MATHEMATICS OF COMPUTATION
Volume 66, Number 217, January 1997, Pages 451–455
S 0025-5718(97)00793-X

A LARGE PAIR OF TWIN PRIMES

TONY FORBES

Abstract. We describe an efficient integer squaring algorithm (involving the fast Fourier transform modulo F_8) that was used on a 486 computer to discover a large pair of twin primes.

In this article we discuss some of the methods that resulted in the discovery of the pair of twin primes, $6797727 \times 2^{15328} \pm 1$, found on 25th July 1995 and first reported in [1]. The computer equipment used was surprisingly modest by today's standards for this type of work; an ordinary IBM-compatible PC with an Intel 486 DX microprocessor running at 33 MHz, later upgraded to a 486 DX4 running at 100 MHz. At the time of writing (October, 1995) there are only two larger known prime pairs, $697053813 \times 2^{16352} \pm 1$, discovered by K.-H. Indlekofer and A. Ja'rai in 1994, and a very recent new record, $570918348 \times 10^{5120} \pm 1$, announced by Harvey Dubner.

By restricting the search for twin primes to integers of the form $m2^n \pm 1$, with m not too large, we take advantage of the well-known methods of J. Brillhart, D. H. Lehmer and J. L. Selfridge [2] for verifying the primality of a large number N where it is possible to factorize the major part of either $N-1$ or $N+1$. If, further, m does not exceed 2^{32}, then we can reduce a number x modulo $m2^n + \varepsilon$ very rapidly. Let $x = x_0 + 2^n x_1$, $0 \le x_0 < 2^n$, $x_1 = u_0 + mu_1$, $0 \le u_0 < m$, where u_0 and u_1 are obtained by dividing x_1 by m, a straightforward operation involving repeated use of the processor's 32-bit integer division instruction. Then we have $x \equiv x_0 - \varepsilon u_1 + u_0 2^n \pmod{m2^n + \varepsilon}$.

The Fermat test. A positive integer N, chosen more or less at random, is likely to be prime if it satisfies

$$2^{N-1} \equiv 1 \pmod{N}. \tag{1}$$

After substituting $m2^n + \varepsilon$ for N and rearranging, (1) can be written as

$$(2^{2^n})^m \equiv 2^{1-\varepsilon} \pmod{N}.$$

The computation of 2^{2^n} is performed by repeated squaring and reduction modulo N. In preference to the school method of computing x^2, we used a procedure—which we describe in some detail—similar to the Schönhage-Strassen algorithm for the fast multiplication of large integers (Aho, Hopcroft and Ullman [3, p. 270]).

Received by the editor October 9, 1995 and, in revised form, December 6, 1995 and January 26, 1996.

1991 *Mathematics Subject Classification.* Primary 11A41; Secondary 11A51.

The finite Fourier transform. We wish to compute x^2, $0 \le x < 2^{15360}$. Let F be the eighth Fermat number $2^{256}+1$ and let $M = 2^{120}$. Given a 256-dimensional vector $Z = (Z_0, Z_1, \ldots, Z_{255})$, we define the Fourier transform $\mathbf{F}[Z]$ modulo F as the vector with components $\mathbf{F}[Z]_k$, $k = 0, 1, \ldots, 255$, satisfying $0 \le \mathbf{F}[Z]_k < F$ and

$$\mathbf{F}[Z]_k \equiv \sum_{i=0}^{255} Z_i 4^{ki} \pmod F. \tag{2}$$

Let $Z \otimes Z$ be the vector with components satisfying $0 \le [Z \otimes Z]_i < F$, $i = 0, 1, \ldots, 255$,

$$[Z \otimes Z]_0 \equiv Z_0^2 \pmod F$$

and

$$[Z \otimes Z]_i \equiv Z_{256-i}^2 \pmod F, \qquad i = 1, 2, \ldots, 255.$$

The number x is represented in the base M by a vector $X = (X_0, X_1, \ldots, X_{255})$,

$$x = \sum_{i=0}^{255} X_i M^i,$$

where $0 \le X_i < M$ for $i = 0, 1, \ldots, 255$. The restriction $x < 2^{15360}$ implies $X_{128} = X_{129} = \cdots = X_{255} = 0$. Finally, let $Y = (Y_0, Y_1, \ldots, Y_{255}) = \mathbf{F}[\mathbf{F}[X] \otimes \mathbf{F}[X]]$, which represents the number

$$y = \sum_{j=0}^{255} \mathbf{F}[\mathbf{F}[X] \otimes \mathbf{F}[X]]_j M^j = \sum_{j=0}^{255} Y_j M^j \tag{3}$$

in base M, although the "digits" Y_j are not necessarily less than M.

Theorem 1. *Let x and y be defined as above. Then $y = 256x^2$.*

Proof. This is a straightforward application of the convolution theorem. (Observe that 4 is a 256th root of unity (mod F) and further, that $(4^k - 1, F) = 1$ for $k = 1, 2, \ldots, 255$.) The rearrangement of $\mathbf{F}[X]$ implicit in $\mathbf{F}[X] \otimes \mathbf{F}[X]$ and second Fourier transform are equivalent to an inverse transformation (up to a scalar multiple) which may be defined as in (2) but with 4^{-ki} instead of 4^{ki}. The parameter $M = 2^{120}$ is small enough to ensure that $256 \sum_{j=0}^{i} X_j X_{i-j} < F$, $i = 0, 1, \ldots, 255$. □

The fast Fourier transform. Let

$$Z_{256,0,i} = Z_i, \qquad i = 0, 1, \ldots, 255.$$

For $d = 128, 64, 32, 16, 8, 4, 2, 1$, let $Z_{d,k,i}$ and $Z_{d,k+128/d,i}$ be the integers in the range 0 to $F-1$ satisfying

$$Z_{d,k,i} \equiv Z_{2d,k,i} + 4^{dk} Z_{2d,k,i+d} \pmod F \tag{4}$$

and

$$Z_{d,k+128/d,i} \equiv Z_{2d,k,i} - 4^{dk} Z_{2d,k,i+d} \pmod F, \tag{5}$$

where $i = 0, 1, \ldots, d-1$ and $k = 0, 1, \ldots, \frac{128}{d} - 1$.

Theorem 2. *We have $\mathbf{F}[Z]_k = Z_{1,k,0}$, $k = 0, 1, \ldots, 255$.*

Proof. The theorem becomes evident when the iterations are performed by hand. □

TABLE 1. Fermat test of $999999 \times 2^n + 1$

n	School method seconds	FFT method seconds	dimension D	root G
5984	51	66	256	4
7008	80	77	256	4
8000	120	88	256	4
8992	170	99	256	4
9984	230	110	256	4
12000	400	130	256	4
13984	630	160	256	4
15328	830	170	256	4
15360	830	350	512	2
16000	940	370	512	2
20000	1800	460	512	2
24000	3200	550	512	2
28000	5000	650	512	2
30688	6600	710	512	2
30720	6600	1900	1024	$2^{192} - 2^{64}$
36000	11000	2300	1024	$2^{192} - 2^{64}$
44000	19000	2800	1024	$2^{192} - 2^{64}$
52000	32000	3300	1024	$2^{192} - 2^{64}$
60000	49000	3800	1024	$2^{192} - 2^{64}$
61408	53000	3900	1024	$2^{192} - 2^{64}$

Programming. The computer programs were written in a combination of Yuji Kida's UBASIC [4] and 486 assembler language; UBASIC and assembler for the preliminary sieving, UBASIC to generate tables for the fast Fourier transform, and assembler for the main body of the Fermat test. We prefer this particular version of the fast Fourier transform because it is easy to implement in 486 assembler language. The technique described in [3] of bit-reversing the k index permits the use of a single array to hold the vector Z; at each stage of the transform the sum $Z_{d,k,i}$ and difference $Z_{d,k+128/d,i}$ in (4) and (5) occupy the same array elements as the operands $Z_{2d,k,i}$ and $Z_{2d,k,i+d}$.

At the upper limit of Theorem 1, we find that we can test a 4624-digit number in about 8.5 minutes on the 486 at 33 MHz in 16-bit mode. At 100 MHz the time reduces to 170 seconds.

Comparison of algorithms. In Table 1 we give running times for the Fermat test of $m2^n + 1$, comparing the two algorithms for computing x^2 on a typical 486 DX4 computer:

(i) *The school method.* Here we simply chop x into digits x_i of size 2^{32} and form the convolution

$$x^2 = \sum_i x_i^2 2^{64i} + 2 \sum_i 2^{32i} \sum_{j<i/2} x_j x_{i-j}.$$

(ii) *The FFT method.* This is based on the fast Fourier transform modulo $F = 2^{256} + 1$ and base $M = 2^{120}$. The dimension of the Fourier transform is D, a power

of two, and G is a primitive Dth root of unity modulo F. The method is applied to numbers less than $M^{D/2} = 2^{60D}$. If we allow a 32-bit word for the multiplier m, the maximum value of the exponent n is $60D-32$. We present results for $D = 256$, 512 and 1024. There is a small complication when $D = 512$ and 1024. Define $X_i = 0$ for $i \ge D$ and let

$$W_j = D\sum_{r=0}^{j} X_r X_{j-r}, \qquad j = 0, 1, \ldots, 2D-1. \tag{6}$$

As in Theorem 1, $Y_j \equiv W_j \pmod F$ (since $X_i = 0$ for $i \ge D/2$), but now it is possible for W_j to exceed $F-1$. However, we also have $W_j \equiv 0 \pmod D$ and $W_j < DF$ from which the true value of W_j may be determined.

The time taken by the FFT method is essentially a linear step function of n with discontinuities at 15360 and 30720, where there is a somewhat more than doubling as the dimension changes. We chose the exponent $n = 15328$ simply because it is the largest multiple of 32 below the first jump. But it was a fortunate choice—the first m making both $m2^{15328}-1$ and $m2^{15328}+1$ prime turns out to be exceptionally small.

One way of smoothing out the steps is to allow x to slightly exceed 2^{60D}. Then the first few components of the vector Y in Theorem 1 will include contributions from the upper half of X because in (6) we no longer have $W_j = 0$ for $j \ge D$. Instead of (3), we have $Y_j \equiv W_j + W_{D+j} \pmod F$ $(0 \le j < D)$ and $y = \sum_{j=0}^{2D-2} W_j M^j$. We calculate any nonzero components W_{D+j} separately and also use them to recover the corresponding W_j from Y_j. Eventually the work involved in computing the W_j for $j \ge D$ will reach a point where it becomes preferable to increase D to the next power of two.

The sieve. Let n be some fixed, not too small integer, and let $L = \frac{(n\log 2)^2}{4C_2}$, where $C_2 \approx 0.660$ is the twin prime constant $\prod_{p>2} \frac{p(p-2)}{(p-1)^2}$. Let $\pi_2(x,n)$ denote the number of $m \le x$ such that both $m2^n+1$ and $m2^n-1$ are prime. Assuming the Hardy-Littlewood conjecture [5] concerning the distribution of twin primes, and assuming that the twin primes are equally distributed among the odd residue classes modulo 2^n, we have $\pi_2(x,n) \approx 4C_2 x(\log x2^n)^{-2}$. Thus $\pi_2(L,n) \approx 1$, and we expect a reasonable chance of finding twin primes $m2^n \pm 1$ with $m \le L$.

We sieve the interval $1 \le m \le L$ by primes up to q; that is, we remove all $m \equiv \pm 2^{-n} \pmod p$ for $2 < p \le q$. Denote the time for this process by $S(n,q)$. To this we add the time required to perform the Fermat test on the remaining numbers. If $T(n)$ is the average time for a single Fermat test of $m2^n+1$, $m < 2^{32}$, then the total time is

$$S(n,q) + LT(n)\prod_{2<p\le q}\frac{p-2}{p} \approx S(n,q) + \frac{4C_2 e^{-2\gamma}L}{(\log q)^2}T(n), \tag{7}$$

by Mertens' theorem.

Our sieving program has three stages. The first uses a difference table of primes $p < 2^{20}$ and residues 2^{-n} modulo p to produce m's satisfying $(m^2 2^{2n}-1, 2^{20}!) = 1$. The second stage takes batches of 15000 from the first stage (corresponding to about 3500000 original m's) and sieves them up to 2^{32}. We use numbers coprime to 30030 rather than primes. The third state is just like the second and covers the range 2^{32} to q.

Assume we have full use of a 100 MHz 486 and let $n = 15328$, so that $T(15328) = 170$ seconds and $L \approx 42700000$. The time for the fist two stages of the sieve is about 5 hours per batch. The third stage requires multiprecision arithmetic for the computation of $2^{-n} \pmod p$, and it takes about 15 seconds on average to sift from p to $p + 10^6$. Thus,

$$S(n,q) \approx \frac{e^{-2\gamma}n^2}{15000}\left(\frac{18000}{20^2} + \frac{0.000015}{32^2}(q - 2^{32})\right) \text{ seconds.}$$

The minimum of (7) occurs when

$$q(\log q)^3 \approx 2048(\log 2)^2 \frac{15000 \times 170}{0.000015} \approx 1.7 \times 10^{14},$$

or when $q \approx 1.3 \times 10^{10}$, corresponding to a total search time of 139 days, 10 days for the sieving and 129 days for about 66000 Fermat tests.

Clearly, we can improve the sieving procedure by operating on larger batches of m's, and if a table of primes up to q were available, we would not have to waste time sifting with composite numbers. However, the effect is relatively slight. For example, with a 20-fold increase in speed, the appropriate value of q turns out to be 1.9×10^{11}, which reduces the search time to 112 days—8 days for sieving and 104 days for the Fermat tests.

Acknowledgement

The author is very grateful to the referee for helpful comments and suggestions.

Postscript. In November, 1995 K.-H. Indlekofer and A. Ja'rai, announced a new, large prime pair, $242206083 \times 2^{38880} \pm 1$.

References

1. Tony Forbes, *Prime k-tuplets−10*, M500 **146** (1995), 8–12.
2. J. Brillhart, D. H. Lehmer, and J. L. Selfridge, *New primality criteria and factorizations of* $2^m \pm 1$, Math. Comp. **29** (1975), 620–647. MR **52:**5546
3. A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, MA, 1975. MR **54:**1706
4. C. K. Caldwell, *UBASIC*, J. Recreational Math. **25** (1993), 47–54.
5. G. H. Hardy and J. E. Littlewood, *Some problems of* '*Partitio Numerorum*'; III: *On the expression of a number as a sum of primes*, Acta Math. **44** (1922), 1–70.

22 St. Albans Road, Kingston upon Thames, Surrey, KT2 5HQ, England
E-mail address: tonyforbes@ltkz.demon.co.uk