Parallelism between the growth of the known Mersenne primes and the development of informatics

Attila Pethő

Department of Computer Science, University of Debrecen

Paris Lodron University of Salzburg
Salzburg, 19 June, 2017.

1. Prolog

On January 6, 2016 the Great Mersenne Prime Search, GIMPS, announced that Curtis Cooper found the actually largest known prime number: $2^{74,207,281}-1$. It has $22,338,618$ decimal digits.

To store such a big number one needs a bit less then 10 MB memory.

The pdf version of the 640 pages book *Unit equations and discriminant equations* of Jan-Hendrik Evertse and Kálmán Győry is 2.6 MB.
The scanned version of the 439 pages *Number Theory* by Z.I. Borevich and I. R. Shafarevich is 14,3 MB!

How is it possible to prove the primality of such a large number?

Why we will prove the primality of such a large number?

I concentrate mainly to the first question.

2. Perfect numbers and results before 1876

An integer is called *perfect*, if the sum of its proper divisors is equal to it. For example 6 and 28 are perfect, because their proper divisors are $1, 2, 3$, and $1, 2, 4, 7, 14$ respectively, and $1 + 2 + 3 = 6$, and $1 + 2 + 4 + 7 + 14 = 28$.

Still not known whether odd perfect numbers exist.

Euklid-Euler: *an even number is perfect if and only if it has the form $2^{p-1}(2^p - 1)$ such that both p and $2^p - 1$ are primes.*

In the sequel I use the notation $M(n) = 2^n - 1$.

Marin Mersenne (1588-1648) claimed: $M(p)$ with $p \leq 257$ is a prime if and only if $p = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$!

The list is not correct: $M(p)$ is not a prime for $p = 67$ and 257, but is prime for $p = 61, 89$ and 107.

Before 1876 only for the primes $p = 2, 3, 5, 7, 13, 17, 19$ and 31 was proved that $M(p)$ is a prime.

They used congruence considerations and hand computation:

Theorem 1. *If p is an odd prime and the prime q divides $M(p)$, then $q \equiv 1 \pmod{2p}$ and $q \equiv \pm 1 \pmod{8}$.*

Theorem 2. [L. Euler(1772)] *Let $p = 4k + 3$ be a prime. The number $q = 2p + 1$ is a prime if and only if $q | M(p)$.*

For example if $p = 251 = 4 \cdot 62 + 3$, then $q = 503$ is a prime, thus $M(251)$ is not a prime.

L. Euler: $2^{31} - 1 = 2147483647$ is a prime.

Peter Barlow (1776-1862) was an influential mathematician and physicist of his time. In 1823, he was made a fellow of the Royal Society. His book *New Mathematical Tables* became known as Barlow's Tables and were regularly reprinted until 1965.

He wrote in 1811 about Euler's discovery:" *[it] is the greatest perfect number known at present, and probably the greatest that ever will be discovered; for, as they are merely curious without being useful, it is not likely that any person will attempt to find one beyond it.*"

Barlow went wrong, because:

- Édouard Lucas discovered in 1876 a very powerful method for the test of primality of Mersenne numbers;
- from the middle of the 20th century many efficient computer implementation of the Lucas test appeared;
- A. Schönhage and V. Strassen (1971) found an algorithm for multiplication of integers, which is practically linear (but is useful only for extremely large numbers);
- the supercomputers and the millions of computers connected to the web give enormous computing capacity.

People like to do useless thinks.

A reporter asked Edmund Hillery, why he climbed Mount Everest. He answered: ”because it is there”.

By my opinion climbing the Mount Everest is as useful as searching for very big Mersenne primes.

3. The Lucas-Lehmer test and its history before 1952

Theorem 3. *Let p be a prime. Set $u_1 = 4$ and $u_{n+1} = u_n^2 - 2, n \geq 1$. The number $M(p) = 2^p - 1$ is a prime if and only if $M(p) | u_{p-1}$.*

Sufficiency: E. Lucas (1876), necessity: D.H. Lehmer (1930).

The Lucas-Lehmer test combined with hand computation proved that the following numbers are primes:

p	year	discoverer
61	1883	I.M. Pervushin
89	1911	R. E. Powers
107	1914	R. E. Powers
127	1876	E. Lucas

4. The Lucas-Lehmer test+computer 1952-1979

The Lucas-Lehmer test can be implemented on a digital computer easily. One has to compute the sequence u_n only modulo $M(p)$! Moreover, one has to implement multiprecision arithmetic, i.e. addition, subtraction, multiplication and division with remainder with arbitrary large integers. The growth of the size of the known Mersenne primes depended only of the growth of the speed and memory size of the computers.

Raphael M. Robinson (5), Hans Riesel (1), Alexander Hurwitz (2), Donald B. Gillies (3), Bryant Tuckerman (1), Landon Curt Noll & Laura Nickel (1), Landon Curt Noll (1) discovered new Meersenne primes.

The values of the exponent was: $521, 607, 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937, 21701, 23209$.

R.M. Robinson, 1954: *At that time, the total memory of the SWAC consisted of 256 words of 36 binary digits each, exclusive of the sign. For the Mersenne test, half of this memory was reserved for commands. Since successive squarings of numbers less than the modulus* $2^n - 1$ *are required, this modulus was restricted to 64 words, so that the condition* $n < 64 \cdot 36 = 2304$ *was imposed.*

D.B. Gillies, 1963: *As an indication of the speed of Illiac II, the residue for* $M(8191)$ *took 100 hours on Illiac I (D. J. Wheeler), 5.2 hours on an IBM 7090 (Hurwitz [2]) and 49 minutes on Illiac II. The three values agree.*

Conjecture. (D.B. Gillies, 1963). *If $A < B \leq \sqrt{M(p)}$ and B/A and $M(p) \to \infty$, then the number of prime divisors of $M(p)$ lying in the interval $[A, B]$ Poisson distribution with mean*

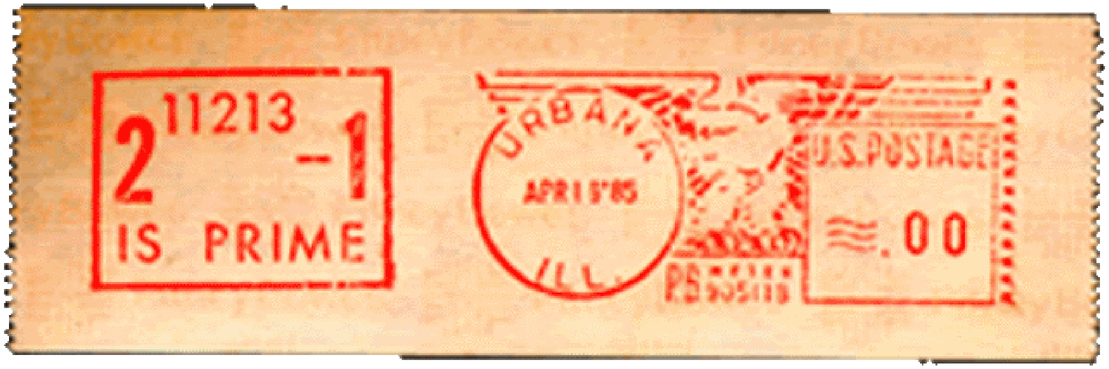
$$\textit{mean} \sim \begin{cases} \log(\log B / \log A), & \textit{if} \quad A \geq 2p \\ \log(\log B / \log 2p), & \textit{if} \quad A < 2p \end{cases}$$

paraméterrel.

Corollary.

- *The number of Mersenne primes, which are less than x is $\sim \frac{2}{\log 2} \log \log x$.*
- *There are in average two primes p in the interval $[x, 2x]$, for which $M(p)$ is a prime.*
- *The probability that $M(p)$ is a prime is $\sim \frac{2 \log 2p}{p \log 2}$.*

The post stamp of the University of Illinois with the discovery of D.B. Gillies in 1963.



C. Noll and L. Nickel, 1979: *The amount of computation needed to check the primality of* $M(p)$ *is* 0(p^3)*. Here, the major computational effort is in squaring the* u_k *since division by* $2^n - 1$ *is readily accomplished by shifting. It may be possible to implement a faster multiplication method. For example, Schoenhage and Strassen [2] have an algorithm, based on fast Fourier transforms, which may be promising.*

5. The Lucas-Lehmer test+supercomputers until 1996

Supercomputer = a computer with several processor units. It can perform computations parallel. At that time Cray and NEC computers.

p	year	disccoverer
44497	1979	H.L. Nelson and D. Slowinski
86243	1982	D. Slowinski
110503	1988	W Colquitt and L. Welsh
132049	1983	D. Slowinski
216091	1985	D. Slowinski
756839	1992	D. Slowinski and P. Gage
859433	1994	D. Slowinski and P. Gage
1257787	1996	D. Slowinski and P. Gage

David Slowinski was a software engineer for Cray Research Inc., and used his codes for Mersenne primes to test the Cray supercomputers. Few input and very complex and long computation detect bugs in the hardware quickly.

In 1985 $2^{216091}-1$ was printed in the Sunday edition of the Haarlem Dagbladet. There was an error in the printing. Next Sunday in Haarlem Daagbladet appeared the correct value.

D. Slowinski used at the beginning the following, Karatsuba type identity to speed up the squaring: if $U = U_2 2^n + U_1$, then

$$U^2 = (2^{2n} + 2^n)U_2^2 - 2^n(U_1 - U_2)^2 + (2^n + 1)U_1^2.$$

It has the advantage against the usual method:

$$U^2 = 2^{2n}U_2^2 + 2^{n+1}U_1U_2 + U_1^2$$

that one can iterate the squaring.

W Colquitt and L. Welsh: *A Fast Fourier Transform (FFT) was used to accelerate the squaring operation, which is the most time-consuming step of the Lucas-Lehmer test... The use of the FFT speeds up the asymptotic time for the Lucas-Lehmer test for M(p) from* $0(p^3)$ *to* $0(p^2 \log p \log \log p)$ *bit operations. An FFT containing* 8192 *complex elements, which was the minimum size required to test* $M(110503)$, *ran approximately 11 minutes on the SX-2.*

A. Schönhage and V. Strassen, 1971: fast multiplication based on FFT. Let $u_j = (u_{j,0}, \ldots, u_{j,k-1}) \in \mathbb{C}^k, j = 1, 2$ and $\omega = \exp(2\pi i/k)$. Then $F(u_j) = (\widehat{u}_{j,0}, \ldots, \widehat{u}_{j,k-1})$-t, where

$$\widehat{u}_{j,h} = \sum_{t=0}^{k-1} u_{j,t}\omega^{ht}$$

is called the finite Fourier transform of $u_j, j = 1, 2$. In this case we have

$$F^{-1}(F(u_1)F(u_2))_\ell = F^{-1}(\widehat{u}_{1,0}\widehat{u}_{2,0}, \ldots, \widehat{u}_{1,k-1}\widehat{u}_{2,k-1})_\ell = \sum_{h=0}^{\ell} u_{1,h}u_{2,\ell-h}.$$

If $k = 2^K$, then $F(u_1), F(u_2), F^{-1}(\widehat{u}_1\widehat{u}_2)$ can easily be computed.

Let $g = 2^n, u_1, u_2 \in \mathbb{N}$ and $u_j = \sum_{\ell=0}^{k-1} u_{j,\ell} g^\ell, j = 1, 2$.

Goal: the computation of the g-ary digits of $U_1 U_2$:

1) Computation of $F(u_1), F(u_2)$,

2) Computation of $F(u_1)F(u_2)$,

3) Computation of $F^{-1}(F(u_1)F(u_2))$

4) Reconstruction of the g-ary digits of $u_1 u_2$.

The complexity of the algorithm is $O(k \log k \log \log k)$.

6. Lucas-Lehmer test+distributed computing since 1996

GIMPS, the Great Internet Mersenne Prime Search, was formed in January 1996 to discover new world-record-size Mersenne primes. GIMPS harnesses the power of thousands of small computers like yours to search for these "needles in a haystack".

GIMPS was founded in 1996 by George Woltman. The software ran on Intel i386 systems using hand-tuned assembly code for the critical calculations, resulting in highly optimized Lucas-Lehmer code.

In the past twenty years only this group found new Mersenne primes, altogether 15.

p	year	p	year
1996	1,398,269	1997	2,976,221
1998	3,021,377	1999	6,972,593
2001	13,466,917	2003	20,996,011
2004	24,036,583	2005	25,964,951
2005	30,402,457	2006	32,582,657
2008	37,156,667	2009	*42,643,801
2008	*43,112,609	2013	*57,885,161
2016	*74,207,281		

At the beginning GIMPS used for squaring the FFT.

R. Crandall and B. Fagin, 1994, Discrete Wighted Transformation (DWT).

Let $a = (a_0, \dots, a_{k-1}), u = (u_0, \dots, u_{k-1})$, then $DWT(k, a)u = (\hat{u}_0, \dots, \hat{u}_{k-1}) = F(a \cdot x)$, i.e.

$$\hat{u}_\ell = \sum_{h=0}^{k-1} a_h u_h \omega^{\ell h}.$$

If $q = 2^p - 1$ is a Mersenne prime and $k \geq p$, then

$$u_j = \sum_{h=0}^{k-1} u_{j,h} 2^{\lceil ph/k \rceil},\ 0 \leq u_{j,h-1} < 2^{\lceil ph/k \rceil - \lceil p(h-1)/k \rceil}, j = 1, 2,$$

the coordinates of the weigh vector: $a_h = 2^{\lceil ph/k \rceil - ph/k}, h = 0, \dots, k-1$.

The Crandall-Fagin algoritm:

1) $\widehat{u_j} := DWT(k, a)u_j, j = 1, 2,$

2) $\widehat{z} := \widehat{u_1}\widehat{u_2},$

3) $z := DWT^{-1}(k, a)\widehat{z},$

4) $z := Round(z),$

5) Reconstruction of the digits of u_1u_2.

The DWT based multiplication is faster than the FFT based.

The inventions of GIMPS

1) Careful analysis of the procedure to find new Mersenne primes:

- Prove as early and cheep as possible that $M(p)$ is composite.
- Collect and store intermediate data, like divisors, the numbers u_n mod $M(p)$.
- Starts the Lucas-Lehmer test only if all elementary considerations failed.
- Check, double- and triple check the results. Compare the data of the later trial with that of the earlier running.

Most of these principle were used by R.M. Robinson, later became more sophisticated.

2) Implementation of the distributed procedure

- Implementation of algorithms for different kind of computation and for many possible platforms.
- Distribution of the software, which is capable to send the results automatically to the right place.
- Automatic collection, ordering and storing the received data.
- Automatic signal if the system finds an interesting result. (What is interesting?)
- Motive the community. Less powerful computers do important work concerning the early abort.

Today’s (May 25, 2017) Numbers

Teams	1,105
Users	173,519
CPUs	1,491,605
TFLOP/s	316.258
GHz-Days	158,129

7. Uniformly distributed sequences

R.P. Brent and P. Zimmermann, 2011, gave an efficient algorithm for the determination of $\mathbb{F}_2[x]$-irreducible trinoms of degree p provided $M(p)$ is a Mersenne prime.
In 2016 they proved that there exist exactly three such polynoms for $p = 74,207,281$:

$x^{74207281} + x^{9156813} + 1$,

$x^{74207281} + x^{9999621} + 1$, and

$x^{74207281} + x^{30684570} + 1$.

Tamás Herendi, 2016, determined uniformly distributed pseudorandom number generators, which produce 64 bit numbers and have period length $2^{74207345}$.

The characteristic polynomials of the generators are:

$$x^{74207283} - x^{74207281} - x^{9156815} - x^{9156813} - x^2 - 2 \cdot x + 1$$
$$x^{74207283} - x^{74207281} - x^{9999623} - x^{9999621} - x^2 - 2 \cdot x + 1$$
$$x^{74207283} - x^{74207281} - x^{30684572} - x^{30684570} - x^2 - 1$$

One has big freedom in the choice of the initial values.

7. Future

The search for even larger Mersenne primes ”must go on.” Are such innovations in mathematics, in computer science or in computer technology, which can considerably push forward the size of the known Mersenne primes?

The possibilities of the mentioned form of the Lucas-Lehmer test seems to be exhausted. There are variants, for the test of numbers of form $h \cdot 2^n \pm 1, h \cdot 3^n \pm 1$ and for $h \cdot 5^n \pm 1$ as well, where h is small. The 7th largest known prime is $10223 \cdot 2^{31172165} + 1$. I expect that in the future the primality tests of the above and similar numbers will evolve. It is possible that the largest known prime will be not a Mersenne prime. In 1951 his happened already, Aimé Ferrier: $\frac{2^{148}+1}{17}$.

The better understanding the location of Mersenne primes would help considerably. After Google DeepMind's AlphaGo program beat the Chinese Go word champion, The New York Times wrote on May 23, 2017: "In the future, computer scientists hope to use similar techniques to do many things, including improving fundamental scientific research and diagnosing illnesses." Can AI better predict the position of the next Mersenne prime as HI?