PRIMALITY TESTING AND APPLICATIONS

PRANAV VIJAY

1. Introduction

How can we determine if

$$785963102379428822376694789446897396207498568951$$

is prime?[1]

To perform such a calculation, we should use a primality test that is both accurate and time-efficient for large candidates n. Accordingly, this paper will compare characteristics (accuracy, time complexity, issues, variants) of a range of algorithms. Furthermore, it will discuss the applications in cryptography, specifically in generating public keys.

Note that throughout this document, FLT refers to Fermat's *Little* Theorem.

1.1. Brute Force: Trial Division.

The most direct approach is to simply test for divisibility by all primes less than or equal to $\sqrt{n}$. For large n, we can use a prime number sieve to find primes between 1 and $\lfloor\sqrt{n}\rfloor$. Essentially, this involves removing the factors of all primes up to $\sqrt{\lfloor\sqrt{n}\rfloor}$ from the list, leaving only primes. If this upper bound is also large, we apply the sieve again (and so on).

Using the sieve and a fast (Schönhage-Strassen) multiplication algorithm, the time complexity becomes $O(\sqrt{N}\cdot\log N/\log\log N)$, thus inefficient for large N.[1] Using this as a reference for comparison, we seek alternatives.

2. Fermat's Little Theorem

It would make sense to use a theorem involving prime numbers in relation to modular congruences, as we would then have a definite classification system for all numbers. Clearly, exponential/factorial running time (e.g. Wilson's theorem) must be avoided, as we are mostly testing large n. Ideally, we should achieve $O((\log n)^k)$ running time, i.e. terrible for small n but optimal for large n.

Fermat's Little Theorem states that $a^p \equiv a \pmod p$ over all integers a for some prime p. This does *not* necessarily imply that all $p = n$ satisfying this condition are prime (we will discuss "liar" numbers later); however, if some n does not satisfy this test, it is certainly composite. Still, we would like to make the test as accurate as possible for determining primality.

Suppose (for candidate n) we randomly select an integral value of $a \in [0, n-1]$ and compare a^n and a modulo n. Computing $a^n \pmod n$ is a time-consuming procedure, but can be done more efficiently with modular exponentiation techniques:

(1) Find p as a sum of powers of 2.
(2) Use the fact that $a^{2^{k+1}} \equiv (a^{2^k} \pmod p)^2 \pmod p$ to find residues modulo p for all 2^k in the representation of p. Multiply the residues and take modulo p.

We require a time complexity of $O((\log n)^2)$ (maximum of $\log_2 n$ multiplications, each of which is squared) over one value of a.

[1] For the significance of this number, see section 5.2.1.

2.1. **Issues.** As aforementioned, we can determine with certainty that a number not satisfying FLT for this value of a is definitely composite. However, a number of issues allow several composite numbers to pass as primes.

We will address each issue separately:

(1) Most obviously, $n \mid 0^n$ for any n. We can make a similar observation for $a = 1$ and $a \equiv -1 \pmod{n}$ (the latter satisfies FLT whenever n is odd, but we are only concerned with odd numbers here. There is no value in testing $n = 2$. This restricts the interval for a to $[2, p-2]$.
(2) Even excluding these values of a, there are some composite "pseudoprimes" for certain a (such a are known as Fermat "liars"). However, most pseudoprimes have corresponding Fermat "witnesses" (values of a such that n does not satisfy FLT). So, we account for such composite n by testing with all values of a on the interval. Note that this increases the time complexity to $O(k(\log n)^2)$, where k is the number of iterations.
(3) Some of the pseudoprimes in (2), such as $n = 561$, do not have any Fermat "witnesses". These are known as the **Carmichael numbers**, and require their own subsection.

2.2. **Carmichael Numbers.** First, we can better define them as follows:

Definition 2.1. *The increasing sequence c_k of (distinct) Carmichael numbers satisfies*

$$a^{c_k} \equiv a \pmod{c_k}$$

for all integers a. (For example, $c_1 = 561$.)

The basic Fermat primality test cannot account for the Carmichael numbers. However, we can make a few statements about their characteristics and distribution:

Theorem 2.1 (Korselt's Criterion)**.** *A positive composite integer n is a Carmichael number iff n is square-free and $p-1 \mid n-1$ for prime divisors p of n.*

Proof. It follows from Fermat's Little Theorem that if $n-1 = k(p-1)$ for some positive integer k, then (for some integer a such that $\gcd(a, n) = 1$):
$a^{n-1} \equiv (a^{p-1})^k \equiv 1^k \equiv 1 \pmod{p} \to a^n \equiv a \pmod{p}$.
Expressing n as $\prod p_i$:
$a^{p_i - 1} \equiv a^{n-1} \equiv 1 \pmod{p_i} \to a^n \equiv a \pmod{p_i}$
for all i, because $n \equiv 0 \pmod{p_i}$. Thus, $a^n \equiv a \pmod{p \cdot \prod p_i} \to a^n \equiv a \pmod{n}$. Note that this can be generalized to all integers a, as the final conclusion is independent of the $a^{p-1} \equiv 1 \pmod{n}$ formula.
□

Theorem 2.2. *There are infinitely many Carmichael numbers.* [2]

We have also been able to accurately describe the distribution of these numbers. We denote #(Carmichael numbers less than n) = C_n. In 2021, Daniel Larsen proved a definite lower bound on C_n. We also have an upper bound (Pomerance, [3]):

$$C_n \leq n^{1-(1+o(1))\log\log\log n/\log\log n}$$

As $n \to \infty$, this bound grows at an increasingly slower rate, indicating the scarcity of Carmichael numbers for large n. Thus, our modified FLT primality test has a high (but not 100%) accuracy.

3. Other Randomized Algorithms

3.1. **Miller-Rabin.** Until now, all of the algorithms we have discussed are unable to account for Carmichael numbers. The Miller-Rabin primality test overcomes this limitation, and is immune to most (but not all) of these.

The Miller-Rabin test makes use of the following statement, inspired by Euler's criterion:

Theorem 3.1. *The only solutions to $x^2 \equiv 1 \pmod{n}$ are ± 1 iff n is prime.*

Proof. We have $n \mid x^2 - 1 = (x-1)(x+1)$. If n is prime, it must divide exactly one of the two expressions (unless $n = 2$, but the statement is still valid because 1 is the only nonzero residue modulo 2). Thus $x \equiv \pm 1 \pmod{n}$. If n is composite, we can split up the prime factors and let some of them divide $x-1$ while others divide $x+1$. The Chinese Remainder Theorem guarantees a solution modulo n for every such split. □

Notice that we can obtain the same residue modulo p with Fermat's Little Theorem. Taking advantage of this, we can state that $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$ and use this as a "filter" to check all n that pass the FLT test we proposed in 2.1. We can divide the exponent by 2 for as many times as $\frac{p-1}{2^k}$ is still an integer, so we will test for all such k to increase our degree of certainty. Note that if the residue of a term with exponent k is $-1 \pmod{p}$, the term with exponent $k+1$ (if it exists) is not congruent to $\pm 1 \pmod{p}$. However, we must have the former in order for the latter to occur. Now, we can state the best algorithm so far:

Theorem 3.2 (Miller-Rabin)**.** *Given (odd) candidate n, select a random integer $a \in [2, n-2]$. Then, n is "prime" if:*

(i) (a, n) satisfy Fermat's Little Theorem.

(ii) Given the sequence $s_k = \frac{p-1}{2^k}$ for integers $1 \le k \le \nu_2(n-1)$[2], either all $a^{s_k} \equiv 1 \pmod{n}$, or the first term not congruent to $1 \pmod{n}$ has residue $-1 \pmod{n}$. (Subsequent terms, if they exist, can have other residues.)

It is very unlikely that a Carmichael number passes this test. In fact, it has been proven that this test successfully identifies more than $\frac{3}{4}$ (often much more) Carmichael numbers as composite. Running it multiple times with different values of a gives an even higher (essentially 100%) probability of identifying primes. However, this increases the running time to $O(k \log^3(n))$ (for k trials). This can be shrunk to $O(k \log^2(n) \log\log n)$ using multiplication algorithms outside of this handout's scope.

3.2. **Pocklington-Lehmer Test.**

3.2.1. *Pocklington criterion.*

Theorem 3.3 (Pocklington)**.** *Given relatively prime integers a, N that satisfy FLT, prime $p \mid N-1$ such that $p > \sqrt{N} - 1$, N is prime if*

$$\gcd(a^{(N-1)/p} - 1, N) = 1$$

Proof. Again, this is a direct result of FLT. We will prove by contradiction; if N is composite, it must have a prime factor $q \le \sqrt{N}$. Given $\gcd(N-1, q-1) = \gcd(p, q-1) = 1$ and prime q, there must exist multiplicative inverse u of p modulo $q-1$ such that $q \mid up$, meaning

$$a^{up} \equiv a \pmod{q}$$

Recalling that $q \mid N$, FLT gives $a^{N-1} \equiv 1 \pmod{q}$. We would like to obtain a congruence with $a^{(N-1)/p}$, so we can manipulate to obtain

$$(a^p)^{(N-1)/p} \equiv (a^p)^{(N-1)/p} \equiv a^{(N-1)/p} \equiv 1 \pmod{q}$$

Thus $q \mid a^{(N-1)/p} - 1$. But $q \mid N$, so $\gcd \ge q$, and we have a contradiction to the statement that these two are relatively prime. [4] □

Along with Fermat's Little Theorem, this leads us to the following:

Theorem 3.4 (Generalized Pocklington Test)**.** *Now, let $N - 1 = A \cdot B$, where $\gcd(A, B) = 1$, $A > \sqrt{N}$, the prime factorization of A is known, but the factorization of B is not necessarily known.*

Then, N is prime if, for each prime factor $p \mid A$, there exists an integer a_p so that

$$a_p^{N-1} \equiv 1 \pmod{N}$$

and

$$\gcd(a_p^{(N-1)/p} - 1, N) = 1.$$

When $a = 2$, we are led to a relation that will be expanded upon in the next section.

[2]Here, $\nu_p(n)$ denotes the p-adic valuation of n, i.e. the exponent of the largest power of p that divides n.

3.3. **Primes of the Form $2^n - 1$.** Let us now consider a special case. Suppose we wanted to determine if a Mersenne number $M_n = 2^n - 1$ is prime. If we think of M_n as a geometric sequence of n terms, i.e. $2^0 + 2^1 + \dots 2^{n-1}$, it is clear that if n is composite, $M_p \mid M_n$ for all prime factors p. Thus we only need to test M_p.

There exists a definite algorithm to determine the primality of these numbers. An improvised version is stated below:

Theorem 3.5 (Lucas-Lehmer)**.** *For some odd prime p, M_p is prime iff $M_p \mid s_{p-2}$, where* $s_i = \begin{cases} 4 & i = 0 \\ s_{i-1}^2 - 2 & i \neq 0 \end{cases}$.

A proof for this that uses basic principles of quadratic reciprocity and clever manipulation can be found at [5]. The fastest version of the Lucas-Lehmer test has time complexity $O(n^2 \log n)$ for the n^{th} Mersenne number. This is great, because n is the exponent of our number. One can then find extremely large primes of this form fairly quickly. The Lucas-Lehmer test is thus used to find primes larger than those ever discovered before (the record is $2^{82,589,933} - 1$!).

3.4. **Elliptic Curve Primality Proving.** We can take inspiration from the Pocklington criterion (sect 2.3) to state the following:

Proposition 3.1. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve, which we will take over the group $\mathbb{Z}/n\mathbb{Z}$.*

If (i) there exists a prime $q > (\sqrt[4]{N} + 1)^2$ such that $q \mid m$, (ii) $mP = 0$ and (iii) $P \cdot \frac{m}{q}$ is defined and not zero, then n is prime.

Proof. Suppose, by contradiction, that n is composite. Then, n has a prime factor $p < \sqrt{n}$. Consider the finite group $E(\mathbb{F}_p)$. The Hasse-Weil bound for elliptic curves gives us that

$$| \#E(\mathbb{F}_p) - (p+1) | \leq 2\sqrt{p}$$

Thus, the upper bound of p can be found by setting $\#E(\mathbb{F}_p) \geq p + 1$ and solving:

$$\#E(\mathbb{F}_p) \leq p + 2\sqrt{p} + 1$$

The RHS can be rewritten as $(\sqrt{p}+1)^2$. Since $p < \sqrt{n}$, $(\sqrt{p}+1)^2 < (\sqrt[4]{p}+1)^2$, which in turn is less than q. But q is prime, so $\gcd(\#E(\mathbb{F}_p), q) = 1$ and q has an inverse modulo $\#E(\mathbb{F}_p)$.

Evaluating statement (iii) modulo p,

$$(m/q)P \equiv qq^{-1} \cdot (m/q)P \equiv mPq^{-1} \pmod{p}$$

Statement (ii), $mP = 0$, then gives us (if true) that $(m/q)P \equiv 0 \pmod{p}$. However, if we repeated the same procedure with p instead of n, we would obtain from statement (iii) that $(m/q)P \not\equiv 0 \pmod{p}$, a contradiction.

□

It may still seem strange to use elliptic curves here, but it turns out that this test is extremely time-efficient for

3.5. **Alternative Approaches.**

(1) The AKS primality test relies on the following statement:

Theorem 3.6 (AKS)**.** *An integer n is prime if $n \mid (x-1)^n - (x^n - 1)$.*

AKS is thought to be 100% accurate[7] and is one of the most frequently used tests today. Evidently, though, it is extremely slow (time complexity $O(n^6)$) for large n. Its speed can be increased with the similar congruence

$$(x + a)^n - (x^n + a) \equiv 0 \pmod{x^r - 1, n}$$

for some r.[7]

(2) The Baillie-PSW primality test is another highly accurate algorithm, and has the additional advantage of being time-efficient for extremely large n. It is essentially a combination of the Fermat test and Lucas-Lehmer test. Though not proven, no composite number has been shown to pass Baillie-PSW.

(3) Next, we introduce the following:

Definition 3.1. *The n^{th} homogeneous cyclotomic polynomial $\phi_n(x, y)$ is the unique irreducible factor of $x^n - y^n$ with degree $\phi(n)$.*

$\phi_3 = \frac{x^3-y^3}{x-y} = x^2 + xy + y^2$ is one such polynomial of interest to us. Primes of this form are called *cuban primes*, and the case $x = y + 1$ (so $\phi_3 = 3y^2 + 3y + 1$) has led to the discovery/proof of a 3-million digit prime with $y = 3^{3304301} - 1$.[3]

4. Application in Cryptography

4.1. **RSA Key Cipher.** We will illustrate the idea behind RSA with an example. Suppose Alice wishes to encrypt the message "EULER" and send it to Bob. Of course, Alice must first convert to a numeric system. For simplicity, we will use the residues modulo 26 and assign the residue n to the $n + 1^{\text{th}}$ letter of the alphabet (since we are only using letters). This gives us $4|20|11|4|17$.

Let us only consider the first character $E \to 4$. Before discussing encryption, we must introduce a few ideas:

Definition 4.1. *The Carmichael totient function $\lambda(n)$ is defined as the smallest positive integer m such that, for all a, $a^m \equiv 1 \pmod{n}$.*

Proposition 4.1. *$\lambda(n)$ for $n = p_1^{e_1} \cdot p_2^{e_2} \cdot \ldots$ equals lcm $(\lambda(p_1^{e_1}), \lambda(p_2^{e_2}), \ldots)$.*

Proposition 4.2. *For prime p, $\lambda(p) = \phi(p) = p - 1$.*

4.1.1. *Encryption.*

(1) Select two (extremely large) primes p and q, and compute $n = pq$. For the sake of simplicity, though, we will use $p = 5$ and $q = 11$ so that $n = 55$. Note that n must be larger than the numeric assignments of all characters for RSA to work.
(2) Choose a positive integer $e < \lambda(n)$ such that $\gcd(e, \lambda(n)) = \gcd(e, \text{lcm}(p - 1, q - 1)) = 1$. Here, $\lambda(n) = 20$, so let $e = 17$.
(3) Alice thus creates a *public key* (n, e) that Bob is aware of.
(4) Encrypt with the formula $c(m) = m^e \pmod{n}$. In our case, $m = 4, e = 17, n = 55$, so we have $c(m) = 49$. Similarly, $c(m)$ can be generated for "U", "L", and "R".

4.1.2. *Decryption.* Now, Bob has the public key $(n, e) = (55, 17)$ and must decrypt the ciphertext $c = 49$.

First, we need the *private key* (n, d), where $d = e^{-1} \pmod{\lambda(n)}$ $(= 13$ for this scenario). Then, the plaintext m is calculated using the formula $m(c) = c^d \pmod{n} \to 49^{13} \pmod{55} = 4$, as desired.

4.1.3. *Generating n.* In practice, p and q are several tens (sometimes hundreds) of orders of magnitude larger than the values we selected. Its creators also suggested [6] that the primes be a few orders of magnitude apart, so that $\lambda(n) = \text{lcm}(p - 1, q - 1)$ is usually larger for randomly-generated primes, thus allowing e to take on more values.

Primality testing algorithms such as those discussed in this handout are used in generating these primes. Suppose, for example, that we want to select p and q such that $\lfloor \log p \rfloor = 100$ and $\lfloor \log q \rfloor = 96$. We might use a prime number sieve to remove multiples of the first few primes from candidates for p and q, and then use a fast primality test with high certainty (such as Miller-Rabin) to test the primality of randomly-generated numbers from the remaining candidates.

[3] See PrimePages 136214.

4.2. **Diffie-Hellman Key Exchange.** This is similar to (and actually preceded) RSA, except that the same key is used for both encryption and decryption. Diffie-Hellman can also be used with more than two parties, but we will consider the simplest case.

Suppose Alice and Bob each have two keys (public and private). Bob receives Alice's public key and vice versa, so that each person has their own private key and the other person's public key. Then, both are able to generate the same secret key using the following algorithm:

(1) Let α, β respectively be the private keys of Alice and Bob. Let a, b respectively be their public keys. Both people know a common base g and a prime modulus p, such that $a = g^\alpha \pmod p$ and $b = g^\beta \pmod p$.
(2) Alice computes a secret key $s = b^\alpha \pmod p$. Bob finds the *same* secret key with $a^\beta \pmod p$. This works because of the following rule in modular arithmetic:

$$(g^\alpha \pmod p)^\beta \pmod p = (g^\beta \pmod p)^\alpha \pmod p$$

$$\to (a)^\beta \pmod p = (b)^\alpha \pmod p$$

Observe that all of the calculations are dependent on the private keys, thus preventing potential attackers from finding s. To keep this safe from a "brute force" attack (manually testing residues modulo p), the value of p must be sufficiently large. Hence, we require primality testing algorithms to use on large, randomly-generated n.

4.2.1. *Elliptic Curve Diffie-Hellman.* (ECDH) To make this cipher even stronger, we can introduce elliptic curves.

Consider an elliptic curve $E : y^2 = x^3 + Ax + B$ taken over the finite field $\mathbb{F}_p$. Also, consider a "generator" point $G \in E(\mathbb{F}_p)$ with order n (i.e. the integer such that $nG = \infty$). Now, we can assign Alice private key $\alpha : 1 \leq \alpha \leq n - 1$ and Bob private key $\beta : 1 \leq \beta \leq n - 1$.

Alice now computes the point $a = \alpha G$ and makes this public. Bob does the same with $b = \beta G$. Both Alice and Bob are now able to find the secret point $P = \alpha\beta G$ by multiplying the other person's public point with their private key.

Division on an elliptic curve is far more tedious than modular arithmetic, especially when p is large. As an example, Microsoft used ECDH for their Digital Rights Management service (used for many years to enforce copyright on audio/video content) with $p = 0x89abcdef012345672718281831415926141424f7_{16}$.

Note that the base-10 representation of p is the number we mentioned in the introduction!

Citations

[1] https://dspace.mit.edu/bitstream/handle/1721.1/122962/18-783-spring-2017/contents/lecture-notes/MIT18_783S17_lec12.pdf

[2] Alford, W. R.; Granville, Andrew; Pomerance, Carl (1994). "There are Infinitely Many Carmichael Numbers". Annals of Mathematics. 140 (3): 703–722. doi:10.2307/2118576. JSTOR 2118576.

[3] Pomerance, C. (1981). "On the distribution of pseudoprimes". Math. Comp. 37 (156): 587–593. doi:10.1090/s0025-5718-1981-0628717-0. JSTOR 2007448.

[4] Koblitz, Neal (1994). A Course in Number Theory and Cryptography. Graduate Texts in Mathematics. Vol. 144 (2nd ed.). Springer. ISBN 0-387-94293-9.

[5] "Lucas-Lehmer test." Prime-Wiki, 17 Feb 2019. 7 Dec 2023, https://www.rieselprime.de/z/index.php?title=Lucas-Lehmer_test&oldid=1199>.

[6] Gallier, Jean (2022). Notes on Primality Testing And Public Key Cryptography Part 1: Randomized Algorithms Miller–Rabin and Solovay–Strassen Tests. https://www.cis.upenn.edu/~jean/RSA-primality-testing.pdf

[7] Agrawal, Manindra; Kayal, Neeraj; Saxena, Nitin (2002). "PRIMES is in P". https://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf

Other references:

(1) PrimePages database: https://t5k.org/largest.html

(2) https://crypto.stanford.edu/pbc/notes/numbertheory/millerrabin.html

(3) https://www.di-mgt.com.au/rsa_alg.html

(4) Brillhart, John; Lehmer, D. H.; Selfridge, J. L. (April 1975). "New Primality Criteria and Factorizations of 2m ± 1". Mathematics of Computation. 29 (130): 620–647. doi:10.1090/S0025-5718-1975-0384673-1. JSTOR 2005583.