MATHEMATICS OF COMPUTATION
VOLUME 38, NUMBER 158
APRIL 1982, PAGES 639–643

Primes of the Form $n! \pm 1$ and $2 \cdot 3 \cdot 5 \cdots p \pm 1$

By J. P. Buhler, R. E. Crandall and M. A. Penk

Abstract. All primes less than 10^{1000} of the form $n! \pm 1$ or $2 \cdot 3 \cdot 5 \cdots p \pm 1$ are determined. Results of Brillhart, Lehmer, and Selfridge are used together with a fast algorithm that applies to primality tests of integers N for which many factors of $N \pm 1$ are known.

Let $p\#$ denote the product of all primes that are $\leqslant p$; for example $7\# = 2 \cdot 3 \cdot 5 \cdot 7 = 210$. In this paper we settle the question of the primality of all numbers of the form $n!+1$, $n!-1$, $p\# + 1$, or $p\# - 1$ that contain not more than one thousand digits. This means that all $n < 450$ and $p < 2371$ were considered (though the calculations were actually carried somewhat further as described below). Our work confirms the results on numbers of the form $n!+1$ and $p\# + 1$ in [2] (for $n \leqslant 100$ and $p \leqslant 307$) and [5] (for $n \leqslant 230$ and $p \leqslant 1031$).

The basic technique was to find possible primes by a pseudoprimality test and to then prove primality by a variant of the method described in [3]. The first section below contains the results and an outline of the basic methods. The proof of primality for numbers of the form $p\# \pm 1$ was greatly expedited by a fast divide-and-conquer algorithm; this is described in the second section.

All computations were done on a PDP 11-70 equipped with a UNIX operating system; both the C language and assembly language were used.

1. Let $N > 1$ be an odd integer. For the purposes of this paper, we will say that N is a pseudoprime if

$$a^{(N-1)/2} \equiv -1 \bmod N$$

for "several" integers $a > 2$, for which the Jacobi symbol $(\frac{a}{N})$ is equal to -1. It is easy to see that if $N = n! \pm 1$ (respectively $p\# \pm 1$), then the Jacobi symbol $(\frac{a}{N})$ is equal to 1 for all a less than n (respectively p) so the search was started at the first prime larger than n (respectively p).

If N is a pseudoprime, then it is virtually certain to be prime; if N fails the pseudoprimality test, then it is definitely composite. Because of the number of integers considered and their size it was necessary to write efficient computer routines (e.g., computing $a^m \bmod N$ quickly) in assembly language.

The technique used to show that a pseudoprime was actually a prime is a variant of a standard procedure based upon a partial factorization of $N \pm 1$. To consider the most basic example, suppose that the factorization of $N - 1$ is known:

$$N - 1 = \prod p^{a_p}.$$

Received June 15, 1981.
1980 *Mathematics Subject Classification.* Primary 10A25, Secondary 10A10.

Then N is a prime if one can find a set S of primes that divide $N-1$ such that

$$\prod_{p\in S} p^{a_p} > N^{1/2},$$

and for each $p \in S$ there is an x_p such that

$$x_p^{N-1} \equiv 1 \bmod N \quad \text{and} \quad x_p^{(N-1)/p} \not\equiv 1 \bmod N.$$

An important refinement due to Brillhart, Lehmer, and Selfridge [3] adds some auxiliary conditions and allows one to replace $N^{1/2}$ with $N^{1/3}$. In addition, similar ideas involving the Lucas-Lehmer sequences can be used to prove primality if a (partial) factorization of $N+1$ is known (or more generally if a factorization of $F_m(N)$ is known where F_m is a cyclotomic polynomial; these seem to be of less practical importance if $m > 2$).

For the numbers $N = n!+1$ this gives a straightforward feasible method of proving primality (since the factorization of $N-1$ is obvious). The time required depends on the size of the set S of primes p for which an x_p must be determined; in this case S is the set of $p \leqslant P$, where P is the smallest prime such that

$$\prod_{p\leqslant P} p^{v_p(n!)} > N^{1/3},$$

where $v_p(n!)$ denotes the exact power of p dividing $n!$. It can be shown that if n is large, then P is of order $n^{1/3}$. For the numbers given below only the primes $p = 2$, 3, 5, and 7 were needed.

Similar remarks apply to the case $N = n!-1$ since the Lucas-Lehmer sequences give a primality-proving algorithm when a factorization of $N+1$ is known [3].

For numbers of the form $N = p\# \pm 1$ the number of primes in a suitable set S is considerably larger. The amount of time required by a straightforward implementation of the above ideas would be on the order of several months total CPU time for the numbers listed below. However, an easy divide-and-conquer algorithm dramatically decreased the time required (to something on the order of a couple of days CPU time). This procedure is described in the next section.

The results of these computations were as follows:

(A) The number $N = n!+1$ is prime for

$$n = 1, 2, 3, 11, 27, 37, 41, 73, 77, 116, 154, 320, 340, 399, 427$$

and is composite for all other $n < 546$.

(B) The number $N = n!-1$ is prime for

$$n = 3, 4, 6, 7, 12, 14, 30, 32, 33, 38, 94, 166, 324, 379, 469$$

and is composite for all other $n < 546$.

(C) The number $N = p\# + 1$ is prime for the primes

$$p = 2, 3, 5, 7, 11, 31, 379, 1019, 1021, 2657$$

and is composite for all other $p < 3088$.

(D) The number $N = p\# - 1$ is prime for

$$p = 3, 5, 11, 13, 41, 89, 317, 991, 1873, 2053$$

and is composite for all other $p < 2377$. The number $2377\# - 1$ is a pseudoprime whose primality has not been verified.

Remarks. (1) Wilson's Theorem can be used to slightly shorten the search for primes of the form $N = n! \pm 1$; if $n + 1$ is prime then $n! + 1$ is composite (for $n > 2$) and if $n + 2$ is prime then $n! - 1$ is composite (for $n > 3$).

(2) For a general integer N an algorithm due to Miller [4] shows that primality can be verified in a time bounded by a polynomial in $\log(N)$ if a generalized version of the Riemann hypothesis is true. A recent algorithm due to Adleman et al. [1] shows that primality can be verified in a time bounded by

$$\log(N)^{c \cdot \log(\log(\log(N)))}$$

for an effective positive constant c; the method is probably feasible even for integers of the size considered here. As several people have observed, there are some philosophical problems connected with primality proofs—the probability of hardware "glitches" or undetected softward errors is surely far higher than the probability that a pseudoprime is not a prime.

(3) It seems reasonable to conjecture that for each odd prime p there is a prime $q < p$ such that the number

$$2 \cdot 3 \cdot 5 \cdots q \cdot p + 1 = p \cdot q\# + 1$$

is prime; similarly for $p \cdot q\# - 1$. On heuristic grounds it seems that one would expect $O(\log(p))$ such primes; we have found pseudoprimes of this form for all $p < 10000$. The only p that required a "large" q was $p = 673$; the smallest q for which $p \cdot q\# + 1$ was prime was $q = 509$. This value of p redeemed itself, however, in that $673 \cdot 509\# - 1$ was also a pseudoprime. Thus $673 \cdot 509\# \pm 1$ is a 215-digit twin prime pair (primality was verified).

2. The basic task required in the primality proof of a number of the form $N = P\# + 1$ is the computation of

$$x^{(N-1)/p} \bmod N$$

for roughly one-third of the primes $p \leqslant P$. The computation of a power x^q can be accomplished reasonably efficiently (by the usual method of repeated squarings based on the binary expansion of q). One could compute the numbers

$$x^{(N-1)/p} = x^{\prod_{q \neq p} q} \bmod N$$

either by multiplying the q together, using multiple-precision routines and exponentiating, or by repeatedly computing

$$x_1 = x^{q_1}, \quad x_2 = x_1^{q_2}, \quad x_3 = x_2^{q_3}, \ldots .$$

These algorithms fail to exploit the redundancy in the different exponents for the various p. An algorithm is given below that decreases the number of multiplications needed; the idea is an instance of the usual divide-and-conquer strategy found in many "fast" algorithms (e.g., standard versions of the fast Fourier transform).

For numbers of the form $N = P\# - 1$ the same ideas apply. Indeed, one can view the Lucas-Lehmer sequences as an implicit computation of a power x^q in a residue class ring of the ring of integers in a suitable quadratic field. Thus the algorithm below can be appropriately modified.

Suppose that x is an element of some group (written multiplicatively) and that we are given integers $a_1, a_2, \ldots, a_n$ together with a routine $\exp(x, a)$ that computes x^a when a is one of the a_i. We wish to compute the n quantities

$$x^{\prod_{j \neq i} a_j}, \qquad 1 \leqslant i \leqslant n, \tag{1}$$

by repeated calls to exp( ).

If $n = 2^m$, it is easy to devise a recursive scheme that requires $m \cdot 2^m$ calls instead of the $n(n-1) = 2^{2m} - 2^m$ required by the naive approach of iteratively computing $x_{j+1} = \exp(x_j, a_j)$.

The routine POWER(r, s, x) below extends this idea to arbitrary positive integers n; the running time is $O(n \cdot \log(n))$. The initial call should be POWER(1, $n+1$, x); in general POWER(r, s, x) computes the numbers in (1) above for $r \leqslant i < s$.

```
POWER(r, s, x)
  m = s − r;
  if m = 1, then output x and return;
  if m = 2, then output exp(x, a_r), exp(x, a_{r+1}) and return;
  m = m/2;
  y = x;
  for i = r, r + 1,...,r + m − 1, y = exp(y, a_i);
  POWER(r + m, s, y);
  y = x;
  for i = r + m, r + m + 1,...,s − 1, y = exp(y, a_i);
  POWER(r, r + m, y);
  return;
```

Remarks. (1) It is an easy exercise to prove the correctness of this algorithm inductively (first it does the left "half" and then the right "half") and to show that its running time is $O(n \cdot \log(n))$. The running time of either of the naive approaches suggested above is at least $O(n^2)$. The constant implicit in the $O(\ )$ notation depends on the running time of the routine exp( ), which in turn depends on the time required to multiply in the group.

(2) As usual the recursive scheme above can be converted to an iterative routine by introducing an appropriate stack.

(3) This algorithm is useful in the primality verification of integers N, for which the known part of the factorization of $N - 1$ (or $N + 1$) contains many primes.

Department of Mathematics
Reed College
Portland, Oregon 97202

Department of Physics
Reed College
Portland, Oregon 97202

2820 S. E. 20th
Portland, Oregon 97202

1. L. ADLEMAN, C. POMERANCE & R. RUMELY, "On distinguishing prime numbers from composite numbers." (Preprint.)

2. A. BORNING, "Some results for $k!+1$ and $2 \cdot 3 \cdot 5 \cdots p + 1$," *Math. Comp.*, v. 26, 1972, pp. 567–570.

3. J. BRILLHART, D. H. LEHMER & J. L. SELFRIDGE, "New primality criteria and factorizations of $2^m \pm 1$," *Math. Comp.*, v. 29, 1975, pp. 620–647.

4. G. L. MILLER, "Riemann's hypothesis and tests for primality," *J. Comput. Systems Sci.*, v. 13, 1976, pp. 300–317.

5. M. TEMPLER, "On the primality of $k!+1$ and $2 * 3 * 5 * \cdots * p + 1$," *Math. Comp.*, v. 34, 1980, pp. 303–304.