Proc. Camb. Phil. Soc. (1974), **76** 521
PCPS 76–54
Printed in Great Britain

Theorems on factorization and primality testing

By J. M. POLLARD

(*Mathematics Department, Plessey Telecommunications Research, Taplow Court, Taplow, Maidenhead, Berkshire*)

(*Received* 8 *April* 1974)

1. *Introduction.* This paper is concerned with the problem of obtaining theoretical estimates for the number of arithmetical operations required to factorize a large integer n or test it for primality. One way of making these problems precise uses a multi-tape Turing machine (e.g. (1), although we require a version with an input tape). At the start of the calculation n is written in radix notation on one of the tapes, and the machine is to stop after writing out the factors in radix notation or after writing one of two symbols denoting 'prime' or 'composite'. There are, of course, other definitions which could be used; but the differences between these are unimportant for our purpose.

The exercise of proving theoretical estimates of this kind differs considerably from that of devising practical algorithms for immediate use. On the one hand, an algorithm devised in order to prove a theoretical result may well be very badly suited to practical use, and indeed there is no shortage of difficulties associated with those described here. On the other hand, in one respect one could claim that the theoretical exercise is the more difficult. Contrary to some inventors of practical algorithms we insist that the algorithm shall always terminate in the stated number of operations, and that a proof shall be given that it does so.

To see the force of this, consider the problem of calculating the least positive quadratic non-residue of a large prime. This is easily done in practice by finding the quadratic character (mod p) of $2, 3, \ldots$ by means of the Law of Quadratic Reciprocity and stopping at the first non-residue. From the practical point of view (2) the number of operations may be said to be $O(p^\epsilon)$ or even $O((\log p)^c)$ for some c; but apparently the best theoretical estimate we can give is $O(p^{b+\epsilon})$ with $b = 1/(4\sqrt{e})$, quoting Burgess's estimate (3) for the least positive non-residue. This particular difficulty concerning the distribution of the quadratic residues has occurred in other algorithms, e.g. Berlekamp (4).

For these reasons it is not easy to know what are, in fact, the best known theoretical estimates for factorization and primality testing. A candidate for both is the estimate $O(n^{\frac{1}{4}+\epsilon})$ given by Shanks (5) by means of the theory of quadratic forms. I am not competent to say whether this does constitute, or can be made to constitute, a strict proof of this estimate. On the other hand, I understand that the method can be improved, at least in the practical sense, to be of order $O(n^{\frac{1}{5}+\epsilon})$.

A recent method of Lehman (6) clearly does achieve the estimate $O(n^{\frac{1}{3}+\epsilon})$ for factorization. Both (5) and (6) are usable practical methods.

My own previous method (7) gives the estimate $O(n^{\frac{1}{4}+\epsilon})$ for primality testing only,

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

and does not have any theoretical difficulties; it *does*, however, have considerable practical difficulties.

The algorithms given here make use of the 'fast Fourier transform' (9); we use also a device from Shanks' method (5) although without mentioning quadratic forms.

THEOREM 1. *Let $0 < \alpha < 1$ and $\delta > 0$. Then there exists a multi-tape Turing machine which can determine whether or not a large integer n has a prime factor not exceeding n^α and, if so, find such a factor in a total of $O(n^{\frac{1}{2}\alpha+\delta})$ operations.*

COROLLARY. *For any $\delta > 0$ there exists a machine which can completely factorize a large integer n in $O(n^{\frac{1}{2}+\delta})$ operations.*

This follows at once by taking $\alpha = \frac{1}{2}$ and repeating the process of the theorem as often as necessary to obtain a complete factorization – not more than $O(\log n)$ times.

THEOREM 2. *For any $\delta > 0$ there exists a multi-tape Turing machine which can test a large integer n for primality in $O(n^{\frac{1}{8}+\delta})$ operations.*

After proving these theorems we will point out a simple practical device for factorization somewhat related to the proof of Theorem 1.

2. *Proof of Theorem* 1. We require the following preliminary results.

LEMMA 1. *There exists a multi-tape Turing machine which, when given any positive integers M and N, and sequences of integers $a_0, a_1, \ldots, a_{N-1}$ and $b_0, b_1, \ldots, b_{N-1}$ of absolute value less than M, can compute all the numbers*

$$c_k = \sum_{i+j=k} a_i b_j \quad (0 \leqslant k \leqslant 2(N-1)),$$

in a number of operations which is $O(N^{1+\epsilon}M^\epsilon)$ for all $\epsilon > 0$.

Proof. It is well-known that this calculation can be performed efficiently by means of the 'fast Fourier transform' algorithm. We can choose between

(i) the original form of the transform in the complex field (8, 9) working to sufficient accuracy to ensure that the integers c_k are found exactly (as in (10)), or

(ii) the transform in the ring of integers (mod m) ((10, 11)).

Taking the second possibility let us put

$$A = \sum_{i=0}^{N-1} a_i D^i \quad \text{and} \quad B = \sum_{i=0}^{N-1} b_i D^i,$$

where $D = M^2N$. Then we can regard the problem of the lemma as simply that of performing a single multiplication of A and B, the required numbers c_k being easily read off from this product. With $C = \max(A, B)$ the multiplication may be performed in $O((\log C)^{1+\epsilon})$ operations according to (10) (this method is also reproduced by Knuth (12)); indeed since this estimate is relatively weak we can use a simplified version of (10). Since $\log C = O(N(\log N + \log M))$ we have the required result.

LEMMA 2. *There exists a multi-tape Turing machine which, when given positive integers a, n and M where $a < n$ and $(a, n) = 1$, can determine whether or not there is an integer m with $1 \leqslant m \leqslant M$ and*

$$(a^m - 1, n) > 1 \tag{1}$$

and, if so, find the least such integer, in a total of $O(M^{\frac{1}{2}+\epsilon}n^\epsilon)$ operations for any $\epsilon > 0$.

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

We remark that, more generally, we could test for the existence of such an m in M consecutive terms of an arithmetical progression in the same number of operations.

Proof. We may assume that $M = L^2$ and $L = 2^k$. The integers $1, 2, \ldots, M$ can be written in the form $m = -u + vL$ $(0 \leqslant u \leqslant L-1,\ 1 \leqslant v \leqslant L)$. Then (1) is equivalent to

$$(a^{vL} - a^u, n) > 1. \tag{2}$$

To test this we first find the least positive residues $(\mathrm{mod}\, n)$ of the coefficients of

$$f(x) = \prod_{u=0}^{L-1} (x - a^u). \tag{3}$$

We compute the least positive residues $(\mathrm{mod}\, n)$, b_u say, of the numbers a^u in (3). Then we multiply pairs of linear factors $(x - b_u)$, then pairs of the quadratic factors obtained, and so on, reducing the coefficients $(\mathrm{mod}\, n)$ at each stage. At the ith stage we multiply $L/2^i$ pairs of polynomials of degree 2^{i-1} requiring, by Lemma 1,

$$O((L/2^i) \cdot 2^{i(1+\epsilon)} \cdot n^\epsilon) = O(L^{1+\epsilon} n^\epsilon)$$

operations, so that the coefficients of $f(x)$ are obtained in $O(L^{1+\epsilon} n^\epsilon)$ operations.

Now we wish to find the solutions, if any, of

$$(f(a^{vL}), n) > 1 \quad (1 \leqslant v \leqslant L). \tag{4}$$

Put $r \equiv a^{\frac{1}{2}L} (\mathrm{mod}\, n)$ and

$$f(x) \equiv \sum_{i=0}^{L} s_i x^i (\mathrm{mod}\, n).$$

Then

$$f(r^{2v}) \cdot r^{L+v^2} \equiv \sum_{i=0}^{L} s_i r^{L-i^2} \cdot r^{(i+v)^2} \pmod{n} \quad (1 \leqslant v \leqslant L). \tag{5}$$

To evaluate this we define sequences a_i and b_j of nonnegative integers less than n by

$$\begin{aligned} a_i &\equiv s_i r^{L-i^2} \pmod{n} && (0 \leqslant i \leqslant L), \\ a_i &= 0, && (L+1 \leqslant i \leqslant 2L), \\ b_j &\equiv r^{(2L-j)^2} \pmod{n} && (0 \leqslant j \leqslant 2L). \end{aligned}$$

Calculation of these sequences takes $O(Ln^\epsilon)$ operations. Now we apply the method of Lemma 1, requiring $O(L^{1+\epsilon} n^\epsilon)$ operations, and obtaining a sequence of numbers c_k of which $c_{2L-1}, c_{2L-2}, \ldots, c_L$ are congruent $(\mathrm{mod}\, n)$ to the values of (5).

The terms r^{L+v^2} in (5) being coprime to n we can calculate at once the required values

$$(f(r^{2v}), n) \quad (1 \leqslant v \leqslant L),$$

by Euclid's algorithm, requiring only $O(Ln^\epsilon)$ operations more.

Thus we find the solutions of (4). If solutions exist we take the least such solution v and find, in a straightforward way, the greatest u for which (2) holds, thus obtaining the least solution of (1).

From the practical point of view our algorithm is now virtually complete. For suppose that there is a prime factor p of n with $p \leqslant n^\alpha$. Then for any a prime to n there is certainly a solution m of (1) with $1 \leqslant m \leqslant M = [n^\alpha]$; indeed $m = p - 1$ will

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

do since then $a^m \equiv 1 \pmod{p}$. If then the integer $y = (a^m - 1, n)$ is less than n we have achieved a factorization of n. The purpose of the rest of the proof is to deal with the (very unlikely) possibility that $y = n$ should hold for many values of a.

We first make the following definition. For $k \geqslant 2$ and p prime, let $s_k(p)$ be the maximum over the integers $q \geqslant k$ with $q|p-1$ of the least positive qth power non-residue of p, or have value 0 if no such q exist.

LEMMA 3. *For any $\delta > 0$ there is an integer $k \geqslant 2$ such that $s_k(p) = O(p^\delta)$.*

Proof. The lemma follows easily from the fact that the number of positive integers less than X all of whose prime factors are less than X^δ is asymptotic to βX for some $\beta > 0$ (13).

Now we can complete the proof of Theorem 1. Put $M = [n^\alpha] + 1$. Test, by straightforward division, whether there is a factor p of n less than $\sqrt{M}$; if so, we are finished, so assume the contrary which implies that the total number of prime factors on n, counted according to their multiplicity, is less than $2/\alpha$. Given $\delta > 0$ choose $k \geqslant 2$ and $C > 0$ such that $s_k(p) < Cp^{\frac{1}{2}\delta}$ for all primes p and put $A = [Cn^{\frac{1}{2}\delta}]$. Since n is large we may assume that $A < \sqrt{M}$.

For each $a = 2, 3, \ldots, A$ in turn, find whether there is a solution of (1) with

$$1 \leqslant m \leqslant M.$$

If for some a there is no solution then the required prime factor of n certainly does not exist (as remarked above). Otherwise let $m_1, m_2, \ldots, m_A$ be the least solutions of (1) and let H be their least common multiple; put

$$y_i = (a^{m_i} - 1, n) \quad (1 \leqslant i \leqslant A).$$

Suppose first that all the y_i are equal to n. Then we prove that any prime factor p of n with $p \leqslant M$ satisfies

$$p - 1 = zH \tag{6}$$

for some z with $1 \leqslant z \leqslant k-1$. Since

$$a^{m_i} \equiv 1 \pmod{p}, \quad (1 \leqslant i \leqslant A),$$

and m_i is minimal with this property, we have $m_i|p-1$ for each i and so $H|p-1$. Put $z = (p-1)/H$ and suppose that $z \geqslant k$. Then the least zth power non-residue, r say, of p does not exceed A, so that for some m with $1 \leqslant m \leqslant M$ and $m|H$,

$$r^m \equiv 1 \pmod{p}. \tag{7}$$

There is a prime factor q of z such that r is a q^ath power non-residue of p where $q^a \| z$. Then if $q^b \| p-1$, (7) implies $q^{b+1-a}|m|H$ whereas in fact $q^{b-a} \| H$. This contradiction proves that $z \leqslant k-1$ in (6). We test whether any of the numbers p defined by (6) with $z \leqslant k-1$ are factors of n.

Now suppose either that the numbers (6) do give a factor, or that contrary to our earlier assumption at least one of the y_i is less than n, and is therefore a non-trivial factor of n. Then perhaps we have achieved our object, which was to find a prime factor not exceeding M. But if not then we restart the algorithm using one of our factors of n in place of n itself, but keeping the same values for M and A, and continue

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

in this manner until there are no more factors left to investigate. The number of times that we restart is $O(1)$ and so our total number of operations is $O(M^{\frac{1}{2}+\epsilon}n^{\epsilon}A) = O(n^{\frac{1}{2}\alpha+\delta})$ as required.

3. *Proof of Theorem* 2. The following is a more complicated version of Theorem 1 of (7) suggested by Theorem 2 of (14).

LEMMA 4. *Let $k \geqslant 2$ and let $f(x)$ be an increasing function of x such that for all primes p we have $s_k(p) < f(p)$. Put $C = \{3(k-1)^3\}^{\frac{1}{4}}$. Then an integer $n > 9C^2$ is composite if and only if at least one of the following is true*:

(i) *there is an integer m for which $2 \leqslant m \leqslant \min(n-1, Cn^{\frac{1}{4}})$ and $m|n$,*

(ii) *there is an integer l for which $2 \leqslant l \leqslant \min(n-1, f(C^{-1}n^{\frac{3}{4}}))$ and*

$$l^{n-1} \not\equiv 1 \pmod{n},$$

(iii) *there are positive integers m, z and l with $m \leqslant z \leqslant k-1$, $m|z(l-1)$ and*

$$n = l\left(1 + \frac{z}{m}(l-1)\right),$$

(iv) *there are positive integers r_1, r_2, r_3 and d such that*

$$r_i \leqslant Cn^{\frac{1}{4}} \quad \text{and} \quad r_i|(n-1), \quad (i = 1, 2, 3),$$

and

$$(dr_1+1)(dr_2+1)(dr_3+1) = n,$$

(v) *there are positive integers r_1, d and z_1 with $z_1 \leqslant k-1$ such that*

$$r_1 \leqslant Cn^{\frac{1}{4}} + C^4, \quad d \leqslant C^{-3}n^{\frac{1}{4}}, \quad dr_1|z_1(n-1) \quad \text{and} \quad (dr_1+1)|n.$$

Proof. A prime n cannot satisfy (i), (ii), (iii) or (iv), and if it satisfied (v) then

$$n = dr_1 + 1 \leqslant (Cn^{\frac{1}{4}} + C^4).C^{-3}n^{\frac{1}{4}} + 1 < 3Cn^{\frac{1}{2}}$$

contradicting $n > 9C^2$. We suppose that a composite n does not satisfy (i) or (ii) and show that one of the other conditions must hold. We have either

$$n = p_1p_2, \qquad Cn^{\frac{1}{4}} < p_1 \leqslant p_2 < C^{-1}n^{\frac{3}{4}}, \tag{8}$$

or

$$n = p_1p_2p_3, \quad Cn^{\frac{1}{4}} < p_1 \leqslant p_2 \leqslant p_3 < C^{-2}n^{\frac{1}{2}}, \tag{9}$$

where p_1, p_2 and p_3 are primes.

In the first case, when (8) holds, we shall prove that

$$p_2 - 1|z(n-1), \tag{10}$$

where $1 \leqslant z \leqslant k-1$. Define z as the least positive integer for which (10) holds and suppose that $z \geqslant k$. Then the least zth power non-residue, r say, of p_2 satisfies

$$r \leqslant s_k(p_2) < f(p_2) < f(C^{-1}n^{\frac{3}{4}})$$

so that by (ii)

$$r^{n-1} \equiv 1 \pmod{p_1p_2}. \tag{11}$$

There is some prime factor q of z such that r is a q^ath power non-residue of p_2 where $q^a \| z$. Then if $q^b \| p_2 - 1$ (so $b \geqslant a$), (11) implies $q^{b-a+1}|n-1$ whereas by definition of z, $q^{b-a} \| n-1$. This contradiction proves that $z \leqslant k-1$.

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

Now (10) implies that $p_2-1|z(p_1-1)$. Putting $z(p_1-1)=m(p_2-1)$ we have $1\leqslant m\leqslant z\leqslant k-1$ since $p_1\leqslant p_2$, and also

$$n=p_1p_2=p_1\left(1+\frac{z}{m}(p_1-1)\right)$$

so that (iii) holds.

In the second case, when (9) holds, we obtain similarly

$$p_i-1|z_i(p_1p_2p_3-1)$$

where $1\leqslant z_i\leqslant k-1$, $(i=1,2,3)$. Putting $q_i=p_i-1$ this becomes

$$q_i|z_i(q_1q_2q_3+q_2q_3+q_3q_1+q_1q_2+q_1+q_2+q_3)\quad(i=1,2,3), \tag{12}$$

whence

$$(q_1,q_2)|(z_1,z_2)\,q_3,\quad (q_2,q_3)|(z_2,z_3)\,q_1,\quad (q_3,q_1)|(z_3,z_1)\,q_2. \tag{13}$$

Now put $d=(q_1,q_2,q_3)$, $e=(z_1,z_2,z_3)$, $q_i=dr_i$, $z_i=ey_i$, $(i=1,2,3)$; then by (13)

$$(r_1,r_2)|e(y_1,y_2),\quad (r_2,r_3)|e(y_2,y_3),\quad (r_3,r_1)|e(y_3,y_1),$$

and hence

$$[r_1,r_2,r_3]=\frac{r_1r_2r_3}{(r_1,r_2)\times(r_2,r_3)\times(r_3,r_1)}\geqslant\frac{r_1r_2r_3}{e(y_1,y_2)\times(y_2,y_3)\times(y_3,y_1)}. \tag{14}$$

But from (12)

$$[r_1,r_2,r_3]\,|\,[z_1,z_2,z_3]\{d(r_2r_3+r_3r_1+r_1r_2)+(r_1+r_2+r_3)\}$$

$$=\frac{ey_1y_2y_3}{(y_1,y_2)\times(y_2,y_3)\times(y_3,y_1)}.\{d(r_2r_3+r_3r_1+r_1r_2)+(r_1+r_2+r_3)\}$$

and from this and (14),

$$r_1r_2r_3\leqslant e^2y_1y_2y_3\{d(r_2r_3+r_3r_1+r_1r_2)+(r_1+r_2+r_3)\}\leqslant(k-1)^3.3(d+1)\,r_2r_3,$$

so that

$$r_1\leqslant 3(k-1)^3.(d+1)=C^4(d+1).$$

Now $dr_3\leqslant p_3\leqslant C^{-2}n^{\frac{1}{2}}$ and so either $r_3\leqslant Cn^{\frac{1}{4}}$ giving condition (iv) or $d\leqslant C^{-3}n^{\frac{1}{4}}$ giving condition (v).

Now we complete the proof of Theorem 2. By Lemma 3 we choose $k\geqslant 2$ so that $s_k(p)=O(p^{\frac{1}{3}})$, say $s_k(p)<Bp^{\frac{1}{3}}$ for all p. Choosing $f(x)=Bx^{\frac{1}{3}}$ in Lemma 4, we can test condition (ii) of the lemma in $O(n^{\frac{1}{3}+\epsilon})$ operations since each value of l requires only $O(n^{\epsilon})$ operations (just as in (7)). Fix $\delta>0$ and define C as in Lemma 4. By Theorem 1 we can find all the prime factors of n and $(k-1)!\times(n-1)$ which do not exceed $Cn^{\frac{1}{4}}+C^4$ in $O(n^{\frac{1}{3}+\epsilon})$ operations. Then, assuming that we did not find a factor of n, we test the remaining conditions (iii), (iv) and (v) of Lemma 4 in $O(n^{\epsilon})$ operations as in (7).

4. *A remark on practical factorization.* We will describe a simple device having some relation to the method of Theorem 1 and which is immediately usable.

We are attempting to factorize an integer n known to be composite. Choose integers L and M with $1<L<M<n^{\frac{1}{4}}$ and $M<L^2$. Our method will find only those prime factors q of n (if any) for which $q-1$ has one of the forms

$$q-1=A\quad\text{or}\quad q-1=Ap \tag{15}$$

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

where A is a product of primes not exceeding L and p is a single prime with $L < p < M$. Further, as will become clear, it will not in exceptional cases split 2 factors q_1 and q_2 both of this form. The method is best suited to numbers of such forms as $n = 2^m \pm k$ (k small) for which reduction modulo n is particularly simple.

Choose some small integer $a > 1$. Our algorithm has two stages. But if we know in advance that the prime factors satisfy $q \equiv 1 \pmod{k}$, with some $k > 2$, then it is worthwhile to begin by replacing a by $a^k \pmod{n}$.

Step 1. Compute the integer b defined by

$$b \equiv a^P \pmod{n}, \quad (0 < b < n), \tag{16}$$

where P is the product of all the primes $p_i \leqslant L$ each to some power $c_i \geqslant 1$. There is some freedom in the choice of the c_i but the smallest primes should certainly occur to some power higher than the first. In (16) we employ the familiar power algorithm using the binary representation of the exponent (2) and can choose between using the primes p_i in succession or computing P in advance and performing a single power operation. Next we compute the g.c.d. $d = (b-1, n)$. If $1 < d < n$ we have found a factor of n, namely d. If $d = n$ we repeat Step 1 with a smaller value of L. If $d = 1$ we proceed to Step 2.

Step 2. Now we wish to compute the sequence of numbers

$$F_m \equiv b^m - 1 \pmod{n}, \tag{17}$$

for m prime in the range $L < m < M$, and hope to find an m with

$$g_m = (F_m, n) > 1. \tag{18}$$

The simplest way is to compute (17) for all odd integers m by successive multiplications of b^m by $b^2 \pmod{n}$. To avoid the work of computing the g.c.d. in (18) for each m, we accumulate the product $\pmod{n}$ of the F_m and at regular intervals compute the g.c.d. of this product and n, obtaining some sequence G_j ($j = 1, 2, \ldots$). If $1 < G_j < n$ then we have found a factor. If $G_j = n$ and $G_{j-1} = 1$ then we go back and compute the numbers g_m in the relevant range individually.

To restrict m in (17) to the prime values we compute in advance a table of the residues

$$b^2, b^4, \ldots, b^{2D} \pmod{n}$$

up to the largest occurring difference of two consecutive primes $p_{k+1} - p_k = 2D$ $p_{k+1} \leqslant M$. By consulting this table we can compute the F_m, m prime, in succession; we accumulate their product $\pmod{n}$ as before.

We give a few numerical examples. A difficult factorization from (14) is

$$2^{107} + 2^{54} + 1 = 843589 \cdot 8174912477117 \cdot 23528569104401 \quad (= p_1 \cdot p_2 \cdot p_3, \text{ say}).$$

Since $p_3 - 1 = 2^4 . 5^2 . 67 . 107 . 199 . 41231$ this is accessible by our method if $L \geqslant 199$ and $M \geqslant 41231$. Particularly favourable to our method is (also from (14)) $2^{150} + 2^{75} + 1$ for which we need only Step 1 with $L \geqslant 61$. For an extreme example let $p_1 = 27! + 1$, which is prime (15) and has 29 digits, and let p_2 be another prime of similar magnitude. Then the product $p_1 p_2$ is easily factorized. Equally, one could give numbers of, say, 15 digits which would cause great trouble.

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press

This device seems most likely to be useful for searching for fairly small factors of very large composite numbers – by which we mean factors a few digits larger than those which can be found by trial division.

Of course this algorithm is only of order $O(n^{\frac{1}{2}+\epsilon})$ in the worst cases; for a more sophisticated algorithm we would theoretically use the fast Fourier transform on Step 2, reducing this to $O(M^{\frac{1}{2}}n^{\epsilon})$ operations and regaining an algorithm of order $O(n^{\frac{1}{4}+\epsilon})$. I cannot say at what point this becomes a practical proposition.

REFERENCES

(1) HARTMANIS, J. and STEARNS, R. E. On the computational complexity of algorithms. *Trans. Amer. Math. Soc.* **117** (1965), 285–306.

(2) LEHMER, D. H. Computer technology applied to the theory of numbers. *Studies in Number Theory* (W. J. LeVeque, Editor) (Prentice-Hall, Englewood Cliffs, N.J. 1969), 117–151.

(3) BURGESS, D. A. On character sums and primitive roots. *Proc. London Math. Soc.* **12** (1962), 179–192.

(4) BERLEKAMP, E. R. Factoring polynomials over large finite fields. *Math. Comp.* **24** (1970), 713–735.

(5) SHANKS, D. Class number, a theory of factorisation, and genera. 1969 *Number Theory Institute, Proc. Sympos. Pure Math.* **20**, Amer. Math. Soc., Providence, Rhode Island (1970), 415–440.

(6) LEHMAN, R. S. Factoring large integers. To appear in *Math. Comp.*

(7) POLLARD, J. M. An algorithm for testing the primality of any integer. *Bull. London Math. Soc.* **3** (1971), 337–340.

(8) GOOD, I. J. The interaction algorithm and practical Fourier analysis. *J. Roy. Statist. Soc. Ser. B* **20** (1958), 361–372.

(9) COOLEY, J. W. and TUKEY, J. W. An algorithm for the machine computation of complex Fourier series. *Math. Comp.* **19** (1965), 297–301.

(10) SCHONNAGE, A. and STRASSEN, V. Schnelle Multiplikation grosser Zahlen. *Computing* (Arch. Elektron Rechnen), **7** (1971), 281–292.

(11) POLLARD, J. M. The fast Fourier transform in a finite field. *Math. Comp.* **25** (1971), 365–374.

(12) KNUTH, D. E. *The art of computer programming*, Volume 2 (revised edition 1971), *Semi-numerical algorithms*. Addison-Wesley, New York, 1971.

(13) NORTON, KARL K. Numbers with small prime factors and the least kth power non-residue. *Memoirs of the Amer. Math. Soc.* **106**, Amer. Math. Soc., Providence, Rhode Island, 1971.

(14) BRILLHART, JOHN and SELFRIDGE, J. L. Some factorizations of $2^n \pm 1$ and related results. *Math. Comp.* **21** (1967), 87–96.

(15) BORNING, ALAN. Some results for $k! \pm 1$ and $2.3.5...p \pm 1$. *Math. Comp.* **26** (1972), 567–570.

https://doi.org/10.1017/S0305004100049252 Published online by Cambridge University Press