MATHEMATICS OF COMPUTATION
VOLUME 36, NUMBER 154
APRIL 1981

Factorization of the Eighth Fermat Number

By Richard P. Brent and John M. Pollard

Abstract. We describe a Monte Carlo factorization algorithm which was used to factorize the Fermat number $F_8 = 2^{256} + 1$. Previously F_8 was known to be composite, but its factors were unknown.

1. Introduction. Brent [1] recently proposed an improvement to Pollard's Monte Carlo factorization algorithm [4]. Both algorithms can usually find a prime factor p of a large integer in $O(p^{1/2})$ operations.

In this paper we describe a modification of Brent's algorithm which is useful when the factors are known to lie in a certain congruence class. To test its effectiveness, the algorithm was applied to the Fermat numbers $F_k = 2^{2^k} + 1$, $5 \leqslant k \leqslant 13$. The least factors of all but F_8 were known [2], and F_8 was known to be composite. The algorithm rediscovered the known factors and also found the previously unknown factor 1,238,926,361,552,897 of F_8.*

2. The Factorization Algorithm and a Conjecture. To factor a number N, we consider a sequence defined by a recurrence relation

$$x_i = f(x_{i-1}) \pmod N, \qquad i = 1, 2, \ldots,$$

where f is a polynomial of degree at least 2, with some suitable x_0. One variant of Brent's algorithm computes $\mathrm{GCD}(x_i - x_j, N)$ for $i = 0, 1, 3, 7, 15, \ldots$ and $j = i + 1, \ldots, 2i + 1$ until either $x_i = x_j \pmod N$ (in which case a different f or x_0 must be tried) or a nontrivial GCD (and hence a factor of N) is found. As in [1], [4] we can reduce the cost of a GCD computation essentially to that of a multiplication mod N, and this is assumed below.

If nothing is known about the factors, we normally choose a quadratic polynomial $x^2 + c$ ($c \neq 0, -2$). However, it is conjectured in [4] that the expected number of steps for Pollard's algorithm can be reduced by a factor $\sqrt{m-1}$ if the factors p are known to satisfy $p = 1 \pmod m$ and we use a polynomial of the form $x^m + c$. This conjecture is equally applicable to the algorithms of [1].

We sketch the informal argument leading to the conjecture. Suppose we are given a function $g(x)$ on a set U of p elements and define a sequence of elements by $x_i = g(x_{i-1})$, $i = 1, 2, \ldots$. Suppose that the elements of the set $S = \{x_0, \ldots, x_{n-1}\}$ are distinct. For a random function g, the probability that the next

Received September 18, 1980.

1980 *Mathematics Subject Classification.* Primary 10–04, 10A25, 10A40; Secondary 10A05, 10A10, 10A35, 65C05, 68–04.

Key words and phrases. Fermat numbers, factorization, Monte Carlo methods.

* The epigram "I am now entirely persuaded to employ the method, a handy trick, on gigantic composite numbers" may appeal to readers who wish to memorize this factor.

element x_n is in S is just n/p (from which the formulae of [1], [4] are derived). We require the corresponding probability when g is chosen at random out of a subset of the functions on our set, namely those producing a graph in which, for each i, a fraction q_i of nodes have in-degree i: here the q_i are any given nonnegative numbers with $\sum_i q_i = \sum_i iq_i = 1$. (For the application to factorization, the argument could be simplified, but as presented it applies to wider classes of functions such as those of [5], at least in the first approximation.)

Let T be the set of elements $y \in U \setminus S$ with $g(y) \in S$. To estimate the expected size of T, we argue that the probability of any node appearing in S is proportional to the node's in-degree i. Thus T has the expected size

$$n\sum_i iq_i(i-1) = n\sum_i q_i(i-1)^2 = nV,$$

where V is the variance of the in-degree. If $x_n \notin S$, we shall have $x_{n+1} \in S$ if and only if $x_n \in T$, an event with probability $nV/(p-n) \simeq n/(p/V)$ (since we are concerned with the situation $n = O(p^{1/2})$, p large).

For a random mapping, the in-degree has a Poisson distribution with mean and variance 1, and the two arguments agree. For the application to factorization, we take $g(x) = f(x) \pmod p$, $f(x) = x^m + c \pmod N$. Since $p = 1 \pmod m$, the in-degree is m for a fraction $1/m$ of the nodes, and zero for the remainder (neglecting one node, c), so the variance of the in-degree is essentially $V = m - 1$. This motivates the conjecture.

Our conjecture must clearly be applied with discretion. Consider, for example, the function $g(x) = x + 1$ or $x + 2 \pmod p$ according as x is a quadratic residue or a nonresidue of p: since the cycle is of order p (in fact $2p/3 + O(p^{1/2}\log^2 p)$) it benefits us little to compute $V \simeq \frac{1}{2}$.

3. Behavior of the Polynomial $x^m + 1$. To illustrate our conjecture, we give some numerical results for the polynomial $g(x) = x^m + 1 \pmod p$, $m = 2^k$, for $1 \leq k \leq 10$. For each k, we give in Table 1 the mean values of $t(p)/\sqrt{p/(m-1)}$ and $c(p)/\sqrt{p/(m-1)}$ for the 10^4 smallest primes $p > 10^6$ satisfying $p = 1 \pmod m$; here $t(p)$ and $c(p)$ denote, respectively, the length of the tail (nonperiodic part) and of the cycle (periodic part) of the sequence (x_i), starting with $x_0 = 1$. The conjectured expectations are $(\pi/8)^{1/2} \simeq 0.627$.

TABLE 1

Behavior of polynomials $x^m + 1$ *for* 10^4 *primes with* $p = 1 \pmod m$, $m = 2^k$

k	mean $t(p)/\sqrt{p/(m-1)}$	mean $c(p)/\sqrt{p/(m-1)}$
1	0.619	0.618
2	0.627	0.619
3	0.625	0.620
4	0.625	0.626
5	0.629	0.619
6	0.628	0.617
7	0.629	0.622
8	0.630	0.618
9	0.625	0.625
10	0.619	0.625

A more obvious conjecture replaces our $\sqrt{m-1}$ by $\sqrt{m}$; this results from the idea that the recurrence relation corresponding to $g(x) = x^m + 1 \pmod p$ operates on a set of $(p-1)/m$ residues when $p = 1 \pmod m$. The difference is important when $m = 2$, as in the standard form of Brent's and Pollard's algorithms. The empirical results of Brent [1] (for $m = 2$ and all odd primes $p < 10^8$) and Table 1 discredit this conjecture.

4. Application to Factorization of Fermat Numbers. The factors p_k of a Fermat number $F_k = 2^{2^k} + 1$ $(k > 1)$ satisfy $p_k = 1 \pmod{2^{k+2}}$, so to factorize F_k we took $f(x) = x^{2^{k+2}} + 1 \pmod{F_k}$ and $x_0 = 3$ in the algorithm of Section 2 ($x_0 = 0$ or 1 is not satisfactory here). By the conjecture of Section 2, compared to Brent's algorithm [1, Section 5], the expected number of steps is reduced by a factor $(2^{k+2} - 1)^{1/2}$, but the number of multiplications (mod F_k) per step is increased from 2 to $k + 3$. Thus, from [1, Eq. (6.2)], the expected number of multiplications (mod F_k) to find the least prime factor p_k of F_k is

$$E_k = (k+3)(\pi p_k/8)^{1/2}(3/\ln 4 + 1)/(2^{k+2} - 1)^{1/2}, \tag{1}$$

and for $k = 8$ this is $0.682 p_k^{1/2}$. For the algorithm of [4] (with a quadratic polynomial), the corresponding number is $4(\pi/2)^{5/2} p_k^{1/2}/3 \simeq 4.123 p_k^{1/2}$, larger by a factor of six.

We did not employ the modification of [1, Section 7] which is not worthwhile unless m is small. Some improvements might have been achieved in other ways, but we preferred to keep the method as simple as possible.

In Table 2, p_k is the least prime factor of F_k, M_k is the number of multiplications (mod F_k) required to find it (by the algorithm just described), and E_k is given by (1). The computation for F_7 took 6 hours 50 minutes on a Univac 1100/82 computer, comparable to the time required by the continued fraction algorithm [3]; that for F_{13} took 3 hours 20 minutes on the same machine. The factorization of F_8 took 2 hours on a Univac 1100/42 computer (a slightly slower machine). The other computations took only a few seconds.

TABLE 2

Least prime factors p_k of Fermat numbers $F_k = 2^{2^k} + 1$

k	p_k	M_k	M_k/E_k
5	641	16	0.45
6	274,177	855	1.46
7	59,649,589,127,497,217	2.67×10^8	1.24
8	1,238,926,361,552,897	2.29×10^7	0.95
9	2,424,833	420	0.51
10	45,592,577	1,521	0.56
11	319,489	112	0.65
12	114,689	30	0.38
13	2,710,954,639,361	38,896	0.13

The application of more than 100 trials of Rabin's probabilistic algorithm lead us to suspect that the cofactor $q_8 = F_8/p_8 = 93{,}461{,}639{,}715{,}357{,}977{,}769{,}163{,}558{,}199{,}606{,}896{,}584{,}051{,}237{,}541{,}638{,}188{,}580{,}280{,}321$ was prime. Professor H. C.

Williams kindly proved the primality of q_8, using the methods of [7] and the partial factorizations

$$\begin{aligned}
q_8 - 1 &= 2^{11} \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot r_1,\\
q_8 + 1 &= 2 \cdot r_2,\\
q_8^2 + 1 &= 2 \cdot 17 \cdot 21649 \cdot 31081 \cdot 2347789 \cdot r_4,\\
q_8^2 + q_8 + 1 &= 3 \cdot r_3,\\
q_8^2 - q_8 + 1 &= 37 \cdot 1459 \cdot 266401 \cdot r_6,
\end{aligned}$$

where r_1, r_2, r_3, r_4, r_6 are composite but have no factors less than 5×10^7. (D. H. Lehmer found that their factors exceed 2×10^9, but this is more than is required for the proof of primality of q_8.) Thus, the factorization of F_k is now complete for $k \le 8$ (F_k is prime for $1 \le k \le 4$, composite with two prime factors for $5 \le k \le 8$).

We are currently applying a slight modification of the algorithm in an attempt to factorize $q_9 = F_9/p_9$, a number of 148 decimal digits which is known to be composite, and F_{14}. The algorithm could also be used to factorize Mersenne numbers $M_k = 2^k - 1$ (k prime), whose prime factors p satisfy $p = 1 \pmod{2k}$.

Acknowledgement. We thank H. C. Williams for proving the primality of q_8, D. H. Lehmer and Daniel Shanks for their assistance, and the Australian National University for the provision of computer time.

Note Added in Proof. A simpler proof of the primality of q_8 is possible, using the factorization $r_1 = 31618624099079 \cdot r_1'$, where r_1' is a 43-digit prime. The factorization of r_1 was obtained by the method of [1].

Department of Computer Science
Australian National University
Canberra, A. C. T. 2600, Australia

Plessey Telecommunications
Taplow Court, Maidenhead
Berkshire, England

1. R. P. Brent, "An improved Monte Carlo factorization algorithm," *BIT*, v. 20, 1980, pp. 176–184.

2. J. C. Hallyburton & J. Brillhart, "Two new factors of Fermat numbers," *Math. Comp.*, v. 29, 1975, pp. 109–112.

3. M. A. Morrison & J. Brillhart, "A method of factoring and the factorization of F_7," *Math. Comp.*, v. 29, 1975, pp. 183–208.

4. J. M. Pollard, "A Monte Carlo method for factorization," *BIT*, v. 15, 1975, pp. 331–334. MR **50** #6992.

5. J. M. Pollard, "Monte Carlo methods for index computation (mod p)," *Math. Comp.*, v. 32, 1978, pp. 918–924. MR **52** #13611.

6. M. Rabin, "Probabilistic algorithms," *Algorithms and Complexity* (J. F. Traub, Ed.), Academic Press, New York, 1976, pp. 31–40.

7. H. C. Williams & J. S. Judd, "Some algorithms for prime testing using generalized Lehmer functions," *Math. Comp.*, v. 30, 1976, pp. 867–886.