Factoring Large Integers

By R. Sherman Lehman

Abstract. A modification of Fermat's difference of squares method is used for factoring large integers. This modification permits factoring \( n \) in \( O(n^{1/9}) \) elementary operations, where addition, subtraction, multiplication, division, or the extraction of a square root is considered as an elementary operation. A principal part is played by the use of a dissection of the continuum similar to the Farey dissection. This has been programmed for \( n \leq 1.05 \times 10^{20} \) on the CDC 6400.

1. Introduction. Fermat's method for factoring an odd positive integer \( n \) consists of finding \( n = x^2 - y^2 \) where \( x \) and \( y \) are positive integers. We find in succes-

\[
x = [n^{1/2}] + 1, \quad x = [n^{1/2}] + 2, \ldots
\]

and determine whether the difference \( x^2 - n \) is a square or not. If \( p \) and \( q \) are primes and \( n = pq \), then Fermat's method is quite efficient if \( p/q \) is near 1, but it requires a large number of trials if \( p/q \) is not near 1. Lawrence [2] used a method which is designed to be efficient if \( p/q \) is near \( a/b \) where \( a \) and \( b \) are small relatively prime integers.

We consider \( x^2 - y^2 = 4kn \), \( k = ab \) with \( 1 \leq k \leq r \). The idea we wish to use is to divide up the interval \([0, 1]\) into parts. Each part will correspond to a fraction \( a/b \), and these parts will fill the interval \([0, 1]\). This means that, for each \( r \), we find a sequence \( S \), which includes \( a/b \) when \( 0 \leq a \leq b, b > 0 \) and \( ab \leq r \). This is reminiscent of the Farey sequence of order \( r \). We prove in Section 3 that many of the ideas go over to the new sequence \( S \). In particular, one obtains a dissection of the continuum similar to the Farey dissection of \([0, 1]\).

The main theorem is given in Section 2. Its proof is contained in Section 4. Numerical results were obtained by a computation on the CDC 6400 of the Computer Center of the University of California at Berkeley. An Algol program is also given in Section 5.

2. The Theorem. We shall use \( \gcd(a, b) \) for the greatest common divisor of \( a \) and \( b \).

Theorem. Suppose that \( n \) is a positive odd integer and \( r \) is an integer such that \( 1 \leq r < n^{1/2} \). If \( n = pq \) where \( p \) and \( q \) are primes and

\[
(n/(r + 1))^{1/2} < p \leq n^{1/2},
\]

Received April 21, 1973.


Key words and phrases. Factorization, Farey series, minimum operations, Fermat's method.

Copyright © 1974, American Mathematical Society

637
then there are nonnegative integers \(x, y\) and \(k\) such that
\[
x^2 - y^2 = 4kn, \quad 1 \leq k \leq r,
\]
\[
x \equiv k + 1 \pmod{2},
\]
\[
x \equiv k + n \pmod{4} \quad \text{if } k \text{ is odd},
\]
\[
0 \leq x - (4kn)^{1/2} \leq (1/4(r + 1))(n/k)^{1/2}
\]
and
\[
p = \min(\gcd(x + y, n), \gcd(x - y, n)).
\]

If \(n\) is a prime, then there are no integers satisfying (2.1).

Let us see how many elementary operations are required to obtain the primes \(p\) and \(q\) when \(n = pq\). First, there are a constant times \((n/(r + 1))^{1/2}\) divisions involved to determine whether there is a small prime factor less than \((n/(r + 1))^{1/2}\). We find that there are
\[
O((n/r)^{1/2}) + \sum_{1 \leq k \leq r} O((1/r)(n/k)^{1/2} + 1)
\]
elementary operations, where the extraction of a square root is counted as one operation. We have
\[
O((n/r)^{1/2}) + O((1/r)n^{1/2}r^{1/2}) + O(r)
\]
operations. Here, if we choose \(r\) to be a constant times \(n^{1/3}\), we find \(O(n^{1/3})\) elementary operations are required.

3. The Sequence \(S_r\). If \(r\) is a positive integer, then we denote by \(S_r\) the sequence of rational numbers \(a/b\) where \(0 \leq a \leq b, b > 0\) and \(ab \leq r\) with \(a\) and \(b\) relatively prime integers. We suppose that the sequence is arranged in order of increasing size. For example, \(S_{15}\) is the sequence
\[
0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 3 \quad 2 \quad 3 \quad 1
\]
\[
1 \quad 15 \quad 14 \quad 13 \quad 12 \quad 11 \quad 10 \quad 9 \quad 8 \quad 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 2 \quad 3 \quad 4 \quad 1.
\]

**Lemma 1.** If \(a/b\) and \(a'/b'\) are two successive terms of \(S_r\), then
\[
a'b - ab' = 1 \quad \text{and} \quad (a + a')(b + b') > r.
\]

**Proof.** It is well known that the Farey series of order \(n\), which consists of all reduced fractions between 0 and 1 whose denominators do not exceed \(n\), can be generated starting from 0/1, 1/1 by the following process: Between two successive terms of the sequence generated, say \(a/b\) and \(a'/b'\), insert their mediant \((a + a')/(b + b')\), which is always a reduced fraction, whenever \(b + b'\) does not exceed \(n\). A similar method can be used to generate \(S_r\)—we insert the mediant \((a + a')/(b + b')\) whenever \((a + a')(b + b') \leq r\). It follows that two successive terms of \(S_r\) are successive terms in a Farey series of some order and thus \(a'b - ab' = 1\). To avoid insertion of the mediant \((a + a')/(b + b')\) between them, we must have \((a + a')(b + b') > r\). This completes the proof.

We now use a dissection of the interval \([0, 1]\) which is analogous to the Farey dissection of the continuum (see [1, p. 29]). We take the sequence \(S_r\) and form the mediants between each two successive terms. We then cut up the interval \([0, 1]\) into
FACTORSING LARGE INTEGERS

639

pieces using the mediants as division points. Thus, we obtain a subinterval corresponding to each term of \( S_r \). It will be convenient to use closed subintervals. Corresponding to \( 0/1 \), we have the subinterval \([0, 1/(r + 1)]\), and, corresponding to \( 1/1 \), we have the subinterval \([a^*/b^* + 1, 1]\) where \( a^*/b^* \) is the term preceding 1/1 in \( S_r \). If \( a'/b', a/b, \) and \( a''/b'' \) are three successive terms of \( S_r \), then, corresponding to \( a/b \), we have the subinterval

\[
\left[ \frac{a + a'}{b + b'}, \frac{a + a''}{b + b''} \right].
\]

By Lemma 1, we have

\[
\frac{a + a'}{b + b'} = \frac{a}{b} - \frac{1}{b(b + b')}, \quad \frac{a + a''}{b + b''} = \frac{a}{b} + \frac{1}{b(b + b'')}. \tag{3.1}
\]

We shall call this dissection with subintervals corresponding to \( S_r \) the dissection of order \( r \).

**Lemma 2.** If \( \alpha \) is in the subinterval corresponding to \( a/b \) with \( a > 0 \) in the dissection of order \( r \), then

\[
\frac{a}{b} \left( 1 - \delta(1 + \frac{1}{2} \delta^2)^{1/2} + \frac{1}{2} \delta^2 \right) \leq \alpha \leq \frac{a}{b} \left( 1 + \delta(1 + \frac{1}{2} \delta^2)^{1/2} + \frac{1}{2} \delta^2 \right)
\]

where \( \delta = \{|ab(r+1)|^{-1/2} \).  

*Proof.* Let \( a'/b' \) be the term preceding and \( a''/b'' \) the term following \( a/b \) in \( S_r \), and suppose that \( \alpha \) is in the subinterval corresponding to \( a/b \) with \( b \geq a \geq 1 \). Since the mediant \( (a + a')/(b + b') \) is not in \( S_r \), we have, by (3.1) and Lemma 1,

\[
r + 1 \leq (a + a')(b + b') = \frac{a + a'}{b + b'} (b + b')^2 = \frac{a}{b} (b + b')^2 - \frac{(b + b')}{b}.
\]

Similarly, we have

\[
r + 1 \leq \frac{a}{b} (b + b')^2 + \frac{(b + b')}{b}.
\]

Using the first of these quadratic inequalities and that \( b + b' > 0 \), we obtain

\[
b + b' \geq \left\{ 1 + (1 + 4ab(r + 1))^{1/2} \right\}/2a
\]

and

\[
\frac{1}{b(b + b')} \leq \frac{a}{b} \left\{ \frac{2}{1 + (1 + 4ab(r + 1))^{1/2}} \right\} = \frac{a}{b} \left\{ -\frac{1}{2} \delta^2 + \delta(1 + \frac{1}{2} \delta^2)^{1/2} \right\}.
\]

Hence, by (3.1), we obtain the first inequality of the lemma. Similarly, using the second of these quadratic inequalities, we obtain

\[
b + b'' \geq \left\{ -1 + (1 + 4ab(r + 1))^{1/2} \right\}/2a
\]

and

\[
\frac{1}{b(b + b'')} \leq \frac{a}{b} \left\{ \frac{1}{2} \delta^2 + \delta(1 + \frac{1}{2} \delta^2)^{1/2} \right\}.
\]

From this, we obtain the second inequality of the lemma. This completes the proof.
4. Proof of Theorem. Let \( n \) be an odd prime or let \( n = pq \) where \( p \) and \( q \) are two odd primes with \( p \leq n^{1/2} \leq q \). Consider the equation

\[
(x + y)(x - y) = x^2 - y^2 = 4kn
\]

where \( x \) and \( y \) are nonnegative integers and \( k \) is a positive integer. Then

\[
x + y = sa'n, \quad x - y = tb' \quad \text{or} \quad x + y = tb', \quad x - y = sa'n,
\]

where \( s, t, a' \) and \( b' \) are positive integers and \( st = 4, a'b' = k \);

\[
x + y = sa'q, \quad x - y = tb'p \quad \text{or} \quad x + y = tb'p, \quad x - y = sa'q,
\]

where \( s, t, a' \) and \( b' \) are positive integers and \( st = 4, a'b' = k \).

To consider (4.2), we add the two equations and we get in either case \( 2x = sa'n + tb' \). There are three possible cases: \( s = 4, t = 1; s = 1, t = 4; s = 2, t = 2 \). These give

\[
x = 2a'n + \frac{1}{2}b', \quad x = \frac{1}{2}a'n + 2b', \quad x = a'n + b'.
\]

In the first case, we see \( b' \) is even. Setting \( a = 2a', b = \frac{1}{2}b' \), we get

\[
x = an + b \quad \text{with} \quad ab = k.
\]

In the second case, we see that \( a'n \) is even, and because \( n \) is odd, \( a' \) must be even. Setting \( a = \frac{1}{2}a', b = 2b' \), we again get (4.4). In the last case, we obtain (4.4) with \( a = a', b = b' \).

We prove that if \( r \) is an integer such that \( 1 \leq r < n^{1/2} \), then

\[
x - (4kn)^{1/2} > \frac{1}{4(r + 1)} \left(\frac{n}{k}\right)^{1/2}, \quad 1 \leq k \leq r,
\]

is correct. This contradicts one of the inequalities in (2.1). Actually, we prove the stronger inequality

\[
x - (4kn)^{1/2} > \frac{1}{4(k + 1)} \left(\frac{n}{k}\right)^{1/2}, \quad 1 \leq k \leq n - 2.
\]

It is equivalent to

\[
an + b > 2k^{1/2}n^{1/2} + \frac{1}{4(k + 1)} \left(\frac{n}{k}\right)^{1/2}, \quad 1 \leq k \leq n - 2, \quad ab = k,
\]

by (4.4). Squaring both sides, we obtain

\[
a^2n^2 - 2kn + b^2 > n/(k + 1) + n/16(k + 1)^2k.
\]

We see that it can be reduced to a special case \( a = 1, b = k \) when the left side is \( n^2 - 2kn + k^2 \). For, if \( a \geq 2 \) and \( 1 \leq k \leq n - 2 \), then

\[
(a^2 - 4)n^2 + (n^2 - k^2) + b^2 + 2n^2 \geq 0
\]

or

\[
a^2n^2 - 2kn + b^2 \geq n^2 - 2kn + k^2.
\]

Thus, it is sufficient to consider \((n - k)^2 > n/(k + 1) + n/16(k + 1)^2k\) where \(1 \leq k \leq n - 2\). A stronger inequality is
\[(n - k)^2 - \left(1 + \frac{1}{16}\right) \frac{n}{k + 1} \geq 0 \quad \text{or} \quad G(k, n) = (k + 1)(n - k)^2 - \left(1 + \frac{1}{16}\right)n \geq 0,\]

where \(1 \leq k \leq n - 2, n \geq 3\) for all real values of \(k\) and \(n\). Differentiating \(G(k, n)\) with respect to \(k\), we have

\[
\frac{\partial G(k, n)}{\partial k} = (n - k)^2 - 2(k + 1)(n - k) = (n - k)(n - 2 - 3k).
\]

For fixed \(n\), we see that \(\partial G/\partial k = 0\) at \(k = n\) and \(k = (n - 2)/3\), and that \(G(k, n)\) is increasing for \(0 \leq k < (n - 2)/3\) and decreasing for \((n - 2)/3 < k < n\). Thus, it is sufficient to check it on the two rays \(k = 1, n \geq 3\) and \(k = n - 2, n \geq 3\). We have

\[
G(1, n) = 2(n - 1)^2 - \left(1 + \frac{1}{16}\right)n = 2n^2 - 4n + 2 - \left(1 + \frac{1}{16}\right)n
\]

\[\geq 6n - 4n - \left(1 + \frac{1}{16}\right)n + 2 > 2\]

and

\[
G(n - 2, n) = 4(n - 1) - \left(1 + \frac{1}{16}\right)n \geq 8 - 3\left(1 + \frac{1}{16}\right) > 4.
\]

It follows that it remains positive, and thus (4.5) is proved. We have shown that there is no solution to (2.1) when \(n\) is a prime.

Now, we consider (4.3). Adding the two equations, we get

\[2x = sa'q + tb'p.\]

There are three possible cases: \(s = 4, t = 1; s = 1, t = 4; s = 2, t = 2\). These give

\[x = 2a'q + \frac{1}{2}b'p, \quad x = \frac{1}{2}a'q + 2b'p, \quad x = a'q + b'p.\]

In the first case, we see that \(b'p\) is even and because \(p\) is odd, \(b'\) must be even. Setting \(a = 2a', b = \frac{1}{2}b',\) we have

\[
(4.6) \quad x = aq + bp, \quad y = |aq - bp|, \quad k = ab,
\]

with \(a\) and \(b\) positive integers. In the second case, we see that \(a'q\) is even and because \(q\) is odd, \(a'\) must be even. Setting \(a = \frac{1}{2}a', b = 2b',\) we get (4.6). In the last case, we get (4.6) with \(a = a', b = b'\).

Let \(d = \gcd(a, b).\) Then

\[
(4.7) \quad x = aq + bp = d(a_1q + b_1p), \quad y = |aq - bp| = d |a_1q - b_1p|
\]

where \(a_1\) and \(b_1\) are positive integers, and

\[
(x/d)^2 - (y/d)^2 = 4a_1b_1n, \quad k = d^2a_1b_1.
\]

Thus, it can be reduced to the case in which \(a\) and \(b\) are relatively prime positive integers.

If \(a\) and \(b\) are relatively prime, then we can prove that \(x \equiv k + 1 \pmod{2}.\) If \(k = ab\) is even, then one of the integers \(aq\) and \(bp\) is even and the other is odd since, by assumption, \(p\) and \(q\) are odd while \(a\) and \(b\) are relatively prime. It follows that \(x = aq + bp\) is odd. On the other hand, if \(k\) is odd, then the integers \(aq\) and \(bp\) are odd. It follows that \(x\) is even.

We can prove that if \(k\) is odd, then \(x \equiv k + n \pmod{4}.)\) We consider \(p\) and \(q\)
which are odd and also $a$ and $b$ which are odd. Then $p - a$ is even and $q - b$ is even. Hence, $(p - a)(q - b) = pq + ab - aq - bp = n + k - x$ is divisible by 4.

Since $ab = k < n^{1/2} \leq q$, we have

$$p = \gcd(2bp, n) = \min(\gcd(x + y, n), \gcd(x - y, n))$$

where any solution of (4.6) is used.

It remains to prove that

$$0 \leq x - (4kn)^{1/2} \leq \frac{1}{4(r + 1)} \left( \frac{n}{k} \right)^{1/2}$$

where

$$p > (n/(r + 1))^{1/2}.$$ (4.9)

Let $m = 4kn = 4abn$, and let $\tau = x - m^{1/2}$. Because the arithmetic mean is not less than the geometric mean

$$x = aq + bp \geq 2(abpq)^{1/2} = m^{1/2}$$

and thus $\tau \geq 0$ which proves the left half of (4.8).

Letting $\epsilon = \tau m^{-1/2}$, we have

$$x = (1 + \epsilon)m^{1/2}, \quad y = (2\epsilon + \epsilon^2)^{1/2}m^{1/2}.$$ (4.10)

The right half of the inequality (4.8) translates into

$$\epsilon \leq \frac{1}{2}\delta^2$$

where $\delta = \{ab(r + 1)\}^{-1/2}$.

We now show that the point $\alpha = p/q$ lies in the subinterval corresponding to $a/b$ in the dissection of order $r$ discussed in Section 3. In applying Lemma 2, we must show that $p/q$ does not lie in the interval $[0, 1/(r + 1)]$. This follows from

$$\frac{p}{q} = \frac{p}{n/p} = \frac{p^2}{n} > \frac{1}{r + 1}$$

by (4.9). We obtain

$$\xi_1 a/b \leq p/q \leq \xi_2 a/b$$

where

$$\xi_1 = 1 - \frac{1}{2}(1 + \frac{1}{4}\delta^{2})^{1/2} + \frac{1}{2}\delta^2,$$

$$\xi_2 = 1 + \frac{1}{2}(1 + \frac{1}{4}\delta^{2})^{1/2} + \frac{1}{2}\delta^2$$

are the two positive roots of the equation

$$\left(1 - \varepsilon\right)^2 = \xi \delta^2.$$ (4.11)

We consider separately two cases depending on whether $p/q \leq a/b$ or $p/q > a/b$. First, if $p/q \leq a/b$ then $aq \geq bp$, and, by (4.6), we have

$$\frac{p}{q} = \frac{a(x - y)}{b(x + y)} = \frac{a}{b} \left(1 + \epsilon - (2\epsilon + \epsilon^2)^{1/2}\right) = \frac{a}{b} \left(1 + \epsilon - (2\epsilon + \epsilon^2)^{1/2}\right)^2.$$
Thus,
\[ \xi_1 \leq (1 + \epsilon - (2\epsilon + \epsilon^2)^{1/2})^2 \quad \text{or} \quad \xi_1^{1/2} + (2\epsilon + \epsilon^2)^{1/2} \leq 1 + \epsilon. \]
Squaring both sides, we have \(2\xi_1^{1/2}(2\epsilon + \epsilon^2)^{1/2} \leq 1 - \xi_1.\) Using that \(\xi_1\) is a root of (4.11), we find
\[
(4.12) \quad 2\epsilon + \epsilon^2 \leq \frac{1}{4}\delta^2.
\]
Solving this inequality for \(\epsilon\), we obtain
\[ \epsilon \leq -1 + (1 + \frac{1}{4}\delta^2)^{1/2} \leq -1 + (1 + \frac{1}{8}\delta^2) = \frac{1}{8}\delta^2, \]
which proves (4.10).
Second, if \(p/a > a/b\), then \(bp > aq\) and
\[ \frac{p}{q} = \frac{a}{b} (1 + \epsilon + (2\epsilon + \epsilon^2)^{1/2})^2. \]
Then \(\xi_2 \geq (1 + \epsilon + (2\epsilon + \epsilon^2)^{1/2})^2.\) From this, we obtain (4.10). This completes the proof of the theorem.

5. The Program and Results. The program was first written in Algol without use of any recursive procedures. It was planned that after testing the program, it would be transferred over to Fortran IV which has available double precision routines. This transfer was feasible because the computation preserves integers.

A dissection of order \(r\) is given by a sequence \(S_r.\) Therefore, \(r\) must be chosen appropriately. We chose \(r = [0.1 n^{1/3}]\) which is nearly the optimal value. Consequently, we are looking for factors which are greater than \((n/(r + 1))^{1/2} \approx 10^{1/2}n^{1/3}.\) We obtained a Fortran routine which is valid for \(n \leq 1.05 \times 10^{20}\) and which requires at most \(1.4 \times 10^{-4}n^{1/3}\) seconds on the CDC 6400.

Professor René DeVogelaere furnished me with some integers of from 17 to 21 digits which he wished to factor. In Table I, they are given with the results. We give, along with the factor, the resulting \(k\) where \(x^2 - y^2 = 4kn\) and \(x\) and \(y\) are integers. The time is given in seconds for the final version.

In our discussion of the program, we give only the Algol procedures. The first procedure is for finding \(x \equiv a \pmod{b}\) where \(x\) is the least nonnegative residue of \(a\) modulo \(b\) where \(a\) and \(b\) are positive integers. The second is for finding the gcd\((a, b)\) where \(a\) and \(b\) are positive integers. The third is a procedure \(\text{isqrt}(n, u)\) which gives as its value the smallest positive integer \(j\) such that \(j^2 \geq n\) and gives to \(u\) the corresponding value of \(j^2 - n.\) This procedure uses the real procedure \(\text{sqrt}(n)\) hence it may be in error. It is designed to correct this error.

We give the procedure \(\text{factor}(n, r, f)\). We enter the procedure by giving \(n\) and \(r\) and leave it with \(f\) assigned a factor. Also, if no factor has been found, then \(f\) is set to be equal to 1.

In going through the integers \(k\) from 1 to \(r\), there is an advantage in going through them in a prescribed order. Let \(d(k)\) be the number of positive divisors of \(k\). If \(a/b\) is closest to the ratio of the divisors of \(n\), which \(k = ab\) should we try first? As an example we take from Table I the first example
\[ k = 23220 = 2^2 \cdot 3^3 \cdot 5 \cdot 43. \]
Thus, there are \( d(k) = 3 \cdot 4 \cdot 2 \cdot 2 = 48 \) different representations \( a/b \) that we look at simultaneously. Clearly, it is better to first choose \( k \) with \( d(k) \) large. For that reason, we chose to look at multiples of

\[
30 = 2 \cdot 3 \cdot 5, \quad 24 = 2^3 \cdot 3, \quad 12 = 2^2 \cdot 3, \quad 18 = 2 \cdot 3^2, \quad 6 = 2 \cdot 3, 2, 1.
\]

The program is designed to go through these sequences.

We have set the Boolean array \( qr \) so that \( qr[i] \) is true if \( i \) is a quadratic residue modulo \( 729 = 3^6 \) and is false otherwise. We have picked \( 729 \) so that the proportion that is true is only \( 274/729 = 0.38 \). For this proportion, we must do the additional work of finding \( \text{isqrt}(u, t) \).

### Table I

<table>
<thead>
<tr>
<th>Number</th>
<th>Factor</th>
<th>( k )</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1123877887715932507</td>
<td>299155897</td>
<td>( 2^{2} \cdot 3^{3} \cdot 5 \cdot 43 )</td>
<td>2.6</td>
</tr>
<tr>
<td>1129367102454866881</td>
<td>25869889</td>
<td>( 2 \cdot 3^{3} \cdot 5^{3} )</td>
<td>1.3</td>
</tr>
<tr>
<td>29742315699406748437</td>
<td>37217342</td>
<td>( 2 \cdot 11 \cdot 1181 )</td>
<td>122.6</td>
</tr>
<tr>
<td>3524967993119848438</td>
<td>59138501</td>
<td>( 2 \cdot 19 \cdot 383 )</td>
<td>17.8</td>
</tr>
<tr>
<td>208127657340093535</td>
<td>430470917</td>
<td>( 2 \cdot 3 \cdot 5 \cdot 23 \cdot 31 )</td>
<td>1.9</td>
</tr>
<tr>
<td>33143253770013787</td>
<td>114098219</td>
<td>( 2^{3} \cdot 3 \cdot 13 \cdot 47 )</td>
<td>6.0</td>
</tr>
<tr>
<td>3070282504055021783</td>
<td>1436222173</td>
<td>( 2^{2} \cdot 3^{2} \cdot 5 \cdot 13 \cdot 43 )</td>
<td>7.2</td>
</tr>
<tr>
<td>375755062726077891</td>
<td>16053127</td>
<td>( 3^{2} \cdot 7 \cdot 2083 )</td>
<td>175.5</td>
</tr>
<tr>
<td>2492881699809468479</td>
<td>347912923</td>
<td>( 2^{2} \cdot 3 \cdot 13 \cdot 73 )</td>
<td>8.3</td>
</tr>
<tr>
<td>10188337563435517819</td>
<td>70901851</td>
<td>( 2^{6} \cdot 3 \cdot 5 \cdot 19 )</td>
<td>3.0</td>
</tr>
</tbody>
</table>

integer procedure \( \text{mod}(a, b) \); value \( a, b \); integer \( a, b \); \( \text{mod} := a - (a \bmod b) \times b \); integer procedure \( \text{gcd}(a, b) \); value \( a, b \); integer \( a, b \);
begin integer \( i \);
if \( a < b \) then begin \( i := a; a := b; b := i \) end;

\( l: i := \text{mod}(a, b); a := b; b := i; \)
if \( i \neq 0 \) then go to \( l \); \( \text{gcd} := a \)
end \( \text{gcd}; \)
integer procedure \( \text{isqrt}(n, u) \); value \( n \); integer \( n, u \);
begin integer \( j, j1, j2 \);

\( j := \) if \( n = 0 \) then \( 1 \) else \( \text{entier}(\sqrt{n}) + 1 \);

\( j1 := j \times j - n \);

\( f: \) if \( j1 < 0 \) then

begin \( j1 := j1 + 2 \times j + 1 \); \( j := j + 1 \); go to \( f \) end;

\( l: j2 := j1 - 2 \times j + 1 \);
if \( j2 \geq 0 \) then

begin \( j1 := j2 \); \( j := j - 1 \); go to \( l \) end;

\( \text{isqrt} := j; u := j1 \)
end \( \text{isqrt}; \)
procedure factor(n, r, f); value n, r; integer n, r, f;
begin
integer i, j, p;
integer array c[1 : 8];
Boolean array qr[0 : 728];
procedure large(m, m0); value m, m0; integer m, m0;
begin
integer i, il, j, jump, k, s, t, u, x, y; Boolean odd;
s := 1; k := m0;
start:
k := k + c[s]; s := if s = m then 1 else s + 1;
if k \leq r then
begin
    x := isqrt(4 \times k \times n, u); j := (isqrt(n \div k, t) - 1) \div (4 \times (r + 1));
    if mod(x + k, 2) = 0 then
    begin
        il := 1; u := u + 2 \times x + 1; x := x + 1 end else il := 0;
        odd := mod(k, 2) = 1; jump := if odd then 4 else 2;
    if odd then
    begin
        if mod(k + n, 4) = mod(x, 4) then
        begin
            il := il + 2; u := u + 4 \times (x + 1); x := x + 2 end
        end;
    for i := il step jump until j + 1 do
    begin
        if qr[mod(u, 729)] then
        begin
            y := isqrt(u, t);
            if t = 0 then
            begin
                p := gcd(n, x - y); if p > n \div p then p := n \div p;
                go to exit
            end;
            comment When a factor p is found, we leave the
            procedure by going to exit;
        end;
        if odd then begin u := u + 8 \times (x + 2); x := x + 4 end
        else
        begin u := u + 4 \times (x + 1); x := x + 2 end
    end;
    go to start
end large;
for i := 0 step 1 until 728 do qr[i] := false;
for i := 0 step 1 until 364 do
begin
j := mod(i \times i, 729); qr[j] := true end;
c[1] := 30; large(1, 0);
\[ c[1] := 4; \ c[2] := 2; \ \text{large}(2, \ -2); \]
\[ c[1] := 2; \ \text{large}(1, \ -1); \]

comment No factor has been found;

\[ p := 1; \]

\[ \text{exit: } f := p \]

end factor;

Department of Mathematics
University of California
Berkeley, California 94720