MATHEMATICS OF COMPUTATION
VOLUME 47, NUMBER 176
OCTOBER 1986, PAGES 703-711

The Primality of $R1031$

By H. C. Williams* and Harvey Dubner

Dedicated to D. H. Lehmer on the occasion of his 80th birthday

Abstract. A description is given of a technique for proving $R1031$ ($= (10^{1031} - 1)/9$) a prime.

1. Introduction. The repunit numbers, numbers of the form $R_n = (10^n - 1)/9$, have been of great interest to mathematical recreationists for many years. One rather peculiar feature of these numbers is the apparent scarcity of primes among them. Indeed, before the writing of this paper only $R2$, $R19$, $R23$, and $R317$ had been identified as primes. In [16] a search for further primes of this form for all $n \leqslant 2000$ turned up only one further candidate for primality, $R1031$. Unfortunately, the methods of primality testing available at the time were not adequate to prove $R1031$ a prime. Dubner has since conducted an extended search for repunit primes and, as a result of this work, it is now known that the only possible repunit primes R_n for all $n \leqslant 10000$ are the 5 numbers mentioned above. This made the resolution of the question of the primality of $R1031$ of greater interest.

Since the publication of [16] a great deal of research has been devoted to the problem of primality testing. The very important results of Adelman, Pomerance, and Rumely [1] have been modified by Cohen and Lenstra [5] and have resulted in the tests described by Cohen and Lenstra [6]. It is now possible to test the primality of numbers of up to 200 digits in about 10 minutes of computer time. However, it has been estimated that the application of these rapid, general purpose tests to the problem of determining the primality of a number as large as $R1031$ might require as many as several hundred hours of computer time.

Since the methods [5], [6] are somewhat complicated to program and would also be rather time-consuming to run on $R1031$, even on a very fast machine, we decided to make use of the earlier methods of Williams and Judd [13], [14]. The main difficulty in using these techniques is that they are not general purpose methods and require, in order to test a given N for primality, that we first obtain a sufficient number of prime factors from among those which divide, $N \pm 1$, $N^2 + 1$, $N^2 \pm N + 1$. Also, running the final tests can still be very time-consuming. In this paper we describe how we overcame these difficulties and were able to prove $R1031$ a prime.

Received December 9, 1985.
1980 *Mathematics Subject Classification.* Primary 10A25.
*Research supported by NSERC of Canada grant A7649.

2. The Strategy. Let $N = R1031$; we have

$$N - 1 = 10(10^{103} + 1) \cdot \frac{10^{103} - 1}{9} \cdot \frac{10^{515} + 1}{10^{103} + 1} \cdot \frac{10^{515} - 1}{10^{103} - 1}.$$

Now $(10^{103} - 1)/9$ is completely factored in [16] and the complete factorization of $10^{103} + 1$ can be obtained from Brillhart et al. [4] and results of Atkin and Rickert mentioned at the end of Williams [12]. Also $(10^{515} + 1)/(10^{103} + 1)$ has $7211 \cdot 9091 \cdot 497491 \cdot 569836171 \cdot 2013681931$ as a factor and $T = (10^{515} - 1)/(10^{103} - 1)$ has $41 \cdot 271 \cdot 5905014721$ as a factor (see [16]). With this information we have a completely factored part F of $N - 1$, where F is approximately 2.73×10^{251}. While this represents a very large factor of $N - 1$, it does not furnish us with sufficient information for developing a proof that N is a prime.

Dubner and Dubner [7] have developed a special purpose microcomputer that performs multiple-precision operations very quickly; in fact, this very small machine is about as fast as IBM's latest mainframe. This device was used to search for more factors of $N - 1$. By utilizing the techniques of Brent and Pollard [2], the factor 326345481191 of T was found and an additional factor 2702394989404991 of T was found by using the $p - 1$ method described by Pollard [9]. These factors were determined by the Dubner machine in a matter of a few minutes. With this information we have $N - 1 = F_1 C_1$, where F_1, the completely factored part of $N - 1$, exceeds 2.40×10^{278}.

Now the following theorem is well known and true for an arbitrary N (see, for example, Brillhart, Lehmer, and Selfridge [3]).

THEOREM. *If for each distinct prime q_i which divides F_1, there exists an integer a_i such that*

$$\gcd\left(a_i^{(N-1)/q_1} - 1, N\right) = 1 \tag{2.1}$$

and

$$a_i^{N-1} \equiv 1 \pmod{N}, \tag{2.2}$$

then any prime divisor of N must have the form $mF_1 + 1$. □

If N is a prime, the probability that (2.1) does not hold is about $1/q_i$. Thus, we elected to try a single a value such that $(a/N) = -1$ (Jacobi symbol). This means that (2.1) must hold if $q_i = 2$ and N is a prime. We determine

$$b \equiv a^{C_1} \pmod{N}.$$

Let $Q_i = F_1/q_i$, where q_i is a prime divisor of F. For each distinct q_i, we then evaluate

$$c_i \equiv b^{Q_i} \pmod{N}.$$

We see that (2.1) and (2.2) hold if

$$\gcd(c_i - 1, N) = 1 \quad \text{and} \quad c_i^{q_1} \equiv 1 \pmod{N}.$$

Of course, if $\gcd(c_i - 1, N) \neq 1$ for some q_i, we would be compelled to try a different value of a for that particular q_i.

By using only a single value of a $(= 7)$ and our particular N value ($R1031$), we were able to run these final tests (for the factors of F_1) and verify in about 18 minutes of AMDAHL 5850 CPU time that if p is a prime factor of N, then $p \equiv 1$

(mod F_1). All programming for the AMDAHL was done in FORTRAN with special assembler language subroutines used for the multiprecise operations of addition, subtraction, multiplication, and division.

It follows from this that if N is the product of three nontrivial factors f_1, f_2, f_3, then

$$N = (m_1F_1 + 1)(m_2F_2 + 1)(m_3F_3 + 1),$$

where $f_i = m_iF_1 + 1$ and $m_i > 0$ $(i = 1, 2, 3,)$. Hence,

$$m_1 + m_2 + m_3 \equiv C_1 \pmod{F_1}.$$

Let L be the remainder on dividing C_1 by F_1. Since

$$m_1 + m_2 + m_3 \geqslant L,$$

we must have $m_i > L/3$ for some $i \in \{1, 2, 3\}$. Thus, if N is the product of three nontrivial factors, we must have $N > LF_1^3/3$. In fact, we found by evaluating L that

$$LF_1^3/3 > N;$$

hence N must be the product of at most two prime factors.

Lenstra [8] has shown that if we know that a factor f of N must have the form $f = ms + r$, where $s > N^{1/3}$, then a fast algorithm, given in [8], can be used to find f. Unfortunately, our value for F_1 is much less than $N^{1/3}$; thus, we decided to search for factors of $N + 1$, $N^2 + 1$, and $N^2 \pm N + 1$. By using trial division, the $p - 1$ method, and Pollard's [10] Monte Carlo method, we found

$$N + 1 = 2^3 \cdot 3 \cdot 78869803 \cdot C_2 = F_2C_2,$$

$$N^2 + 1 = 2 \cdot 101 \cdot 2184509 \cdot 134089273 \cdot 1124225381 \cdot C_4 = 2F_4C_4,$$

$$N^2 + N + 1 = 7 \cdot 13 \cdot 193 \cdot 54223873993 \cdot C_3 = F_3C_3,$$

$$N^2 - N + 1 = 3 \cdot 37 \cdot 661 \cdot 4236022699 \cdot C_6 = 3F_6C_6.$$

Also, C_1, C_2, C_3, C_4, C_6, are composite integers. These new factors were also found in a matter of a few minutes on the Dubner machine. We need

$$K' = (F_1F_2F_3F_4F_6)/2 > N^{1/3};$$

however, $K' \simeq 7.47 \times 10^{341}$ and $N^{1/3} \simeq 2.23 \times 10^{343}$; hence, we require one more factor.

After using the Dubner machine to search each of C_1, C_2, C_3, C_4, C_6 for another factor (spending, on the average about 30 hours per C value), we still had found no additional factor. As C_6 was the last number to be tested and was still in the machine, we simply allowed the machine to keep running on it. After using the $p - 1$ method for 110 hours, the factor

$$\pi = 2211993420324463$$

of C_6 was discovered. Replacing, F_6 by πF_6, we now have

$$K = (F_1F_2F_3F_4F_6)/2 > N^{1/3}.$$

Let S be any factor of K such that $S > N^{1/3}$ and let

$$\bar{F}_i = \gcd(S, F_i) \qquad (i = 1, 2, 3, 4, 6).$$

Once the final tests for the primes dividing S given in [3], [13], and [14] have been executed (see Section 3 for a full description of these), we know that since N can be the product of at most two primes, we must, if N is composite, have one of 6 possibilities for one of these primes $p < N$. (See Williams and Holte [15].)

By using the Chinese Remainder Theorem, we first solve the system

$$x \equiv 1 \left(\text{mod } \bar{F}_1\right), \qquad x \equiv -1 \left(\text{mod } \bar{F}_2\right)$$

for y. We then solve each of the 6 systems

(i) $x \equiv y \pmod{\bar{F}_1\bar{F}_2}$, $x \equiv N \pmod{\bar{F}_4\bar{F}_3\bar{F}_6}$ for r_1;

(ii) $x \equiv y \pmod{\bar{F}_1\bar{F}_2}$, $x \equiv -N \pmod{\bar{F}_4}$, $x \equiv N \pmod{\bar{F}_3\bar{F}_6}$ for r_2;

(iii) $x \equiv y \pmod{\bar{F}_1\bar{F}_2}$, $x \equiv N \pmod{\bar{F}_4}$, $x \equiv 1 \pmod{\bar{F}_3}$, $x \equiv -1 \pmod{\bar{F}_6}$ for r_3;

(iv) $x \equiv y \pmod{\bar{F}_1\bar{F}_2}$, $x \equiv -N \pmod{\bar{F}_4}$, $x \equiv 1 \pmod{\bar{F}_3}$, $x \equiv -1 \pmod{\bar{F}_6}$ for r_4;

(v) $x \equiv y \pmod{\bar{F}_1\bar{F}_2}$, $x \equiv N \pmod{\bar{F}_4}$, $x \equiv -N - 1 \pmod{\bar{F}_3}$, $x \equiv -N + 1 \pmod{\bar{F}_6}$ for r_5;

(vi) $x \equiv y \pmod{\bar{F}_1\bar{F}_2}$, $x \equiv -N \pmod{\bar{F}_4}$, $x \equiv -N - 1 \pmod{\bar{F}_3}$, $x \equiv -N + 1 \pmod{\bar{F}_6}$ for r_6.

If N is composite, it must have a prime factor p such that $p \equiv r_i \pmod{S}$ where $i \in \{1, 2, 3, 4, 5, 6\}$ and $S > N^{1/3}$.

We can now use Lenstra's algorithm to attempt to find p. The only difficulty which arises in using this algorithm lies in the determination of whether or not there are integer zeros of a quadratic polynomial $X^2 - AX + B$ when A and B are large. But, since we usually do not expect to have such integral zeros, the best way to test for this is to find a collection of small primes $t_1, t_2, t_3, \ldots, t_m$ such that $t_i \nmid S$ (in this case) and determine for each t_i its set of $(t_i - 1)/2$ quadratic nonresidues $(\text{mod } t_i)$. We need, then, only to test whether or not $\Delta = A^2 - 4B \equiv$ to a quadratic nonresidue of $t_i \pmod{t_i}$. Of course, if Δ is a quadratic residue for each t_i, then Newton's method can be employed to determine whether or not Δ is truly a perfect square. We found that with $m = 15$ this method worked very well.

We implemented Lenstra's algorithm on the AMDAHL 5850 and tested it thoroughly on a large number of multiprecise composite numbers for which we knew the form of the factors. This algorithm works surprisingly rapidly and when used on the problem of finding a factor of $R1031$ required only 18 CPU seconds for each of the 6 (S, r_i) pairs.

3. The Final Tests. It remains to describe the final tests for the factors of S. We have already performed these tests for the factors of $\bar{F}_1$; hence, we need only perform them for factors of $\bar{F}_2$, $\bar{F}_3$, $\bar{F}_4$, $\bar{F}_6$. As was done in [3], we give the final tests for the factors of $\bar{F}_2$ in terms of Lucas Functions.

Let P, Q be integers such that $\gcd(P, Q) = 1$. Define the Lucas Functions by

$$V_n(P, Q) = \alpha^n + \beta^n, \qquad U_n(P, Q) = (\alpha^n - \beta^n)/(\alpha - \beta),$$

where α, β are the zeros of $x^2 - Px + Q$. Put $\Delta = P^2 - 4Q$.

Final Tests for the Factors of $\bar{F}_2$. Let D be a fixed integer such that the Jacobi symbol $(D/N) = -1$. For each distinct prime q_i such that $q_i | \bar{F}_2$, find a pair P, Q such that $D = P^2 - 4Q$,

$$\gcd\left(N, U_{(N+1)/2}(P, Q)\right) = 1,$$

and

$$N \mid U_{N+1}(P, Q).$$

In [3] it is shown how $U_m(P, Q) \pmod N$ can be computed rapidly, even when m is large. However, when N is very large, these final tests as stated here may consume more computer time than necessary. We describe below another means by which these tests can be conducted.

We first note that from the definition of the Lucas Functions we can show that**

$$V_{2m} = V_m^2 - 2Q^m, \tag{3.1}$$

$$V_{4n+2} = V_{2n}V_{2n+2} - Q^{2n}V_2, \tag{3.2}$$

$$V_m^2 - \Delta U_m^2 = 4Q^m, \tag{3.3}$$

$$V_m(V_n, Q^n) = V_{nm}(P, Q). \tag{3.4}$$

If we define $W_n \equiv V_{2n}Q^{-n} \pmod N$, we have

$$W_1 \equiv P^2Q^{-1} - 2 \pmod N. \tag{3.5}$$

Also, from (3.1), (3.2), and (3.3) we get

$$W_{2n} \equiv W_n^2 - 2 \pmod N, \tag{3.6}$$

$$W_{2n+1} \equiv W_nW_{n+1} - W_1 \pmod N, \tag{3.7}$$

$$\Delta\left(U_{2n}Q^{-n}\right)^2 \equiv W_n^2 - 4 \pmod N. \tag{3.8}$$

Since $W_0 = 2$, we see from (3.5) and (3.6) that the value of W_n depends only on that of W_1; hence, we may regard W_n to be a function of W_1 and n only. When we do this, we can use (3.4) to deduce

$$W_n(W_m) \equiv W_{nm}(W_1) \pmod N. \tag{3.9}$$

Let $\mathscr{T}_m$ denote the pair $\{W_m, W_{m+1}\}$, where each entry of $\mathscr{T}_m$ is reduced modulo N. Let $m = (b_0, b_1, b_2, \ldots, b_t)_2$ be the binary representation of m, and define $f_1(x) = x^2 - 2$, $f_2(x, y) = xy - W_1$. If we also define $c_0 = b_0 = 1$, $c_{j+1} = 2c_j + b_{j+1}$, $\mathscr{G}_i = \mathscr{T}_{c_i}$, then $\mathscr{G}_t = \mathscr{T}_m$. By (3.6) and (3.7) we can easily deduce that if $\mathscr{G}_i = \{x, y\}$, then

$$\mathscr{G}_{i+1} = \begin{cases} \{f_1(x), f_2(x, y)\} & \text{when } b_{i+1} = 0, \\ \{f_2(x, y), f_1(y)\} & \text{when } b_{i+1} = 1. \end{cases}$$

Thus, if we are given W_1, we now have a fast $O(\log m)$ algorithm for computing $W_m(W_1) \equiv V_{2m}(P, Q)Q^{-m} \pmod N$, when W_1 is given by (3.5).

We now give a different formulation of the final tests for the factors of $\bar{F}_2$. As before, we let D be a fixed integer such that the Jacobi symbol $(D/N) = -1$. We find P, Q such that $P^2 - 4Q \equiv D \pmod N$ and we determine W_1 by (3.5). Since Q is usually small, the computation of $Q^{-1} \pmod N$ can be done quickly. Compute

$$W_1^* \equiv W_A(W_1) \pmod W \qquad \left(A = (N+1)/2\bar{F}_2\right) \tag{3.10}$$

by using the algorithm given above. For each $q_i \mid \bar{F}_2$, put $Q_i = \bar{F}_2/q_i$,

$$X_i \equiv W_{Q_i}\left(W_1^*\right) \pmod N. \tag{3.11}$$

Note that $X_i \equiv W_{(N+1)/2q_i}(W_1) \pmod N$ by (3.9). Determine that

$$\gcd\left(X_i^2 - 4,\ N\right) = 1, \tag{3.12}$$

**When there is no doubt as to the arguments P, Q of V_n and U_n, we will omit them.

and for

$$Y_i \equiv W_{q_i}(X_i) \pmod N \tag{3.13}$$

($W_{q_i}(X_i) \equiv W_{(N+1)/2}(W_1) \pmod N$), that

$$N \mid Y_i^2 - 4. \tag{3.14}$$

Since $\gcd(DQ, N) = 1$, we see by (3.8) that this formulation of the final tests is correct. If (3.12) should not be true for some prime q_i, then the tests would have to be repeated with a different P, Q pair, but we would only test those q_i's for which (3.12) failed to be true. Since the probability that (3.12) will not hold when N is a prime is about $1/q_i$, this event is not very likely.

In fact, with $\bar{F}_2 = F_2/8$, $D = 21$, $P = 5$, $Q = 1$, we found that $N = R1031$ passed these final tests in about 4 CPU minutes of AMDAHL time.

The ideas developed here for the final tests for the factors of $\bar{F}_2$ can be extended to the relevant final tests for the factors of $\bar{F}_4$, $\bar{F}_3$, $\bar{F}_6$ given in [13] and [14]. For the generalized Lehmer Functions defined in [11] we now define

$$W_{j,n} \equiv V_{j,2n}Q^{-n} \pmod N,$$

$\mathscr{T}_m = \{W_{0,m}, W_{1,m}, W_{0,m+1}, W_{1,m+1}\}$ when $k = 2$ (used when dealing with the factors of $\bar{F}_4$), and $\mathscr{T}_m = \{W_{0,m}, W_{1,m}, W_{2,m}, W_{0,m+1}, W_{1,m+1}, W_{2,m+1}\}$ when $k = 3$ (used when dealing with the factors of $\bar{F}_3$ or $\bar{F}_6$). As before, the entries of $\mathscr{T}_m$ are reduced modulo N; also, for a given m, we put $\mathscr{G}_i = \mathscr{T}_{c_i}$.

If we define

$$\begin{aligned}
&g_1(x, y) = x^2 - P_2y^2 - 2, \quad g_2(x, y) = 2xy + P_1y^2,\\
&g_3(x, y, z, w) = xz - P_2yw - W_{0,1},\\
&g_4(x, y, z, w) = yz + xw + P_1yw - W_{1,1}\\
&\qquad = (y + x)(z + w) - xz + (P_1 - 1)yw - W_{1,1},
\end{aligned}$$

then when $k = 1$, we have $W_{0,0} = 2$, $W_{1,0} = 1$, $W_{0,1} \equiv +P_2Q^{-1} - 2$, $W_{1,1} \equiv P_1Q^{-1} \pmod N$ and

$$\begin{aligned}
&W_{0,2n} \equiv g_1(W_{0,n}, W_{1,n}), \qquad W_{1,2n} = g_2(W_{0,n}, W_{1,n}),\\
&W_{0,2n+1} \equiv g_3(W_{0,n}, W_{1,n}, W_{0,n+1}, W_{1,n+1}),\\
&W_{1,2n+1} \equiv g_4(W_{0,n}, W_{1,n}, W_{0,n+1}, W_{1,n+1}) \pmod N.
\end{aligned}$$

These forms can be easily verified by using the methods of [11] and the simple results (analogous to (3.1) and (3.2)) that

$$\begin{aligned}
&v_{2n}(P, Q) = v_n(P, Q) - 2Q^n,\\
&v_{4n+2}(P, Q) = v_{2n+1}(P, Q)v_{2n}(P, Q) - Q^{2n}v_2(P, Q).
\end{aligned}$$

Thus, if $\mathscr{G}_i = \{x, y, z, w\}$, then

$$\mathscr{G}_{i+1} = \begin{cases} \{g_1(x, y), g_2(x, y), g_3(x, y, z, w), g_4(x, y, z, w)\} & \text{if } b_{i+1} = 0,\\ \{g_3(x, y, z, w), g_4(x, y, z, w), g_1(z, w), g_2(z, w)\} & \text{if } b_{i+1} = 1. \end{cases}$$

Also, if we write

$$W_{0,n} = W_{0,n}(W_{0,1}, W_{1,1}, P_1, P_2), \qquad W_{1,n} = W_{1,n}(W_{0,1}, W_{1,1}, P_1, P_2),$$

then

$$W_{0,mn} \equiv W_{0,m}(W_{0,n}, W_{1,n}, P_1, P_2), \quad W_{1,mn} \equiv W_{1,m}(W_{0,n}, W_{1,n}, P_1, P_2) \pmod N.$$

By noting that the U_m in final test α (the final tests for the factors of $\bar{F}_4$) of [13] is the same as $V_{1,m}$, we can reformulate test α here in an analogous manner to our reformulation of the final tests for the factors of $\bar{F}_2$ given in [3]. We use the previous value of D and select a fixed C value as described in [13]. We then select H, K and compute P_1, P_2, Q by the formulas given in [13]. We compute $W_{0,1}$, $W_{1,1} \pmod N$. The test has the same structure of that for the factors of $\bar{F}_2$, but we replace (3.10) by

$$W_{0,1}^* \equiv W_{0,A}(W_{0,1}, W_{1,1}, P_1, P_2) \pmod N \quad \left(A = (N^2+1)/2\bar{F}_4\right),$$
$$W_{1,1}^* \equiv W_{1,A}(W_{0,1}, W_{1,1}, P_1, P_2) \pmod N,$$

(3.11) by

$$X_{0,i} \equiv W_{0,Q_i}\left(W_{0,1}^*, W_{1,1}^*, P_1, P_2\right) \pmod N,$$
$$X_{1,i} \equiv W_{1,Q_i}\left(W_{0,1}^*, W_{1,1}^*, P_1, P_2\right) \pmod N,$$

(3.12) by

$\gcd(X_{1,i}, N) = 1$ (if this does not occur, we must change the (H, K) pair for q_i),

(3.13) by

$$Y_{1,i} \equiv W_{1,q_i}(X_{0,i}, X_{1,i}, P_1, P_2) \pmod N,$$

and (3.14) by

$$N \mid Y_{1,i}.$$

Here, of course, $q_i \mid \bar{F}_4$ and $Q_i = \bar{F}_4/q_i$.

Using this reformulation of test α with $\bar{F}_4 = F_4$ and $D = 21$, $C = 29$, $H = 1$, $K = 0$, we found that N passed these final tests in about 16 minutes of AMDAHL CPU time.

If, when $k = 3$, we further define

$$h_1(x,y,z) = x^2 + 2P_3yz + P_1P_2z^2 - 2,$$
$$h_2(x,y,z) = 2xy - 2P_2yz + (P_3 - P_1P_2)z^2,$$
$$h_3(x,y,z) = y^2 + 2zx + 2P_1yz + \left(P_1^2 - P_2\right)z^2,$$
$$h_4(x,y,z,u,v,w) = xu + P_3(y+z)(v+w) + (P_1 - P_2P_3)zw - P_3yv - W_{0,1},$$
$$h_5(x,y,z,u,v,w) = (y+x)(u+v) - xu + (P_2 - 1)yv - P_2(z+y)(v+w) + (P_3 - P_1P_2 + P_2)zw - W_{1,1},$$
$$h_6(x,y,z,u,v,w) = (z+x)(u+w) - xu - (P_1 - 1)yv + \left(P_1^2 - P_2 - P_1 - 1\right)zw + P_1(y+z)(v+w) - W_{2,1},$$

then we can show that for $i = 0, 1, 2$, we have

$$W_{i,2n} \equiv h_{i+1}(W_{0,n}, W_{i,n}, W_{2,n}),$$
$$W_{i,2n+1} \equiv h_{i+4}(W_{0,n}, W_{i,n}, W_{2,n}, W_{0,n+1}, W_{1,n+1}, W_{2,n+1}) \pmod N.$$

Also, $W_{0,1} \equiv -2$, $W_{1,1} \equiv 0$, $W_{2,1} \equiv Q^{-1} \pmod N$, and

$$W_{i,mn} \equiv W_{i,m}(W_{0,n}, W_{1,n}, W_{2,n}, P_1, P_2, P_3) \pmod N.$$

Thus, we can reformulate the final tests (1) and (3) of [14] in a manner analogous to our other reformulations.

When we implemented these new revisions of tests (1) and (3), using $a = 21$, $b = 7$, $h_i = 1$, $k_i = 1$, $l_i = 0$, $G = 1$ (for test (1)) and $G = 7$ (for test (3)) to determine P_1, P_2, P_3, Q as discussed in [14], we required 39 minutes of AMDAHL CPU time to run each test. Here $\bar{F}_3 = F_3/7$, $\bar{F}_6 = F_6$. With $\bar{F}_1 = F_1$, $S = \bar{F}_1\bar{F}_2\bar{F}_3\bar{F}_4\bar{F}_6 > N^{1/3}$, we then ran Lenstra's test as described in Section 2. As we found no nontrivial factor of N, we now know that N must be a prime. This was done in a total of $18 + 4 + 16 + 39 + 39 + 6(18/60) \approx 118$ minutes of AMDAHL time.

It might be argued here that even though only 2 hours of AMDAHL time was needed to prove $R1031$ a prime, we still needed in excess of 200 hours of time on the Dubner machine to find the final prime factor π which we needed in order to begin our primality proof. In fact, because of the (unexpected) speed of Lenstra's algorithm, we could have proved $R1031$ a prime without this final factor. Had we used slightly different parameters (select Q in the final tests for the factors of F_2 such that $(Q/N) = -1$; select a, b, for tests (1) and (3) such that P in [14] is not 7) in some of our final tests, we could have put $\bar{F}_1 = F_1$, $\bar{F}_2 = F_2$, $\bar{F}_3 = F_3$, $\bar{F}_4 = F_4$, $\bar{F}_6 = F_6/\pi$ and have $S = K'$, where K' is defined in Section 2.

Had these tests been performed, we would know that if N is composite, it must have a prime factor p such that $p \equiv r_i \pmod{S}$, $i \in \{1, 2, 3, 4, 5, 6\}$. But for this value of S we have

$$N^{1/3}/S < 31$$

and $31 \nmid S$. Thus, if we use Lenstra's algorithm on the 6×30 pairs (S', r_i') $(i = 1, 2, 3, 4, 5, 6)$, where $S' = 31S$, $r_i' \equiv r_i \pmod{S}$, and $r_i' \equiv 1, 2, 3, \ldots, 30 \pmod{31}$, we could have demonstrated the primality of N in an additional $29 \times 6 \times 18$ CPU seconds or 53 extra CPU minutes. Nevertheless, we prefer the proof given here as it is more succinct. We do, however, wish to recommend highly the implementation of Lenstra's algorithm in any general or special purpose primality testing routine where its use is relevant. It is very easy to program, executes rapidly, and saves a great deal of time that might otherwise have to be spent on computationally more expensive final tests.

4. Acknowledgment. The authors gratefully acknowledge the enthusiasm and interest displayed in this project by Samuel Yates.

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

Dubner Computer Systems, Inc.
158 Linwood Plaza
Fort Lee, New Jersey 07024

1. L. M. ADELMAN, C. POMERANCE & R. S. RUMELY, "On distinguishing prime numbers from composite numbers," *Ann. of Math.* (2), v. 117, 1983, pp. 173–206.

2. R. P. BRENT & J. H. POLLARD, "Factorization of the eighth Fermat number," *Math. Comp.*, v. 36, 1981, pp. 627–630.

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

4. J. BRILLHART, D. H. LEHMER, JOHN SELFRIDGE, B. TUCKERMAN & S. S. WAGSTAFF, JR., *Factorizations of $b^n + 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ Up to High Powers*, Contemp. Math., vol. 22, Amer. Math. Soc., Providence, R. I., 1983.

5. H. COHEN & W. H. LENSTRA, JR., "Primality testing and Jacobi sums," *Math. Comp.*, v. 42, 1984, pp. 297–330.

6. H. COHEN & A. K. LENSTRA, "Implementation of a new primality test," *Math. Comp.* (To appear.)

7. H. DUBNER & R. DUBNER, "The development of a powerful, low-cost computer for number theory application," *J. Recreational Math.* (To appear.)

8. H. W. LENSTRA, JR., "Divisors in residue classes," *Math. Comp.*, v. 42, 1984, pp. 331–340.

9. J. M. POLLARD, "Theorems on factorization and primality testing," *Proc. Cambridge Philos. Soc.*, v. 76, 1974, pp. 521–528.

10. J. M. POLLARD, "A Monte Carlo method for factorization," *BIT*, v. 15, 1975, pp. 331–334.

11. H. C. WILLIAMS, "A generalization of Lehmer's functions," *Acta Arith.*, v. 29, 1976, pp. 315–341.

12. H. C. WILLIAMS, "Factoring on a computer," *Math. Intelligencer*, v. 6, 1984, pp. 29–36.

13. H. C. WILLIAMS & S. JUDD, "Determination of the primality of N by using factors of $N^2 \pm 1$," *Math. Comp.*, v. 30, 1976, pp. 157–172.

14. H. C. WILLIAMS & S. JUDD, "Some algorithms for prime testing using generalized Lehmer functions," *Math. Comp.*, v. 30, 1976, pp. 867–886.

15. H. C. WILLIAMS & R. HOLTE, "Some observations on primality testing," *Math. Comp.*, v. 32, 1978, pp. 905–917.

16. H. C. WILLIAMS & E. SEAH, "Some primes of the form $(a^n - 1)/(a - 1)$," *Math. Comp.*, v. 33, 1979, pp. 1337–1342.