# On Primes Recognizable in Deterministic Polynomial Time

Sergei Konyagin[1]* and Carl Pomerance[2]**

[1] Department of Mechanics and Mathematics, Moscow State University, 119899 Moscow, Russia
[2] Department of Mathematics, University of Georgia, Athens, Georgia 30602, U.S.A.

For Paul Erdős on his eightieth birthday

## 1. Introduction

In this paper we present several algorithms that can find proofs of primality in deterministic polynomial time for some primes. In particular we show this for any prime $p$ for which the complete prime factorization of $p-1$ is given. We can also show this when a completely factored divisor of $p-1$ is given that exceeds $p^{1/4+\epsilon}$. And we can show this if $p-1$ has a factor $F$ exceeding $p^{\epsilon}$ with the property that every prime factor of $F$ is at most $(\log p)^{2/\epsilon}$. Finally, we present a deterministic polynomial time algorithm that will prove prime more than $x^{1-\epsilon}$ primes up to $x$. The key tool we use is the idea of a smooth number, that is, a number with only small prime factors. We show an inequality for their distribution that perhaps has independent interest.

It is known that if one assumes the Riemann hypothesis for Dirichlet $L$-functions, then the prime recognition problem is in the complexity class $P$. Thus, from Miller and Bach we know that for every odd composite number $n$ there is some integer $a$ in the range $1 < a < \min\{n, 2(\log n)^2\}$ such that $n$ is not a strong probable prime to the base $a$, and so $n$ is proved composite. If an odd number $n$ is a strong probable prime to every base $a$ in the above range, then $n$ is prime. Thus assuming the above extended Riemann hypothesis, every prime $p$ can be deterministically supplied with a proof of its primality in $O((\log p)^3)$ arithmetic steps with integers at most $p$.

The results in this paper do not rely on the truth of any unproved hypotheses.

It has been known since Lucas that it is easy to find a proof of primality for a prime $p$ if the complete factorization of $p-1$ is known. Indeed one merely has to present a primitive root for $p$ and prove it is one using the prime factorization of $p-1$. Though we know no fast deterministic algorithm for finding a primitive root for a prime $p$, the probabilistic method of just choosing random integers until a primitive root is found works very well in

* Supported in part by the Cultural Initiative Fund and the Russian Academy of Natural Sciences
** Supported in part by an NSF grant

In: The Mathematics of Paul Erdős I, R.L. Graham and J. Nešetřil, eds., Springer-Verlag, Berlin Heidelberg, 1997, pp. 176 - 198.

practice. The expected number of tries is $O(\log\log p)$. In fact, one can show that the expected number of random choices to find a set of numbers which generate $(\mathbb{Z}/p\mathbb{Z})^*$ as a group is $O(1)$. (Note also that it is a simple matter to deterministically fashion a primitive root out of a set of generators with knowledge of the complete prime factorization of $p-1$.) Our algorithm requires $O((\log p)^{10/7})$ tries, and does not guarantee that it will find a primitive root or a set of generators, but it does prove primality and it is deterministic.

It is also known (see Brillhart, Lehmer, Selfridge [5]) that if one has a fully factored divisor $F$ of $p-1$, where $F > p^{1/3}$, then one can quickly decide if $p$ is prime or composite. Again, this involves choosing numbers at random. We show how the prime or composite nature of $p$ can be decided deterministically and in polynomial time. In addition, we only require $F > p^{1/4+\epsilon}$. In another algorithm we only need $F > p^{\epsilon}$, but for the method to be fast, $F$ must be smooth.

While many of the algorithms in this paper are only of theoretical interest, it is likely that at least some of the ideas have practical value. In particular, an algorithm we present below which allows one to decide whether $n$ is prime or composite, when it is known that all prime factors of $n$ are 1 mod $F$ with $F \geq n^{3/10}$, should be a practical addition to the Brillhart, Lehmer, Selfridge "$n-1$ test".

It is to be expected that some of the ideas presented here would be of use in the "$n+1$ test" and the combined "$n^2-1$ test". These elementary tests are often used in conjunction with the Jacobi sums test (see [3] and references there), and it is possible that a few ideas presented here will be of use in that context as well. However, as stated above, our primary emphasis in this paper is theoretical and not practical.

Let $\psi(x,y)$ denote the number of integers $n \leq x$ free of prime factors exceeding $y$. In [9], Erdös and van Lint show that in some sense $\psi(x,y)$ can be approximated by the binomial coefficient $\binom{\pi(y)+[u]}{[u]}$ where $u = \log x/\log y$ and $\pi(y)$ denotes the number of primes not exceeding $y$. In fact an elementary combinatorial argument shows that $\psi(x,y) \geq \binom{\pi(y)+[u]}{[u]}$, so one side of the approximation is easy. In this paper we obtain the lower bound $x/(\log x)^u = x^{1-\log\log x/\log y}$, which is valid whenever $x \geq 4$ and $2 \leq y \leq x$. This lower bound for $\psi(x,y)$ is attractive for its simplicity and near universality. However, one should note that it is a good approximation to $\psi(x,y)$ only in the range $(\log x)^{1+\epsilon} \leq y \leq \exp((\log x)^{\epsilon})$. The inequality in the special case $y = (\log x)^2$ was previously established by Lenstra [12], and for a similar purpose.

Our main idea in this paper is to build up a large subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ using a small set of generators. Specifically, if $p$ is the least prime factor of $n$ and $a$ is an integer with $1 < a < p$, let $\mathcal{G}_n(a)$ denote the subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ generated by $j$ mod $n$ for $j = 2, 3, \ldots, a$. From the above estimate for $\psi(x,y)$, we have

$$\#\mathcal{G}_n([(\log n)^c]) \geq \psi(n, (\log n)^c) > n^{1-1/c},$$

whenever $n \geq 4$ and $2 \leq (\log n)^c < p$, with $p$ the least prime factor of $n$. Thus we can create an "exponentially large" subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ with a "polynomially sized" set of generators. The idea of using smooth number estimates to show that one has built up a large subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$ for $p$ prime was first used in 1926 by Vinogradov [19] to estimate the least positive residue mod $p$ that is not a $k$-th power.

It was previously shown by Pintz, Steiger and Szemerédi [14] that there are infinitely many primes $p$ that can be proved prime in deterministic polynomial time. They require for their primes $p$ that $p-1$ has a divisor which is a power of 3 and exceeds $p^{1/3}$. Thus they could only show there are more than $x^{2/3-\epsilon}$ such primes up to $x$. As mentioned above, we replace the "2/3" with 1.

Our result in Section 3 on deciding if $n$ is prime or composite in deterministic polynomial time, when the complete prime factorization of $n-1$ is given, was anticipated by Fellows and Koblitz [10], though their algorithm is not as fast as ours.

We mention a few other results that are somewhat relevant. Adleman and Huang [1] have given a probabilistic algorithm for primality proving that has expected polynomial time. Much earlier, Solovay and Strassen [18] had given a probabilistic algorithm for compositeness proving that has expected polynomial time. In [15], the second author showed that for every prime $p$ there is a proof that $p$ is prime that can be verified in $O(\log p)$ arithmetic steps with integers at most $p$. Previously, Pratt [16] had shown via Lucas's test the existence of a primality proof that requires $O((\log p)^2)$ arithmetic steps. These two papers show only the existence of these proofs; they do not show how to find them quickly.

We wish to thank W. R. Alford, Ronald Burthe, Andrew Granville, Hendrik Lenstra and Jeff Shallit for some helpful remarks.

## 2. A lower bound for the distribution of smooth numbers

We say an integer $n$ is $y$-smooth if no prime factor of $n$ exceeds $y$. Let $\psi(x,y)$ denote the number of integers $n$ in $[1,x]$ that are $y$-smooth. In this section we are going to prove the following theorem.

**Theorem 2.1.** *If $x \geq 4$ and $2 \leq y \leq x$, then $\psi(x,y) > x^{1-\log\log x/\log y}$.*

We begin with a few lemmas. Let $\pi(x)$ denote the number of primes $p$ with $p \leq x$.

**Lemma 2.1.** *For $x \geq 37$ we have $\pi(x) - \pi(x^{1/2}) > (7/9)x/\log x$.*

This lemma follows from Rosser and Schoenfeld [17, Theorem 1] and a simple calculation. The next lemma is well known; we give the proof for completeness.

**Lemma 2.2.** *Let $p_k$ denote the $k^{th}$ prime. For $x \geq 1$ we have*

$$\psi(x, p_k) > \frac{(\log x)^k}{k!} \prod_{j=1}^{k} \frac{1}{\log p_j}.$$

*Proof.* An integer $n \leq x$ which is $p_k$-smooth has its prime factorization in the form $p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$ where $a_1, a_2, \dots, a_k$ are non-negative integers and $\sum a_j \log p_j \leq \log x$. Thus $\psi(x, p_k)$ is the number of lattice points $(a_1, a_2, \dots, a_k) \in \mathbb{Z}^k$ with each $a_j \geq 0$ and $\sum a_j \log p_j \leq \log x$. Putting each such lattice point at the "lower left" corner of a unit cube with edges parallel to the axes, a region is described which is strictly larger than the simplex

$$\left\{ (y_1, \dots, y_k) \in \mathbb{R}^k : \text{ each } y_j \geq 0, \sum_{j=1}^{k} y_j \log p_j \leq \log x \right\}.$$

Thus $\psi(x, p_k)$ exceeds the $k$-dimensional volume of this simplex, which is

$$(\log x)^k (k!)^{-1} \prod_{j=1}^{k} (\log p_j)^{-1}.$$

*Proof (of Theorem 2.1).* We verify the theorem directly for pairs $x$, $y$ with $2 \leq y < 37$ and $x < 120$. Assume now that $2 \leq y < 37$ and $x \geq 120$. Since the theorem is trivial when $\log y < \log \log x$, we may assume $y \geq 3$. It is not hard to show that

$$x^{1 - \log\log x / \log 37} < \frac{(\log x)^2}{\log 3 \log 4} \tag{2.1}$$

for $x \geq 120$. But the left side of (2.1) is greater than $x^{1 - \log\log x / \log y}$ and the right side of (2.1) is less than $\psi(x, 3)$ by Lemma 2.2. Since $\psi(x, 3) \leq \psi(x, y)$, the theorem holds in this range.

Now assume $37 \leq y \leq x$. Let $u = \log x / \log y$ and let $\{u\} = u - [u]$ denote the fractional part of $u$. If $m$ is a positive integer with $m \leq y^{\{u\}}$ and $n$ is a product of $[u]$ not necessarily distinct primes in the interval $(y^{1/2}, y]$, then $N = mn$ is $y$-smooth and $N \leq y^{\{u\}} y^{[u]} = y^u = x$. Moreover, since $m$ has at most one prime factor in $(y^{1/2}, y]$, it follows that the number of representations of $N$ as a product $mn$ in this way is at most $[u]+1$. In fact, if $\{u\} \leq 1/2$, then $N$ has at most one representation as $mn$. We conclude that

$$\psi(x, y) \geq \begin{cases} \left[y^{\{u\}}\right] (\pi(y) - \pi(y^{1/2}))^{[u]} / ([u]+1)!, & \text{if } \{u\} > 1/2 \\ \left[y^{\{u\}}\right] (\pi(y) - \pi(y^{1/2}))^{[u]} / [u]!, & \text{if } \{u\} \leq 1/2. \end{cases} \tag{2.2}$$

Note that using $y \geq 37$, we have

$$\left[y^{\{u\}}\right] > \begin{cases} \frac{6}{7} y^{\{u\}}, & \text{if } \{u\} > 1/2 \\ \frac{1}{2} y^{\{u\}}, & \text{if } \{u\} \leq 1/2. \end{cases} \tag{2.3}$$

Also, from Lemma 2.1, we have

$$(\pi(y) - \pi(y^{1/2}))^{[u]} > \left(\frac{7}{9} \cdot \frac{y}{\log y}\right)^{[u]} = \left(\frac{7u}{9}\right)^{[u]} \frac{y^{[u]}}{(\log x)^{[u]}}.$$

Thus

$$\begin{aligned} y^{\{u\}}(\pi(y) - \pi(y^{1/2}))^{[u]} &> \left(\frac{7u}{9}\right)^{[u]} (\log x)^{\{u\}} \frac{y^u}{(\log x)^u} \\ &= \left(\frac{7u}{9}\right)^{[u]} (\log x)^{\{u\}} x^{1-\log\log x/\log y}. \end{aligned}$$

Using this inequality with (2.2) and (2.3) we have that the theorem will hold if we show

$$\left(\frac{7u}{9}\right)^{[u]} (\log x)^{\{u\}} \geq \begin{cases} \frac{7}{6}([u]+1)!, & \text{if } \{u\} > 1/2 \\ 2[u]! & \text{if } \{u\} \leq 1/2. \end{cases} \tag{2.4}$$

We now show

$$\left(\frac{7k}{9}\right)^k > 2(k+1)! \text{ for every integer } k \geq 6. \tag{2.5}$$

This holds by inspection for $k = 6, 7, 8, 9$. For any non-negative integer $k$, the arithmetic-geometric mean inequality implies that

$$\left(\frac{k+2}{2}\right)^{k+1} \geq (k+1)!.$$

Using this and the easily verified inequality

$$\left(\frac{7k}{9}\right)^k > 2\left(\frac{k+2}{2}\right)^{k+1} \text{ for } k \geq 10,$$

we have (2.5). Note that (2.5) implies (2.4) when $u \geq 6$.

Suppose that $3 \leq u < 6$ and $\{u\} \leq 1/2$. We verify for $k = 3, 4, 5$ that

$$\left(\frac{7k}{9}\right)^k > 2k!,$$

so that (2.4) holds for these values of $u$.

Suppose now that $2.5 < u < 6$ and $\{u\} > 1/2$. We verify for $k = 2, 3, 4, 5$ that

$$\left(\frac{7(k+1/2)}{9}\right)^k \left(\log(37^{k+1/2})\right)^{1/2} > \frac{7}{6}(k+1)!,$$

so that (2.4) holds for these values of $u$. (We use that $x = y^u \geq 37^u$.)

Now suppose $2.2 \leq u \leq 2.5$. We have

$$\left(\frac{7u}{9}\right)^{[u]}(\log x)^{\{u\}} \ge \left(\frac{7(2.2)}{9}\right)^{2}(\log(37^{2.2}))^{0.2} > 4 = 2[u]!,$$

so the theorem holds here too.

In the range $1 \le u < 2.2$ we use another estimate for $\psi(x,y)$. The number of integers up to $x$ divisible by a prime $p$ is $[x/p]$. Thus

$$\psi(x,y) \ge [x] - \sum_{y<p\le x}\left[\frac{x}{p}\right] > x - 1 - x\sum_{y<p\le x}\frac{1}{p} \tag{2.6}$$

where $p$ runs over primes.

First assume that $1.6 \le u < 2.2$. Then $x \ge 37^{1.6} > 286$. It follows from Theorem 5 in Rosser and Schoenfeld [17] that

$$\begin{aligned}\sum_{y<p\le x}\frac{1}{p} &< \log\log x - \log\log y + \frac{1}{2(\log x)^2} + \frac{1}{2(\log y)^2}\\ &\le \log 2.2 + \frac{1}{2(\log(37^{1.6}))^2} + \frac{1}{2(\log 37)^2} < 0.85.\end{aligned}$$

Thus from (2.6) we have

$$\psi(x,y) > 0.15x - 1 > 0.14x.$$

But

$$x^{1-\log\log x/\log y} = \frac{x}{(\log x)^u} \le \frac{x}{(\log(37^{1.6}))^{1.6}} < 0.07x,$$

so the theorem holds in this range.

Finally assume $1 \le u < 1.6$. Then from Theorem 5 and its Corollary in [17] we have that

$$\begin{aligned}\sum_{y<p\le x}\frac{1}{p} &< \log\log x - \log\log y + \frac{1}{(\log x)^2} + \frac{1}{2(\log y)^2}\\ &\le \log 1.6 + \frac{1}{(\log 37)^2} + \frac{1}{2(\log 37)^2} < 0.59,\end{aligned}$$

so that from (2.6) we have

$$\psi(x,y) > 0.41x - 1 > 0.38x.$$

But

$$x^{1-\log\log x/\log y} \le \frac{x}{\log 37} < 0.28x,$$

so we have the theorem here as well. This concludes the proof of Theorem 2.1.

## 3. When $n-1$ is fully factored

In this section we present and analyze two deterministic algorithms that will decide if a positive integer $n$ is prime or composite when the complete prime factorization of $n-1$ is known. The first algorithm uses the Brillhart, Lehmer, Selfridge "$n-1$ test" (see [5]). The second algorithm is somewhat faster and uses a new result presented below.

We begin with a factorization algorithm that is very fast, but unfortunately is usually unsuccessful in factoring composite numbers.

*The base $B$ factorization method.* We are input integers $n, B$ with $n > B \geq 2$. This algorithm attempts to find a nontrivial factorization of $n$.

**Step 1** Write $n$ in the base $B$ : $n = c_d B^d + c_{d-1}B^{d-1} + \cdots + c_0$, where $c_0, \cdots, c_d$ are integers in the interval $[0, B-1]$ and $c_d > 0$.

**Step 2** Compute $c = \gcd(c_0, \cdots, c_d)$. If $c > 1$, return $c$ as a proper factor of $n$ and stop.

**Step 3** Factor $f(x) = c_d x^d + \cdots + c_0$ into irreducible polynomials in $\mathbb{Z}[x]$ with the algorithm of [11].

**Step 4** If $f(x)$ is irreducible in $\mathbb{Z}[x]$, the algorithm has failed, so stop. If $f(x) = g_1(x)g_2(x)\cdots g_k(x)$ where each $g_i(x)$ is irreducible in $\mathbb{Z}[x]$, then return $g_1(B)g_2(B)\cdots g_k(B)$ as a nontrivial factorization of $n$ and stop.

That each $g_i(B)$ is a proper factor of $n$ in Step 4 follows from [4]. Thus the algorithm is correct. From the analysis in [11], it follows that the running time of the algorithm is $(\log n)^{O(1)}$.

We shall only be applying the base $B$ factorization method in the cases $d = 2, 3$ and in these cases it should be considered "overkill" to use the algorithm of [11] to factor $f(x)$ in Step 3. In particular, if $d = 2$, then $c_2x^2 + c_1x + c_0$ factors if and only if $c_1^2 - 4c_0c_2$ is a square, in which case it is trivial to write down the factorization. Further, it is easy to detect squares and take square roots of squares with a binary search. Thus the time for Step 3 in the case $d = 2$ is $O(\log n)$ arithmetic steps with integers at most $n$. (Newton's method is even better than a binary search; its complexity is $O(\log\log n)$ arithmetic steps with integers at most $n$.)

When $d = 3$ we can again use a binary search in Step 3. In particular, $f(x)$ factors if and only if it has a rational root, and if one rational root is found, we can reduce the problem to the quadratic case. It is more convenient to replace $f(x)$ with $c_3^2 f(x) = g(c_3x)$, since $g(x)$ factors if and only if it has an integer root. However, every integer root of $g$ divides $g(0)$, so if $g(0) \neq 0$, then every integer root is in the interval $[-|g(0)|, |g(0)|]$ and may be located with essentially a binary search. Thus again Step 3 can be accomplished in $O(\log n)$ arithmetic steps with integers at most $n$. (Note that Newton's method could be applied here as well.)

We now describe an algorithm based on the Brillhart, Lehmer, Selfridge $n-1$ test.

**Algorithm 3.1.** *We are input an integer $n > 4$ and the complete prime factorization of $n-1$. This deterministic algorithm decides if $n$ is prime or composite.*

*Let $F(1) = 1$. For $a = 2, 3, \ldots, [(\log n)^{3/2}]$ do the following:*

**Step 1** *If $a$ is composite, let $F(a) = F(a-1)$ and go to Step 7. If $a^{F(a-1)} \equiv 1 \bmod n$, let $F(a) = F(a-1)$ and go to Step 7. Verify that $a^{n-1} \equiv 1 \bmod n$. If not, declare $n$ composite and stop.*

**Step 2** *Using the prime factorization of $n-1$, find the least positive divisor $E(a)$ of $n-1$ with $a^{E(a)} \equiv 1 \bmod n$.*

**Step 3** *Verify that $(a^{E(a)/q} - 1, n) = 1$ for each prime factor $q$ of $E(a)$. If not, declare $n$ composite and stop.*

**Step 4** *Let $F(a) = \operatorname{lcm}\{F(a-1), E(a)\}$. Compute $F(a)$.*

**Step 5** *If $F(a) \geq n^{1/2}$, declare $n$ prime and stop.*

**Step 6** *If $n^{1/3} \leq F(a) < n^{1/2}$, attempt to factor $n$ by the base $F(a)$ factorization method. If $n$ is factored nontrivially, declare $n$ composite and stop. If $n$ is not factored, declare $n$ prime and stop.*

**Step 7** *If $a < [(\log n)^{3/2}]$, get the next $a$. Otherwise declare $n$ composite and stop.*

*Proof (of correctness).* Since $(\log n)^{3/2} < n$ for every integer $n > 1$, Step 1 is correct by Fermat's little theorem. It is clear that Step 3 is correct from the definition of $E(a)$.

Suppose $r$ is a prime factor of $n$ and we have reached Step 4 of the algorithm for a particular $a$. Consider the subgroup $\mathcal{G}_r(a)$ of $(\mathbb{Z}/r\mathbb{Z})^*$ defined in the Introduction. We shall show that $\#\mathcal{G}_r(a) = F(a)$. For each prime $j$ with $j \leq a$ we have $j^{F(a)} \equiv 1 \bmod n$, so that $j^{F(a)} \equiv 1 \bmod r$. Thus $\#\mathcal{G}_r(a) | F(a)$. Further, if $j$ is a prime with $j \leq a$ and $F(j) > F(j-1)$, then from Step 3 the order of $j$ in $(\mathbb{Z}/r\mathbb{Z})^*$ is $E(j)$. Since $F(a)$ is the least common multiple of those numbers $E(j)$ with $j$ prime, $j \leq a$, and $F(j) > F(j-1)$, we have $F(a) | \#\mathcal{G}_r(a)$. Thus $\#\mathcal{G}_r(a) = F(a)$, as asserted.

We conclude that if we have reached Step 4 of the algorithm for a particular $a$, then for each prime factor $r$ of $n$ we have $r \equiv 1 \bmod F(a)$. Thus the correctness of Steps 5 and 6 follows from [5].

Suppose now that $a = [(\log n)^{3/2}]$ and we have reached Step 7. Thus $F(a) < n^{1/3}$. Suppose $n$ is prime. For every $a$-smooth integer $m$ in the range $1 \leq m \leq n$ we have $m \bmod n \in \mathcal{G}_n(a)$. Thus

$$F(a) = \#\mathcal{G}_n(a) \geq \psi(n, a) = \psi(n, (\log n)^{3/2}) > n^{1/3},$$

where the last inequality follows from Theorem 2.1 and the fact that $(\log n)^{3/2} > 2$ for $n > 4$. This is a contradiction and so Step 7 is correct.

We conclude that Algorithm 3.1 is correct.

*Analysis of runtime.* We measure the runtime by the number of arithmetic steps with integers no larger than $n$. By an arithmetic step we mean addition, subtraction, multiplication, division with remainder, greatest common divisor, and finding an inverse for a member of $(\mathbb{Z}/n\mathbb{Z})^*$. Using naive arithmetic, an arithmetic step can be accomplished in $O((\log n)^2)$ bit operations. Using the FFT, an arithmetic step can be accomplished in $O_\epsilon((\log n)^{1+\epsilon})$ bit operations for each $\epsilon > 0$.

One can use the sieve of Eratosthenes to prepare a list of all of the primes up to $(\log n)^{3/2}$ in time $O((\log n)^{3/2} \log\log n)$. For each prime number $a$, Step 1 can be accomplished in $O(\log n)$ arithmetic steps. Since the number of such primes is $O((\log n)^{3/2}/\log\log n)$, an upper bound for the time spent in Step 1 is $O((\log n)^{5/2}/\log\log n)$.

To do Step 2 we use a variation of the algorithm of [6]. First consider the case where $n-1$ is squarefree; say $n-1 = q_1 \dots q_k$ with $q_1, \dots, q_k$ distinct primes. Then to find $E(a)$ it suffices to find the set of $q_i$ which divide $E(a)$. But $q_i | E(a)$ if and only if $a^{(n-1)/q_i} \not\equiv 1 \bmod n$. The algorithm of [6] computes all of the residues $a^{\prod_{j\neq i} q_j} \bmod n = a^{(n-1)/q_i} \bmod n$ for $i = 1, \dots, k$. It breaks the computation into steps where at a particular step we are taking a residue $x \bmod n$ and computing $x^{q_j} \bmod n$ for some $j$. Each $q_j$ is used $O(\log(k+1))$ times, so that the total number of arithmetic operations with integers at most $n$ is

$$O\left(\sum_{j=1}^{k} \log q_j \log(k+1)\right) = O(\log n \log(k+1)).$$

Now consider the general case where we no longer assume that $n-1$ is squarefree. Say $n-1 = q_1^{\alpha_1} \dots q_k^{\alpha_k}$ with the $q_i$'s distinct primes and the $\alpha_i$'s positive integers. We have $q_i^{\beta_i} \| E(a)$ if and only if $\beta_i$ is the least non-negative integer with $a^{(n-1)q_i^{\beta_i - \alpha_i}} \equiv 1 \bmod n$. To compute the $\beta_i$'s we combine the ideas from the squarefree case with a binary search. In the first step, we let $m_1 = q_1^{[\alpha_1/2]} \dots q_k^{[\alpha_k/2]}$ and let $a_1 = a^{m_1} \bmod n$. We use the algorithm of [6] with $a_1$ and the numbers $q_i^{\alpha_i - [\alpha_i/2]}$ to decide if $\beta_i \le [\alpha_i/2]$ or $[\alpha_i/2] < \beta_i \le \alpha_i$ for each $i = 1, \dots, k$. In the first case we replace $q_i^{[\alpha_i/2]}$ in $m_1$ with $q_i^{[\alpha_i/4]}$. In the second case we replace $q_i^{[\alpha_i/2]}$ in $m_1$ with $q^{\alpha_i - [\alpha_i/4]}$. Thus we have a number $m_2$, we form $a_2 = a^{m_2} \bmod n$ and again we use the algorithm of [6], this time with the $q_i$'s raised to exponents about $\alpha_i/4$. Continuing in this fashion we compute the $\beta_i$'s and thus $E(a)$. In the $l$-th step of this algorithm we are using the algorithm of [6] with numbers whose product is about $n^{2^{-l}}$. Thus the number of arithmetic operations for the $l$-th step is $O(2^{-l} \log n \log(k+1))$. Moreover, we can compute $a_l$ from $a_{l-1}$ in $O(2^{-l} \log n)$ arithmetic operations. Thus summing over $l$, the number of steps for Step 2 for a particular value of $a$ is $O(\log n \log(k+1))$. The number of values of $a$ for which we perform Step 2 is $O(\log n)$. (To see this, note that we only perform

Step 2 when $F(a) > F(a-1)$ and that $F(a)$ is the product of the integers $F(j)/F(j-1)$ for $2 \le j \le a$.) Since $k = O(\log n)$, we have that the total time spent in Step 2 is $O((\log n)^2 \log\log n)$ arithmetic steps with integers at most $n$.

For Step 3, note that to compute the greatest common divisor it is sufficient to work with $a^{E(a)/q} \bmod n$ rather than $a^{E(a)/q}$. If $E(a) = q_1^{\beta_1} \dots q_{k'}^{\beta_{k'}}$ where $q_1, \dots, q_{k'}$ are distinct primes and $\beta_1, \dots, \beta_{k'}$ are positive integers, let $a_1 = \prod a^{q_i^{\beta_i - 1}} \bmod n$. We use the algorithm of [6] to compute $a_1^{\prod_{j \ne i} q_j} \bmod n = a^{E(a)/q_i} \bmod n$ for each $i$. The number of steps is $O(\log E(a) \log(k'+1)) = O(\log n \log(k+1))$, where $k \ge k'$ is the number of distinct prime factors of $n-1$. Thus as with Step 2, the total time spent in Step 3 is $O((\log n)^2 \log\log n)$ arithmetic steps with integers at most $n$.

Steps 4, 5, and 7 are each $O(1)$ arithmetic steps for each $a$ and, as remarked above, Step 6 is $O(\log n)$ arithmetic steps for each $a$. Note that we visit Steps 4, 5, and 6 for $O(\log n)$ values of $a$ and we visit Step 7 for $O((\log n)^{3/2})$ values of $a$. Thus the total time for all of these steps is $O((\log n)^2)$ arithmetic steps with integers at most $n$.

We conclude that in the worst case, Algorithm 3.1 runs in $O((\log n)^{5/2}/\log\log n)$ arithmetic steps with integers at most $n$. We have proved the following theorem.

**Theorem 3.1.** *Given an integer $n > 4$ and the complete prime factorization of $n-1$, Algorithm 3.1 correctly decides if $n$ is prime or composite. Further, Algorithm 3.1 uses at most $O((\log n)^{5/2}/\log\log n)$ arithmetic steps with integers at most $n$.*

We remark that in some cases when Algorithm 3.1 declares $n$ composite, a nontrivial factorization of $n$ may also be found. In particular, this is true in Steps 3 and 6. However most composite inputs will be proved composite in Step 1 with $a = 2$, in which case no nontrivial factorization of $n$ is produced.

The next algorithm may be considered an extension of the Brillhart, Lehmer, Selfridge $n-1$ test. We shall use it as a subroutine in an improved version of Algorithm 3.1 we present below.

**Algorithm 3.2.** *This deterministic algorithm finds the complete prime factorization of $n$ when input integers $n, F$ such that $F \ge n^{3/10} > 1$ and each prime factor of $n$ is 1 mod $F$.*

**Step 1** *If $n \le 243$, factor $n$ by trial division and stop.*

**Step 2** *If $F \ge n^{1/3}$, use the method of [5] and stop. (That is, if $F \ge n^{1/2}$, declare $n$ prime; if $n^{1/3} \le F < n^{1/2}$, use the base $F$ factorization method to factor $n$. Note that if the base $F$ factorization succeeds in factoring $n$, then it produces the prime factorization of $n$, while if it fails, then $n$ is prime.)*

**Step 3** *We have $n^{3/10} \le F < n^{1/3}$. Attempt to factor $n$ by the base $F$ factorization method. If this succeeds in splitting $n$, report it as the*

*complete prime factorization of $n$ and stop. Let $c_1, c_2, c_3$ be the base $F$ "digits" of $n$, so that $n = c_3F^3 + c_2F^2 + c_1F + 1$.*

**Step 4** *Let $c_4 = c_3F + c_2$ so that $n = c_4F^2 + c_1F + 1$. If either $c_4x^2 + c_1x + 1$ or $(c_4 - 1)x^2 + (c_1 + F)x + 1$ are reducible in $\mathbb{Z}[x]$, this may lead to a factorization of $n$ as in the base $F$ factorization method. If so, report this factorization as the prime factorization and stop.*

**Step 5** *Develop the continued fraction for $c_1/F$ and let $u/v, u'/v'$ be consecutive convergents with $v < F^2/\sqrt{n} \leq v'$. Let $u_0 = \pm u', v_0 = \pm v'$ be such that $uv_0 + u_0v = 1$.*

**Step 6** *For each integer $d$ with $|d - c_4v/F| < 2n^{3/2}/F^5$ $(\leq 2)$ do the following. Find all integral roots $s$ of the polynomial*

$$f_d(x) = y^3 - c_1y^2 + c_4y + Fzy + z$$

*where $y = dv_0 + vx$, $z = -du_0 + ux$. For any integral root $s$ found with $(dv_0 + vs)F + 1$ a nontrivial factor of $n$, report this number and its cofactor in $n$ as the prime factorization of $n$ and stop. If $n$ is not split in this step, then declare $n$ prime and stop.*

*Proof (of correctness).* We first show that if $n$ is factored in Step 3, then this step produces the complete prime factorization of $n$. First, it is clear that $n$ has at most three prime factors. Thus if $f(x) = c_3x^3 + c_2x^2 + c_1x + 1$ factors into three linear factors in $\mathbb{Z}[x]$, then these give, upon substituting $F$ for $x$, the prime factorization of $n$. Suppose conversely that $n$ has three prime factors. Thus there are positive integers $a_1, a_2, a_3$ with $n = (a_1F + 1)(a_2F+1)(a_3F+1)$. Since $n > 243$ we have $F^4 > 3n$, so that $a_1a_2a_3 < F/3$. Thus $a_1a_2 + a_1a_3 + a_2a_3 \leq 3a_1a_2a_3 < F$ and $a_1 + a_2 + a_3 < F$. We conclude that $c_3 = a_1a_2a_3$, $c_2 = a_1a_2 + a_1a_3 + a_2a_3$, $c_1 = a_1 + a_2 + a_3$ and that $f(x) = (a_1x+1)(a_2x+1)(a_3x+1)$. That is, the base $F$ factorization method will find the complete prime factorization of $n$.

Suppose now that $n$ has exactly two prime factors so that there are positive integers $a_1, a_2$ with $n = (a_1F + 1)(a_2F + 1)$. Assume $a_1 \leq a_2$. If we obtain any nontrivial splitting of $n$ in any step of the algorithm, evidently this gives the complete prime factorization of $n$. We now show that if $n$ has not been factored in Steps 3 and 4 of the algorithm and if $n$ is composite, then it will be factored in Step 6.

Since $n = c_4F^2 + c_1F + 1 = a_1a_2F^2 + (a_1 + a_2)F + 1$, there is some integer $t \geq 0$ with

$$a_1a_2 = c_4 - t, \quad a_1 + a_2 = c_1 + tF. \tag{3.1}$$

From the failure of Step 4 to find $a_1, a_2$, we have $t \geq 2$. Thus

$$a_2 \geq \frac{a_1 + a_2}{2} \geq \frac{c_1 + 2F}{2} \geq F \tag{3.2}$$

and

$$a_1 < \frac{n}{a_2F^2} \leq \frac{n}{F^3}. \tag{3.3}$$

We have from (3.1) that

$$t \le \frac{a_1 + a_2}{F} \le \frac{a_1 a_2 + 1}{F} < \frac{c_4}{F} < \frac{n}{F^3}. \tag{3.4}$$

Also (3.1) gives us the equation

$$a_1 c_1 + a_1 t F = a_1^2 + c_4 - t. \tag{3.5}$$

From the elementary theory of continued fractions we have

$$\left|\frac{c_1}{F} - \frac{u}{v}\right| \le \frac{1}{vv'} \le \frac{\sqrt{n}}{vF^2}. \tag{3.6}$$

Using (3.5) we have

$$\begin{aligned} a_1 u + a_1 t v - \frac{c_4 v}{F} &= a_1 v \left(\frac{u}{v} - \frac{c_1}{F}\right) + (a_1 c_1 + a_1 t F)\frac{v}{F} - \frac{c_4 v}{F} \\ &= a_1 v \left(\frac{u}{v} - \frac{c_1}{F}\right) + (a_1^2 + c_4 - t)\frac{v}{F} - \frac{c_4 v}{F} \\ &= a_1 v \left(\frac{u}{v} - \frac{c_1}{F}\right) + (a_1^2 - t)\frac{v}{F}. \end{aligned}$$

Thus from (3.3), (3.4), (3.6) and the fact that $v < F^2/\sqrt{n}$ we have that

$$\left|a_1 u + a_1 t v - \frac{c_4 v}{F}\right| < a_1 v \frac{\sqrt{n}}{vF^2} + \left(\frac{n}{F^3}\right)^2 \frac{v}{F} < \frac{n}{F^3}\frac{\sqrt{n}}{F^2} + \frac{n^2}{F^7}\frac{F^2}{\sqrt{n}} \le \frac{2n^{3/2}}{F^5}. \tag{3.7}$$

Let $d = a_1 u + a_1 t v$. Note that the general solution to $yu + zv = d$ is given by

$$y = dv_0 + vs, \quad z = du_0 - us,$$

where $s$ runs over the integers. Let $s$ be the unique integer with

$$a_1 = dv_0 + vs, \quad a_1 t = du_0 - us.$$

From (3.5) we have that $s$ satisfies

$$(dv_0 + vs)^2 + c_4 - \frac{du_0 - us}{dv_0 + vs} = (dv_0 + vs)c_1 + (du_0 - us)F.$$

Thus from (3.7) we see that $s$ is an integral root for one of the polynomials $f_d(x)$ presented in Step 6. This concludes the proof of correctness of Algorithm 3.2.

Since the computation of the convergents $u/v$ and $u'/v'$ in Step 5 of the algorithm can be made part of the extended Euclidean algorithm for $c_1$ and $F$, it is clear that the runtime of Algorithm 3.2 is dominated by the calculations of the possible integer roots of the four cubic polynomials in Step 6. Thus Algorithm 3.2 runs in $O(\log n)$ arithmetic operations with integers at most $n$ if a binary search is used to find the roots as discussed in connection with the base $B$ factorization method above.

We now use Algorithm 3.2 in the framework of Algorithm 3.1.

**Algorithm 3.3.** *We are input an integer $n > 5$ and the complete prime factorization of $n-1$. This deterministic algorithm decides if $n$ is prime or composite.*

*Let $F(1) = 1$. For $a = 2, 3, \ldots, [(\log n)^{10/7}]$ do the following:*

**Steps 1–4** *These are exactly the same as in Algorithm 3.1 except that when $F(a) = F(a-1)$ we go to Step 6.*

**Step 5** *If $F(a) \geq n^{3/10}$, find the complete prime factorization of $n$ with Algorithm 3.2 and stop.*

**Step 6** *If $a < [(\log n)^{10/7}]$, get the next $a$. Otherwise declare $n$ composite and stop.*

We have already proved in connection with Algorithm 3.1 that if we do Steps 1-4 for $j = 2, 3, \ldots, a$, and we have not stopped, then every prime factor of $n$ is 1 mod $F(a)$. Thus Algorithm 3.2 is appropriate to use in Step 5. Suppose $a = [(\log n)^{10/7}]$ and we are in Step 6. If $n$ is prime and $\mathcal{G}_n(a)$ is as before, then, as with the proof of correctness of Algorithm 3.1, we have

$$F(a) = \#\mathcal{G}_n(a) \geq \psi(n, a) = \psi(n, (\log n)^{10/7}) > n^{3/10}$$

by Theorem 2.1. Thus Step 6 is correct. We conclude that Algorithm 3.3 is correct.

We have the following theorem.

**Theorem 3.2.** *Given an integer $n > 5$ and the complete prime factorization of $n-1$, Algorithm 3.3 correctly decides if $n$ is prime or composite. Moreover, it uses at most $O((\log n)^{17/7}/\log\log n)$ arithmetic operations with integers at most $n$.*

## 4. When $n-1$ is partially factored

In this section we describe a deterministic polynomial time algorithm that decides if $n$ is prime or composite when $n$ and a divisor $F$ of $n-1$ are input with $F > n^{1/4+\epsilon}$.

**Algorithm 4.1.** *We are input an integer $n$ and a number $\epsilon$ with $n > e^3$, $0 < \epsilon \leq 3/4$ and $(\log n)^{5/(4\epsilon)} < n$. We are also input integers $F$, $R$ with $n - 1 = FR$ and $F > n^{1/4+\epsilon}$, and we are input the complete prime factorization of $F$. This deterministic algorithm decides if $n$ is prime or composite.*

*Let $F(1) = 1$. For $a = 2, 3, \ldots, [(\log n)^{5/(4\epsilon)}]$ do the following:*

**Step 1** *If $a$ is composite, let $F(a) = F(a-1)$ and go to Step 7. If $a^{RF(a-1)} \equiv 1 \bmod n$, let $F(a) = F(a-1)$ and go to Step 7. Verify that $a^{n-1} \equiv 1$ mod $n$. If not, declare $n$ composite and stop.*

**Step 2** *Using the prime factorization of $F$, compute $E(a)$, the order of $a^R$ mod $n$ in $(\mathbb{Z}/n\mathbb{Z})^*$. Thus $E(a)$ is the least positive divisor of $F$ with $a^{RE(a)} \equiv 1 \bmod n$.*

**Step 3** *For each prime factor $q$ of $E(a)$, verify that $(a^{RE(a)/q} - 1, n) = 1$. If not, declare $n$ composite and stop.*

**Step 4** *Let $F(a) = \operatorname{lcm}\{F(a-1), E(a)\}$. Compute $F(a)$.*

**Step 5** *If $F(a) \geq n^{3/10}$, get the complete prime factorization of $n$ by Algorithm 3.2. In particular, if $n$ is prime, declare it so and stop; if $n$ is composite, declare it so and stop.*

**Step 6** *If $F(a) > n^{1/4+\epsilon/5}$, attempt to factor $n$ by the base $F(a)$ factorization method. If this succeeds in splitting $n$, then declare $n$ composite and stop. Let $c_1, c_2, c_3$ be the base $F(a)$ "digits" of $n$ so that $n = c_3F(a)^3 + c_2F(a)^2 + c_1F(a) + 1$. Let $c_4 = c_3F(a) + c_2$. If either $c_4x^2 + c_1x + 1$ or $(c_4 - 1)x^2 + (c_1 + F(a))x + 1$ are reducible in $\mathbb{Z}[x]$, this may lead to nontrivial factorization of $n$ by substituting $F(a)$ for $x$. If so, declare $n$ composite and stop.*

**Step 7** *If $a < [(\log n)^{5/(4\epsilon)}]$ get the next $a$. If $a = [(\log n)^{5/(4\epsilon)}]$ and $F(a) \leq n^{1/4+\epsilon/5}$ declare $n$ composite and stop. If $a = [(\log n)^{5/(4\epsilon)}]$ and $F(a) > n^{1/4+\epsilon/5}$, declare $n$ prime and stop.*

*Proof (of correctness).* Since $(\log n)^{5/(4\epsilon)} < n$, Step 1 is correct. Step 3 is clearly correct. We recall the definition of $\mathcal{G}_r(a)$ from the Introduction. If we have passed Step 3 of the algorithm for $2, 3, \ldots, a$, then $F(a) | \#\mathcal{G}_r(a)$ and $\#\mathcal{G}_r(a) | RF(a)$ for each prime factor $r$ of $n$. In particular $r \equiv 1 \bmod F(a)$. Thus it is appropriate to use Algorithm 3.2 in Step 5.

It is clear that Step 6 is correct since it only declares $n$ composite when it succeeds in splitting $n$. Suppose we are in Step 7 and $a = [(\log n)^{5/(4\epsilon)}]$. If $n$ is prime, then as in the analysis of Algorithm 3.1, we have

$$RF(a) \geq \#\mathcal{G}_n(a) \geq \psi(n, a) = \psi(n, (\log n)^{5/(4\epsilon)}) > n^{1-4\epsilon/5}$$

by Theorem 2.1. Since $R < n^{3/4-\epsilon}$, we thus have $F(a) > n^{1/4+\epsilon/5}$. We conclude that if $F(a) \leq n^{1/4+\epsilon/5}$, then Step 7 is correct in declaring $n$ composite. Suppose finally we are in Step 7, $a = [(\log n)^{5/(4\epsilon)}]$, $F(a) > n^{1/4+\epsilon/5}$ and $n$ is composite. Since Step 6 was not able to split $n$, we have as with the analysis of Algorithm 3.2 that $n$ is the product of two primes. (It is here where we use the hypothesis $n > e^3$ since this assures that $F(a)^4 > n^{1+4\epsilon/5} > n((\log n)^{5/(4\epsilon)})^{4\epsilon/5} = n \log n > 3n$.) Say $n = r_1r_2$ where $r_1 \leq r_2$ are primes and $r_1 = b_1F(a) + 1$, $r_2 = b_2F(a) + 1$, where $b_1, b_2$ are positive integers. Again from the failure of Step 6 to factor $n$ and the argument for Algorithm 3.2 (see (3.5) and (3.3)) we have

$$b_2 \geq \frac{b_1 + b_2}{2} \geq F(a), \quad b_1 < \frac{n}{b_2F(a)^2} \leq \frac{n}{F(a)^3}.$$

Thus

$$\begin{aligned}\#\mathcal{G}_{r_2}(a) &\leq (RF(a), r_2 - 1) \leq (n-1, r_2 - 1)\\ &= (b_1 b_2 F(a)^2 + (b_1 + b_2)F(a), b_2 F(a))\\ &= (b_1 b_2 F(a) + b_1 + b_2, b_2)F(a) = (b_1, b_2)F(a)\\ &\leq b_1 F(a) = \frac{b_1}{b_2} b_2 F(a) < \frac{n}{F(a)^4} r_2 < n^{-4\epsilon/5} r_2.\end{aligned}$$

On the other hand, as before, we have

$$\#\mathcal{G}_{r_2}(a) \geq \psi(r_2, a) = \psi(r_2, (\log n)^{5/(4\epsilon)}) \geq \psi(r_2, (\log r_2)^{5/(4\epsilon)}) \geq r_2^{1-4\epsilon/5}$$

by Theorem 2.1. These last two displays are incompatible. Thus Step 7 is correct in declaring $n$ prime when $a = [(\log n)^{5/(4\epsilon)}]$ and $F(a) > n^{1/4+\epsilon/5}$. This concludes the proof of correctness for Algorithm 4.1.

The runtime analysis for Algorithm 4.1 is argued analogously to that of Algorithm 3.1. We have the following theorem.

**Theorem 4.1.** *On input of an integer $n > e^3$, a number $\epsilon$ in the range $0 < \epsilon \leq 3/4$ with $(\log n)^{5/(4\epsilon)} < n$, integers $F$, $R$ with $n - 1 = FR$ and $F > n^{1/4+\epsilon}$, and the complete prime factorization of $F$, Algorithm 4.1 correctly decides if $n$ is prime or composite. The runtime is $O((\log n)^{1+5/(4\epsilon)}/\log\log n)$ arithmetic operations with integers at most $n$.*

## 5. Primes recognizable in deterministic polynomial time

The algorithms in the preceding two sections to determine whether $n$ is prime or not all hypothesized substantial information about the factorization of $n-1$ (or knowledge about the prime factors of $n$). In this section we present an algorithm that can be applied to any number $n$. For most numbers, this algorithm will not be very efficient – in fact it will take exponential time. However there are also many primes for which the algorithm will work in polynomial time – more than $x^{1-\epsilon}$ of them up to $x$. Before we give this result, we first state the algorithm.

**Algorithm 5.1.** *Suppose we are input a positive integer $n > 5 \times 10^{14}$. This deterministic algorithm decides if $n$ is prime or composite.*

**Step 1** *Continue using trial division on $n-1$ until a fully factored divisor $F$ of $n-1$ is found with $F > n^{1/3}$.*

**Step 2** *Use Algorithm 4.1 with inputs $n$, $\epsilon = 1/12$, $F$, $R = (n-1)/F$.*

It is clear that Algorithm 5.1 is correct. It is also clear that for some numbers it is a terrible algorithm. For example, if $n$ is even, one might well spend exponential time discovering that $n$ is composite. Nevertheless, Algorithm 5.1 is able to prove prime quite a few numbers in polynomial time.

**Theorem 5.1.** *For each $\epsilon > 0$ there are numbers $k$ and $x_0$ such that if $x \geq x_0$, then the number of primes $p \leq x$ which Algorithm 5.1 proves prime in at most $(\log p)^k$ arithmetic steps with integers at most $p$ exceeds $x^{1-\epsilon}$.*

The proof of this theorem depends strongly on the distribution of primes $p$ for which $p-1$ has a large smooth divisor. We establish such a result now.

**Theorem 5.2.** *There are effectively computable positive constants $c_1$, $x_1$ with the following property. Suppose $x \geq x_1$, $\log x \leq y \leq x^{1/20}$ and $N(x,y)$ is the number of primes $p \leq x$ such that $p-1$ has a $y$-smooth divisor exceeding $x^{1/3}$. Then $N(x,y) \geq x/(c_1 \log x)^{1+u/3}$, where $u = \log x/\log y$.*

*Proof.* Let $x_2$ be the number $x_{\epsilon,\delta}$ in Theorem 2.1 of [2], where $\epsilon = 1/11$ and $\delta = 1/60$. If $x \geq x_2$, let $d_1, d_2, \ldots, d_k$ be the possible "exceptional moduli" corresponding to $x$ in this theorem, so that they all exceed $\log x$ and $k = k(x) = O(1)$. Let $q_i$ denote the greatest prime factor of $d_i$ for $i = 1, \ldots, k$. Let

$$\mathcal{P} = \{q \text{ prime } : y/2 < q \leq y\} \setminus \{q_1, \ldots, q_k\}.$$

Thus if an integer $d$ is composed solely of primes from $\mathcal{P}$, then no $d_i$ divides $d$. From Mertens's theorem we have that if $x$ is sufficiently large, then

$$\frac{1}{2\log y} < \sum_{q\in\mathcal{P}} \frac{1}{q} < \sum_{q\in\mathcal{P}} \frac{1}{q-1} < \frac{1}{\log y}. \tag{5.1}$$

Let $v = \lceil (\log(x^{1/3}))/\log(y/2) \rceil$ and let $\mathcal{D}(\mathcal{P}, v)$ denote the set of integers $d$ composed of $v$ not necessarily distinct primes from $\mathcal{P}$. If $d \in \mathcal{D}(\mathcal{P}, v)$, then clearly $d > x^{1/3}$ and $d$ is $y$-smooth. Further, if $x$ is sufficiently large, then

$$d \leq y^{1+(\log(x^{1/3}))/\log(y/2)} = y(x^{1/3})^{\log y/\log(y/2)} \leq x^{2/5}.$$

Thus if $x$ is sufficiently large and $d \in \mathcal{D}(\mathcal{P}, v)$, we have from Theorem 2.1 in [2] that

$$\pi(x, d, 1) := \sum_{p\leq x, d|p-1} 1 \geq \frac{9}{10\varphi(d)} \cdot \frac{x}{\log x}, \tag{5.2}$$

where $p$ runs over primes and $\varphi$ denotes Euler's function.

For $n$ a positive integer, let $(n, \mathcal{P})$ denote the largest divisor of $n$ composed of only primes from $\mathcal{P}$. For $p$ a prime, let $d(p, v)$ denote the number of divisors of $p-1$ which come from $\mathcal{D}(\mathcal{P}, v)$. Note that $d(p, v) = 1$ if and only if there is some $d \in \mathcal{D}(\mathcal{P}, v)$ with $d|p-1$ and $((p-1)/d, \mathcal{P}) = 1$. Thus

$$\begin{aligned} N(x,y) &\geq \sum_{\substack{p\leq x\\ d(p,v)>0}} 1 \geq \sum_{\substack{p\leq x\\ d(p,v)=1}} 1 = \sum_{d\in\mathcal{D}(\mathcal{P},v)} \sum_{\substack{p\leq x, d\mid p-1\\ ((p-1)/d,\mathcal{P})=1}} 1 \\ &= \sum_{d\in\mathcal{D}(\mathcal{P},V)} \sum_{\substack{p\leq x\\ d\mid p-1}} 1 - \sum_{d\in\mathcal{D}(p,v)} \sum_{\substack{p\leq x, d\mid p-1\\ ((p-1)/d,\mathcal{P})>1}} 1 \\ &\geq \sum_{d\in\mathcal{D}(\mathcal{P},v)} \sum_{\substack{p\leq x\\ d\mid p-1}} 1 - \sum_{d\in\mathcal{D}(p,v)} \sum_{q\in\mathcal{P}} \sum_{\substack{p\leq x\\ dq\mid p-1}} 1 \qquad (5.3) \\ &= \sum_{d\in\mathcal{D}(\mathcal{P},v)} \pi(x,d,1) - \sum_{d\in\mathcal{D}(\mathcal{P},v)} \sum_{q\in\mathcal{P}} \pi(x,dq,1). \end{aligned}$$

From (5.2) we have

$$\sum_{d\in\mathcal{D}(\mathcal{P},v)} \pi(x,d,1) \geq \frac{9}{10}\cdot\frac{x}{\log x} \sum_{d\in\mathcal{D}(\mathcal{P},v)} \frac{1}{\varphi(d)} \qquad (5.4)$$

if $x$ is sufficiently large. From the Brun-Titchmarsh inequality we have

$$\begin{aligned} \sum_{d\in\mathcal{D}(\mathcal{P},v)} \sum_{q\in\mathcal{P}} \pi(x,dq,1) &\ll \sum_{d\in\mathcal{D}(\mathcal{P},v)} \sum_{q\in\mathcal{P}} \frac{x}{\varphi(dq)\log(x/(dq))} \\ &\ll \frac{x}{\log x}\Bigg(\sum_{d\in\mathcal{D}(\mathcal{P},v)} \frac{1}{\varphi(d)}\Bigg) \sum_{q\in\mathcal{P}} \frac{1}{q-1} \\ &< \frac{x}{\log x \log y} \sum_{d\in\mathcal{D}(\mathcal{P},v)} \frac{1}{\varphi(d)}, \end{aligned}$$

where we use (5.1) for the last inequality. Putting this estimate and (5.4) into (5.3) we have for all sufficiently large $x$ that

$$N(x,y) \geq \frac{4}{5}\cdot\frac{x}{\log x} \sum_{d\in\mathcal{D}(\mathcal{P},v)} \frac{1}{\varphi(d)}. \qquad (5.5)$$

We now estimate this last sum. We have from (5.1) that

$$\begin{aligned} \sum_{d\in\mathcal{D}(\mathcal{P},v)} \frac{1}{\varphi(d)} &> \sum_{d\in\mathcal{D}(\mathcal{P},v)} \frac{1}{d} \geq \frac{1}{v!}\Bigg(\sum_{q\in\mathcal{P}} \frac{1}{q}\Bigg)^v \\ &= \exp(-v\log v - v\log\log y + O(v)) \\ &= \exp(-\frac{1}{3}u\log u - \frac{1}{3}u\log\log y + O(u)) \\ &= \exp(-\frac{1}{3}u\log\log x + O(u)). \end{aligned}$$

Putting this in (5.5) we have the theorem.

*Proof (of Theorem 5.1).* First note that from Theorem 5.2, if $k \geq 1$ is arbitrary, then the number of primes $p \leq x$ for which $p-1$ has a $(\log x)^k$-smooth divisor $F$ with $F > x^{1/3}$ is at least $x^{1-1/(3k)+o(1)}$ as $x \to \infty$. For such primes $p$, Step 1 of Algorithm 5.1 takes at most $O((\log x)^k)$ arithmetic steps with integers at most $p$. The number of arithmetic steps with integers at most $p$ to complete Step 2 of Algorithm 5.1 is $O((\log p)^{16}/\log\log p)$.

It suffices to establish the theorem for values of $\epsilon$ satisfying $0 < \epsilon < 1/48$. In this case let $K > K' > 1/(3\epsilon)$ be arbitrary. If $p > x^{1/3}$ and $c$ is any constant, then $(\log p)^K > c(\log x)^{K'}$ for all large $x$. Thus if $x$ is large, $p$ is a prime with $p \leq x$ and $p-1$ has a $(\log x)^{K'}$-smooth divisor $F > x^{1/3}$, then Algorithm 5.1 takes at most $(\log p)^K$ steps with integers at most $p$ to prove $p$ prime. By the above, the number of such primes is at least $x^{1-1/(3K')+o(1)}$ for $x \to \infty$. As $1 - 1/(3K') > 1 - \epsilon$, the theorem follows.

## 6. More primes recognizable in deterministic polynomial time

In this section we describe a deterministic algorithm that recognizes many more primes in polynomial time than our previous methods. Covered is any prime $n$ with a divisor $F$ of $n-1$ exceeding $n^\epsilon$ and such that all of the prime factors of $F$ are at most $(\log n)^k$. The running time is about $O((\log n)^{2/\epsilon} + (\log n)^k)$. We also show that for most such primes, the $2/\epsilon$ can be reduced to $1/\epsilon$. A corollary of this algorithm is an improved version of Theorem 5.1. There if we were willing to spend time $(\log n)^k$ on trying to prove $n$ prime, we would succeed for about $x^{1-(3k)^{-1}}$ primes up to $x$. With the methods of this section we will succeed for about $x^{1-k^{-2}}$ primes up to $x$.

If $n$ is prime and the order of $b$ in $(\mathbb{Z}/n\mathbb{Z})^*$ is $E$, then for any $a$ with $a^E \equiv 1 \bmod n$, there is some exponent $j \in \{1, 2, \cdots, E\}$ with $a \equiv b^j \bmod n$. Thus if it is shown that no such $j$ exists, then it is proved that $n$ is composite. How difficult is it to do this test? If we have already prepared the complete set $\{b^j \bmod n : j = 1, 2, \cdots, E\}$, then testing if there is some $j$ with $a \equiv b^j \bmod n$ can be accomplished by a binary search in $O(\log E)$ steps. Thus we have the initial step of preparing the set of powers of $b$, which takes $E$ steps, and then each subsequent test takes $O(\log E)$ steps.

In the case when $E = q^\beta$ with $q$ prime and $\beta \geq 1$, we can do a precomputation taking $q$ steps, with each subsequent test taking $O(\beta^2 \log q)$ steps. Here is how. Suppose the order of $b \bmod n$ in $(\mathbb{Z}/n\mathbb{Z})^*$ is $q^\beta$. Assuming that we have already computed $b^{q^{\beta-1}} \bmod n$ (if not, this takes an additional $O(\beta \log q)$ steps for the precomputation), we can compute the set

$$\mathcal{B} = \{b^{jq^{\beta-1}} \bmod n : j = 1, \cdots, q\}$$

in $q$ steps. Now suppose we are presented with some integer $a$ with the order of $a \bmod n$ in $(\mathbb{Z}/n\mathbb{Z})^*$ equal to $q^\alpha$, with $0 \leq \alpha \leq \beta$, and we wish to see if

there is some integer $j$ with $a \equiv b^j \bmod n$. If $\alpha = 0$ or 1, then if $j$ exists it is a multiple of $q^{\beta-1}$, and so we test for membership of $a \bmod n$ in $\mathcal{B}$ by a binary search. As an induction hypothesis suppose $1 < \alpha \leq \beta$ and we have already described how to find $j$ for any $a'$ for which the order of $a' \bmod n$ in $(\mathbb{Z}/n\mathbb{Z})^*$ properly divides $q^\alpha$. Note that the order of $a^q \bmod n$ is $q^{\alpha-1}$, so we may use our inductively described algorithm to search for some integer $j_0$ with $a^q \equiv b^{j_0} \bmod n$. Suppose we have found $j_0$. Then it must be that $q^{\beta-(\alpha-1)}$ divides $j_0$ and, in particular, $q|j_0$. Then $ab^{-j_0 q^{-1}} \bmod n$ has order dividing $q$. But we have already described how in this case we may search for an integer $j_1$ with $ab^{-j_0 q^{-1}} \equiv b^{j_1} \bmod n$. If these searches are successful, we may take $j = j_1 + j_0 q^{-1}$. Totaling up the time spent, we have used $\alpha$ binary searches in the set $\mathcal{B}$, we have done $\alpha - 1$ modular multiplications, and we have done $\alpha - 1$ modular powerings with exponents at most $q^\beta$ (in fact, at most $q^{\beta-1}$). The latter computation dominates, taking $O(\alpha\beta \log q) = O(\beta^2 \log q)$ arithmetic steps with integers the size of $n$.

The computation of $\mathcal{B}$, which is the precomputation step of this method, we call "Set up $(b, q^\beta)$". The subsequent search for an exponent $j$ we call "Test $(b, q^\beta; a)$". With these subroutines we are now ready to describe the main algorithm of this section.

**Algorithm 6.1.** *We are given positive integers $n$, $F$, $R$ and a positive number $\epsilon$ such that $n > 4$, $2 \leq (\log n)^{2/\epsilon} < n$, $n - 1 = FR$, and $F > n^\epsilon$. This algorithm decides if $n$ is prime or composite.*

*Let $F(1) = 1$. For $a = 2, 3, \cdots, \lceil (\log n)^{2/\epsilon} \rceil$ do the following.*

**Step 1.** *Check if $n$ is even or if $n$ is a nontrivial power. If so, declare $n$ composite and stop.*

**Step 2.** *Verify that $a^{n-1} \equiv 1 \bmod n$ holds. If not, declare $n$ composite and stop.*

**Step 3.** *Compute $E(a)$, the order of $a^R \bmod n$ in $(\mathbb{Z}/n\mathbb{Z})^*$. Let $F(a) = \operatorname{lcm}\{E(a), F(a-1)\}$. For each prime $q|F(a)/F(a-1)$, verify that $(a^{RE(a)/q} - 1, n) = 1$ holds, declaring $n$ composite and stopping if not.*

**Step 4.** *For each prime $q$ and positive integers $\alpha, \beta$ with $q^\alpha \| (E(a), F(a-1))$ and $q^\beta \| F(a-1)$ do Test $(b_q, q^\beta; a^{RE(a)q^{-\alpha}})$. If this test proves $n$ composite, then declare this and stop.*

**Step 5.** *For each prime $q$ and positive integer $\beta$ with $q|F(a)/F(a-1)$ and $q^\beta \| F(a)$, let $b_q = a^{RE(a)q^{-\beta}} \bmod n$ and do Set up $(b_q, q^\beta)$.*

**Step 6.** *If $F(a)^{\log a} > n^{\log\log n}$, declare $n$ prime and stop. If $a < \lceil (\log n)^{2/\epsilon} \rceil$, get the next $a$. If $a = \lceil (\log n)^{2/\epsilon} \rceil$, declare $n$ composite and stop.*

*Proof (of correctness).* Suppose we have made it to Step 6 and $a = \lceil (\log n)^{2/\epsilon} \rceil$. Suppose $n$ is prime. From Theorem 2.1 we have $\#\mathcal{G}_n(a) > n^{1-\epsilon/2}$. Every $b \in \mathcal{G}_n(a)$ satisfies $b^{(n-1)F(a)/F} \equiv 1 \bmod n$, so $\#\mathcal{G}_n(a) \leq (n-1)F(a)/F < n^{1-\epsilon}F(a)$. Thus $F(a) > n^{\epsilon/2}$, so that $F(a)^{\log a} > n^{\log\log n}$. Hence it is correct to declare $n$ composite when this inequality fails.

Suppose $n$ is composite and suppose we have made it to Step 6 for a particular $a$. From Step 1 we know that $n$ is divisible by at least two distinct odd primes. From Step 3 we know that each prime factor of $n$ is 1 mod $F(a)$. Let $\mathcal{F} = \{b \bmod n : b^{n-1} \equiv 1 \bmod n\}$. By the Chinese remainder theorem, this subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ is isomorphic to the direct product of the groups $\mathcal{F}_p = \{b \bmod p^\alpha : b^{n-1} \equiv 1 \bmod p^\alpha\}$ where $p^\alpha$ runs over the prime powers with $p^\alpha \| n$. Since $p$ is an odd prime, $(\mathbb{Z}/p^\alpha\mathbb{Z})^*$ is cyclic so that $\#\mathcal{F}_p = (n-1, \varphi(p^\alpha)) = (n-1, p-1)$. Thus for each prime power $q^\beta$ with $q^\beta \| F(a)$ we have $q^\beta | \#\mathcal{F}_p$. Since $n$ has at least two distinct prime factors $p$, the number of $b \bmod n \in \mathcal{F}$ with $b^{q^\beta} \equiv 1 \bmod n$ is at least $q^{2\beta}$. From Step 2, $\mathcal{G}_n(a)$ is a subgroup of $\mathcal{F}$. And from Step 4 we have that $\mathcal{G}_n(a)$ has exactly $q^\beta$ members $b \bmod n$ with $b^{q^\beta} \equiv 1 \bmod n$. Thus the index of $\mathcal{G}_n(a)$ in $\mathcal{F}$ is at least the product of the prime powers $q^\beta$ with $q^\beta \| F(a)$, which is $F(a)$. We have from Theorem 2.1 that

$$n^{1-\log\log n/\log a} < \#\mathcal{G}_n(a) \le \frac{1}{F(a)} \prod_{p|n} (n-1, p-1) < \frac{n}{F(a)},$$

so that $F(a) < n^{\log\log n/\log a}$. Hence it is correct to declare $n$ prime when $F(a)^{\log a} > n^{\log\log n}$. This concludes the proof of correctness.

*Analysis of runtime.* For each integer $k \le \log n/\log 2$ we can check if $n$ is a $k$-th power by computing $[n^{1/k}]$ with a binary search and seeing if $[n^{1/k}]^k = n$. When $k \ge 3$, the binary search may begin with $[n^{1/(k-1)}]$. Thus the number of arithmetic operators to see if $n$ is a $k$-th power is $O(\frac{\log k}{k} \log n)$, so the total number of arithmetic operations for Step 1 is $O(\log n (\log\log n)^2)$.

The time for Step 2 is at most $O((\log n)^{1+2/\epsilon})$.

For each $a$, the time for Step 3 is $O(\log n \log\log n)$. Thus the total time for Step 3 is $O((\log n)^{1+2/\epsilon} \log\log n)$.

As we have seen, each implementation of Test $(b_q, q^\beta; a)$ in Step 4 takes time $O(\beta^2 \log q)$. Thus the total time for Step 4 is $O((\log n)^{2+2/\epsilon})$.

Each time we do Set up $(b_q, q^\beta)$ in Step 5, it takes time $O(q + \beta \log q)$. Thus if $\Omega$ is the total number of prime factors of $F$, counted with multiplicity, and if each prime factor $q$ of $F$ satisfies $q \le B$, then the total time for Step 5 is $O(B\Omega + \Omega^2 \log n) = O(B \log n + (\log n)^3)$.

Thus the total number of arithmetic steps with integers at most $n$ is $O((\log n)^{2+2/\epsilon} + B \log n)$.

We have the following theorem.

**Theorem 6.1.** *On input of positive integers $n$, $F$, $R$ and a positive number $\epsilon$ with $n > 4$, $2 \le (\log n)^{2/\epsilon} < n$, $n - 1 = FR$ and $F > n^\epsilon$, Algorithm 6.1 correctly decides if $n$ is prime or composite. The running time is $O((\log n)^{2+2/\epsilon} + B \log n)$ arithmetic steps with integers at most $n$, where $B$ is the largest prime factor of $F$.*

We remark that the term $(\log n)^{2+2/\epsilon}$ in the running time may be replaced with $\epsilon(\log n)^{2+2/\epsilon}/\log\log n$ if we perform Steps 2 to 4 only for prime values of $a$.

The time bound in Theorem 6.1 is only an upper bound. With the aid of the next result we will be able to show that most primes for which Algorithm 6.1 is applicable are proved prime in a considerably shorter time.

**Theorem 6.2.** *For $2 \le y \le x$ let $R(x,y)$ denote the number of primes $p$ in the range $y < p \le x$ such that $(\mathbb{Z}/p\mathbb{Z})^*$ is not generated by the set $\{a \bmod p : 1 \le a \le y\}$; that is, such that $G_p([y]) \ne (\mathbb{Z}/p\mathbb{Z})^*$. Then $R(x,y) < 8x^{2\log\log(x^2)/\log y}$.*

*Proof.* Let $\mathcal{Z}$ denote the set of $y$-smooth numbers up to $x^2$. Suppose $p$ is a prime counted by $R(x,y)$. Then $\#G_p([y])$ divides $(p-1)/q$ for some prime $q$. Thus the set $\mathcal{Z}$ occupies at most $(p-1)/q$ residue classes mod $p$, so there are at least $p/2$ residue classes mod $p$ free of elements of $\mathcal{Z}$. Thus by the large sieve (see Theorem 3, p. 159 in [8]) we have

$$\#\mathcal{Z} \le \frac{4x^2}{\frac{1}{2}\cdot R(x,y)}.$$

Thus from Theorem 2.1 we have

$$R(x,y) \le \frac{8x^2}{\#\mathcal{Z}} = \frac{8x^2}{\psi(x^2,y)} < 8x^{2\log\log(x^2)/\log y}, \tag{6.1}$$

which completes the proof of the theorem.

Because of the use of Theorem 2.1 in the proof, Theorem 6.2 is only fairly good when $y$ is a power of $\log x$. If we let $v = 2\log x/\log y$ and use a stronger estimate for $\psi(x^2,y)$ (see [7]) we may obtain the following result which is valid in the same range as is Theorem 6.2:

$$R(x,y) \le \exp(v(\log v + \log\log v - 1 + (\log\log v - 1)/\log v) + O(v(\log\log v/\log v)^2)).$$

This estimate is implicit in the dissertation of Pappalardi [13, Section 3.3]. The estimate has an inexplicit constant, though an actual numerical value could be provided in principle. We remark that Vinogradov, Linnik and Fridlender have discussed problems related to the estimation of $R(x,y)$.

**Theorem 6.3.** *Let $x$, $\epsilon > 0$ be arbitrary and let $E(x,\epsilon)$ denote the number of primes $p$ with $(\log p)^{2/\epsilon} < p \le x$ for which there is some integer $F > p^{\epsilon}$ with $F \mid p-1$ and such that if Algorithm 6.1 is run on $n = p$, $F$, $\epsilon$, then $F(a)^{\log a} \le p^{\log\log p}$ for $a = \lceil(\log p)^{1/\epsilon}\rceil$. Then $E(x,\epsilon) < 9x^{3\epsilon} + e^{32}$.*

*Proof.* Suppose Algorithm 6.1 is run on $n = p$, $F$, $\epsilon$ where $F|p-1$ and $F > p^{\epsilon}$. If $F(a) = F$ with $a = \lceil (\log p)^{1/\epsilon} \rceil$, then $F(a)^{\log a} > p^{\log\log p}$. Thus we may assume that if $p$ is counted by $E(x,\epsilon)$, then $F(a) < F$ with $a = \lceil (\log p)^{1/\epsilon} \rceil$. Thus $(\mathbb{Z}/p\mathbb{Z})^*$ is not generated by $\{j \bmod p : 1 \le j \le a\}$. We conclude from Theorem 6.2 that

$$\begin{aligned} E(x,\epsilon) - E(x^{1/2},\epsilon) &\le R(x, \lceil (\log x^{1/2})^{1/\epsilon} \rceil) \\ &\le R(x, (\log x^{1/2})^{1/\epsilon}) \\ &< 8x^{2\epsilon \log\log(x^2)/\log\log(x^{1/2})}. \end{aligned}$$

Note that if $x \ge e^{32}$, then $\log\log(x^2)/\log\log(x^{1/2}) \le 3/2$. Thus if $k$ is that positive integer with $x^{2^{-k}} < e^{32} \le x^{2^{-(k-1)}}$ then

$$\begin{aligned} E(x,\epsilon) &= \sum_{i=0}^{k-1} (E(x^{2^{-i}},\epsilon) - E(x^{2^{-i-1}},\epsilon)) + E(x^{2^{-k}},\epsilon) \\ &< \sum_{i=0}^{k-1} 8(x^{2^{-i}})^{3\epsilon} + E(e^{32},\epsilon) \\ &< 9x^{3\epsilon} + e^{32}, \end{aligned}$$

which concludes the proof of the theorem.

We remark that Theorem 6.2 can also be used in the context of Algorithm 4.1 to give a small improvement in that algorithm for most primes.

We now give an algorithm similar to Algorithm 5.1.

**Algorithm 6.2.** *Suppose we are input a positive number $\epsilon$ and an integer $n > 4$ with $2 \le (\log n)^{2/\epsilon} < n$. This algorithm attempts to decide if $n$ is prime or composite.*

**Step 1.** *Using trial division, find the largest divisor $F$ of $n-1$ composed of primes up to $(\log n)^{1+1/\epsilon}$. If $F \le n^{\epsilon}$, then stop for the algorithm has failed.*

**Step 2.** *Use Algorithm 6.1 on $n$, $F$, $\epsilon$, terminating with failure if the parameter $a$ exceeds $\lceil (\log n)^{1/\epsilon} \rceil$.*

From the proof of Theorem 5.2, for each number $\epsilon$ with $0 < \epsilon < 1$, there is a number $x_2(\epsilon)$, such that if $x \ge x_2(\epsilon)$, the number $N$ of primes $p \le x$ for which $p-1$ has a $(\log p)^{1+1/\epsilon}$-smooth divisor exceeding $p^{\epsilon}$ satisfies $N > 2x^{1-\epsilon^2}$. For such primes $p$ we make it past Step 1 of Algorithm 6.2. From Theorem 6.3 we have that if $0 < \epsilon \le 1/4$, then at least half of these primes are proved prime in Step 2 of Algorithm 6.2, though $x_2(\epsilon)$ may have to be adjusted. We thus have the following result.

**Theorem 6.4.** *Let $\epsilon$ be any number with $0 < \epsilon \le 1/4$. There is a number $x_2(\epsilon)$ such that if $x \ge x_2(\epsilon)$ then the number of primes $p \le x$ which Algorithm 6.2 proves prime exceeds $x^{1-\epsilon^2}$.*

We remark that the running time of Algorithm 6.2 is $O((\log n)^{2+1/\epsilon})$ arithmetic steps with integers at most $n$. Thus Theorem 6.4 improves on Theorem 5.1 since there if one wants to prove prime $x^{1-\epsilon^2}$ primes up to $x$, the bound for the running time is about $(\log n)^{1/(3\epsilon^2)}$.

## References


1. L. M. Adleman and M.-D. Huang, Primality testing and abelian varieties over finite fields, Lecture Notes in Mathematics **1512**, (1992), Springer-Verlag, Berlin.
2. W. R. Alford, A. Granville, and C. Pomerance, There are infinitely many Carmichael numbers, Ann. of Math. **140** (1994), 703–722.
3. W. Bosma and M.-P. van der Hulst, Primality proving with cyclotomy, Ph.D. thesis, Amsterdam (1990).
4. J. Brillhart, M. Filaseta, and A. Odlyzko, On an irreducibility theorem of A. Cohn, Can. J. Math. **33** (1981) 1055–1099.
5. J. Brillhart, D. H. Lehmer and J. L. Selfridge, New primality criteria and factorizations of $2^m \pm 1$, Math. Comp. **29** (1975) 620–647.
6. J. P. Buhler, R. E. Crandall and M. A. Penk, Primes of the form $n! \pm 1$ and $2 \cdot 3 \cdot 5 \cdots p \pm 1$, Math. Comp. **38** (1982) 639–643.
7. E. R. Canfield, P. Erdős and C. Pomerance, On a problem of Oppenheim concerning "factorisatio numerorum", J. Number Theory **17** (1983) 1–28.
8. H. Davenport, *Multiplicative number theory*, 2nd ed., Springer-Verlag, New York, 1980.
9. P. Erdős and J. H. van Lint, On the number of positive integers $\leq x$ and free of prime factors $> y$, Simon Stevin, **40** (1966/67) 73-76.
10. M. R. Fellows and N. Koblitz, Self-witnessing polynomial-time complexity and prime factorization, Designs, Codes and Cryptography, **2** (1992) 231-235.
11. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. **261** (1982) 515–534.
12. H. W. Lenstra, Jr., Miller's primality test, Information Processing Letters **8** (1979), 86–88.
13. F. Pappalardi, On Artin's conjecture for primitive roots, Ph.D. thesis, McGill University, (1993).
14. J. Pintz, W. L. Steiger and E. Szemerédi, Infinite sets of primes with fast primality tests and quick generation of large primes, Math. Comp. **53** (1989) 399–406.
15. C. Pomerance, Very short primality proofs, Math. Comp. **48** (1987) 315–322.
16. V. R. Pratt, Every prime has a succinct certificate, SIAM J. Comput. **4** (1975) 214–220.
17. J. B. Rosser and L. Schoenfeld, Approximate formulas for some functions of prime numbers, Illinois J. Math. **6** (1962) 64–94.
18. R. Solovay and V. Strassen, A fast Monte Carlo test for primality, SIAM J. Comput. **6** (1977) 84–85; *erratum* **7** (1978), 118.
19. I. M. Vinogradov, On the bound of the least non-residue of $n$th powers, Bull. Acad. Sci. USSR **20** (1926) 47–58 (= Trans. Amer. Math. Soc. **29** (1927), 218–226).