Finding primes & proving primality

How to Prove Primality (single page version)

File: prove1.html "Primality Proving 1: A Quick Introduction" (**Chapter One: Introduction**)

1: Introduction

How do we go about finding primes? And once we have found them, how do we prove they are truly prime? The answer depends on the size of the primes and how sure we need to be of their primality. In these pages we present the appropriate answers in several sections. Let us preview these chapters one at a time.

Chapter Two: The quick tests for small numbers and probable primes

For very small primes we can use the Sieve of Eratosthenes or trial division. These methods are sure, and are the best methods for small numbers, but become far too time consuming before our numbers reach thirty digits.

If we are going to use our primes for "industrial" uses (e.g., for RSA encryption) we often do not need to prove they are prime. It may be enough to know that the probability they are composite is less than 0.000000000000000000000001%. In this case we can use (strong) probable primality tests.

These probable primality tests can be combined to create a very quick algorithm for *proving* primality for integers less than 340,000,000,000,000.

Chapter Three: The classical tests

A quick look at the list of largest known primes shows numbers with hundreds of thousands (even millions) of digits--and these are all **proven** primes (not probable primes)! So how can we know they are prime? Look at a portion of this list and decide what the numbers all have in common.

```
111  189*2^34233-1                  10308 Z  89
112  15*2^34224+1                   10304 D  93 .
113  (5452545+10^5153)*10^5147+1    10301 D  90 Palindrome
114  23801#+1                       10273 C  93 primorial plus one
115  63*2^34074+1                   10260 Y  95
116  213819*2^33869+1               10201 Y  93
```

They are all trivial to factor if we either add, or subtract, one! This is no accident.

It is possible to turn the probable-primality tests of chapter two for an integer *n* into primality proofs, if we know enough factors of either *n*+1 and/or *n*-1. These proofs are called **the classical tests** and we survey them in our third chapter.

These tests have been used for over 99.99% of the largest known primes. They include special cases such as the Lucas-Lehmer test for Mersenne primes and Pepin's Test for Fermat primes.

Chapter Four: The General Purpose Tests

Finally, the obvious problem with the classical tests is that they depend on factorization--and it appears factoring is much harder than primality proving for the "average" integer. In fact this is the key assumption behind the popular RSA encryption method!

Using complicated modern techniques, the classical tests have been improved into tests for general numbers that require no factoring such as the APR, APRT-CL and the ECPP algorithms. In chapter four we say a few words about these methods, discuss which of these test to use (classical, general purpose...), and then leave you with a few references with which to pursue these tests.

In 2002 a long standing question was answered: can integers be prove prime in "polynomial time" (that is, with time bounded by a polynomial evaluated at the number of digits). Some of the previous algorithms come close (ECPP is almost always polynomial, and is conjectured to always be polynomial). Agrawal, Kayal and Saxena answered this question in the affirmative by giving a "simple" polynomial time algorithm. We present this algorithm in chapter four.

File: prove2.html "Primality Proving: Contents of Section Two "The quick tests for small numbers and probable primes"" (Chapter Two (contents))

Chapter 2: The quick tests for small numbers and probable primes

Contents of this Chapter:

1. Finding Very Small Primes
2. Fermat, Probable-Primality and Pseudoprimes
3. Strong Probable-Primality and a Practical Test

This is one of four chapters on finding primes and proving primality. The first is a short introduction and table of contents. The second (these pages) chapter discusses finding small primes and the basic probable primality tests. The third chapter cover the classical primality tests that have been used to prove primality for 99.99% of the numbers on the largest known prime list. The last chapter introduces the general purpose tests that do not require factorization.

File: prove2_1.html "Primality Proving 2.1: Finding very small primes" (Chapter Two > Small Primes)

2.1: Finding very small primes

For finding all the small primes, say all those less than 10,000,000,000; one of the most efficient ways is by using **the Sieve of Eratosthenes** (ca 240 BC):

> Make a list of all the integers less than or equal to *n* (greater than one) and strike out the multiples of all primes less than or equal to the square root of *n*, then the numbers that are left are the primes. (See also our glossary page.)

For example, to find all the odd primes less than or equal to 100 we first list the odd numbers from 3 to 100 (why even list the evens?) The first number is 3 so it is the first odd prime--cross out all of its multiples. Now the first number left is 5, the second odd prime--cross out all of its multiples. Repeat with 7 and then since the first number left, 11, is larger than the square root of 100, all of the numbers left are primes.

This method is so fast that there is no reason to store a large list of primes on a computer--an efficient implementation can find them faster than a computer can read from a disk.

Bressoud has a pseudocode implementation of this algorithm [Bressoud89, p19] and Riesel a PASCAL implementation [Riesel94, p6]. It is also possible to create an even faster sieve based on quadratic forms.

To find individual small primes **trial division** works well. To test *n* for primality (to see if it is prime) just divide by all of the primes less than the square root of *n*. For example, to show is 211 is prime, we just divide by 2, 3, 5, 7, 11, and 13. (Pseudocode [Bressoud89, pp 21-22], PASCAL [Riesel94, pp 7-8].) Sometimes the form of the number *n* makes this especially effective (for examples, Mersenne divisors have a special form).

Rather than divide by just the primes, it is sometimes more practical to divide by 2, 3 and 5; and then by all the numbers congruent to 1, 7, 11, 13, 17, 19, 23, and 29 modulo 30--again stopping when you reach the square root. This type of factorization is sometimes called **wheel factorization**. It requires more divisions (because some of the divisors will be composite), but does not require us to have a list of primes available.

Suppose *n* has twenty-five or more digits, then it is impractical to divide by the primes less than its square root. If *n* has two hundred digits, then trial division is impossible--so we need much faster tests. We discuss several such tests below.

File: prove2_2.html "Primality Proving 2.2: Fermat, probable-primality and pseudoprimes" (Chapter Two > Probable Primes)

2.2: Fermat, probable-primality and pseudoprimes

Fermat's "biggest", and also his "last" theorem states that $x^n + y^n = z^n$ has no solutions in positive integers x, y, z with $n > 2$. This has finally been proven by Wiles in 1995 [Wiles95]. What concerns us here is his "little" theorem:

> **Fermat's (Little) Theorem:** If p is a prime and if a is any integer, then $a^p = a \pmod p$. In particular, if p does not divide a, then $a^{p-1} = 1 \pmod p$. ([proof])

Fermat's theorem gives us a powerful test for compositeness: Given $n > 1$, choose $a > 1$ and calculate a^{n-1} modulo n (there is a very easy way to do quickly by repeated squaring, see the glossary page "binary exponentiation"). If the result is not one modulo n, then n is composite. If it is one modulo n, then n *might* be prime so n is called a weak **probable prime base *a*** (or just an ***a*-PRP**). Some early articles call all numbers satisfying this test pseudoprimes, but now the term **pseudoprime** is properly reserved for composite probable-primes.

The smallest examples of pseudoprimes (composite PRPs) are the following. (There are more examples on the glossary page "probable prime ".)

- 341 = 11.31 is a 2-PRP, (Sarrus 1819)
- 91 = 7.13 is a 3-PRP,
- 217 = 7.31 is a 5-PRP and,
- 25 = 5.5 is a 7-PRP.

There are 1,091,987,405 primes less than 25,000,000,000; but only 21,853 pseudoprimes base two [PSW80], so Henri Cohen joked that 2-PRP's are "industrial grade primes" [Pomerance84, p5]. Fortunately, the larger n, the more likely (on the average) that a PRP test is correct--see the page "How probable?".

It is interesting to note that in 1950 Lehmer, using the weaker definition $a^n = a \pmod n$ for probable/pseudo-prime, discovered 2*73*1103 = 161038 is an even "pseudoprime" base two. See [Ribenboim95 Chpt. 2viii] for a summary of results and history--including a debunking of the Chinese connection. Richard Pinch lists the pseudoprimes to 10^{21} (by various definitions) in at his website.

There may be relatively few pseudoprimes, but there are still infinitely many of them for every base a>1, so we need a tougher test. One way to make this test more accurate is to use multiple bases (check base 2, then 3, then 5,...). But still we run into an interesting obstacle called the Carmichael numbers.

> **Definition:** The composite integer n is a **Carmichael number** if $a^{n-1}=1 \pmod n$ for every integer a relatively prime to n.

Here is the bad news: repeated PRP tests of a Carmichael number will fail to show that it is composite until we run across one of its factors. Though Carmichael number are 'rare' (only 2,163 are less than 25,000,000,000), it has recently been shown that there are infinitely many [AGP94]. The Carmichael numbers under 100,000 are

561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657, 52633, 62745, 63973, and 75361.

Richard Pinch lists the Carmichael's to 10^{16} at his FTP site (see [Pinch93]).

Note: Jon Grantham developed the idea of **Frobenius Pseudoprime** [Grantham2000] to generalize many of the standard types (Fermat, Lucas...), and to make the tests more accurate. His papers are available on-line.

File: prove2_3.html "Primality Proving 2.3: Strong probable-primality and a practical test" (Chapter Two > Strong PRPs)

2.3: Strong probable-primality and a practical test

A better way to make the Fermat test more accurate is to realize that if an odd number n is prime, then the number 1 has just two square roots modulo n: 1 and -1. So the square root of a^{n-1}, $a^{(n-1)/2}$ (since n will be odd), is either 1 or -1. (We actually could calculate which it should be using the Jacobi symbol, see the glossary page on Euler PRP's, but we wish to develop a stronger test here.) If $(n-1)/2$ is even, we can easily take another square root... Let's make this into an algorithm:

> Write $n-1 = 2^s d$ where d is odd and s is non-negative: n is a **strong probable-prime base *a*** (an ***a*-SPRP**) if either $a^d = 1 \pmod n$ or $(a^d)^{2^r} = -1 \pmod n$ for some non-negative r less than s.

Again all integers $n > 1$ which fail this test are composite; integers that pass it *might* be prime. The smallest odd composite SPRP's are the following.

- 2047 = 23.89 is a 2-SPRP,
- 121 = 11.11 is a 3-SPRP,
- 781 = 11.71 is a 5-SPRP and,
- 25 = 5.5 is a 7-SPRP.

A test based on these results is quite fast, especially when combined with trial division by the first few primes. If you have trouble programming these results Riesel [Riesel94, p100] has PASCAL code for a SPRP test, Bressoud has pseudocode [Bressoud89, p77], and Langlois offers C-Code. See the glossary page "Strong PRP" for more information.

It has been proven ([Monier80] and [Rabin80]) that the strong probable primality test is wrong no more than 1/4th of the time (3 out of 4 numbers which pass it will be prime). Jon Grantham's "Frobenius pseudoprimes" can be used to create a test (see [Grantham98]) that takes three times as long as the SPRP test, but is far more than three times as strong (the error rate is less than 1/7710).

Combining these tests to prove primality

Individually these tests are still weak (and again there are infinitely many a-SPRP's for every base $a>1$ [PSW80]), but we can combine these individual tests to make powerful tests for small integers $n>1$ (these tests *prove* primality):

- If $n < 1{,}373{,}653$ is a both 2 and 3-SPRP, then n is prime [PSW80].
- If $n < 25{,}326{,}001$ is a 2, 3 and 5-SPRP, then n is prime [PSW80].
- If $n < 25{,}000{,}000{,}000$ is a 2, 3, 5 and 7-SPRP, then either $n = 3{,}215{,}031{,}751$ or n is prime [PSW80]. (This is actually true for $n < 118{,}670{,}087{,}467$ [Jaeschke93].)
- If $n < 2{,}152{,}302{,}898{,}747$ is a 2, 3, 5, 7 and 11-SPRP, then n is prime [Jaeschke93].
- If $n < 3{,}474{,}749{,}660{,}383$ is a 2, 3, 5, 7, 11 and 13-SPRP, then n is prime [Jaeschke93].
- If $n < 341{,}550{,}071{,}728{,}321$ is a 2, 3, 5, 7, 11, 13 and 17-SPRP, then n is prime [Jaeschke93].

The first three of these are due to Pomerance, Selfridge and Wagstaff [PSW80], the parenthetical remark and all others are due to Jaeschke [Jaeschke93]. (These and related results are summarized in [Ribenboim95, Chpt 2viiib].) In the same article Jaeschke considered other sets of primes (rather than just the first primes) and found these slightly better results:

- If $n < 9{,}080{,}191$ is a both 31 and 73-SPRP, then n is prime.
- If $n < 4{,}759{,}123{,}141$ is a 2, 7 and 61-SPRP, then n is prime.

Finally, Worley (on-web, unpublished) suggests the following,

- If $n < 170{,}584{,}961$ is a 350 and 3958281543-SPRP, then n is prime.
- If $n < 75{,}792{,}980{,}677$ is a 2, 379215, and 457083754-SPRP, then n is prime.
- If $n < 21{,}652{,}684{,}502{,}221$ is a 2, 1215, 34862, and 574237825-SPRP, then n is prime.

To make a quick primality test from these results, start by dividing by the first few primes (say those below 257); then perform strong primality tests base 2, 3, ... until one of the criteria above is met. For example, if $n < 25{,}326{,}001$ we need only check bases 2, 3 and 5. This is much faster than trial division because someone else has already done much of the work, but will only work for small numbers ($n < 10^{16}$ with the data above).

Note that these results can be strengthened by not treating them as separate tests, but rather realizing we are finding square root of -1. For example, $n = 46{,}856{,}248{,}255{,}981$ is a 2 and 7 pseudoprime, but

$$2^{(n-1)/4} = 34456063004337 \pmod{n}, \text{ and}$$
$$7^{(n-1)/4} = 21307242304265 \pmod{n}.$$

The square of both of these is -1. If n were prime, then it would have only two square root and the above would be equal or negatives of each other;

yet gcd(n,34456063004337-21307242304265) = 4840261 and we have factored n.

Finally, there is a fair amount more that could (and should) be said. We could discuss Euler pseudoprimes and their relationship with SPRP's. Or we could switch

to the "plus side" and discuss Lucas pseudoprimes, or Fibonacci pseudoprimes, or the important combined tests... but that would take a chapter of a book--and it has already been well written by Ribenboim [Ribenboim95]. Let us end this section with one last result:

> **Miller's Test** [Miller76]**:** *If the extended Riemann hypothesis is true,* then if n is an a-SPRP for all integers a with $1 < a < 2(\log n)^2$, then n is prime.

The extended Riemann hypothesis is far too complicated for us to explain here--but should it be proven, then we would have a very simple primality test. Until it is proven, we can at least expect that if n is composite, we should be able to find an a that shows it is composite (a witness) without searching "too long." Most surveys cover Miller's test (often with the constant 70 from [Osterle1979] as Miller's article just said $O((\log n)^2)$); the improvable constant 2 is due to Bach [Bach85], see also [CP2001, pp. 129-130]. Note that *heuristically* Bach and Huelsbergen [BH1993] argue that we should be able to replace the bound in Miller's test with a bound near:

$$(\log 2)^{-1} \log n \log \log n.$$

Note that there is no finite set of bases that will work in Miller's test. In fact, if for n composite we let $W(n)$ denote the least witness for n (the least a which shows n is composite), then there are infinitely many composite n with

$$W(n) > (\log n)^{1/(3 \log \log \log n)} \quad [AGP94]$$

File: prove3.html "Primality Proving: Contents of Section Three" (Chapter Three (contents))

Chapter 3: The classical tests

Contents of this Chapter:

1. n-1 Tests (and Pepin's Test for Fermats)
2. n+1 Tests (and the Lucas-Lehmer Test for Mersennes)
3. A Combined Test -- and more

This is one of four chapters on finding primes and proving primality. The first is a short introduction and table of contents. The second chapter

discusses finding small primes and the basic probable primality tests. The third chapter (these pages) cover the classical primality tests that have been used to prove primality for 99.99% of the numbers on the largest known prime list. The last chapter introduces the general purpose tests that do not require factorization.

File: prove3_1.html "Primality Proving 3.1: n-1 tests and Pepin's Test for Fermats" (Chapter Three > *n*-1 Tests)

3.1: *n*-1 tests and Pepin's tests for Fermats

Have you ever looked at the list of largest known primes? The most obvious feature of the largest few thousand primes p is that in almost every case either p-1 or p+1 is trivially factored. Why is that? Because these are the numbers easiest to prove prime! In this section we will show how we can use Fermat like tests for n if we know enough factors of n-1. These are tests that **prove** primality, they do not just suggest that primality is (however highly) probably.

In 1891 Lucas turned Fermat's Little Theorem into a practical primality test. Here is Lucas' test as strengthened by Kraitchik and Lehmer (see [BLS75]):

> **Theorem 1:** Let $n > 1$. If for every prime factor q of n-1 there is an integer a such that
>
> - $a^{n-1} = 1 \pmod n$, and
> - $a^{(n-1)/q}$ is not $1 \pmod n$;
>
> then n is prime.

We will prove this theorem because we have a great deal to learn from it. (If you lose your way here, then just move on to the next theorem--since in this case you must be taking me at my word anyway.)

> **Proof:** To show n is prime we need only show phi(n) = n-1 (here phi(n) is Euler totient function), or more simply, that n-1 divides phi(n). Suppose this is not the case, then there is a prime q and exponent $r>0$ such that q^r divides n-1, but not phi(n). For this prime q we must have an integer a that satisfies the conditions above. Now let m be the order of a modulo n, then m divides n-1 (first condition), but not $(n-1)/q$ (second condition). So q^r divides m which divides phi(n)--a contradiction which proves the theorem.

What did we do in this proof? We looked at a group, $(\mathbf{Z}/n\mathbf{Z})^*$, which, if it had the correct size, n-1, would show n was prime. We then collected enough information (the two conditions) to show the group had the correct size! **This is the basis of all modern primality tests** whether they are as simple as the test above or something as elaborate such as the methods using elliptic curves or number fields.

Theorem 1 requires a complete factorization of n-1. The key to strengthening this result into a form that only requires the factored part of n-1 to be roughly the square root of n-1 was discovered by Pocklington:

> **Pocklington's Theorem** (1914): Let $n-1 = q^k R$ where q is a prime which does not divide R. If there is an integer a such that $a^{n-1} = 1 \pmod n$ and $\gcd(a^{(n-1)/q}-1, n) = 1$, then each prime factor p of n has the form $q^k r+1$.
>
> **Proof.** Let p be any prime divisor of n, and let m be the order of a modulo p. As above m divides n-1 (first condition on a), but not $(n-1)/q$ (second condition); so q^k divides m. Of course m divides p-1 so the conclusion follows.

The result of applying Pocklington's theorem to each prime power factor of n (plus a little more work) is:

> **Theorem 2:** Suppose $n-1 = FR$, where $F>R$, $\gcd(F,R)$ is one and the factorization of F is known. If for every prime factor q of F there is an integer $a>1$ such that
>
> 1. $a^{n-1} = 1 \pmod n$, and
> 2. $\gcd(a^{(n-1)/q}-1, n) = 1$;
>
> then n is prime.

(Notice that different a's can be used for each prime q.) Theorem 2 can be improved even more: if $F<R$, but either every factor of R is greater than $\sqrt{R/F}$; or $n<2F^3$, $R=rF+s$, $0<s<F$, and r is odd or s^2-4r is not a square; then n is prime. If you are interested in these theorems, then it is well worth going to the source: [BLS75].

Before we switch to the plus side tests, let me quote a few classical cases of theorem 2.

> **Pepin's Test** (1877): Let F_n be the nth Fermat number (so $F_n = 2^{2^n}+1$) with $n>1$. F_n is prime if and only if $3^{(F_n-1)/2} = -1 \pmod{F_n}$.
>
> **Proof**. If $3^{(F_n-1)/2} = -1 \pmod{F_n}$, then F_n is prime by theorem 2 with $a = 3$. If instead F_n is prime, then $3^{(F_n-1)/2} = (3|F_n) \pmod{F_n}$ where $(3|F_n)$ is the Jacobi symbol. It is easy to check that $(3|F_n) = -1$.
>
> **Proth's Theorem** (1878): Let $n = h\cdot 2^k+1$ with $2^k > h$. If there is an integer a such that $a^{(n-1)/2} = -1 \pmod n$, then n is prime.
>
> **Theorem 3** ("Well Known"): Let $n = h\cdot q^k+1$ with q prime and $q^k > h$. If there is an integer a such that $a^{n-1} = 1 \pmod n$, and $\gcd(a^{(n-1)/q}-1, n) = 1$, then n is prime.

Perhaps the best single source source of information on the classical tests is Hugh Williams book "ï¿½douard Lucas and Primality Testing"

[Williams98]. Other useful sources include "the" n^2-1 article: [BLS75], and the standard surveys (such as [BLSTW88], [Ribenboim95] and [Riesel94]). These surveys include pointers to the results which use the factorization of other polynomials in n such as n^6-1, most developed by Williams and his associates [Williams78, Williams98].

These theorems have been implemented and are available for you to use on most computer platforms. For example, look at Yves Gallot's Proth.exe and Chris Nash's PrimeForm).

File: prove3_2.html "Primality Proving 3.2 n+1 tests and the Lucas-Lehmer test" (Chapter Three > n+1 Tests)

3.2: n+1 tests and the Lucas-Lehmer test

About half of the primes on the list of the largest known primes are of the form N-1, where N (the prime plus one) is trivial to factor, why is that? It is because there is a theorem similar to Fermat's Little theorem that we can use here--but first we must do a little ground work. Again you may skip the details and go straight to the theorem if you must, but you'll miss most of the fun!

Suppose we choose integers p and q such that p^2-4q is **not a square** modulo n, then the polynomial x^2-px+q has distinct zeros, one of which is r = (p+sqrt(p^2-4q))/2, and it is easy (by induction) to show r's powers have the form

Lemma 1: $r^m = (V(m) + U(m)\text{sqrt}(p^2\text{-}4q))/2$

where U and V are defined recursively by

$U(0) = 0, \quad U(1) = 1, \quad U(m) = pU(m\text{-}1) - qU(m\text{-}2)$
$V(0) = 2, \quad V(1) = p, \quad V(m) = pV(m\text{-}1) - qV(m\text{-}2)$

These are the **Lucas sequences** associated with p and q. A well known special case is given by letting p=1, q=-1, then U(m) is the sequence of Fibonacci numbers.

These Lucas sequences have many properties (such as the following) which make them very fast to calculate (in a way analogous to how we calculate x^m by repeated squarings):

$U(2m) = U(m)V(m)$
$V(2m) = V(m)^2\text{-}2q^m$

(See [BLSTW88] or better [Ribenboim95, chpt2, iv].)

Now we are ready to state our analog to Fermat's Little Theorem (keep lemma 1 in mind while reading this theorem):

> **Lemma 2:** (With p, q and r as above so p^2-4q is not a square mod n), let $2r = a+b\text{sqrt}(p^2-4q) \pmod n$ for integers a and b of the same parity. If n is prime, then $2r^n = a-b\text{sqrt}(p^2-4q) \pmod n$.

That's too messy, lets restate it using our sequence U (the coefficient of sqrt(p^2-4q)) from above. To do this notice that lemma 2 essentially says that r^n is the complex conjugate of r^1 modulo n, so multiply them together.

> **Lemma 3:** (With p, q as above) if n is prime, then $U(n+1) = 0 \pmod n$.

Now we can restate theorem 1 for the plus side:

> **Theorem 4:** Let $n > 1$ be an odd integer. If there is an integer d for which the Jacobi symbol $(d|n) = -1$ and for every prime factor r of $n+1$ there are relatively prime integers p and q with $p^2-4q = d$ such that
>
> - $U(n+1) = 0 \pmod n$, and
> - $U((n+1)/r)$ is not $0 \pmod n$;
>
> then n is prime.

Note that you may use different p's and q's as long as the discriminant d does not change. One way to alter p and q (but not d) is to replace (p,q) by $(p+2,p+q+1)$.

An interesting example of this test is found by setting $S(k) = V(2^{k+1})/2^{2^k}$

> **Lucas-Lehmer Test** (1930): Let n be an odd prime. The Mersenne number $M(n) = 2^n-1$ is prime if and only if $S(n-2) = 0 \pmod{M(n)}$ where $S(0) = 4$ and $S(k+1) = S(k)^2-2$.

(The proof of sufficiency is found on a separate page.) This test is exceptionally fast on a binary computer because it requires no division. It is also so easy to program that in 1978 two high school students, with little understanding of the mathematics behind the test, were able to use it to find the then record Mersenne prime $2^{21701}-1$ (see our page on Mersennes).

It is also easy to give a test paralleling Pocklington's theorem using Lucas sequences. This was first done by D. H. Lehmer in 1930 (in the same article he introduced the Lucas-Lehmer test: [Lehmer30]). See [BLSTW88] or [BLS75] or ... for more information on these tests.

Joerg Arndt notes that a striking (but computationally useless) way to state this test is as follows:

> **Theorem:** $p=2^n-1$ is prime if and only if p divides $\cosh(2^{n-2}\log(2+\sqrt{3}))$.

Lucas also stated one case of his theorem in this manner.

File: prove3_3.html "Primality Proving 3.3: Combined Tests" (Chapter Three > Combined Tests)

3.3: Combined Tests

In previous sections we have pointed out if the factored portion of n-1 or of n+1 is larger than the cube root of n, then we can prove n is prime. In this section we discuss the case that the product of these two factored portions is greater than the cube root of n, then we can prove n prime using a combined test. (If we can not find enough factors to prove n prime this way, then we must use the generalized tests of the following chapter.)

Let $n > 1$ be an odd integer. Let $n-1 = F_1R_1$ and $n+1 = F_2R_2$ where F_1 and F_2 are completely factored, and $\gcd(F_1,R_1) = \gcd(F_2,R_2) = 1$. The two types of tests we have applied to n in the previous sections are as follows:

> **Condition I**. For each prime p dividing F_1 there is an integer a such that
>
> - $a^{n-1} = 1 \pmod n$, but
> - $\gcd(a^{(n-1)/p}-1, n) = 1$.
>
> **Condition II.** Let $(d|n) = -1$. For each prime p dividing F_2 there is a Lucas sequence with discriminant d such that
>
> - $U(n+1) = 0 \pmod n$, and
> - $\gcd(U((n+1)/p), n)=1$.

Pocklington's theorem tells us that if (I) is true, then each prime factor q of n has the form $k\cdot F_1+1$. About 60 years later Morrison proved that if (II) held, then each prime factor q of n has the form $k\cdot F_2\pm1$ [Morrison75]. Together these give us the following:

> **Combined Theorem 1:** Suppose n, F_1, F_2, R_1, R_2 are as above and conditions (I) and (II) are satisfied. If $n < \max(F_1^2F_2/2 , F_1F_2^2/2)$, then n is prime.
>
> **Proof.** Let q be a prime factor of n and let $n = mq$. From Condition I we have that $q = 1 \pmod{F_1}$, so since n is 1 (mod F_1), so is m. From Condition (II) we know $q = \pm1 \pmod{F_2}$, so since n is -1 (mod F_2), either q or m is 1 (mod F_2). We may assume that $m = 1$

(mod F_2), because if every prime factor q of n was 1 (mod F_2), we'd have the contradiction that $n = 1 \pmod{F_2}$. Finally $\gcd(F_1,F_2)=2$, so we can combine these to get that $m = 1 \pmod{F_1F_2/2}$. So for n to be composite we must have both

- $n = qm > (1 + F_1)(1 + F_1F_2/2) > F_1^2F_2/2$, and
- $n = qm > (-1 + F_2)(1 + F_1F_2/2) > F_1F_2^2/2$.

This completes the proof of the theorem.

Adding in a factoring bound

Sometimes, if n is small enough that we have almost enough factors to use the above (or similar) results, it can be helpful to bring in information about how far we have tried to factor $n\pm1$. Suppose, for example, that all of the prime factors of R_1 and R_2 are greater than B. Next apply the conditions above to R_1 and R_2

Condition III. There is an integer a such that

- $a^{n-1} = 1 \pmod{n}$, but
- $\gcd(a^{(n-1)/R_1} - 1, n) = 1$.

Condition IV. Let $(d|n) = -1$. There is a Lucas sequence with discriminant d (same d as used in condition II) such that

- $U(n+1) = 0 \pmod{n}$, and
- $\gcd(U((n+1)/R_2), n)=1$.

These two conditions inform us respectively that every prime factor q of n has the form $k\cdot u+1$ where u is a prime factor of R_1; and every prime factor q also has the form $k\cdot v\pm1$ where v is a prime factor of R_2. (Note that the factors u and v are dependent on q.) Of course u and v must each be larger than the factoring bound B. With the above notation we can now state our final classical theorems. (For the first the proof is virtually identical to the proof above.)

Combined Theorem 2: Suppose n, F_1, F_2, R_1, R_2, B are as above and conditions (I) through (IV) are satisfied. Define integers r and s by $R_1 = sF_2/2 + r$ with $0 \le r < F_2/2$. If

$$n < \max(B\cdot F_1 + 1, B\cdot F_2 - 1)\,(B^2F_1F_2/2 + 1)$$

then n is prime.

Combined Theorem 3: Suppose n, F_1, F_2, R_1, R_2, B are as above and conditions (I) through (IV) are satisfied. Again define integers r and s by $R_1 = sF_2/2 + r$ with $0 \le r < F_2/2$. If for some integer m

$$n < (m \cdot F_1F_2 + r \cdot F_1 + 1)\,(B^2F_1F_2/2 + 1)$$

then either n is prime or $kF_1F_2 + rF_1 + 1$ divides n for some non-negative integer $k < m$.

Both of these results (and more) can be found in "the" paper on the classical results: [BLS75]. Another excellent source on these theorems and their extensions is the excellent text H. Williams "Édouard Lucas and Primality testing" [Williams98]

How much further can we go? It is possible to consider higher powers such as the factors of

$$n^6 - 1 = (n - 1)(n^2 + n + 1)(n + 1)(n^2 - n + 1).$$

(See [Williams78] for the theory and examples of these techniques). But the cost in terms of mathematical complication is very high. So in practice adding a few terms such as n^2+n+1 or n^2-n+1 is rarely worth the effort. Rather it makes sense to just move on to the general primality proving methods of the next chapter.

File: prove4.html "Primality Proving Section Four "The General Purpose Tests"" (Chapter Four (contents))

Chapter 4: The general purpose tests

Contents of this Chapter:

1. The Neoclassical Tests, especially APR and APR-CL
2. Using Elliptic Curves, especially the ECPP Test
3. A Polynomial Time Algorithm
4. Conclusion and Suggestions

This is one of four chapters on finding primes and proving primality. The first is a short introduction and table of contents. The second chapter discusses finding small primes and the basic probable primality tests. The third chapter cover the classical primality tests that have been used to prove primality for 99.99% of the numbers on the largest known prime list. The last chapter (these pages) introduces the general purpose tests that do not require factorization.

File: prove4_1.html "Primality Proving 4.1: Extending the classical tests" (Chapter Four > APR and APR-CL)

4.1 Neoclassical tests: APR and APR-CL

How do we improve on the tests of the previous chapter? The way Williams and others began in the 70's was to use the factors of other polynomials of n such as n^2+1, n^2+n+1 and n^2-n+1 [Williams78]. But why stop there? Why not try n^m-1 for a higher exponent m such as 5040, then every prime q such that $q-1$ divides 5040 (which does not divide n) must divide $n^{5040}-1$ (by Fermats Little Theorem).

This (with much more cleverness) makes (for $m = 5040$) a product of primes q which is greater than 10^{52}. So after we show there are theorems similar to the classical theorems which only require a factorization to the square root of n, then using this same m, 5040, we will be done for all numbers with less than 100 digits--without any (explicit) factoring (the same q's work for all of these n's).

What about even larger N's? It is always possible to find the necessary factors. In fact it has been shown that there is always an integer m with

$$m < (\log n)^{\log\log\log n}$$

for which the factors q dividing n^m-1 with $q-1$ dividing m, have a product at least the size of the square root of n. Usually m is around 100,000,000 for numbers n with about 3,000 digits.

This is roughly (very roughly!) how Adleman, Pomerance and Rumely began the modern age of primality testing by introducing the **APR** primality test [APR83] in 1979. The running time of their method is almost polynomial--its running time t is bounded as follows

$$(\log n)^{(c_1 \log\log\log n)} < t < (\log n)^{(c_2 \log\log\log n)}$$

(recognize those bounds?)

Soon Cohen and Lenstra [CL84] improved this test into a practical version called **APRT-CL** that handles 100 digit numbers in a matter of seconds (see also [CL87], [Mihailescu98], and [BH90]). (They improved it by replacing the general reciprocity law for the power residue symbol with much easier to calculate Jacobi sums).

It is also possible to mix the two approaches (the classical assuming large factors of $n\pm1$ and the neoclassical above assume many small factors of n^m-1). One example of this mixed approach is Tony Forbes VFYPR (currently limited to 2982 digits).

File: prove4_2.html "Primality Proving 4.2: Elliptic curves and the ECPP test" (Chapter Four > Elliptic Curves)

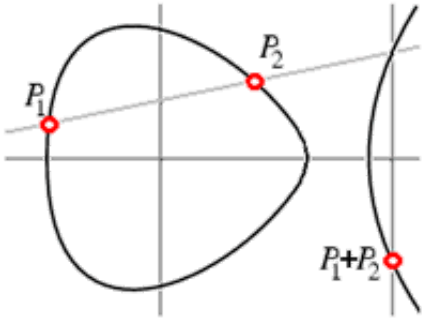
4.2: Using elliptic curves and the ECPP test

What is the next big leap in primality proving? To switch from Galois groups to some other, perhaps easier to work with groups--in this case the points on Elliptic Curves modulo n. An Elliptic curve is a curve of genus one, that is a curve that can be written in the form

E(*a*,*b*) : $y^2 = x^3 + ax + b$ (with $4a^3 + 27b^2$ not zero)

They are called "elliptic" because these equations first arose in the calculation of the arc-lengths of ellipses.

The rational points on such a curve form a group with addition defined using the "chord and tangent method:" That is, if the two points P_1 and P_2 are rational (have rational coefficients), then the line through P_1 and P_2 intersects the curve again in a third rational point which we call $-(P_1+P_2)$ (the negative is to make the associative law work out). Reflect through the x-axis to get P_1+P_2. (If P_1 and P_2 are not distinct, then use the tangent line at P_1.)



If we then reduce this group modulo a prime p we get a small group $E(a,b)/p$ whose size can be used in roughly the way we use the size of $(\mathbf{Z}/p\mathbf{Z})^*$ in the first of the classical tests. Let |E| be the order (size) of the group E:

> **Theorem:** $|E(a,b)/p|$ lies in the interval $(p+1-2\sqrt{p}, p+1+2\sqrt{p})$ and the orders are fairly uniformly distributed (as we vary a and b).

Obviously we are again getting out of our depth, but perhaps you see that we now have replaced the groups of order n-1 and n+1 used in the classical test with a far larger range of group sizes. We can keep switching curves until we find one we can "factor". This improvement comes at the cost of having to do a great deal of work to find the actual size of these groups.

About 1986 S. Goldwasser & J. Kilian [GK86] and A. O. L. Atkin [Atkin86] introduced elliptic curve primality proving methods. Atkin's method, ECPP, was implemented by a number of mathematicians, including Atkin & Morain [AM93]. François Morain's C-code (discussed in [AM93]) is available on the web for many platforms. For Windows based platforms the Primo implementation is easier to use.

Heuristically, the best version of ECPP is $O((\log n)^{4+eps})$ for some $eps>0$ [LL90] (see also D. J. Bernstein's page http://cr.yp.to/primetests.html). It has been proven to be polynomial time for almost all choices of inputs.

File: prove4_3.html "Primality Proving 4.3: A polynomial-time algorithm" (Chapter Four > A Polynomial-Time Algorithm)

4.3: A Polynomial-Time Algorithm

As we mentioned before, many of the primality proving methods are conjectured to be polynomial-time. For example, Miller's test is polynomial if ERH is true (and Rabin gave a version of this test that was unconditionally randomized polynomial-time [Rabin80]). Adleman and Hang [AH1992] modified the Goldwasser-Killian algorithm [GK86] to produce a randomized polynomial time algorithm that always produced a certificate of primality... So it is not surprising that there exists a polynomial-time algorithm for proving primality. But what is surprising is that in 2002 Agrawal, Kayal and Saxena [AKS2002] found a relatively simple *deterministic* algorithm which relies on *no unproved assumptions*. We present this algorithm below then briefly refer to a related algorithm of Bernstein.

The key to AKS' result is another simple version of Fermat's Little Theorem:

> **Theorem:** Suppose that a and p are relatively prime integers with $p > 1$. p is prime if and only if
>
> $$(x-a)^p = (x^p-a) \pmod p$$
>
> **Proof.** If p is prime, then p divides the binomial coefficients ${}_pC_r$ for $r = 1, 2, \ldots p-1$. This shows that $(x-a)^p = (x^p-a^p) \pmod p$, and the equation above follows via Fermat's Little Theorem. On the other hand, if $p > 1$ is composite, then it has a prime divisor q. Let q^k be the greatest power of q that divides p. Then q^k does not divide ${}_pC_q$ and is relatively prime to a^{p-q}, so the coefficient of the term x^q on the left of the equation in the theorem is not zero, but it is on the right.

(This result was used to create a randomized polynomial-time algorithm by Agrawal and Biswas [AB1999].)

Of course in this form it is too difficult to use because there are just far too many coefficients to check. Their idea was to look at the simpler condition:

$$(x-a)^p = (x^p-a) \pmod{x^r-1, p}$$

This must hold if p is prime and it is conjectured (see [BP2001, KS2002]) that if $r > 1$ does not divide p and the above congruence holds, then either p is prime or p^2 is 1 modulo r.

Agrawal, Kayal and Saxena managed to reformulate this into the following algorithm which they proved would run in at most $O((\log n)^{12} f(\log \log n))$ time where f is a polynomial. (This means the time it takes to run the algorithm is at most a constant times the number of digits to the twelfth power times a polynomial evaluated at the log of the number of digits.)

```
Input: Integer n > 1

if (n is has the form a^b with b > 1) then output COMPOSITE
```

```
r := 2
while (r < n) {
    if (gcd(n,r) is not 1) then output COMPOSITE
    if (r is prime greater than 2) then {
        let q be the largest factor of r-1
        if (q > 4sqrt(r)log n) and (n^((r-1)/q) is not 1 (mod r)) then break
    }
    r := r+1
}

for a = 1 to 2sqrt(r)log n {
    if ( (x-a)^n is not (x^n-a) (mod x^r-1,n) ) then output COMPOSITE
}

output PRIME;
```

The proof [AKS2002] is relatively straightforward, and perhaps the most advanced result necessary is a sieve result required to show the necessary q exists for each composite ([F1985], [BH1996]). (Note that the first step, determining if the number is a perfect power, can be done in essentially linear time [Bernstein1998b].)

AKS also showed that if Sophie Germain primes have the expected distribution [HL23] (and they certainly should!), then the exponent 12 in the time estimate can be reduced to 6, bringing it much closer to the (probabilistic) ECPP method. But of course **when actually finding primes it is the unlisted constants[1] that make all of the difference**! We will have to wait for efficient implementations of this algorithm (and hopefully clever restatements of the painful `for loop`) to see how it compares to the others for integers of a few thousand digits. Until then, at least we have learned that there is a polynomial-time algorithm for all integers that both is deterministic and relies on no unproved conjectures!

Note: D. J. Bernstein's exposition of the Agrawal-Kayal-Saxena theorem (mentioned above) contains improvements by many diferent researchers which reduce the constants involved in the time analysis by at least a factor of 2,000,000. This is perhaps the best source for the present state of the algorithm.

Related Approaches and Recent News!

Berrizbeitia [Berrizbeitia2003] found a way to save time in AKS-type primality proofs for some primes n, reducing the exponent from 6+o(1) to 4+o(1). Cheng [Cheng2003] extended Berrizbeitia's idea to more primes n, and Bernstein [Bernstein2003] extended it to all primes n. The algorithm for finding these proofs relies on some randomness, unlike the original AKS algorithm.

It seems plausible that a variant of AKS may soon compete in practice with ECPP for 'general' primality proofs. This field is in great deal of flux at this time!

Other useful links:

- Primes in P little faq by Anton Stiglic
- Links to things relevant to the AKS algorithm by Phil Carmody
- Distinguishing prime numbers from composite numbers by D. J. Bernstein (an excellent comparision of many methods).

File: prove5.html "Primality Proving 5: Conclusion and suggestions" (Chapter Five: Conclusion)

4.3: Conclusion and suggestions

In practice it is easy to decide what method of primality proof to use:

- If all you want to do is find any number of large enough to make the list of largest known primes, use a version of Proth's Theorem such as theorem 3 (or an equivalent plus side theorem).
- If you are aiming for the money, then either join GIMPS (as Mersenne's have held te record for quite awhile now) or look for a very large generalized Fermat.
- If instead of looking for n, you are given n, first check for small factors. If it has none, try a Fermat test to see if it is a probable prime. If so, try briefly to factor $n+1$, $n-1$... If these factor substantially use the classical methods, if not, then reach for a modern method.

When programming the classical methods, the most difficult aspect is multiplying quickly. Fortunately someone has done much of the work for us! There are several large free libraries for arithmetic with large integers as well as for proving the primality of large integers.

With the classical methods you can easily handle a 100,000 digit number. With the modern methods you will work very hard to handle a 5,000 digit number! Unless you are very brave I would suggest you look for an already coded version of the modern algorithms, they are quite difficult to implement.

At this site we keep a list of the 5000 largest known primes, so if you do find new record primes, why not let us know?

File: references.html "Primality Proving: References" (References)

References for primality proving pages

These are the references used in our primality proving pages. They are a subset of the Prime Pages' references.

AB1999
M. Agrawal and **S. Biswas**, *Primality and identity testing via Chinese remaindering*. In "40th Annual Symposium on Foundations of Computer Science (New York, 1999)," IEEE Computer Soc., Los Alamitos, CA, 1999. pp. 202--208, **MR1917560**

AGP94
W. R. Alford, **A. Granville** and **C. Pomerance**, "There are infinitely many Carmichael numbers," *Ann. of Math. (2)*, **139** (1994) 703--722. **MR 95k:11114**

AH1992
L. M. Adlemann and **M. D. Huang**, *Primality testing and two dimensional Abelian varieties over finite fields*., Lecture Notes in Mathematics Vol, 1512, Springer-Verlag, Berlin, 1992. pp. viii+142, ISBN 3-540-55308-8. **MR 93g:11128**

AKS2002
M. Agrawal, **N. Kayal** and **N. Saxena**, "PRIMES in P," *Ann. of Math. (2)*, **160**:2 (2004) 781--793. Available from http://www.cse.iitk.ac.in/users/manindra/. **MR2123939**

> **Abstract:** We present a deterministic polynomial-time algorithm that determines whether an input number *n* is prime or composite.

AM93
A. O. L. Atkin and **F. Morain**, "Elliptic curves and primality proving," *Math. Comp.*, **61**:203 (July 1993) 29--68. **MR 93m:11136**

APR83
L. M. Adleman, **C. Pomerance** and **R. S. Rumely**, "On distinguishing prime numbers from composite numbers," *Ann. Math.*, **117**:1 (1983) 173--206. **MR 84e:10008** [The first of the modern primality tests.]

Atkin86
A. O. L. Atkin, "Lecture notes of a conference," Boulder Colorado, (August 1986) Manuscript. [See also [AM93].]

Bach85
E. Bach, *Analytic methods in the analysis and design of number-theoretic algorithms*, A.C.M. Distinguished Dissertations The MIT Press, Cambridge, MA, 1985. pp. xiii+48, ISBN 0-262-02219-2. **MR 87i:11185**

Bernstein1998b
D. Berstein, "Detecting perfect powers in essentially linear time," *Math. Comp.*, **67**:223 (1998) 1253--1283. Available from http://cr.yp.to/papers.html. **MR 98j:11121** (Abstract available)

Bernstein2003
D. J. Bernstein, "Proving primality in essentially quartic random time," (2003) Draft available from http://cr.yp.to/papers.html.

> **Abstract:** This paper presents an algorithm that, given a prime *n*, finds and verifies a proof of the primality of *n* in random time $(lg\ n)^{4+o(1)}$.

Berrizbeitia2003
P. Berrizbeitia, "Sharpening "Primes is in P" for a large family of numbers," (2003) Available from

http://arxiv.org/abs/math.NT/0211334. (Annotation available)

BH1993

E. Bach and **L. Huelsbergen**, "Statistical evidence for small generating sets," *Math. Comp.*, **61**:203 (1993) 69--82. **MR1195432**

BH1996

R. C. Baker and **G. Harman**, *The Brun-Titchmarsh theorem on average*. In "Proc. Conf. in Honor of Heini Halberstam (Allerton Park, IL, 1995)," Progr. Math. Vol, 138, Birkhäuser Boston, 1996. Boston, MA, pp. 39--103, **MR 97h:11096**

BH90

W. Bosma and **M. P. van der Hulst**, *Faster primality testing*. In "Advances in Cryptology--EUROCRYPT '89 Proceedings," J. J. Quisquater and J. Vandewalle editors, Springer-Verlag, 1990. pp. 652--656,

BLS75

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. **MR 52:5546** [*The* article for the classical (n^2 -1) primality tests. Table errata in [Brillhart1982]]

BLSTW88

J. Brillhart, **D. H. Lehmer**, **J. L. Selfridge**, **B. Tuckerman** and **S. S. Wagstaff, Jr.**, *Factorizations of $b^n \pm 1$, b=2,3,5,6,7,10,12 up to high powers*, Amer. Math. Soc., 1988. Providence RI, pp. xcvi+236, ISBN 0-8218-5078-4. **MR 90d:11009** (Annotation available)

BP2001

R. Bhattacharjee and **P. Pandey**, "Primality testing," IIT Kanpur, (2001)

Bressoud89

D. M. Bressoud, *Factorizations and primality testing*, Springer-Verlag, 1989. New York, NY, ISBN 0387970401. **MR 91e:11150** [QA161.F3B73]

Cheng2003

Q. Cheng, "Primality proving via one round of ECPP and one iteration in AKS," Crypto 2003, Santa Barbara, (2003) Available from http://www.cs.ou.edu/~qcheng/.

CL84

H. Cohen and **Lenstra, Jr., H. W.**, "Primality testing and Jacobi sums," *Math. Comp.*, **42** (1984) 297--330. **MR 86g:11078** [APRT-CL test introduced.]

CL87

H. Cohen and **A. K. Lenstra**, "Implementation of a new primality test," *Math. Comp.*, **48** (1987) 103--121. **MR 88c:11080** [APRT-CL test implemented.]

CP2001

R. Crandall and **C. Pomerance**, *Prime numbers: a computational perspective*, Springer-Verlag, 2001. New York, NY, pp. xvi+545, ISBN 0-387-94777-9. **MR 2002a:11007** (Abstract available) [This is a valuable text written by true experts in two different areas: computational and theoretical respectively. There is now a second edition [CP2005].]

F1985

E. Fouvry, "Théorème de Brun-Titchmarsh; application au théorèm de Fermat," *Ivent. Math.*, **79**:2 (1985) 383--407. **MR**

86g:11052

GK86

S. Goldwasser and **J. Kilian**, *Almost all primes can be quickly certified*. In "STOC'86, Proceedings of the 18th Annual ACM Symposium on the Theory of Computing (Berkeley, CA, 1986)," ACM, New York, NY, May 1986. pp. 316--329,

Grantham2000

J. Grantham, "Frobenius pseudoprimes," *Math. Comp.*, **70** (2001) 873--891. **MR 2001g:11191** (Abstract available)

Grantham98

J. Grantham, "A probable prime test with high confidence," *J. Number Theory*, **72** (1998) 32--47. **MR 2000e:11160**

HL23

G. H. Hardy and **J. E. Littlewood**, "Some problems of `partitio numerorum' : III: on the expression of a number as a sum of primes," *Acta Math.*, **44** (1923) 1-70. Reprinted in "Collected Papers of G. H. Hardy," Vol. I, pp. 561-630, Clarendon Press, Oxford, 1966.

Jaeschke93

G. Jaeschke, "On strong pseudoprimes to several bases," *Math. Comp.*, **61** (1993) 915-926. **MR 94d:11004**

KS2002

N. Kayal and **N. Saxena**, "Towards adeterministic polynomial-time test," (2002) Available from http://www.cse.iitk.ac.in/research/btp2002/primality.html.

Lehmer30

D. N. Lehmer, "An extended theory of Lucas' functions," *Ann. Math.*, **31** (1930) 419-448. Reprinted in *Selected Papers*, D. McCarthy editor, v. **1**, Ch. Babbage Res. Center, St. Pierre, Manitoba Canada, pp. 11-48 (1981).

LL90

Lenstra, Jr., A. K. and **Lenstra, Jr., H. W.**, *Algorithms in number theory*. In "Handbook of Theoretical Computer Science, Vol A: Algorithms and Complexity," The MIT Press, Amsterdam and New York, 1990. pp. 673-715, **MR 1 127 178**

Mihailescu98

P. Mihailescu, *Cyclotomy primality proving -- recent developments*. In "Proceedings of the III Applied Number Theory Seminar, ANTS III, Portland, Oregon 1998," Lecture Notes in Computer Science Vol, 1423, 1998. pp. 95--110, **MR 2000j:11195**

Miller76

G. Miller, "Riemann's hypothesis and tests for primality," *J. Comput. System Sci.*, **13** (1976) 300--317. **MR 58:470a**

Monier80

L. Monier, "Evaluation and comparsion of two efficient probablistic primality testing algorithms," *Theoretical Computer Science*, **12**:1 (1980) 97--108. **MR 82a:68078**

Morrison75

M. Morrison, "A note on primality testing using Lucas sequences," *Math. Comp.*, **29** (1975) 181--182. **MR 51:5469**

Osterle1979

Oesterlé, J., *Versions effectives du théorème de chebotarev sous l'hypothèse de riemann généralisée*. In "Journ{\'e}es Arithm{\'e}tiques de Luminy (20 Juin--24 Juin 1978)," Astérisque 61 Société Mathématique de France, 1979. Paris, pp. 165--167,

Pinch93

R. Pinch, "The Carmichael numbers up to 10^{15}," *Math. Comp.*, **61**:203 (1993) 381-391. **MR 93m:11137** [A preprint and several data files may be found in the Carmichael directory of his FTP site. For example, he lists the Carmichaels to 10^{17}.]

Pomerance84

C. Pomerance, *Lecture notes on primality testing and factoring (notes by G. M. Gagola Jr.)*, Notes Vol, 4, Mathematical Association of America, 1984. pp. 34 pages,

PSW80

C. Pomerance, **J. L. Selfridge** and **Wagstaff, Jr., S. S.**, "The pseudoprimes to $25 \cdot 10^9$," *Math. Comp.*, **35**:151 (1980) 1003-1026. **MR 82g:10030** [See Richard Pinch's lists of pseudoprimes and [Jaeschke93].]

Rabin80

M. O. Rabin, "Probabilistic algorithm for testing primality," *J. Number Theory*, **12** (1980) 128--138. **MR 81f:10003**

Ribenboim95

P. Ribenboim, *The new book of prime number records*, 3rd edition, Springer-Verlag, 1995. New York, NY, pp. xxiv+541, ISBN 0-387-94457-5. **MR 96k:11112** [An excellent resource for those with some college mathematics. Basically a Guinness Book of World Records for primes with much of the relevant mathematics. The extensive bibliography is seventy-five pages.]

Riesel94

H. Riesel, *Prime numbers and computer methods for factorization*, Progress in Mathematics Vol, 126, Birkhäuser Boston, Boston, MA, 1994. ISBN 0-8176-3743-5. **MR 95h:11142** [An excellent reference for those who want to start to program some of these algorithms. Code is provided in Pascal. Previous edition was vol. 57, 1985.]

Wiles95

A. Wiles, "Modular elliptic curves and Fermat's last theorem," *Ann. Math.*, **141**:3 (1995) 443--551. **MR 96d:11071** (Annotation available)

Williams78

H. C. Williams, "Primality testing on a computer," *Ars Combin.*, **5** (1978) 127--185. **MR 80d:10002** [A survey of the classical primality tests.]

Williams98

H. C. Williams, *Édouard Lucas and primality testing*, Canadian Math. Soc. Series of Monographs and Adv. Texts Vol, 22, John Wiley \& Sons, New York, NY, 1998. pp. x+525, ISBN 0-471-14852-0. **MR 2000b:11139** (Annotation available)