MATHEMATICS OF COMPUTATION
VOLUME 53, NUMBER 187
JULY 1989, PAGES 399–406

Infinite Sets of Primes with Fast Primality Tests and Quick Generation of Large Primes

By János Pintz*, William L. Steiger**, and Endre Szemerédi

Abstract. Infinite sets P and Q of primes are described, $P \subset Q$. For any natural number n it can be decided if $n \in P$ in (deterministic) time $O((\log n)^9)$. This answers affirmatively the question of whether there exists an infinite set of primes whose membership can be tested in polynomial time, and is a main result of the paper. Also, for every $n \in Q$, we show how to randomly produce a proof of the primality of n. The expected time is that needed for $1\frac{1}{2}$ exponentiations mod n. We also show how to randomly generate k-digit integers which will be in Q with probability proportional to k^{-1}. Combined with the fast verification of $n \in Q$ just mentioned, this gives an $O(k^4)$ expected time algorithm to generate and certify primes in a given range and is probably the fastest method to generate large certified primes known to belong to an infinite subset. Finally, it is important that P and Q are relatively dense (at least $cn^{2/3}/\log n$ elements less than n). Elements of Q in a given range may be generated quickly, but it would be costly for an adversary to search Q in this range, a property that could be useful in cryptography.

1. Introduction. A leading problem in computational number theory is primality testing: given an integer $n > 1$, decide whether or not n is prime. The crudest algorithm checks each integer m, $2 \le m \le \sqrt{n}$. As soon as one divides n it answers "no"; otherwise n is prime. This algorithm is poor because it can take $O(\sqrt{n})$ "steps", which is exponential in the input size of $\lceil \log n \rceil$ bits (all logs are base 2). Asymptotically it would be too slow to be practical. A good algorithm from the complexity standpoint would have to run in time that is polynomial in the input size, or in $O((\log n)^k)$ steps, k fixed.

Miller [8] gives an algorithm which, assuming the truth of the General Riemann Hypothesis (GRH), can check if n is prime in $O((\log n)^5)$ steps. Although the running time is acceptable, a major disadvantage is that we do not know if GRH is true. The best unconditional result is by Adleman, Pomerance and Rumely [2], in which the primality of n may be decided in $O((\log n)^{c \log\log\log n})$ steps.

In light of the discouraging fact that we do not know whether primality may be tested quickly, two different approaches arise, one deterministic, the other probabilistic. The first seeks subsets of primes whose membership may be decided in polynomial time. The second seeks random algorithms which have fast expected running times.

Received December 28, 1987; revised June 14, 1988.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 11Y11, 11Y16.
*Research partially supported by Hungarian National Foundation for Scientific Research Grant #1811.
**This author acknowledges the Institut für Ökonometrie und Operations Research, Universität Bonn, where some of this work was done.

With regard to the first, it is not yet known whether there even exists an infinite subset of the primes whose membership may be decided quickly. The first main result of the present paper gives an affirmative answer. In the next section we exhibit a reasonably dense set of primes P for which "$n \in P$?" may be answered in $O((\log n)^9)$ steps. To be more precise, a step will now be taken as an arithmetic operation on binary digits. Thus, to add integers x and y, $x, y \leq n$ takes $O(\log n)$ steps, and to multiply x and y, $O((\log n)^2)$ steps. In fact, using the fast Fourier transform, Schönhage and Strassen [12] show how to multiply x and y in $O(\log n \log_2 n \log_3 n)$ steps, where $\log_j$ is the jth iterated logarithm. We will often use this fact when we reiterate a complexity statement, and put the improved bound implied by [12] in a pair of curly brackets $\{\{-\}\}$. To simplify the writing, we will use $O((\log n)^{1+\varepsilon})$ for the bit complexity of fast multiplication of x and y.

The second direction seeks probabilistic algorithms to decide primality that have fast expected running times. Rabin [11] proposed a Monte-Carlo method to quickly test primality. Given an integer n, in expected time $O((\log n)^3)$ $\{\{O((\log n)^{2+\varepsilon})\}\}$ it either correctly declares n composite via a witness of length $O(\log n)$, or else concludes that n is probably prime. The latter case could occur if n were really composite, although this is very unlikely. Because the assertion "n is composite" is proved by a witness, the test is really one for compositeness. It could be a drawback for some purposes that the algorithm may erroneously accept n as prime.

The Las Vegas algorithm of Goldwasser and Kilian [5] uses properties of elliptic curves to decide the primality of n. Either n is correctly declared composite or for almost all primes, a sequence $p_1, \ldots, p_k$ of primes is generated, $n > p_1 > \cdots > p_k$, $k = O(\log n)$. The sequence, called the certificate, may be independently checked for primality and the primality of n verified. It was shown in [5] that for almost all primes the algorithm runs in time $O((\log n)^{9+t})$, $t \geq 1$. Unfortunately, it is not known how this algorithm behaves on the exceptional set of primes. Finally, Adleman and Huang [1] have given an algorithm that can decide primality in all cases in expected polynomial time, but at the expense of a slower expected running time than that of [5].

The second main result of the present paper merges the two approaches. We describe a reasonably dense set of primes, Q, from which elements in a given range may be quickly generated and proved prime. The main ingredient is a Las Vegas algorithm for testing $n \in Q$. If n really is in Q it will find a certificate of length $O(\log n)$ in expected time $O((\log n)^3)$, the time needed for $1\frac{1}{2}$ exponentiations mod n. This algorithm is used in conjunction with a simple method to randomly generate k-digit integers which will belong to Q with probability proportional to k^{-1}. Altogether then, in expected time $O(k^4)$, we will generate a k-digit element of Q and also give a certificate to prove its primality. Rabin's compositeness test could also be used in generating large probable primes, if a random odd integer in the appropriate range were tested by his algorithm. The advantage is that *any* prime might be "generated" in this way and the complexity is of the same order as our method. The drawback is that a composite number might be produced.

Because the density of Q is relatively high ($|\{n \in Q\colon n \leq x\}| \geq cx^{2/3}/\log x$, $x > x_0$), this set Q could be useful in cryptographic applications. For example, if the encryptor were using primes in Q with about 100 digits, the code breaker would

need to search about 10^{63} members of Q in this range. Other easily tested subsets of primes do not have the nice density property. For example, the code breaker could easily test the Mersenne primes ($n = 2^p - 1$ for certain primes p) using the Lucas-Lehmer test (see [13]), since there are only 13 Mersenne primes of 100 digits or less and it is not even known if they form an infinite subset. Other subsets of primes have been used in fast generation of large, certified primes. The costs may be less than with our method, but only by a small factor of proportionality. However, none of these sets has so far been proved infinite. Section 5 discusses the fast generation of elements of Q.

Only elementary number theory is needed to show that membership of P and Q may be decided quickly. The main tool is the Brillhart, Lehmer, Selfridge $n-1$ test [4]. The details appear in Section 3. But to establish the density of P and Q, we needed to use deep results from analytic number theory. Section 4 is devoted to these arguments.

2. The Sets P and Q. Define the sets

$$
\begin{aligned}
I_m &= \{n\colon 27^{m-1} < n < 27^m,\ n \equiv 1\ (3^m)\},\\
Q_m &= \{\text{primes } p \in I_m\},\\
P_m &= \{p \in Q_m\colon \exists l,\ 1 < l \le c\log^6 p,\ l^{(p-1)/3} \not\equiv 1\ (p)\},
\end{aligned}
\tag{1}
$$

where c is a suitably chosen absolute constant, and write

$$
Q = \bigcup_{m>3} Q_m, \qquad P = \bigcup_{m>3} P_m. \tag{2}
$$

We will show that P is an infinite set of primes whose membership can be tested quickly and that Q is a larger set from which primes in a given range may be quickly generated.

First, as a brief motivation for these definitions, we note that Dirichlet (1837) showed that if $(c,d) = 1$, the arithmetic progression $a_j = c + jd$ contains infinitely many primes. Taking $c = 1$ and d to be a prime power p^k (as in I_m), the a_j may be tested quickly for primality as long as j is not too large. For example, $j \le p^{2k}$ implies that p^k, the factored part of $a_j - 1$, satisfies $p^k \ge a_j^{1/3}$, and then the Brillhart, Lehmer, Selfridge test may be applied to quickly check a_j for primality.

This would not be useful to us unless there would be a reasonable density of primes in the beginning segments of such progressions. It is perhaps fortuitous that a result of Barban, Linnik, and Tshudakov implies that once $a_j > c(p^k)^{8/3+\varepsilon}$, the sequence already has a high density of primes. Taking $p = 3$ and noting that $c(3^m)^{8/3+\varepsilon} < 27^{m-1}$ for large enough m, we see that I_m has two crucial properties. First, its elements can be tested quickly for primality. Secondly, many primes will be found. The first assertion is easily established in the next section. The other is dealt with in Section 4.

3. Testing Membership in P and Q. The following statements appear in a more general form in Lenstra [7]. They help elucidate some of the properties of these sets.

LEMMA A. *Let $n \le 27^m$. If there is an integer a satisfying $a^{3^m} \equiv 1\ (n)$ and $(a^{3^{m-1}} - 1, n) = 1$, then n is a prime or a product of two primes $\equiv 1\ (3^m)$.*

Proof. If r is a prime dividing n, then $a^{3^m} \equiv 1\ (r)$, and $a^{3^{m-1}} \not\equiv 1\ (r)$. Thus 3^m divides $r-1$. The condition $n \leq 27^m$ guarantees that n has at most 2 prime factors $\equiv 1\ (3^m)$. □

Remark 1. If $n = k3^m + 1$ is prime and $l^{(n-1)/3} \not\equiv 1\ (n)$, then $a = l^k$ satisfies the conditions of Lemma A (and such an l exists if n is a prime). It is called a certificate of the primality of n.

A simple fact about composite $n \in I_m$ which satisfy the conditions of Lemma A is

LEMMA B. *Suppose* $n = (x3^m + 1)(y3^m + 1) \leq 27^m$, $x, y \geq 1$, *and also that* $n = C \cdot 9^m + D \cdot 3^m + 1$, *where* $0 < C < 3^m$ *and* $1 \leq D \leq 3^m$. *Then* $xy = C$ *and* $x + y = D$.

Proof. The relations $(x-1)(y-1) \geq 0$ and $n \leq (3^m)^3$ imply that $0 < x + y \leq xy + 1 \leq 3^m - 1 + 1 = 3^m$. On the other hand, it is clear that $n \equiv D3^m + 1 \pmod 9^m$ and also $n \equiv (x+y)3^m + 1 \pmod 9^m$, so $x + y \equiv D \pmod 3^m$ and therefore $x + y = D$. Also, $xy9^m = n - 1 - (x+y)3^m = n - 1 - D \cdot 3^m = C9^m$, so $C = xy$ as well. □

The point of Lemma B is that, if $n \leq 27^m$ has both representations, $D^2 - 4C$ is the square $(x-y)^2$, and conversely. Thus, we have the

COROLLARY. *If* $n = C \cdot 9^m + D \cdot 3^m + 1$, *where* $0 < C < 3^m$, $1 \leq D \leq 3^m$, *then* n *can be written with natural numbers* x *and* y *in the form* $n = (x3^m + 1)(y3^m + 1)$ *if and only if* $D^2 - 4C$ *is a square.*

The facts we need from [7] may now be summarized by the following statement:

THEOREM A. *Let* $n = C \cdot 9^m + D \cdot 3^m + 1$ *with* $0 < C < 3^m$, $1 \leq D \leq 3^m$. *Then* n *is a prime if and only if both of the following conditions hold:*

(3)
(i) $D^2 - 4C$ *is not a square.*
(ii) *There exists* l *with* $l^{n-1} \equiv 1\ (n)$ *and* $(l^{(n-1)/3} - 1, n) = 1$.

Condition (i) distinguishes primes satisfying (ii) from products of such integers. To test if $n \in P_m$ for the appropriate m, the following procedure exploits Theorem A.

ALGORITHM 1.

[1] Find C, D: $n = C9^m + D3^m + 1$, $0 < C < 3^m$, $1 \leq D \leq 3^m$.
[2] If $D^2 - 4C$ is a square, "n is composite".
[3] Else test each l, $1 < l \leq c(\log n)^6$, for $l^{n-1} \equiv 1\ (n)$ and $(l^{(n-1)/3} - 1, n) = 1$.
If yes, "$n \in P_m$" by certificate l.
[4] Else "$n \notin P_m$".

If $n \in Q_m$, $m > 3$, it has a representation as in line [1]. Line [2] eliminates composite n which could satisfy line [3].

For every l tested in line [3], $l^{(n-1)/3}$ may be computed in $O(\log n)$ multiplications and the gcd needs $O(\log n)$ additions/subtractions. Using $O((\log n)^2)$ steps as the cost of each multiplication and the fact that there are $O((\log n)^6)$ values of l to check, we see that the algorithm terminates in time $O((\log n)^9)$. If fast multiplication were used, the cost could be reduced to $O((\log n)^{8+\varepsilon})$ steps, which we

denote by $\{\{O((\log n)^{8+\varepsilon})\}\}$. To test $n \in P$, find the appropriate m and then use Algorithm 1. Once l has been found, the assertion "n is prime because of l" may be verified in $O((\log n)^3)$ steps $\{\{O((\log n)^{2+\varepsilon})\}\}$. Therefore:

THEOREM 1. *Algorithm* 1 *terminates in* $O((\log n)^9)$ *time* $\{\{O((\log n)^{8+\varepsilon})\}\}$ *and either*

(a) *shows that* $n \notin P$, *or*

(b) *gives a certificate* l *of length* $O(\log_2 n)$ *for the primality of* n, *which can be verified correct in time* $O((\log n)^3)\{\{O((\log n)^{2+\varepsilon})\}\}$.

The most interesting point remains to be proved, namely that P is infinite. This will follow from

· Q *is an infinite set* (see Lemma 1 of the next section), and

· *most primes in* Q *are also in* P (see Lemma 4 of the next section).

If $n = p$ is a prime, condition (ii) of Theorem A requires a value l satisfying $(l^{(p-1)/3} - 1, p) = 1$. Taking any primitive root g and writing $l \equiv g^b \ (p)$, this condition is equivalent to $3 \nmid b$, a relation satisfied by $2/3$ of all reduced residue classes mod p. Given $n \in Q_m$, a random choice of l from the uniform distribution on $1, \ldots, n-1$ would give an l which satisfies $(l^{(n-1)/3} - 1, n) = 1$ with probability $2/3$. This motivates the procedure for checking if $n \in Q_m$.

ALGORITHM 2.

[1] Find C, D: $n = C9^m + D3^m + 1$, $0 < C < 3^m$, $1 \leq D \leq 3^m$.
[2] If $D^2 - 4C$ is a square, "n is composite" and STOP.
[3] Repeat (at most) k times:
    Randomly choose l, $1 \leq l \leq n-1$.
    If (3) holds, "$n \in Q_m$" and STOP.
[4] Else "n is probably composite".

Once again, the cost of lines [1] and [2] is less than that of the loop in line [3]. Therefore, using this procedure, every element of Q_m can be proved to be prime in $O((\log n)^3)$ expected time.

THEOREM 2. *Algorithm* 2 *terminates for every* $n \in Q$ *in expected time* $O((\log n)^3)$, $\{\{O((\log n)^{2+\varepsilon})\}\}$ *and gives a certificate* l *of length* $O(\log n)$ *for the primality of* n, *which can be verified correct in* $O((\log n)^3)$ *time* $\{\{O((\log n)^{2+\varepsilon})\}\}$.

Remark 2. The expected number of exponentiations needed to obtain a certificate l for the primality of $n \in Q_m$ is $3/2$. This may be improved by using an odd prime $p > 3$ in place of 3: Now I_m would be the integers between $(p^3)^{m-1}$ and $(p^3)^m$ which are congruent to 1 $(\text{mod } p)^m$; a randomly chosen residue class l will satisfy $l^{n-1} \equiv 1 \ (n)$ and $(l^{(n-1)/p} - 1, n) = 1$ with probability $(p-1)/p$, so we expect to find a certificate for the primality of $n \in Q_m$ in $p/(p-1)$ exponentiations. It is hard to imagine a method that could certify the primality of n with less work than one exponentiation. Incidentally, Pomerance [10] has recently shown that every prime may be proved to be prime with one exponentiation. However, no fast algorithm to find Pomerance's certificate is known.

The set Q may be used in a very quick random algorithm to generate primes of a given order of magnitude. This will be discussed in Section 5.

4. Density of P and Q. In this section we show that both P and Q ultimately have about $cx^{2/3}/\log x$ elements up to x. If we assume the GRH,

(i) Q_m contains the expected number of primes ($|Q_m| \sim c9^m/m$), and

(ii) $P_m = Q_m$, once m is large enough.

Statement (i) is the prime number theorem for arithmetic progressions; (ii) was proved by Ankeny and Montgomery (cf. [9]), even if $(\log p)^2$ replaced $(\log p)^6$ in the definition of P_m in (1).

The GRH is not necessary for (i). A result of Barban, Linnik and Tshudakov implies

LEMMA 1 (Barban, Linnik, Tshudakov, [3]). *The cardinality of Q_m is given by*

$$|Q_m| = \frac{27^m - 27^{m-1}}{\ln 27^m \phi(3^m)}\left(1 + O\left(\frac{1}{m}\right)\right) = \frac{13/9}{m \ln 27} 9^m \left(1 + O\left(\frac{1}{m}\right)\right),$$

where as usual $\phi(D)$ denotes the number of reduced residue classes mod D. *Thus, $|p \in Q, p \le x| \ge cx^{2/3}/\log x$, $x \ge x_0$, c an absolute constant.*

Therefore, Q is infinite. Moreover, for large m, the density of Q_m in I_m is close to that of the primes in the integers:

$$\frac{|Q_m|}{|I_m|} = \frac{c}{m}(1 + O(m^{-1})).$$

To see that P is also infinite, we relate $|P_m|$ to $|Q_m|$. It seems to be hopeless to establish (ii) by current methods, without using any unproved hypotheses. Instead, we will show (Lemma 4) that $|P_m| = |Q_m|(1 + O(\frac{1}{m}))$, using a modification of the Ankeny-Montgomery argument and known density theorems.

LEMMA 2. *Let χ be a nonprincipal character* mod p. *If $L(s,\chi) \ne 0$ for $s = \sigma + it$, $\sigma > 1 - h$, $|t| \le \log^2 p$, then there exists an l, $1 \le l \le C_1(\log p)^{1/h}$, with $\chi(l) \ne 1$.*

Proof. Write $\rho = \beta + i\gamma$ for the zeros of $L(s,\chi)$. If $\chi(m) = 1$ for all $m \le N$, where $N < p$, we get, as in [9, Theorem 13.1],

$$\begin{aligned}\sum_{n \le N} \chi(n)\left(1 - \frac{n}{N}\right) \wedge (n) &= -\sum_{\rho} \frac{N^\rho}{\rho(\rho+1)} + O(\log p) \\ &= -\sum_{|\gamma| \le \log^2 p} \frac{N^\rho}{\rho(\rho+1)} + O\left(\frac{N}{\log^2 p} \log p\right) + O(\log p) \\ &\ll N^{1-h} \log p + \frac{N}{\log p}.\end{aligned}$$

The prime number theorem implies that the left-hand side is $N/2 + o(N)$, because $\chi(n) = 1$ for $n \le N$, and so we obtain $N^h \ll \log p$, which implies the assertion. □

To see how to use this result, fix any primitive root g_0 mod p and choose the character χ_1 as $\chi_1(g_0) = e^{i \cdot 2\pi/3}$. Letting $l \equiv g_0^a$ (mod p), we notice that

$$l^{(p-1)/3} \not\equiv 1 \ (p) \Leftrightarrow g_0^{a(p-1)/3} \not\equiv 1 \ (p) \Leftrightarrow 3 \nmid a \Leftrightarrow \chi_1(l) \ne 1.$$

In this way, we can demonstrate that an element p of Q_m definitely belongs to P_m as long as the condition of Lemma 2 holds for all nonprincipal characters mod p,

with $h = \frac{1}{6} - C_2/\log_2 x$. Since a nonprincipal character mod p is a primitive one, it is sufficient for $|P_m| = |Q_m|(1 + O(\frac{1}{m}))$ to show, for example, that with $x = 27^m$

$$|Q_m \backslash P_m| \leq \sum_{q \leq x} \sum_{\chi(q)}^{*} N(1-h, \log^2 x, \chi) \ll \frac{x}{3^m \log^2 x}, \tag{4}$$

where $N(\sigma, T, \chi) = \sum 1$, the sum extending over $\{\rho = \beta + i\gamma : L(\rho, \chi) = 0, \beta \geq \sigma, |\gamma| \leq T\}$ and $\sum_{\chi(q)}^{*}$ denotes summation over all primitive characters χ mod q. But this relation is a direct consequence of the following density theorem of Huxley and Jutila.

LEMMA 3 (Huxley and Jutila [6]). *If $\sigma \geq 4/5$, $K \geq 1$, $T \geq 3$, then*

$$\sum_{q \leq K} \sum_{\chi(q)}^{*} N(\sigma, T, \chi) \ll (K^2 T)^{2(1-\sigma)} (\log KT)^{C_3}.$$

Applying this result to the middle term of (4) with $h = \frac{1}{6} - \varepsilon$, $\varepsilon = (\frac{2}{3} + C_3/4) \log\log x / \log x$, we see that

$$\begin{aligned} |Q_m \backslash P_m| &\ll (x^2 \log^2 x)^{1/3 - 2\varepsilon} (\log x)^{C_3} \ll x^{2/3 - 4\varepsilon} (\log x)^{2/3 + C_3} \\ &= \frac{x^{2/3}}{(\log x)^2} = \frac{x}{3^m (\log x)^2}, \end{aligned}$$

as required. This now proves

LEMMA 4. *One has $|P_m| = |Q_m|(1 + O(\frac{1}{m}))$.*

Combined with Lemma 1, this gives

THEOREM 3. *If $x > x_0$ then $|\{p \in Q; p \leq x\}| \geq |\{p \in P; p \leq x\}| > cx^{2/3}/\log x$.*

5. Fast Generation of Primes. We now discuss the generation of k-digit primes. To obtain elements from Q in a given range, we choose the relevant value of m and then randomly search I_m. The density result in Lemma 1 implies that we expect to succeed quickly, namely in $O(m^{-1})$ steps.

ALGORITHM 3.

[1] Randomly choose C, D, $3^{m-3} \leq C < 3^m$, $1 \leq D \leq 3^m$.
[2] Test $n = C9^m + D3^m + 1$ using Algorithm 2.
If n prime, STOP
Else REPEAT [1].

The probability of generating $n \in Q$ in line [1] is $O(\log n)^{-1}$, so we expect to find a prime in $O(\log n)$ queries. Since line [2] takes $O((\log n)^3)$, we have

THEOREM 4. *A k-digit prime may be generated in expected time $O(k^4)$ $\{\{O(k^{3+\varepsilon})\}\}$ along with a certificate l of size $O(k)$, which may be verified in $O(k^3)$ steps $\{\{O(k^{2+\varepsilon})\}\}$.*

Remark 3. Rabin's algorithm might be considered the method of choice in generating large primes. A k-digit odd integer n is chosen at random and tested for primality as follows: A sequence of at most r random integers $a_i < n$ is generated. For each, a test is performed to check if a_i is a "witness" to the compositeness of n. The test is reliable in that a "yes" answer occurs only if n *is* composite. The

test takes $O((\log n)^3)$ steps. If no witness is found in the sequence, n is declared "probably prime". If n were actually composite, the test could accept it as prime, but with probability at most 4^{-r}. If the randomly chosen n really *is* prime, the test would declare "n is prime with probability $\geq 1 - 4^{-r}$". The time for this is that of r exponentiations. By way of contrast, if the randomly chosen $n \in I_m$ is a prime, our method needs the expected time of $3/2$ exponentiations to assert "n is certainly a prime". As mentioned in Remark 2, the constant $3/2$ can be replaced by any $c > 1$. There exist other methods (Pepin's test for Fermat primes, or the Lucas-Lehmer test for Mersenne primes) which can be used in the generation of certified primes and which have expected costs possibly lower than ours, but only by a small multiplicative factor. However, the primes that may be generated are not known to comprise an infinite set.

Remark 4. In an actual implementation of Algorithm 3 one would sieve out those n which have a prime divisor less than $\log n$, say. This would reduce the running time by a factor of $c \log\log n$, or $d \log k$ if n has k digits.

Acknowledgment. We thank the referee, who made many useful comments that helped improve the paper.

Mathematical Institute
Hungarian Academy of Sciences
Realtanoda u. 13–15
Budapest V, Hungary (Pintz and Szemerédi)

Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903 (Steiger and Szemerédi)

1. L. M. ADLEMAN & M. A. HUANG, *Recognizing Primes in Random Polynomial Time*, Proc. 19th Ann. ACM Sympos. on Theory of Computing, 1987, pp. 462–469.

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

3. M. B. BARBAN, JU. V. LINNIK & N. G. TSHUDAKOV, "On prime numbers in an arithmetic progression with a prime-power difference," *Acta Arith.*, v. 9, 1964, pp. 375–390.

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

5. S. GOLDWASSER & J. KILIAN, *Almost All Primes Can Be Quickly Certified*, Proc. 18th Ann. ACM Sympos. on Theory of Computing, 1986, pp. 316–329.

6. M. HUXLEY & M. JUTILA, "Large values of Dirichlet polynomials IV", *Acta Arith.*, v. 32, 1977, pp. 297–312.

7. H. W. LENSTRA, JR., "Primality testing," *Computational Methods in Number Theory*, Math. Centrum Tracts, no. 154, Math. Centrum, Amsterdam, 1982, pp. 55–77.

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

9. H. L. MONTGOMERY, *Topics in Multiplicative Number Theory*, Lecture Notes in Math., Vol. 227, Springer-Verlag, Berlin, 1971.

10. C. POMERANCE, "Very short primality proofs," *Math. Comp.*, v. 48, 1987, pp. 315–322.

→ M. RABIN, "Probabilistic algorithms for testing primality," *J. Number Theory*, v. 12, 1980, pp. 128–138.

12. A. SCHÖNHAGE & V. STRASSEN, "Schnelle Multiplikation grosser Zahlen," *Computing*, v. 7, 1971, pp. 281–292.

13. H. C. WILLIAMS, "Primality testing on a computer," *Ars Combin.* v. 5, 1978, pp. 127–185.