its results and get a new work assignment.

The Great Internet Mersenne Prime Search (GIMPS), which was founded in 1996, has a goal of methodically testing all Mersenne numbers up to achievable limits. Slower machines look for small factors of Mersenne numbers, and so perform the kind of sieving mentioned in the Introduction. Faster machines run the Lucas-Lehmer primality tests. Medium speed machines rerun primality tests to make sure the first test ran properly (the last 64 bits of the last Lucas-Lehmer iteration are compared). The distributed nature of the search process means that there are always "gaps" in the testing and double-checking process. There are over 12000 exponents below 6972593 that have not finished testing and over 60000 exponents that have not been double-checked including some below 2976221.

To run a successful distributed project, the central server must do more than hand out assignments and track results. The Entropia.com server must reassign work that is not completed in a timely manner. To help users running on several computers, the server provides reports so the user can keep track of each computer's progress. Finally, many users enjoy seeing where they or their team stand in a "top producers" report. The server also sends out periodic newsletters to interested users. It is important to provide these "extras" to keep an all-volunteer work force enthused and up-to-date on current events.

4. THE FUTURE

The Electronic Frontier Foundation, www.eff.org, is offering awards of $100,000 and up for the discovery of primes with 10 million, 100 million, and 1 billion digits. While the 10 million digit award may be within GIMPS reach—it is "only" 125 times more difficult than finding a 2 million digit prime—the larger primes will require significant breakthroughs in primality testing or multiplication algorithms.

There are several factors that affect when the next Mersenne prime will be found. Obviously, the size of the exponent is one. Based on past percentage distances between Mersenne primes, the next exponent could be as high as 28.6 million! The number of computers participating in GIMPS is always growing and the speed of the average computer gets faster every year. While the next Mersenne prime could be found tomorrow, the chances are GIMPS will find the next one sometime during the next 2 years. It will be hard to sustain GIMPS' track record of 4 Mersenne primes found in less than 4 years.

5. CONCLUSIONS

It would appear that many supercomputing chores of the future will no doubt be performed by vast networks of computers. Many other fields of computation—ranging from medical data processing to searches for extraterrestrial intelligence—may well benefit from such massive parallelism. It is hoped that this Mersenne discovery serves as a kind of example of what is possible using these techniques.

In many fields, new research will be needed to develop algorithms that can be run on such a massively parallel network using a minimum of bandwidth. Many businesses and universities will eventually harness power of the many small computers they own to do computationally intensive research more economically.

ACKNOWLEDGMENTS

While there are far too many people to list here, the author wishes to thank the huge number of people that have contributed to the GIMPS project. While most have contributed computer time, many have also contributed by writing software, recruiting others, running mailing lists, helping new users, and hosting web pages.

REFERENCES

1. D. H. Bailey. "FFTs in External or Hierarchical Memory." *NAS Technical Reports* RNR-89-004, 1989.
2. R. Crandall & B. Fagin. "Discrete Weighted Transforms and Large-Integer Arithmetic." *Mathematics of Computation* **62** (1994):305-24.
3. R. Crandall & C. Pomerance. *Prime Numbers: A Computational Perspective.* New York: Springer-Verlag, 2000.
4. G. H. Hardy & E. M. Wright. *An Introduction to the Theory of Numbers*. 5th ed. Oxford: Clarendon Press 1979.
5. D. E. Knuth. *Seminumerical Algorithms*. Vol. 2: *The Art of Computer Programming*. 3rd ed. Reading, MA: Addison-Wesley, 1998.
6. P. Ribenboim. *The New Book of Prime Number Records*. New York: Springer, 1996.
7. A. Schönhage & V. Strassen. "Schnelle Multiplikatioin großer Zahlen." *Computing* 7 (1971): 208-13.
8. S. Wagstaff, Jr. "Divisors of Mersenne Numbers." *Mathematics of Computation* **40** (1983): 385-97.

AMS Classification Numbers: 11-04, 11A41

❖❖❖