Lucas primality test

From Wikipedia, the free encyclopedia

"Lucas–Lehmer test" redirects here. For the test for Mersenne numbers, see Lucas–Lehmer primality test. For the Lucas–Lehmer–Riesel test, see Lucas–Lehmer–Riesel test. For the Lucas probable prime test, see Lucas pseudoprime.

In computational number theory, the **Lucas test** is a primality test for a natural number *n*; it requires that the prime factors of *n* − 1 be already known.[1][2] It is the basis of the Pratt certificate that gives a concise verification that *n* is prime.

Contents [hide]

Concepts [edit]

Let *n* be a positive integer. If there exists an integer 1 < *a* < *n* such that

$$a^{n-1} \equiv 1 \pmod n$$

and for every prime factor *q* of *n* − 1

$$a^{(n-1)/q} \not\equiv 1 \pmod n$$

then *n* is prime. If no such number *a* exists, then *n* is either 1 or composite.

The reason for the correctness of this claim is as follows: if the first equivalence holds for *a*, we can deduce that *a* and *n* are coprime. If *a* also survives the second step, then the order of *a* in the group (**Z**/*n***Z**)* is equal to *n*−1, which means that the order of that group is *n*−1 (because the order of every element of a group divides the order of the group), implying that *n* is prime. Conversely, if *n* is prime, then there exists a primitive root modulo *n*, or generator of the group (**Z**/*n***Z**)*. Such a generator has order |(**Z**/*n***Z**)*| = *n*−1 and both equivalences will hold for any such primitive root.

Note that if there exists an *a* < *n* such that the first equivalence fails, *a* is called a Fermat witness for the compositeness of *n*.

Example [edit]

For example, take *n* = 71. Then *n* − 1 = 70 and the prime factors of 70 are 2, 5 and 7. We randomly select an *a=17* < *n*. Now we compute:

$$17^{70} \equiv 1 \pmod{71}.$$

For all integers *a* it is known that

$$a^{n-1} \equiv 1 \pmod n \text{ if and only if } \operatorname{ord}(a)|(n-1).$$

Therefore, the multiplicative order of 17 (mod 71) is not necessarily 70 because some factor of 70 may also work above. So check 70 divided by its prime factors:

$$17^{35} \equiv 70 \not\equiv 1 \pmod{71}$$
$$17^{14} \equiv 25 \not\equiv 1 \pmod{71}$$
$$17^{10} \equiv 1 \equiv 1 \pmod{71}.$$

Unfortunately, we get that 17^{10}≡1 (mod 71). So we still don't know if 71 is prime or not.

We try another random *a*, this time choosing *a* = 11. Now we compute:

$$11^{70} \equiv 1 \pmod{71}.$$

Again, this does not show that the multiplicative order of 11 (mod 71) is 70 because some factor of 70 may also work. So check 70 divided by its prime factors:

$$\begin{aligned}11^{35} &\equiv 70 \not\equiv 1 \pmod{71}\\ 11^{14} &\equiv 54 \not\equiv 1 \pmod{71}\\ 11^{10} &\equiv 32 \not\equiv 1 \pmod{71}.\end{aligned}$$

So the multiplicative order of 11 (mod 71) is 70, and thus 71 is prime.

(To carry out these modular exponentiations, one could use a fast exponentiation algorithm like binary or addition-chain exponentiation).

Algorithm [edit]

The algorithm can be written in pseudocode as follows:

```
Input: n > 2, an odd integer to be tested for primality; k, a parameter that
determines the accuracy of the test
Output: prime if n is prime, otherwise composite or possibly composite;
determine the prime factors of n−1.
LOOP1: repeat k times:
   pick a randomly in the range [2, n − 1]
      if a^(n-1) ≢ 1 (mod n) then return composite
      otherwise
         LOOP2: for all prime factors q of n−1:
            if a^((n-1)/q) ≢ 1 (mod n)
               if we did not check this equality for all prime factors of n−1
                  then do next LOOP2
               otherwise return prime
            otherwise do next LOOP1
return possibly composite.
```

See also [edit]

- Édouard Lucas
- Fermat's little theorem

Notes [edit]

1. ^ Crandall, Richard; Pomerance, Carl (2005). *Prime Numbers: a Computational Perspective (2nd edition)*. Springer. p. 173. ISBN 0-387-25282-7.
2. ^ Křížek, Michal; Luca, Florian; Somer, Lawrence (2001). *17 Lectures on Fermat Numbers: From Number Theory to Geometry*. CMS Books in Mathematics **9**. Canadian Mathematical Society/Springer. p. 41. ISBN 0-387-95332-9.

V · T · E	Number-theoretic algorithms [hide]
Primality tests	AKS test · APR test · Baillie–PSW · ECPP test · Elliptic curve · Pocklington · Fermat · **Lucas** · *Lucas–Lehmer* · *Lucas–Lehmer–Riesel* · *Proth's theorem* · *Pépin's* · Quadratic Frobenius test · Solovay–Strassen · Miller–Rabin
Prime-generating	Sieve of Atkin · Sieve of Eratosthenes · Sieve of Sundaram · Wheel factorization
Integer factorization	Continued fraction (CFRAC) · Dixon's · Lenstra elliptic curve (ECM) · Euler's · Pollard's rho · *p* − 1 · *p* + 1 · Quadratic sieve (QS) · General number field sieve (GNFS) · *Special number field sieve (SNFS)* · Rational sieve · Fermat's · Shanks' square forms · Trial division · Shor's
Multiplication	Ancient Egyptian · Long · Karatsuba · Toom–Cook · Schönhage–Strassen · Fürer's
Discrete logarithm	Baby-step giant-step · Pollard rho · Pollard kangaroo · Pohlig–Hellman · Index calculus · Function field sieve
Greatest common divisor	Binary · Euclidean · Extended Euclidean · Lehmer's
Modular square root	Cipolla · Pocklington's · Tonelli–Shanks
Other algorithms	Chakravala · Cornacchia · Integer relation · Integer square root · Modular exponentiation · Schoof's
	Italics indicate that algorithm is for numbers of special forms · Smallcaps indicate a deterministic algorithm

Categories: Primality tests