The New Largest Known Prime is $2^p - 1$ With $p = 43112609$. Who Cares?

Sam Wagstaff
Computer Sciences and Mathematics

November 18, 2008

On August 23, 2008, a computer at UCLA found the first known prime number with more than 10,000,000 digits and won a $100,000 prize from the Electronic Frontier Foundation.

We explain how it was found and give some applications of this achievement.

We will give the answer to this problem later in this talk.

Let $b(n)$ equal the number of 1 bits in the binary representation of the positive integer n.

In other words, $b(n)$ is the sum of the bits in the binary number n.

Problem: Prove that for every integer $k > 1$ there exist positive integers m and n such that

- $b(m) = 3$,
- $b(n) = k$, and
- $b(m \cdot n) = 2$.

Hint: Since $b(2^r n) = b(n)$ for positive integers n and r, we may assume WNLG that m and n are odd integers in the problem.

Many people have searched for Mersenne primes $2^p - 1$ during the past 400 years. The current search is organized by GIMPS, the Great Internet Mersenne Prime Search, led by George Woltman, Scott Kurowski and others.

One can download programs from GIMPS at
`http://www.mersenne.org`
to search for new large primes.

The program uses the Lucas-Lehmer test to determine whether the number is prime. Most of the time is spend multiplying very large integers.

Each candidate number is tested two or three times. An online database lists the numbers to be tested. The GIMPS program contacts the database and is assigned a candidate to test. When finished, it gets another one and so on.

Theorem: If $2^n - 1$ is prime, then n is prime.

Numbers $M_p = 2^p - 1$ with p prime are called *Mersenne numbers*.

Primes $M_p = 2^p - 1$ are called *Mersenne primes*. Forty-six of them are known. Probably there are infinitely many.

Theorem: If $2^n + 1$ is prime, then $n = 2^k$ for some $k \geq 0$.

Primes $F_k = 2^{2^k} + 1$ are called *Fermat primes*. Only five of them are known: $0 \leq k \leq 4$. Probably there are no more.

As a teenager, Gauss proved that a regular n-gon can be constructed with ruler and compass if and only if the largest odd divisor of n is a product of distinct Fermat primes.

Let $M_p = 2^p - 1$.

Theorem (Lucas-Lehmer Test): Define a sequence $S_1 = 4$ and $S_{n+1} = S_n^2 - 2$ for $n \geq 1$. Let p be a prime number. Then M_p is prime if and only if M_p divides S_{p-1}.

When using the Lucas-Lehmer Test to decide whether M_p is prime, one can reduce each S_n modulo M_p to keep the numbers smaller than M_p.

```
LucasLehmerTest(p)

S = 4

for (n=2; n<p; n++)
      S = (S*S - 2) modulo M_p

if (S = 0) then M_p is prime
else M_p is composite
```

Lucas-Lehmer test for $p = 7$, $M_7 = 127$:

n	S_n	S_n mod M_p
1	4	4
2	14	14
3	194	67
4	37634	42
5	1416317954	111
6	2005956546822746114	0

This shows that $M_7 = 127$ is prime.

Lucas-Lehmer test for $p = 11$, $M_{11} = 2047$:

n	S_n mod M_p
1	4
2	14
3	194
4	788
5	701
6	119
7	1877
8	240
9	282
10	1736

This shows that $M_{11} = 2047$ is not prime.
In fact, $2047 = 23 \cdot 89$.

At first, it looks like the algorithm runs in time $O(p)$. But the numbers inside the loop have length p bits, so it might take time $O(p^2)$ to multiply and divide them. So maybe the algorithm takes $O(p^3)$ steps.

A polynomial time primality test was discovered in 2003 by Agrawal, Kayal and Saxena. The latest improvements of it test n for primeness in roughly $O((\log n)^7)$ or $O((\log n)^8)$ steps. With $n = 2^p - 1$ this would be roughly time complexity $O(p^7)$.

There are probabilistic tests for the primality of n that run in $O((\log n)^3)$ steps and almost always give the correct answer.

The numbers $n = M_p$ are rare among numbers n in that there is a deterministic test for their primality with time complexity $O((\log n)^3)$ or even $O((\log n)^2 \log\log n)$. Few other numbers can be tested for primality so swiftly.

First, division by M_p to get a remainder is fast because of the special form $M_p = 2^p - 1$. Just shift the bits of higher order than the first p bits under the first p bits and add, perhaps twice. This works because $2^p \equiv 1 \bmod M_p$.

Example with $p = 3$: 44 mod 7 = 2.

In binary, 101100 mod 111 = 010.

```
101 100
 +  101
  -----
  1 001
 +    1
  -----
    010
```

The reduction modulo M_p (of a number $< M_p^2$) can be done in $O(p)$ steps this way.

How about the multiplication—actually a squaring in our algorithm, which is a bit easier? We will consider general multiplication here.

The multiplication algorithm you learned in grade school multiplies two D-digit numbers in $O(D^2)$ steps.

Example: multiply $792 \times 648 = 513216$.

```
       7 9 2        Multiplicand
       6 4 8        Multiplier
------------
    6  3 3 6  = 8 * 792
  3 1  6 8    = 4 * 792
4 7 5  2      = 6 * 792
------------
5 1 3  2 1 6  Final product
```

There are several fast multiplication methods. We explain one of them for the same multiplication problem (multiply $792\times 648 = 513216$).

First, multiply pairs of one-digit numbers and write their products in the proper column.
Second, add the columns.
Finally, “release the carry.”

```
           7  9  2
           6  4  8
------------------
          56 72 16  Products of
      28  36 08     two one-digit
   42 54  12        numbers
-- -- -- --- -- --
 0 42 82 104 80 16  Sum the columns
 5  9 11   8  1     Release
-- -- -- --- -- --  the
 5 51 93 112 81  6  carry

 5  1  3   2  1  6  Final product
```

The column sums are a convolution of the digits of the numbers being multiplied.

Prepend 0s to the numbers being multiplied to make them as long as their product. For example, 792 and 648 become

```
0 0 0 7 9 2
0 0 0 6 4 8
```

After these 0s are prepended, let the numbers x, y to multiply be written in base B as

$$x = (x_{D-1}x_{D-2}\ldots x_2x_1x_0)_B = \sum_{i=0}^{D-1} x_iB^i$$

and likewise for y. The n-th column sum is

$$z_n = \sum_{i+j\equiv n \bmod D} x_iy_j.$$

The sequence $z = z_{D-1}, z_{D-2}, \ldots, z_1, z_0$ of these column sums is called the *cyclic convolution* $x \times y$ of x and y. (Electrical engineers call sequences like x, y and z “signals.”)

It takes $O(D^2)$ operations to compute a convolution by the definition.

However, it takes only $O(D)$ operations to compute the pointwise multiplication $x * y = \{x_i y_i\}$ of two sequences $\{x_i\}$ and $\{y_i\}$.

There is an invertible operation on sequences of length D, called a *discrete Fourier transform*, with the property that

$$x \times y = DFT^{-1}(DFT(x) * DFT(y)).$$

This means that we can compute all D column sums by doing three DFTs and D multiplications.

If you compute $DFT(x)$ from its definition, it takes $O(D^2)$ steps. But there is an algorithm, called a *fast Fourier transform*, that computes $DFT(x)$ in $O(D \log D)$ steps.

Discrete Fourier transform (DFT)

Let x be a sequence of length D of elements of a number system in which D^{-1} exists and there is a primitive D-th root of unity g. This means that $g^k = 1$ if and only if k is a multiple of D.

For example, the number system might be the complex numbers with $g = e^{2\pi i/D}$.

Or it might be the integers modulo B, the number base for the digits of the numbers to multiply. In this case, g would be a primitive D-th root of 1.

Then the discrete Fourier transform $X = DFT(x)$ and its inverse $x = DFT^{-1}(X)$ are defined by

$$X_k = \sum_{j=0}^{D-1} x_j g^{-jk}, \qquad x_j = \frac{1}{D} \sum_{k=0}^{D-1} X_k g^{jk}.$$

Fast Fourier transform (FFT)

Assume the sequence x has length $D = 2^d$.

The following algorithm computes $FFT(x) = DFT(x)$ via the Danielson-Lanczos identity.

```
FFT(x)
  n = length(x) // length of current sequence
  if (n = 1) return x
  m = n/2
  X = x[2j], j = 0..m-1   // even subscripts
  Y = x[2j+1], j = 0..m-1 // odd subscripts
  X = FFT(X)
  Y = FFT(Y)
  U = X[k mod m], k = 0..n-1
  V = g^{-k}Y[k mod m], k=0..n-1 // order g = n
  return U + V  // butterfly operation
```

The time complexity is $O(D \log D)$.

Large integer multiplication with FFTs

Input: two integers x, y, each with $\leq k$ digits in base B.

Output: the digits of xy in base B.

```
Prepend 0s to x and y up to length D = 2*k
X = FFT(x)
Y = FFT(y)
for (i=0; i<D; i++)
      Z[i] = X[i]*Y[i] // pointwise product
z = FFT_INVERSE(Z)
carry = 0
for (i=0; i<D; i++) // release the carry
      v = z[i] + carry
      z[n] = v mod B
      carry = floor(v/B)
delete any leading 0s in z
return z
```

Who Cares?

Taken from Chris Caldwell's Prime FAQ page
`http://primes.utm.edu/notes/faq/why.html`

1. Tradition: Euclid, Fermat, Mersenne, Cunningham, Lucas, Lehmer, Gillies, Tuckerman

2. For the by-products: FFT, large integer arithmetic, fast multiplication

3. Rare and beautiful objects: $<$ 50 known

4. Glory: like climbing Mount Everest

5. Test hardware: Slowinski tested Crays

6. Study the distribution of Mersenne primes

7. For money: EFF offers $150,000 for 10^8-digit prime, $200,000 for 10^9-digit prime

Euclid was interested in Mersenne primes because of perfect numbers.

A positive integer n is perfect if it equals the sum of all of its divisors $< n$.

Examples: $6 = 3 + 2 + 1$ is perfect.

$4 > 2 + 1$ is deficient, not perfect.

$12 < 6+4+3+2+1$ is abundant, not perfect.

$28 = 14 + 7 + 4 + 2 + 1$ is perfect.

Theorem (Euclid) If p and $2^p - 1$ are both prime, then $2^{p-1}(2^p - 1)$ is perfect.

Theorem (Euler) If n is perfect and even, then $n = 2^{p-1}(2^p - 1)$ for some prime number p.

Each new Mersenne prime gives a new (even) perfect number.

No odd perfect number is known.

Table of known Mersenne primes

p	digits	year	discoverer
2	1	–	–
3	1	–	–
5	2	–	–
7	3	–	–
13	4	1456	anonymous
17	6	1588	Cataldi
19	6	1588	Cataldi
31	10	1772	Euler
61	19	1883	Pervushin
89	27	1911	Powers
107	33	1914	Powers
127	39	1876	Edward Lucas
521	157	1952	Raphael Robinson
607	183	1952	Robinson
1279	386	1952	Robinson
2203	664	1952	Robinson
2281	687	1952	Robinson SWAC
3217	969	1957	Hans Riesel BESK
4253	1281	1961	Alex Hurwitz
4423	1332	1961	Hurwitz 7090

Table of known Mersenne primes

p	digits	year	discoverer
9689	2917	1963	Don Gillies
9941	2993	1963	Gillies ILLIAC II
11213	3376	1963	Gillies
19937	6002	1971	Bryant Tuckerman
21701	6533	1978	L C Noll & L Nickel
23209	6987	1979	Noll
44497	13395	1979	Nelson & Slowinski
86243	25962	1982	David Slowinski
110503	33265	1988	Colquitt & Welsh
132049	39751	1983	Slowinski
216091	65050	1985	Slowinski
756839	227832	1992	Slowinski & Gage
859433	258716	1994	Slowinski & Gage
1257787	378632	1996	Slowinski & Gage

Table of known Mersenne primes

p	digits	year	discoverer
1398269	420921	1996	GIMPS (Armengaud)
2976221	895932	1997	GIMPS (Spence)
3021377	909526	1998	GIMPS (Clarkson)
6972593	2098960	1999	GIMPS (Hajratwala)
13466917	4053946	2001	GIMPS (Cameron)
20996011	6320430	2003	GIMPS (Shafer)
24036583	7235733	2004	GIMPS (Findley)
25964951	7816230	2005	GIMPS (Nowak)
30402457	9152052	2005	GIMPS (Cooper)
32582657	9808358	2006	GIMPS (Cooper)
37156667	11185272	2008	GIMPS (Elvenich)
43112609	12978189	2008	GIMPS (Smith)

Note: $p = 13466917$ is known to be the 39-th Mersenne prime in order of size. Seven larger Mersenne primes are known, but not all p between 13466917 and 43112609 have been checked yet. Thus, $p = 43112609$ is the 45-th Mersenne prime discovered, but not necessarily the 46-th Mersenne prime in order of size.

Conjectures about Mersenne primes

Conjecture: There are infinitely many Mersenne primes.

Conjecture: There are about $(e^{\gamma}/\ln 2)\ln\ln x$ Mersenne primes $\leq x$. Note $(e^{\gamma}/\ln 2) \approx 2.5695$.

Conjecture (Mersenne, 1644): $2^p - 1$ is prime for $p =$

2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257

and for no other $p \leq 257$.

Mersenne was wrong. The list should be

2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127

This led to the "New Mersenne Conjecture:"

Conjecture (Bateman, Selfridge, Wagstaff, 1989): If any two of the following statements about an odd positive integer p are true, then the third one is also true.

- $p = 2^k \pm 1$ or $p = 4^k \pm 3$.

- $2^p - 1$ is prime.

- $(2^p + 1)/3$ is prime.

The numbers $W_p = (2^p + 1)/3$ are now called Wagstaff numbers. They seem to be prime about as often as Mersenne numbers are prime. W_p is prime for $p =$ 3, 5, 7, 11, 13, 17, 19, 23, 31, 43, 61, 79, 101, 127, 167, 191, 199, 313, 347, 701, 1709, 2617, 3539, 5807, 10501, 10691, 11279, 12391, 14479, 42737, ...

In a 2008 posting to MersenneForum, Anton Vrba claimed a fast test for primality of Wagstaff numbers. If correct, his test would be as fast as the Lucas-Lehmer test for the primality of Mersenne numbers.

Theorem? (Vrba): Define a sequence $S_0 = 6$ and $S_{n+1} = S_n^2 - 2$ for $n \geq 1$. Let p be an odd prime number. Then W_p is prime if and only if W_p divides $S_p - S_2 = S_p - 1154$.

People objected to his "proof" of this statement and he withdrew it. However, the test works correctly for all $p < 42738$ and the statement may be true. A program for this test would compute the S_n modulo $2^p + 1$ until the end of the loop, at which time the remainder would be reduced modulo $W_p = (2^p + 1)/3$. All the tricks used in programming the Lucas-Lehmer test would work here, too.

More Applications

Many 2^p-1 have small prime factors. A search for new Mersenne primes begins by eliminating those 2^p-1 having a small prime factor. The factors themselves have uses.

There are many tables of factorizations of 2^n-1, as well as factors of $b^n \pm 1$ for small b. See the Cunningham table by Brillhart, Lehmer, Selfridge, Tuckerman and Wagstaff at `http://www.ams.org/online_bks/conm22`

The factors of $b^n \pm 1$ have many uses. For example, they are used in proofs of unlikely properties that an odd perfect number must have, such as $> 10^{300}$ and has ≥ 8 distinct prime factors. If $p > 5$, then the length of the period of the repeating decimal fraction for $1/p$ is the smallest n for which p divides 10^n-1. The factors of p^n-1 are important in constructing elliptic curves with small embedding degree to do Weil pairings efficiently for cryptography.

A *linear feedback shift register* is a device that generates a pseudorandom bit stream sometimes used in cryptography. It consists of an n-bit shift register and an exclusive-or gate whose inputs come from two or three selected bit positions in the register (called taps). At each clock cycle, the bits in the register move one position to the right. The right-most bit is the next output bit to the stream. The output of the exclusive-or gate is shifted into the left-most bit position of the register.

The bit stream is periodic. The period can be as long as $2^n - 1$, depending on the taps sent to the exclusive-or gate. To choose these tap positions to maximize the period, one must know the complete prime factorization of $2^n - 1$. This construction is especially easy when $2^n - 1$ is a Mersenne prime. The new Mersenne prime, $M_{43112609}$, would give a linear feedback shift register of length 43112609 bits and period length $M_{43112609}$.

Still More Applications

Sometimes one can discover new identities by examining tables of factored numbers.

For example, here is an excerpt from a table of factors of numbers $2^n + 1$:

n	$2^n + 1$	factored	again	$2^{n/2} + 1$
2	5	5	$1 \cdot 5$	3
6	65	$5 \cdot 13$	$5 \cdot 13$	9
10	1025	$5^2 \cdot 41$	$25 \cdot 41$	33
14	16385	$5 \cdot 29 \cdot 113$	$113 \cdot 145$	129

It is easy to observe that the average of the two factors shown in the penultimate column equals the number in the last column. This leads to the identity

$$2^{4k-2} + 1 = (2^{2k-1} - 2^k + 1)(2^{2k-1} + 2^k + 1),$$

which is easy to prove once it is noticed.

There is a similar identity for each b that is not a power. It algebraically factors either $b^n - 1$ or $b^n + 1$, depending on b, for all n in a certain arithmetic progression. These identities are named after Aurifeuille, who discovered some of them.

In terms of the binary representation of integers, the formula

$$2^{4k-2} + 1 = (2^{2k-1} - 2^k + 1)(2^{2k-1} + 2^k + 1)$$

shows that there exist an integer with any number of 1 bits which can be multiplied times a number with exactly three 1 bits to give a product with exactly two 1 bits.

This identity solves the

Problem: Prove that for every integer $k > 1$ there exist positive integers m and n such that

- $b(m) = 3$,
- $b(n) = k$, and
- $b(m \cdot n) = 2$.

Solution: Let $m = 2^{2k-1} + 2^k + 1$ and
$n = 2^{2k-1} - 2^k + 1$.

$$2^{4k-2} + 1 = (2^{2k-1} + 2^k + 1)(2^{2k-1} - 2^k + 1)$$