Primality Tests

Why do we need Primality Testing?

Primality Testing is a test which we use to determine whether a given number is prime or not. Primality testing has a major application in cryptography, where prime numbers form the base of encrypting your data safely.

Obviously, primality testing sounds like a test that's very straightforward, just check all the numbers that are less than the inputted number n and check if any of them divide it without any remainder. If there exists such a number, then we say that the number is **composite**, else it's **prime.**

A simple optimization can be made however, to reduce the time complexity from $O(n)$ to $O(\sqrt{n})$ by just checking the first $\sqrt{n}$ numbers.

An $O(\sqrt{n})$ solution can be very fast, and sufficient for many use cases as well. However, in fields such as cryptography, even an $O(\sqrt{n})$ is not sufficient, as cryptography deals with **very** large primes, beyond the range of 10^{18}, hence igniting the search for a faster solution.

Probabilistic Algorithms

Instead of only looking at algorithms which are guarantee the answer to the primality test, we can widen our search range for solutions by looking for solutions that give a very high probability of giving the correct answer. One of these solutions is the **Miller-Rabin Primality Test.**

The **Miller-Rabin primality test** is a test that does not guarantee that the answer it gives is correct. However, it is an algorithm which provides a bound on the probability of the answer being correct, and can easily be tweaked to increase the probability at the cost of efficiency, or vice versa.

Miller-Rabin Algorithm

The Miller-Rabin primality test is guaranteed to say that a number is prime if it actually is prime, and has a very high probability of saying that a composite number is composite. Hence, the error that might occur is that the algorithm may call a composite number a prime number with very low probability.

However, an adversary might try to come up with worst case input to break the algorithm, hence the primality test makes use of randomization, which makes the algorithm much harder to break due to lack of a uniform pattern.

Steps of Algorithm

Input : An integer p.

1. If $p = 2$, then accept (claim that the number is prime). Else if p is even, then reject (claim the number is composite).
2. Pick k numbers, $a_1, a_2, \ldots, a_k$ from $\mathbb{Z}_p^+$ (Numbers $0, 1, \ldots, p-2, p-1$).
3. For each i from 1 to k
    1. Compute a_i^{p-1} mod p and reject if different from 1. This is in accordance with Fermat's Little Theorem, which states that $a^{p-1} \equiv 1 \bmod p$ if p is prime.
    2. Since we are only dealing with odd value of p, since we accepted/rejected all even numbers in step 1, we can say that $p-1$ is even. Hence, we can factorize $p - 1 = 2^k \cdot s$, where s is some odd number.
    3. We then compute the sequence $a_i^{s \cdot 2^0}, a_i^{s \cdot 2^1}, \ldots, a_i^{s \cdot 2^k}$ mod p.
    4. We read from right to left, and try to find an element in this sequence that is not 1. If the first number we come across a number which is not 1, and that number isn't -1, then we reject p (claim that it's composite).
4. If we haven't rejected the number p till now, it implies that we have passed all the tests, hence we accept the number p (claim that it's prime).

Time Complexity

If we are given a number p, and we pick k numbers from $\mathbb{Z}_p^+$, then the time complexity of the test is $O(k \cdot log^3 p)$, a significant improvement over the simple $O(\sqrt{n})$ solution, provided we pick a reasonable k

Probability of Correctness

We made a claim earlier that if a number is prime, then the test will claim that the given number is prime with 100% probability. This claim is proven below

Proof a prime number is always accepted with probability = 1

If the prime number is even and 2, then we automatically accept it in the first step itself.

Else the prime number is odd.

In step 3a, we make use of Fermat's Little Theorem that states that $a^{p-1} \equiv 1 \bmod p$, if p is prime. Hence, if the given input is prime, we won't ever reject it on this step as it will always satisfy this condition.

In step 3d, we need to prove that the first number we come across from right to left that's not 1 has to be -1 if the given number is prime.

Proof By Contradiction:

We can notice that the sequence that we calculate is just $a_i^{s \cdot 2^0}$ squared repeatedly. Hence an element in the sequence is just the square of the previous term (except for the first term).

Take the first term that is not 1 as b. We'll formally state that $b \not\equiv 1 \mod p$ trivially, and make the assumption $b \not\equiv -1 \mod p$ to arrive at a contradiction later.

Since b is the first term that is not equal to 1, we know the next term is.

Hence $b^2 \equiv 1 \mod p \implies b^2 - 1 \equiv 0 \mod p$. Any number that is equivalent to $0 \mod p$ implies that the number is divisible by p.

We can factorize $b^2 - 1$ as

$$b^2 - 1 \equiv (b-1)(b+1) \equiv 0 \mod p$$

Since $(b-1)(b+1) \equiv 0 \mod p$, this implies that at least one of the two, $(b-1)$ or $(b+1)$ are divisible by p, as p is prime.

If $(b-1)$ is divisible by p, it implies that $(b-1) \equiv 0 \mod p$. Hence, $b \equiv 1 \mod p$. However, this goes against our earlier assumption that $b \not\equiv 1 \mod p$.

If $(b+1)$ is divisible by p, it implies that $(b+1) \equiv 0 \mod p$. Hence, $b \equiv -1 \mod p$. However, this goes against our other assumption that $b \equiv -1 \mod p$

Hence, we arrive at a contradiction. Hence, if $b \not\equiv 1 \mod p$, then $b \equiv -1 \mod p$ when p is an odd prime number.

However, if a number is an odd composite number, than there is a chance that the number may or may not be correctly identified as composite. We prove below that with one run, the probability that it will be identified as composite is $\geq 1/2$.

Proof a Odd Composite Number will be rejected with Probability $\geq \frac{1}{2}$

There are two nontrivial witnesses, 1 and -1. There are two types of witnesses as well, those which generate a sequence of only 1s, and those which have some numbers which aren't 1. -1 falls into the second kind.

We will find the non-witness of the second kind that generates a sequence with the -1 coming at the rightmost position possible. We consider that number as h, and the position of the -1 at position j, assuming 0 based indexing.

Therefore $h^{s \cdot 2^j} \equiv -1 \mod p$.

Since p is composite, it must either be a power of a prime, or it can be represented as a product of two numbers that are relatively prime, q, r.

Using the Chinese Remainder Theorem, we know there exists a number t such that

$$t \equiv h \mod qt \equiv 1 \mod r$$

From this we can state that

$$t^{s\cdot 2^j} \equiv -1 \mod q t^{s\cdot 2^j} \equiv 1 \mod r$$

We know that $h^{s\cdot 2^j}$ can be stated in the form $a(qr)-1$. Therefore, we can restate it as $ar(q)-1$, hence $h^{s\cdot 2^j} \equiv -1 \mod q$ as well. Since $t \equiv h \mod q$, we can clearly conclude that $t^{s\cdot 2^j} \equiv -1 \mod q$.

The second equivalence is trivial as $t \equiv 1 \mod r$, hence raising it to any power will give 1.

Now, we'll make the claim that t is a witness.

If $t^{s\cdot 2^j} \equiv 1 \mod p$, Then we can write $t^{s\cdot 2^j} = kp+1 = (kr)q+1$. This implies $t^{s\cdot 2^j} \equiv 1 \mod q$, which is incorrect. Hence $t^{s\cdot 2^j} \not\equiv 1 \mod p$

If $t^{s\cdot 2^j} \equiv -1 \mod p$, Then we can write $t^{s\cdot 2^j} = kp-1 = (kq)r-1$. This implies that $t^{s\cdot 2^j} \equiv -1 \mod r$, which is incorrect.

Hence $t^{s\cdot 2^j} \not\equiv \pm 1 \mod p$.

However, since $t^{s\cdot 2^{j+1}} \equiv 1 \mod q$, and $t^{s\cdot 2^{j+1}} \equiv 1 \mod r$, this implies that $t^{s\cdot 2^{j+1}} - 1$ is divisible by both q and r. Since q and r are relatively prime, we know that $t^{s\cdot 2^{j+1}} \equiv 1 \mod p$

Since $t^{s\cdot 2^j} \not\equiv \pm 1 \mod p$ and $t^{s\cdot 2^{j+1}} \equiv 1 \mod p$, this implies that t is a witness.

Mapping each non-witness to a unique witness

We can prove that $dt \mod p$ is a unique witness for each unique non-witness d.

Since d is a non-witness, it implies that $d^{s\cdot 2^j} \equiv \pm 1 \mod p$. However, $t^{s\cdot 2^j} \not\equiv \pm 1 \mod p$ as t is a witness.

Hence, $(dt)^{s\cdot 2^j} \equiv d^{s\cdot 2^j} \cdot t^{s\cdot 2^j} \equiv \pm 1 \cdot t^{s\cdot 2^j} \not\equiv \pm 1 \mod p$. Hence dt is a witness as well

Proving uniqueness of witnesses. i.e $d_1 t \equiv d_2 t \mod p \implies d_1 \equiv d_2 \mod p$

If $d_1 t \equiv d_2 t \mod p$, then

$$d_1 t \cdot t^{s\cdot 2^{j+1}-1} \equiv d_2 t \cdot t^{s\cdot 2^{j+1}-1} \mod p d_1 t^{s\cdot 2^{j+1}} \equiv d_2 t^{s\cdot 2^{j+1}} \mod p d_1 \cdot 1 \equiv d_2 \cdot 1 \mod p$$

Hence we've proven that the witnesses are unique. Hence, we've proven that |Witnesses| $\geq$ |Non-witnesses|. Hence, the probability of an odd composite number being rejected is 1/2.

Hence, upon running the Miller-Rabin primality test with k numbers, we get an accuracy of $\frac{2^k-1}{2^k}$ of correctly identifying if the number is prime or not.

Hence, when working with the test in cryptography, the designers pick a k to balance between the speed of the algorithm, and the accuracy of it as well to

suit their needs, hence providing a very fast algorithm, providing great security with very less probability of failure.