MATHEMATICS OF COMPUTATION
Volume 67, Number 224, October 1998, Pages 1735–1738
S 0025-5718(98)00989-2

LARGE PRIMES AND FERMAT FACTORS

JEFF YOUNG

ABSTRACT. A systematic search for large primes has yielded the largest Fermat factors known.

INTRODUCTION

Over the last two years, the author has done a systematic search for large primes of the form $N = k \bullet 2^n + 1$. These primes were then checked to determine if they were Fermat factors. This search has resulted in finding the largest known primes of this type.

IMPLEMENTATION

The test used to determine whether a given N is prime is Theorem 102 from Hardy and Wright [2]. This theorem states.

> *Let $n \geq 2$, $k < 2^n$ and $N = k \bullet 2^n + 1$ be a quadratic non-residue $(\mathrm{mod}\, p)$ for some odd prime p. Then the necessary and sufficient condition for N to be a prime is that*
>
> $$p^{(N-1)/2} \equiv -1 \ (\mathrm{mod}\, N).$$

The program was coded in FORTRAN 77 and run on the CRAY-ELS computer system. This is the entry level system from Cray Research, Inc. It has the full functionality of the larger systems used to find the largest known Mersenne primes, but runs with a 30ns clock.

The first task in showing that N is prime is to find a prime p for which N is a quadratic non-residue. This was done by simply trying the primes in order until a suitable value was found. In all cases, we never had to use a prime greater than 37.

Because of the special form $k \bullet 2^n + 1$ of N, the computation of $p^{(N-1)/2} \equiv -1 \ (\mathrm{mod}\, N)$ consists mostly of repeated squaring of numbers with $n + \log_2(k)$ bits. The squaring was done via the usual Schoenhage-Strassen multiplication algorithm using FFTs. The reduction $(\mathrm{mod}\, N)$ was accomplished by doing a single-precision divide of the upper half of the squared result by k. To see why this is sufficient, let

Received by the editor June 19, 1995 and, in revised form, May 10, 1996.
1991 *Mathematics Subject Classification.* Primary 11A51; Secondary 11B99.

$X = A \bullet 2^n + B$, where $0 \le A < k^2 \bullet 2^n$, $0 \le B < 2^n$. Then we have the following:

$$\begin{aligned} X &= A \bullet 2^n + B \\ &= (k \bullet Q + R) \bullet 2^n + B, \quad \text{where } 0 \le Q < k \bullet 2^n,\ 0 \le R < k \\ &= (2^n k)Q + 2^n R + B \\ &= (N-1)Q + 2^n R + B \\ &= 2^n R + (B - Q) + QN \\ &\equiv 2^n R + (B - Q) \pmod{N}. \end{aligned}$$

The single-precision divide of A by k can thus be used instead of a more complicated algorithm such as Algorithm D from Knuth [3].

Results

The author searched the following areas for primes:

$$\begin{aligned} k = 3, &\quad 40000 \le n < 310000, \\ 5 \le k \le 7, &\quad 40000 \le n < 300000, \\ 9 \le k \le 17, &\quad 40000 \le n < 200000, \\ 19 \le k \le 31, &\quad 400000 \le n < 100000, \\ 33 \le k \le 63, &\quad 12000 \le n < 50000. \end{aligned}$$

k	n
3	42294 42665 44685 48150 55182 59973 80190 157169 213321 303093
5	125413 209787 240937
7	54486 88066 95330 207084 283034
9	43963 47003 49902 67943 114854 127003 145247 147073 149143
11	93279 105741
13	43856 88018 109258 114296
15	43388 48444 61758 184290
17	99231
19	73338
21	47337 55828 91008 94801
23	none found
25	57488 66872
27	44164 50696
29	88117 96947
31	76596 93168
33	37249 42685 47805 61372 74148
35	22645 27321 61963 68281
37	23014 32424 59354 62350
39	20221 21882 25654 28437 30325 31485 33593 37393 46362 62429 66971 77242 82191 94449
41	none found
43	29862 33526
45	29240 32018 38457 48612
47	none found
49	48666
51	20733 20807 26703 45541
53	none found
55	none found
57	20742 25010 26838 29623 45435
59	22455 29115 34437
61	none found
63	20746 24633 30053 34074

Searching these areas extends the work of Dubner and Keller [1] to the indicated values. The table above gives the new values of k and n for which $k \bullet 2^n + 1$ is prime.

All of the pairs (k, n) were tested to determine if they were Fermat factors. The following factors were found:

$$57 \bullet 2^{25010} + 1 \text{ divides } F_{25006},$$
$$21 \bullet 2^{94801} + 1 \text{ divides } F_{94798},$$
$$7 \bullet 2^{95330} + 1 \text{ divides } F_{95328},$$
$$13 \bullet 2^{114296} + 1 \text{ divides } F_{114293},$$
$$5 \bullet 2^{125413} + 1 \text{ divides } F_{125410},$$
$$3 \bullet 2^{157169} + 1 \text{ divides } F_{157167},$$
$$3 \bullet 2^{213321} + 1 \text{ divides } F_{213319},$$
$$3 \bullet 2^{303093} + 1 \text{ divides } F_{303088}.$$

All eight of these primes are larger than the previously largest known Fermat factor $5 \bullet 2^{23473} + 1$ which divides F_{23471}.

Error analysis

There are several ways errors can occur in this search:

(1) An invalid sieve is used and some potential candidates for the primality test are skipped.

(2) A number is declared prime which is really composite.

(3) A number is declared composite which is really a prime.

All pairs (k, n) which were removed via the sieve were then checked with another program written in C and run on a SPARCstation 5. With different hardware and software, we are confident that type-1 errors did not occur.

Several of the numbers reported below (including two of the Fermat factors) were checked by Harvey Dubner using his own software and the Cruncher. We feel that this type of confirmation eliminates type-2 errors.

As for type-3 errors, there is really no way of knowing if they have occurred unless the entire search is rerun and compared against the original residues. However, Harvey Dubner has also checked several composites and we agree on the final residues. We feel that this minimizes the chances of a type-3 error.

When using the FFT squaring technique, care must be taken to avoid overflow and underflow conditions. The CRAY library routines SCFFT and CSFFT were used. These routines have been analyzed previously by Cray Research to determine the maximum number of bits per word which were recoverable for a given length vector. After squaring, a checksum was calculated for further assurance of correctness.

The reduction step can also be checksummed. Let b = bits per word. From above, we have

$$X = 2^n R + (B - Q) + QN.$$

And so,

$$C_X \equiv 2^n C_R + (C_B - C_Q) + C_Q \bullet C_N \pmod{2^b - 1}, \text{ where } C_t \equiv t \pmod{2^b - 1}.$$

Now C_t can be quickly calculated by adding the elements of the vector which comprises t and folding it every b bits.

Acknowledgments

I would like to thank my wife Debra and son Nealon for their patience, Harvey Dubner and Wilfrid Keller for being faithful and inspirational colleagues, Wayne Roiger for keeping the faith, and Cray Research for the machine resources.

References

1. H. Dubner and W. Keller, *Factors of generalized Fermat numbers*, Math. Comp. **64** (1995), 397–405. MR **95c:**11010
2. G. Hardy and E. Wright, *The theory of numbers*, 4th ed., Oxford Univ. Press, 1975.
3. D. Knuth, *The art of computer programming*, vol. 2, Addison-Wesley, 1969. MR **44:**3531

655F Lone Oak Drive, Eagan, Minnesota 55121
E-mail address: jsy@cray.com