PRIMALITY TESTING
A Journey from Fermat to AKS

Shreejit Bandyopadhyay
Mathematics and Computer Science
Chennai Mathematical Institute

Abstract

We discuss the Fermat, Miller-Rabin, Solovay-Strassen and the AKS tests for primality testing. We also speak about the probabilistic or deterministic nature of these tests and their time-complexities.

1 Introduction

We all know what a prime number is – it's a natural number $p \neq 1$ for which the only divisors are 1 and p. This essentially means that for a prime number p, gcd(a, p)=1 $\forall a \in \mathbf{N}$ and1 $\leq a \leq (p-1)$. So the value of Euler's totient function for a prime p, $\phi(p)$ equals $(p-1)$ as all the $(p-1)$ values of a in (1,2,...$p-1$) satisfy gcd(a, p)=1. A primality test, the topic of this paper, is simply an algorithm that tests, either probabilistically or deterministically, whether or not a given input number is prime. A general primality test does not provide us with a prime factorisation of a number not found to be prime, but simply labels it as composite. In cryptography, for example, we often need the generation of large primes and one technique for this is to pick a random number of requisite size and detrmine if it's prime. The larger the number, the greater will be the time required to test this and this is what prompts us to search for efficient primality tests that are polynomial in complexity. Note that the desired complexity is logarithmic in the number itself and hence polynomial in its bit–size as a number n requires O(log n) bits for its binary representation.

There are some primality tests which conclusively determine whether a number is prime or composite and are therefore determinstc, while others, such as the Fermat and the Miller-Rabin tests, despite correctly classifying all prime numbers, may allow some some composites to filter through, incorrectly labelling them as primes or probably primes, and this makes these tests probabilistic. There are usually four criteria which we look for in an efficient primality-testing algorithm : it must be (a) general, (b) unconditional, (c) deterministic and (d) polynomial in complexity. Tests like Pepin's test,working only for some special primes, are not general while tests like the Miller-Rabin are not deterministic. The earliest primality tests man knew were mostly general and deterministic

but were exponential in complexity. The Agrawal-Kayal-Saxena or the AKS test was the first to satisfy simultaneously all these four criteria unlike earlier tests like the Miller-Rabin and the Solovay-Strassen, about which we first discuss before moving on to AKS. These earlier tests satisfied three of these four criteria but were probabilistic.Throughout the rest of this paper, we speak of complexity with respect to the bit-size of a number and so a complexity that is $O(log^k n)$ is basically polynomial in the bit-size of n and a complexity that is $O(kn^p)$ is basically exponential in the bit-size.

2 The First Primality Tests and The Sieve of Eratosthenes

We all know the simplest primality tests from high school. Given any number n, we can check all numbers less than or equal to $\sqrt{n}$ and see if any of them divides n. If at least one of them does, the algorithm outputs composite and otherwise, it outputs prime. This follows from the fact that if n is indeed composite, it must be possible for us to factor it into at least two factors and at least one of them must be less than or equal to $\sqrt{n}$. However, this test, involving $\sqrt{n}$ operations, is $\Omega(\sqrt{n})$ and hence the algorithm is essentially exponential in bit-size. The algorithm can be made more efficient, however, by skipping all even numbers below $\sqrt{n}$ except 2, because non-divisibility by 2 guarantees non-divisibility by all other even even numbers. Another better modification is obtained by noting that all primes exceeding 3 are of the form $6k \pm 1$ and this restriction further narrows down the list of possible primes, as now only 2 of any 6 consequtive integers are candidates for being a prime. This algorithm can now be presented in pseudocode as follows:

1.Input n. If $n = 2$ or $n = 3$, output PRIME. If $n \neq 2$ and $n \equiv 0 \pmod 2$, output COMPOSITE.
2.Elif $n \not\equiv 1 \pmod 6$ or if $n \not\equiv 5 \pmod 6$, output COMPOSITE.
3. Else : for i in range $(2, \lfloor\sqrt{n}\rfloor+1)$ and $i \not\equiv 0 \pmod 2$, if $n \equiv 0 \pmod i$, output COMPOSITE. Break.
4. Else : $i = i + 1$. If i=(int$\sqrt{n}$)+1, output PRIME.

The time-complexity of the algorithm is readily seen to be $\Omega(\sqrt{n})$.
A method of finding prime numbers from antiquity is the Sieve of Eratosthenes. In this method, if we want to generate all primes upto a certain upper bound, say M, we start with the first prime, i.e. 2 and mark all the multiples of 2 upto M in the list of natural numbers from 1 to M. After this, we take 3, the smallest number in the list (1,2...M) not marked as a multiple of 2, and mark all the unmarked multiples of 3 in the list and continue similarly for the other numbers, sieving with the smallest unmarked number in each step. All the numbers used for sieving are candidates for being primes, i.e., all numbers left unmarked after sieving with all numbers below them are primes themselves. Also note that

when sieving with a particular number n, all multiples of n less than n^2 have already been marked(as $2n, 4n$ etc. are multiples of 2 ; $3n, 9n$ are multiples of 3 etc.)and hence we can essentially start marking the multiples of n from n^2 only.

3 Fermat's Little Theorem and Fermat's Test

Fermat's little theorem is a predecessor of many of the later celebrated primality tests including the Miller-Rabin test and even the AKS algorithm is nothing but a generalization of Fermat's little theorem to a field of polynomials. We state and prove this theorem below.

Theorem : For a prime p, and an integer a, $a^p \equiv a \pmod{p}$. If gcd(a, p)=1, which certainly holds for all a with $1 \leq a \leq (p-1)$, then the cancellation rule for congruences yields $a^{p-1} \equiv 1 \pmod{p}$. The converse of this result is, however not true in general. The proof we present is by induction on a.

Proof: For $a = 1$, p$|1^p - 1$, implying the base case is trivially true. We assume inductively that p$|a^p - a$, for some a.
Now,$(a+1)^p - (a+1)$=a^p+$\binom{p}{1}a^{p-1}$+$\binom{p}{2}a^{p-2}$+...$\binom{p}{p-1}a + 1 - (a+1)$

$$=a^p - a + \sum_{i=1}^{p-1}\binom{p}{i}a^{p-i}$$

Now,$\binom{p}{i}$=$\frac{p!}{i!(p-i)!}$=$\frac{p(p-1)(p-2)..(p-i+1)}{i(i-1)(i-2)..2.1}$. Note that p divides the numerator. If it divides the denominator, it must divide one of the factors since if a prime divides a product of some numbers, it must divide one of them. But as $1 \leq i \leq (p-1)$, $p \nmid (i-j)$ for j=0,1,2...(i-1). So p divides only the numerator, implying $p|\binom{p}{i}$ with $1 \leq i \leq (p-1)$. So $\sum_{i=1}^{p-1}\binom{p}{i}a^{p-i}$ is divisible by p and so is $a^p - a$ (by induction hypothesis), implying $p|(a+1)^p - (a+1)$, completing our proof. However, we must note carefully that the converse of this result is not true in general. Thus although $2^{341} \equiv 2 \pmod{341}$ and $3^{91} \equiv 3 \pmod{91}$, 341=11.31 and 91=7.13 are not primes. But as $3^{341} \not\equiv 3 \pmod{341}$ and $2^{91} \not\equiv 2 \pmod{91}$, neither 91 nor 341 will pass Fermat's test if we try out with both 2 and 3 as bases, i.e., as values of a.
Fermat's primality test says that given any number n as input, pick a random number a with $1 \leq a \leq (n-1)$ and see if gcd$(a, n) = 1$. If not, n is certainly composite, as for a prime, all numbers less than it are coprime to it. However, if the chosen a is indeed coprime to n, we proceed to check the congruence $a^{n-1} \equiv 1 \pmod{n}$. If the congruence does not hold, n is certainly composite but if it does, we cannot say with certainty that n is prime. In that case, we may try out another value of a and repeat the test, thereby improving its accuracy and, in general, may try the test for all a with $1 \leq a \leq (n-1)$, but, as we shall see, we are not only compensating on the complexity of the algorithm in this case, but that n cannot always be correctly classified as composite even if the congruence does hold for all a in our range of values. Before going on to see for

what values of n this is the case in the next section, we now present Fermat's algorithm in pseudocode as follows.

1. Input n
2. For a in range (1,$n-1$),
   Choose a. If gcd(a,n) >1, output COMPOSITE.
   Elif gcd(a,n)=1, compute $x = a^{n-1} \pmod{n}$. If $x \neq 1$,output COMPOSITE.
   Else return PROBABLY PRIME.
3. Repeat steps 1 and 2 for k values of a in range (1,$n-1$)

Here k is a parameter denoting the number of bases (values of a) we test and essentially controlling the accuracy as well as the complexity of the test.

For example, let's suppose that we wish to determine whether 299 is prime by Fermat's test. For this purpose, we first randomly choose an $a \in (1, 2...298)$. Let the chosen value of a be 116. We see that gcd(116,299)=1 and that $116^{298} \equiv 1 \pmod{299}$, which shows that 299 is either a prime or a Fermat pseudoprime to base 116. To verify further, we choose $a = 183$ and note that gcd(183,299)=1. On computation, we find that for 183 also, $183^{298} \equiv 1 \pmod{299}$, implying that 299 is indeed a prime or a Fermat pseudoprime to both the bases 116 and 183. However, on putting a=155, we note that gcd(155,299)=1 and $155^{298} \equiv 144 \not\equiv 1 \pmod{299}$ and hence that 299 is composite. Thus, 116 and 183 were indeed Fermat liars for 299. Looking closely, we see that 116 and (299-116)=183 were both Fermat liars. This follows from the fact that $(n-a)^{n-1}-a^{n-1}$ is divisible by $(n-a+a) = n$ since $n-1$ is even and $x+y|x^m-y^m$ for even m. So if a is a Fermat liar for an odd composite number n, i.e., if gcd(a,n)=1 and $a^{n-1} \equiv 1 \pmod{n}$, then evidently gcd$(n-a,n)$=gcd$(a,n)$=1 and since $n|(n-a)^{n-1}-a^{n-1}$, it implies that $(n-a)^{n-1} \equiv a^{n-1} \equiv 1 \pmod{n}$, meaning that $(n-a)$ is also a Fermat liar for n. Since this implies that Fermat liars always occur in pairs, there will always be an even number of such liars for an odd composite number n. So having checked a=116, we need not have checked $a = 183$ at all. Similarly, in Fermat's algorithm, if we find that n is a probable prime to base a, for some $a \in (1, 2...n-1)$ with gcd(a,n)=1, we need never check for $(n-a)$. In fact, 116 and 283 are the only Fermat liars for 299.

4 Flaws in Fermat's Test : The Carmichael Numbers and Korselt's Criterion

In the last section, we saw that even though $3^{91} \equiv 3 \pmod{91}$ and $2^{341} \equiv 2 \pmod{341}$, 91 and 341 are not primes. In fact, these numbers are called pseudoprimes to the chosen base, i.e., the value of a. So we conclude that 91 and 341 are pseudoprimes to base 3 and base 2 respectively. However, since $2^{341} \not\equiv 2 \pmod{341}$ and $3^{91} \not\equiv 3 \pmod{91}$, we may correctly pronounce these two numbers as composite by testing out both the bases 2 and 3 for them and, in general,

we may hope that if we try out all values of a with $1 \leq a \leq (n-1)$, n being the input number, we will always be accurate with our conclusions from this test. This will be the case if there is no number n that is a pseudoprime or a Fermat pseudoprime to all values of a with $a \leq (n-1)$ and gcd(a, n)=1. However, there do exist such numbers and, for these numbers, we will not be able to predict their compositeness even by carrying out Fermat's test for all bases. Examples of such numbers were first found by Carmichael and hence, these numbers are called Carmichael numbers.

Definition : A Carmichael number or an absolute Fermat pseudoprime is a composite natural number n satisfying the congruence $a^{n-1} \equiv 1 \pmod{n}$ for all a with gcd(a, n)=1. Definitely, for all a with $1 \leq a \leq (n-1)$, it satisfies $a^n \equiv a \pmod{n}$.
Korselt's criterion is one which imparts a classifying characteristic to all Carmichael numbers, the first of which is 561.

Fact (Korselt's criterion) : If n is a Carmichael number, i.e., if for all a with gcd(a,n)=1, $a^{n-1} \equiv 1 \pmod{n}$, then it satisfies the following two conditions:

(a) n must be square-free, i.e., for any prime p, if $p|n$, $p^2 \nmid n$.
(b) If p is a prime with $p|n$, then $(p-1)|(n-1)$.

The first Carmichael number is 561=3×11 × 17 and it's easy to see that it indeed satisfies Korselt's criterion. There is also a result which says that any Carmichael number is a product of at least 3 odd primes, which, according to Korselt's criterion (a) above, must be distinct.

5 Probability of Success in Fermat's Test

In the last section, we saw that Carmichael numbers are impossible to detect as composites in Fermat's algorithm but, fortunately enough for us, such numbers are indeed few and far between. The number of Carmichael numbers below 10^6 is just 43 and below 10^{16}, it's just less than 2.5×10^5, although the number of primes in this range is more than 2.5×10^{14}, so that the probability that a Fermat prime is actually a Carmichael number is negligibly small. In this section, we ask ourselves – what's the probability of success (i.e., a composite number being labelled as composite) in Fermat's test and how does this probability increase with increase in the number of bases we test out ? For this, we prove a theorem, taking the following easily verifiable fact from group theory.

Fact : $\mathbf{Z}/n\mathbf{Z}$, which denotes the set of residue classes modulo n and coprime to n, is a group under multiplication.

Definition : A number a is said to be a Fermat witness for a composite n if gcd$(a, n) > 1$ or if $a^{n-1} \not\equiv 1 \pmod{n}$, it's actually a witness for the compos-

iteness of n. On the other hand, if n is composite and, for a particular a with gcd(a, n)=1, $a^{n-1} \equiv 1 \pmod{n}$, a is called a Fermat liar for n, i.e., n is, in this case, a pseudoprime to base a.

Now, we state our theorem and prove it.

Theorem : Given any composite non-Carmichael number n, the number of Fermat witnesses in the set (1,2,3...$(n-1)$) coprime to n is greater than or equal to the number of Fermat liars. So, if we choose an a with gcd$(a, n) = 1$ from this set arbitrarily, the probability that it's a Fermat witness, i.e., the probability that n is correctly labelled as composite, is at least $\frac{1}{2}$.

Proof : Let b be a Fermat liar for a composite number n. If no such b exists, all numbers in (1,2...n-1) are Fermat witnesses and since $n \geq 4$, we are trivially done. Otherwise, there is a b with gcd(b, n)=1 such that $b^{n-1} \equiv 1 \pmod{n}$. Since n is not a Carmichael number, it has at least one Fermat witness a with gcd(a, n)=1 and $a^{n-1} \not\equiv 1 \pmod{n}$. Working in $\mathbf{Z}/\mathrm{n}\mathbf{Z}$, we see that $ab^{n-1} = a^{n-1}b^{n-1} = a^{n-1}.1$ (since $b^{n-1} \equiv 1 \pmod{n}$) and $a^{n-1} \neq 1$ as $a^{n-1} \not\equiv 1 \pmod{n}$, implying $ab^{n-1} \not\equiv 1 \pmod{n}$, so that ab is also a Fermat witness. Since b is not the identity, $ab \neq a$ and we see that given any witness a and any liar b, we can find another witness ab. So the number of Fermat witnsses is at least as many as the number of Fermat liars, implying that our success probability in Fermat's test is at least $\frac{1}{2}$, whenever we are not dealing with a Carmichael number. If we take k different values of a, the probability of failure is $\frac{1}{2^k}$ at most; in many cases, it's much smaller than this.

Fermat's test is probabilistic and certainly forms a basis for the more celebrated algorithms like the Miller-Rabin test, which we now move on to. One advantage of the Miller-Rabin test is, as we shall see, that there is no absolute pseudoprime or Carmichael number equivalent in this test, although it's still probabilistic. Another advantage is the Miller-Rabin's increased success probability as compared to Fermat.

6 The Miller-Rabin Primality Test

The Miller-Rabin primality test, due to Michael O. Rabin and Gary L. Miller, is an unconditional but probabilistic algorithm for primality testing; its probability of success is, however, greater than than that of Fermat's test. Before moving on to the pseudocode itself in the next section, we first describe the algorithm now. Let n be our input number. If $n = 2$, we output prime and for all other even numbers, we output composite, leaving us only with the odd numbers n. We now write $(n-1)$, an even number, as $n - 1 = 2^s.r$, where r is odd. Now, note that if n is a prime, then for any a with gcd(a, n)=1, the congruence $a^{n-1} \equiv 1 \pmod{n}$ must hold, by Fermat's little theorem, implying $a^{2^s.r} \equiv 1 \pmod{n}$. Taking square roots on both sides, we get $a^{2^{s-1}.r} \equiv \pm 1 \pmod{n}$. This is because if $a^{2^s.r} \equiv 1 \pmod{n}$, it implies that

$n|a^{2^s.r} - 1 \Longrightarrow n|(a^{2^{s-1}.r} + 1)(a^{2^{s-1}.r} - 1) \Longrightarrow n|(a^{2^{s-1}.r} + 1)$ or $n|(a^{2^{s-1}.r} - 1)$, since n is a prime. This means that $a^{2^{s-1}.r} \equiv \pm 1 \pmod{n}$. If we continue to take square roots repeatedly, we will get –1 once, i.e., will get the congruence $a^{2^j.r} \equiv -1 \pmod{n}$ for some $j \in (0, 1, 2...s-1)$ or otherwise, we will ger $a^r \equiv 1 \pmod{n}$. So, whenever n is a prime, either $a^r \equiv 1 \pmod{n}$ or $a^{2^j.r} \equiv -1 \pmod{n}$ for some $j \in (0, 1, 2...s-1)$. If both fail to hold, the algorithm outputs composite but if they do, we, much like Fermat's test, cannot say with certainty that n is prime, but must check for another value of a to improve our accuracy.

Before moving on, we now give a definitions.

Definition : Let n be an odd composite number and let $(n-1) = 2^s r$, with r odd, and let $a \in$ (1,2,3..n-1) with gcd(a,n)=1. If $a^r \not\equiv 1 \pmod{n}$ and $a^{2^j.r} \not\equiv -1 \pmod{n} \forall j \in (0, 1, 2..s-1)$, a is called a strong witness for n.

If either of the two congruences holds for some j $\in$ (0,1,2..s-1), a is called a strong liar for n. However, there is no n for which all $a \in$ (1,2,3..$n-1$) with gcd(a,n) are strong liars, implying that there is no absolute pseudoprime analog in the Miller-Rabin test. We now state below a fact without proof.

Fact : If n is an odd composite integer, the number of strong liars for n is less than or equal to $\phi(n)/4$, where $\phi(n)$ is the Euler's totient function for n.

Since n is composite, $\phi(n) < n-1$,implying that n can pass the Miller-Rabin's test, i.e., will be labelled composite by it, for at most $(n-1)/4$ values of a with $1 \leq a \leq (n-1)$ and gcd(a,n)=1. So the probability that a randon a in this set is a strong liar for n is at most $\frac{1}{4}$, and if we carry out the test for k different bases, i.e., k different values of a, the failure probability is at most $\frac{1}{4^k}$. Comparing this with the accuracy in Fermat's test, we find that the Miller-Rabin test basically reduces the number of pseudoprimes by at least half. For example, consider the number 133, which is composite. The set of all Fermat liars for 133 is (1,8,11,12,18,20,26,27,30,31,37,39,45,46,50,58,64,65,68,69,75,83,87,88,94,96,102, 103,106,107,113,115,121,122,125,132) while the set of all strong liars is (1,11,12, 27,30,31,39,58,64,69,75,94,102,103,106,121,122,132). Thus we see that although 133 has 36 fermat liars, it has only 18, or half that number,of strong liars. Note that the number of strong liars of 133, 18, is less than $\phi(133)/4$=108/4=27. Again, as we have already said, though there are Carmichael numbers whichsatisfy $a^{n-1} \equiv 1 \pmod{n}$ for all a with gcd(a, n)=1, even as n is composite, there are no numbers n which satisfy $a^r \equiv 1 \pmod{n}$ and $a^{2^j r} \equiv -1 \pmod{n}$ for all $j \in$ (1,2...$s-1$), where r, s have usual significances, for all $a \in$ (1,2...$n-1$) with gcd(a, n) = 1, so that there is no absolute strong pseudoprime in this test.

7 The Miller-Rabin Algorithm and its Complexity

We now give in pseudocode the Miller-Rabin algorithm for primality testing. Note the underlying Fermat's little theorem and compare it with Fermat's algorithm we have already presented in our previous sections.

1. Input a number n. If $n \equiv 0 \pmod{n}$ and $n \neq 2$, output COMPOSITE. If n=2, output PRIME.
2. Write $n-1 = 2^s.r$, with r an odd number.
3. For a in (1,2...$n-1$), if gcd$(a,n) \neq 1$, output COMPOSITE.
   Elif gcd$(a,n) = 1$, compute $y = a^r \pmod{n}$
   If $y \neq 1$ and $y \neq (n-1)$, go to 4.
4. Put $y = y^{2^j} \pmod{n}$, starting with j=1. If $y = 1$, return COMPOSITE.
5. Else $j = j+1$. Continue upto $j = s-1$.
6. If $y \neq n-1$, return COMPOSITE. Repeat steps 1-6 for k values of $a \in (1, 2...n-1)$
7. Return PROBABLY PRIME.

Here k denotes the number of bases we test for n and hence controls the accuracy of the test.

For example, let's suppose that we wish to determine whether or not n=247 is prime, by the Miller-Rabin test. So we write $n-1$=246 as 246=2^1.123, so that $s = 1$ and $r = 123$, with r and s as defined in step 2 of the algorithm. To determine if 247 is prime, we now choose an $a \in (1, 2...246)$. Let the chosen value of a be $a = 68$. Note that gcd$(68, 247)$=1, and that $68^{123} \equiv 1 \pmod{247}$, implying that 247 is either a prime or else, 68 is a strong liar for it. To verify further, we put $a = 75$ and note that gcd$(75, 247)$=1. But we find that in this case also, $75^{123} \equiv 246 \equiv -1 \pmod{247}$, so that 247 is inded a prime, or both 68 and 75 are strong liars for it. To make sure, we put $a = 105$ and see that gcd$(105, 247)$=1. However, $105^{123} \equiv 27 \not\equiv \pm 1 \pmod{247}$, so that we conclude that 247 is composite and 105 is a strong witness for it. If we write a program to determine all the strong liars of 247, we shall find that the set of all such strong liars of 247 is (1,12,56,68,69,75,87,88,103,144,159,160,172,178,179,191,235,246). Note that if a is a strong liar for the composite number n, so will be $(n-a)$, as can be easily shown,(it's evident in the above list of all strong liars of 247 also) and so having checked for a, and having found n to be a probable prime to base a, we can essentially skip the test with $(n-a)$ as a base.

The complexity of the Miller-Rabin algorithm can be determined from its pseudocode. We see that in this test, multiplication of two numbers is always done modulo n and any such multiplication must essentially involve O(log $n \times$log n) or O($log^2 n$) time for its completion as n can be represented in log n bits. But, for any a, we can compute $a^n \bmod n$ in O(log n) multiplications; we may break

n into powers of 2 (say) and then proceed. For example, if we wish to raise a, say, to power 19 modulo n, we need to carry out only the following multiplications modulo n, namely : $a.a$, $a^2.a^2$, $a^4.a^4$, $a^{16}.a^2$ and $a^{18}.a$. It is thus evident that this method of modular exponentiation can be done in polynomial time and the complexity is essentially O(log $n \times \log^2 n$)=log $^3 n$ for each value of a we test out. This method of multiplication by raising to powers of 2 is called modular exponentiation by repeated squaring. Since this exponentiation has the dominant complexity in the above algorithm, we conclude that the overall time-complexity of the Miller-Rabin algorithm is O($k \times \log^3 n$) = O($klog^3 n$), k being the number of bases we test out, and hence it's inded polynomial in the bit-size of n.

Although the Miller-Rabin test is probabilistic as we have stated it, we can obtain a deterministic version of it by using the unproven Generalized Riemann Hypothesis. It's a known fact that all strong liars for a composite number n are contained in a proper subgroup of the multiplicative group $\mathbf{Z}/\mathrm{n}\mathbf{Z}$, generated by a number $x < 2log^2 n$, if we assume the validity of this hypothesis, and this puts this deterministic version of the Miller-Rabin algorithm at a time-complexity of O($log^4 n$).

However, for all practical purposes, the probabilistic version works better as it involves a lower time-complexity. Theoretically, however, this version might have been of considerable interest, but has largely been replaced by the AKS test, whish is simultaneously deterministic and unconditional. Before we discuss the AKS algorithm, however, we present another probabilistic primality-testing algorithm based on Fermat's little theorem–the Solovay-Strassen primality test.

8 Quadratic Residues and the Legendre Symbol

The concept of quadratic residues and the notation of the Legendre symbol are central to the understanding and application of the Solovay-Strassen primality test, another probabilistic primality-testing algorithm. Before we proceed, we first give a definition.

Definition : An integer a is said to be a quadratic residue modulo another natural number n if there exists a b such that $a \equiv b^2 \pmod{n}$ and a quadratic nonresidue modulo n otherwise.
The Legendre symbol is basically a multiplicative indicator function, denoted by $(\frac{a}{p})$, where a is any integer and p a prime number. It's defined as follows:

$(\frac{a}{p})$=1, if $a \equiv b^2 \pmod{p}$ has a solution, i.e., if a is a quadratic residue modulo p,
and $(\frac{a}{p})$= –1 if a is a quadratic nonresidue modulo p.

In addition, some choose to define $(\frac{a}{p})$=0 if $a \equiv 0 \pmod{p}$. We now state a theorem and prove it, remembering always that p is an odd prime number.

Proposition : For a prime p, the number of quadratic residues mod p, i.e., the number of integers a with $1 \leq a \leq p-1$ such that $a \equiv b^2 \pmod{p}$ has a solution, is $\frac{p-1}{2}$ and the number of quadratic nonresidues mod p, i.e., the number of such a with $1 \leq a \leq p-1$ for which no such b exists is also $\frac{p-1}{2}$. Thus exactly half the numbers a with $1 \leq a \leq p-1$ satisfy $(\frac{a}{p})$=1 while the others have $(\frac{a}{p})$= –1. Note that for $1 \leq a \leq p-1$, $p \not| a$ and hence $(\frac{a}{p}) \neq 0$ for any a.

Proof : Let's take any x with $1 \leq x \leq \frac{p-1}{2}$. Then note that if $a \equiv x^2 \pmod{p}$, we have $a \equiv (-x)^2 \pmod{p}$, i.e., $a \equiv (p-x)^2 \pmod{p}$. So the quadratic residue a is congruent modulo p to not only an integer x with $1 \leq x \leq \frac{p-1}{2}$ but also to another integer $(p-x)$ with $\frac{p+1}{2} \leq p-x \leq p-1$. Again, if $a \equiv x^2 \pmod{p}$, it also satisfies $a \equiv (yp+x)^2 \pmod{p}$ for any integer y. From this, we can conclude that any quadratic residue modulo p is congruent to one of the integers $1^2, 2^2, 3^2 ... (\frac{p-1}{2})^2$ mod p. Since each such residue is congruent to two such numbers in the set $(1^2, 2^2, 3^2 ... (p-1)^2)$, modulo p, the number of quadratic residues can be at most half the numbers in this set, i.e., $\frac{p-1}{2}$. But does this prove that there are exactly $\frac{p-1}{2}$ quadratic residues modulo p? It doesn't, for to complete our proof, we also need to show that any such residue is congruent to exactly one of the $\frac{p-1}{2}$ squares in the set $(1^2, 2^2, 3^2 ... (\frac{p-1}{2})^2)$, i.e., if $a \equiv x^2 \pmod{p}$ and $a \equiv y^2 \pmod{p}$, with both x, y satisfying $1 \leq x,y \leq \frac{p-1}{2}$, then $x = y$. If we do show this, it would imply that one and only one number in the set $(1^2, 2^2, 3^2 ... (\frac{p-1}{2})^2)$ and one and only one number in the set $((\frac{p+1}{2})^2, (\frac{p+3}{2})^2, ... (p-1)^2)$ are congruent to any quadratic residue mod p, thereby implying that there are exactly $\frac{p-1}{2}$ quadratic residues mod p. To show that if $a \equiv x^2 \pmod{p}$ and $a \equiv y^2 \pmod{p}$, with both x, y satisfying $1 \leq x,y \leq \frac{p-1}{2}$, then $x = y$, note that if $a \equiv x^2 \pmod{p}$ and $a \equiv y^2 \pmod{p}$, it implies $x^2 \equiv y^2 \pmod{p} \Longrightarrow p|(x^2 - y^2) \Longrightarrow p|(x+y)(x-y) \Longrightarrow p|(x+y)$ or $p|(x-y)$ as p is a prime. But $x \leq \frac{p-1}{2}$ and, if we assume $x > y$ without loss of generality, $y \leq \frac{p-3}{2} \Longrightarrow (x+y) \leq (p-2)$, implying $p \not| x+y < p-2$. So the only case possible is $p|x-y$ and since $x - y \leq \frac{p-3}{2}$, the only possibility is that $x - y = 0$ or $x = y$. This shows that there are exactly $\frac{p-1}{2}$ quadratic residues. The other $\frac{p-1}{2}$ numbers in the set (1,2,3...$p-1$) are tus quadratic nonresidues, proving our proposition.

For example, if we consider the prime 7, the quadratic residues modulo 7 are 1,2 and 4 since $1 \equiv 1^2 \equiv -1^2 \equiv 6^2 \pmod{7}$, $2 \equiv 3^2 \equiv -3^2 \equiv 4^2 \pmod{7}$ and $4 \equiv 2^2 \equiv -2^2 \equiv 5^2 \pmod{7}$. The remaining 3 numbers 3,5 and 7 in the set (1,2,3...6) are quadratic nonresidues modulo 7. Note that each quadratic residue is congruent to exactly one of the squares in the set $(1^2, 2^2, 3^2)$ and exactly one square in the set $(4^2, 5^2, 6^2)$.
Note that throughout the above discussion, we have excluded the case of 0. This 0 is of course congruent to $(yp)^2$ mod p for any integer y and hence can be regarded as a quadratic residue. If we, like some authors, do include zero as a

possible candidate for being a residue, then we will have $\frac{p+1}{2}$ quadratic residues and $\frac{p-1}{2}$ quadratic nonresidues for any prime number p.

9 Euler's Criterion and the Solovay-Strassen Algorithm

Euler's criterion, forming the basis of the Solovay-Strassen algorithm, is essentially based on Fermat's little theorem. We state and prove his criterion below and then state some of the important properties of the Legendre symbol, about which we have already discussed.

Theorem (Euler's criterion) : Let a be any integer and p any prime number. Then, $a^{\frac{p-1}{2}} \equiv (\frac{a}{p}) \pmod{p}$, where $(\frac{a}{p})$ is the Legendre symbol.

Proof : Case I: Let a be a quadratic residue modulo p. Then, by definition, $(\frac{a}{p})$=1 and there exists an x such that $a \equiv x^2 \pmod{p}$. By Fermat's little theorem, $x^{p-1} \equiv 1 \pmod{p}$, since $x \not| p$, implying $\gcd(x, p) = 1$. But $a^{\frac{p-1}{2}} \equiv (x^2)^{\frac{p-1}{2}} \pmod{p}$ as $a \equiv x^2 \pmod{p}$, implying $a^{\frac{p-1}{2}} \equiv x^{p-1} \pmod{p} \equiv 1 \pmod{p}$. Since $(\frac{a}{p})$=1 in this case, by the definition of Legendre symbol, we conclude that $a^{\frac{p-1}{2}} \equiv (\frac{a}{p}) \pmod{p}$.

Case II : Let a be a number such that $a \equiv 0 \pmod{p}$. Then $(\frac{a}{p})$=0 by definition, and since $p|a$, we have that $p|a^{\frac{p-1}{2}}$. So $a^{\frac{p-1}{2}} \equiv 0 \pmod{p} \equiv (\frac{a}{p}) \pmod{p}$, proving Euler's criterion for this case.

Case III : Let a be a quadratic nonresidue modulo p. Since p is prime, we get $a^{p-1} \equiv 1 \pmod{p}$ by another application of Fermat's little theorem. This implies that $(a^{\frac{p-1}{2}} - 1)(a^{\frac{p-1}{2}} + 1) \equiv 0 \pmod{p}$. Now, by Lagrange's theorem, an integer polynomial in $\mathbf{Z}/\mathrm{p}\mathbf{Z}$ with degree n has at most n roots. So the equation $a^{p-1} \equiv 1 \pmod{p}$ has at most $\frac{p-1}{2}$ zeroes and as we have shown in Case I that the quadratic residues do satisfy this equation and, in the last section that there are exactly $\frac{p-1}{2}$ residues of a prime p, the equation does have exactly $\frac{p-1}{2}$ zeroes, i.e., there are exactly $\frac{p-1}{2}$ values for which $a^{p-1} \equiv 1 \pmod{p}$. But then, the remaining $\frac{p-1}{2}$ numbers (we have already examined the cases of the $\frac{p-1}{2}$ residues and the case corresponding to a number divisible by p, i.e, congruent to 0 mod p) in $\mathbf{Z}/\mathrm{p}\mathbf{Z}$ must make the second factor in the equation $(a^{\frac{p-1}{2}}-1)(a^{\frac{p-1}{2}}+1) \equiv 0 \pmod{p}$ zero because in a field a product of two factors being zero implies that at least one of them must be zero. But these remaining $\frac{p-1}{2}$ numbers in $\mathbf{Z}/\mathrm{p}\mathbf{Z}$ are precisely the quadratic nonresidues modulo p and this requirement forces them to make the second factor in the above equation zero, i.e., to satisfy $(a^{\frac{p-1}{2}} + 1) \equiv 0 \pmod{p}$. So for these nonresidues, $(a^{\frac{p-1}{2}}) \equiv -1 \pmod{p}$ and $(\frac{a}{p})$= –1, by definition. So Euler's criterion $(a^{\frac{p-1}{2}}) \equiv (\frac{a}{p}) \pmod{p}$

holds for these quadratic nonresidues also, prooving the above theorem.

Using Euler's criterion, we can also show the multiplicative nature of the Legendre symbol. Since we have from his criterion that $a^{\frac{p-1}{2}} \equiv (\frac{a}{p}) \pmod{p}$, we get $(\frac{ab}{p}) \equiv (ab)^{\frac{p-1}{2}} \equiv (a^{\frac{p-1}{2}})(b^{\frac{p-1}{2}}) \equiv (\frac{a}{p})(\frac{b}{p}) \pmod{p}$, implying $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ modulo p. Using induction, we can easily extend this to more than two variables, showing the multiplicative nature of the Legendre symbol. We also note that if $a \equiv b \pmod{p}$, then $a \equiv x^2 \pmod{p}$ has a solution if and only if $b \equiv x^2 \pmod{p}$ is solvable which immediately forces the fact that $a \equiv b \pmod{p} \Longrightarrow (\frac{a}{p}) = (\frac{b}{p})$. Euler's criterion forms the basis of the Solovay-Strassen algorithm, which we present next. But first, we introduce the concept of the Jacobi symbol, a generalisation of the legendre symbol.

Definition (Jacobi symbol) : Let a be any integer and n an odd integer. Then the Jacobi symbol, denoted by $(\frac{a}{n})$, is defined as

$(\frac{a}{n}) = \prod_{i=1}^{k} (\frac{a}{p_i})^{\alpha_i}$, where $n = p_1^{\alpha_1}.p_2^{\alpha_2}...p_k^{\alpha_k}$ is the prime decomposition of n and $(\frac{a}{p_i})$ is the Legendre symbol.

This Jacobi symbol is known to possess the following properties. Note that this symbol too is multiplicative just like the Legendre symbol we have already discussed.

1. Let a, b be two integers and let m, n be two odd integers. Then $(\frac{ab}{n}) = (\frac{a}{n})(\frac{b}{n})$ and $(\frac{a}{mn}) = (\frac{a}{m})(\frac{a}{n})$.
2. $(\frac{a}{n}) = (\frac{a \ mod n}{n})$.
3. For odd primes p, q with $p \neq q$, $(\frac{p}{q})=(-1)^{\frac{(p-1)(q-1)}{4}}(\frac{q}{p})$, where $(\frac{p}{q})$ is the Legendre symbol. This is the well-known law of quadratic reciprocity of Gauss. Generalising this to any two relatively prime odd numbers m, n, it can be shown that $(\frac{m}{n}) = (-1)^{\frac{(m-1)(n-1)}{4}}(\frac{n}{m}) \Longrightarrow (\frac{m}{n}) = (\frac{n}{m})$ iff $m = n \equiv 3 \pmod{4}$ and $(\frac{m}{n})=(-\frac{n}{m})$, otherwise, i.e., when either of them is congruent to 1 mod 4. This is a generalisation of the law of quadratic reciprocity to any odd number appearing in the denominator, and is sometimes called Jacobi's law of reciprocity.

4. The following can be derived from the quadratic reciprocity law : $(\frac{2}{n})=(-1)^{\frac{(n^2-1)}{8}}$, $(\frac{1}{n})=1$ and $(\frac{0}{n})=0$, where $n \geq 3$ is any odd integer.

Armed with these results, we now give the following algorithm for calculating the value of the Jacobi symbol, which we shall need for implementing the Solovay-Strassen test. Note the similarity in the recursive formulation of the algorithm given below with that of the familiar Euclidean algorithm for computing the gcd of two integers.

1. Input an integer a and an odd integer n. If $a > n$, put $a = a \bmod n$.

Else go to step 2.

2. If $a = 0$, output 0. If $a = 1$, output 1.
3. Write $a = 2^r.s$, with odd s. If $r > 0$, put $(\frac{a}{n}) = (\frac{2^r.s}{n}) = (\frac{2}{n})^r.(\frac{s}{n})$ and calculate $(\frac{2}{n}) = (-1)^{\frac{(n^2-1)}{8}}$ and then $(\frac{2}{n})^r$. Then go to step 4. Also if $r = 0$, go to 4.
4. Now we only need to compute $(\frac{s}{n})$. If $s = n \equiv 3 \pmod 4$, output $(\frac{n}{s})$. Else output $-(\frac{n}{s})$. Then go to 1 and proceed by recursion.

Having found out a way of computing the value of the Jacobi symbol, we proceed to first describe the Solovay-Strassen algorithm, before later presenting it in pseudocode.

Let's consider an input number n. To determine if n is prime or not, we first of all check whether or not n satisfies Euler's criterion, i.e., we see if the congruence $a^{\frac{n-1}{2}} \equiv (\frac{a}{n}) \pmod n$ holds, where $2 \leq a \leq n-1$ and gcd(a, n)=1. If it does not, the algorithm outputs composite, but if it does, it checks with another value of a in some predetermined range of values, outputting 'probably prime' when all such values have been exhausted.

Definition : Let n be an odd composite integer and a with $1 \leq a \leq n-1$ be an integer. If gcd(a, n) >1 or if $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \pmod n$, a is said to be Euler witness for n, i.e., a is a witness for the compositeness of n. If, however, gcd(a, n)=1 and $a^{\frac{n-1}{2}} \equiv (\frac{a}{n}) \pmod n$, a is said to be an Euler liar for n and n is a pseudoprime or an Euler pseudoprime to base a.

Before we move on, we state and prove two important theorems.

Theorem : Let n be any odd composite number. Then n has at least one Euler witness a, i.e., there's at least one a in the set (1,2...$n-1$) with gcd(a, n)=1 such that $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \pmod n$.

Proof : The proof we give is by contradiction. If the result does not hold, then we must have that for all a in (1,2...$n-1$) with gcd(a, n)=1, $a^{\frac{n-1}{2}} \equiv (\frac{a}{n}) \pmod n \implies a^{n-1} \equiv [(\frac{a}{n})]^2 \equiv 1 \pmod n$ since $(\frac{a}{n}) = \pm 1$. If this indeed holds for all a with $1 \leq a \leq n-1$ and gcd(a, n)=1, n must be a Carmichael number. Thus to complete the proof of our theorem, we only need to show that any such Carmichael number n has always at least one Euler witness. By Korselt's criterion, we know that any Carmichael number is square-free, i.e., there's no prime such that $p|n$ and $p^2|n$. So n, being odd and composite, must be a product of distinct odd primes. Let p be the smallest among the prime factors of n. Then $n = pk$, for some odd integer k. Consider any number x which is a quadratic non-residue modulo p, i.e., for which the congruence $x \equiv y^2 \pmod p$ has no solution for an integer y. Then, by the definition of the Legendre symbol, $(\frac{x}{p}) = -1$. Let's take the two congruences $m \equiv x \pmod p$ and $m \equiv 1 \pmod k$. Can the same m satisfy both these congruences? By the Chinese Remainder Theorem,

there is an m which satisfies both. Now consider $(\frac{m}{n}) = (\frac{m}{pk}) = (\frac{m}{p})(\frac{m}{k})$, by the multiplicative property of the Legendre symbol. Now recollect the property that $m \equiv x \pmod{p} \Longrightarrow (\frac{x}{p}) = (\frac{x}{p})$. So here $(\frac{m}{p}) = (\frac{x}{p}) = -1$ and since $m \equiv 1 \pmod{k}$, $(\frac{m}{k}) = 1$. So, $(\frac{m}{n}) = -1.1 = -1$. Note that x being a quadratic non-residue modulo p, $x < p \Longrightarrow p \not| x$. But $m \equiv x \pmod{p} \Longrightarrow p|(m-x) \Longrightarrow p \not| m$ as $p \not| x$. So $\gcd(m, n) = gcd(m, pk) = 1$ because $gcd(m, k) = 1$ (since $m \equiv 1 \pmod{k}$) and $\gcd(m, p) = 1$ as $p \not| m$. So m satisfies $\gcd(m, n) = 1$ and $(\frac{m}{n}) = -1$. If m is an Euler liar, $m^{\frac{n-1}{2}} \equiv (\frac{m}{n}) \equiv -1 \pmod{n}$. Then, $m^{\frac{n-1}{2}} \equiv -1 \pmod{k}$ as $k|n$. But as $m \equiv 1 \pmod{k}$, $m^{\frac{n-1}{2}} \equiv 1^{\frac{n-1}{2}} \equiv 1 \pmod{k}$ and hence we have a contradiction. So $\gcd(m, n)$=1 and $m^{\frac{n-1}{2}} \not\equiv (\frac{m}{n} \pmod{n}$, implying that m is an Euler witness for n. So we have proved the result that any odd composite number n has at least one Euler witness.

We now state and prove another theorem concerning the number of Euler liars and Euler witnesses.

Theorem : Let n be an odd composite integer. Then the number of Euler witnesses b with $\gcd(b, n)$=1 in the set (1,2...$n-1$), is greater than or equal to the number of Euler liars a, i.e., at least half of the integers x with $\gcd(x, n)$=1 and $x \in (1, 2...n-1)$ satisfy $x^{\frac{n-1}{2}} \not\equiv (\frac{x}{n}) \pmod{n}$.

Proof : Let a be an Euler liar for the odd composite integer n. If there is no such liar, then all numbers coprime to n in the set (1,2...$n-1$) are Euler witnesses and we are trivially done. By the last theorem, we can also find an Euler witness b of n in this set with $\gcd(b, n)$=1. Since $\gcd(a, n)$=1, a being an Euler liar, we must have $\gcd(ab, n)$=1. Again, since a is an Euler liar and b an Euler witness, $(\frac{a}{n}) \equiv a^{\frac{n-1}{2}} \pmod{n}$ and $(\frac{b}{n}) \not\equiv b^{\frac{n-1}{2}} \pmod{n}$. Let, if possible, ab be an Euler liar. Then $(\frac{ab}{n}) \equiv ab^{\frac{n-1}{2}} \pmod{n}$. But $(\frac{ab}{n}) = (\frac{a}{n})(\frac{b}{n})$, by the multiplicative property of the Jacobi symbol, implying $(\frac{a}{n})(\frac{b}{n}) = (\frac{ab}{n}) \equiv (ab)^{\frac{n-1}{2}} \pmod{n} \equiv (a)^{\frac{n-1}{2}}(b)^{\frac{n-1}{2}} \pmod{n}$. But since $(\frac{a}{n}) \equiv a^{\frac{n-1}{2}} \pmod{n}$ and $(\frac{b}{n}) \not\equiv b^{\frac{n-1}{2}} \pmod{n}$, $(\frac{ab}{n}) \not\equiv ab^{\frac{n-1}{2}} \pmod{n}$, forcing a contradiction and proving that ab, satisfying $\gcd(ab, n)$=1, is also an Euler witness. Thus for any witness b and any liar a, we can generate another witness ab, implying that at least half the integers coprime to n in the set (1,2...$n-1$) are actually Euler witnesses for n.

This above theorem imposes a lower bound on the accuracy of the Solovay-Strassen algorithm, as we shall see in the next section. Before we proceed, however, we present the pseudocode of this algorithm.

1. Input n. If n=2, output PRIME. If $n \neq 2$ and $n \equiv 0 \pmod{2}$, output COMPOSITE.
2. Choose a in range (1,$n-1$), and find $\gcd(a, n)$. If $\gcd(a, n) > 1$, output COMPOSITE.

3. Compute $x = a^{\frac{n-1}{2}} \pmod{n}$. If $x \neq 1$ and $x \neq n-1$, output COMPOSITE.
4. Compute $y = (\frac{a}{n})$. Note that we have already given before a recursive algorithm for computing the value of the Jacobi symbol.
5. If $x \not\equiv y \pmod{n}$, output COMPOSITE. Repeat steps 1-4 for some k values of $a \in (1, 2...n-1)$
6. Return PROBABLY PRIME.

For example, let's suppose that we wish to use the Solovay-Strassen algorithm to determine if n=205 is prime. For this, we arbitrarily choose an $a \in (1, 2...204)$. Let the chosen value be $a = 32$. We firstly observe that gcd(32,205)=1, which makes us go to step 3 of the algorithm. Noting that here $\frac{n-1}{2} = \frac{205-1}{2} = 102$, we now compute $x = a^{\frac{n-1}{2}} = 32^{102} \pmod{205}$ and find that $32^{102} \equiv 204 \equiv -1 \pmod{205}$, so that $x = -1$. Now, we go to step 4 of the algorithm and find $y = (\frac{a}{n})$ which in this case is $(\frac{32}{205}) = -1$, as can be easily verified by writing a suitable computer program. So in this case, $x = y$, i.e., $a^{\frac{n-1}{2}} \equiv (\frac{a}{n}) \pmod{n}$, implying that 205 is either a prime or an Euler pseudoprime to base 32. To verify further, we check for $a = 73$. Going through all the above steps, we note that gcd(73,205)=1, $73^{102} \equiv 204 \equiv -1 \pmod{205}$, and that $(\frac{73}{205}) = -1$, which, again, is easily verifiable and so in this case also $a^{\frac{n-1}{2}} \equiv (\frac{a}{n}) \pmod{n}$. So 205 is indeed a prime or else an Euler pseudoprime to both the bases 32 and 73. However, if we choose $a = 84$, we note that gcd(84,205)=1 and that $84^{102} \equiv 86 \not\equiv \pm 1 \pmod{205}$, so that for $a = 84$, the above algorithm outputs composite in step 3 itself. So, 205 is composite and 32 and 73 were indeed Euler liars for it. As in the case of Fermat and strong liars, in case of Euler liars also, we can easily show that if a is an Euler liar for n, so is $(n-a)$. So, having seen that 32 and 73 are Euler liars for 205, we can say that 173 and 132 also satisfy this property. So, given any input number n, if we find n to be a probable prime to base a in the Solovay-Strassen test, we can basically leave out the check with $(n-a)$ as a base.

We now proceed to discuss about the accuracy and the time-complexity of the Solovay-Strassen algorithm in the next section.

10 Accuracy and Complexity of the Solovay-Strassen Algorithm

In the last section, we established that if n is an odd composite number, at least half the numbers x with gcd(x, n)=1 in the set (1,2...$n-1$) are Euler witnesses for n. So, for any a, the probability of failure, i.e, a composite number being incorrectly labelled as prime, is bounded above by $\frac{1}{2}$. So, if we repeat the test with k bases, i.e., with k values of a, the accuracy is expected to be very high and the probability of failure will be $\frac{1}{2^k}$ at most, idendical to the failure probability in Fermat's test but twice that of the Miller-Rabin algorithm. So, in general, the

Miller-Rabin algorithm has a greater accuracy than the Solovay-Strassen test. It has been shown that for an arbitrary integer n, the number of Fermat liars is lesser than the number of Euler liars which in turn is lesser than the number of strong liars. So, accuracy-wise, we may rank the three tests we have described as Miller-Rabin, followed by Solovay-Strassen and then by Fermat.
We can easily analyze the complexity of the Solovay-Strassen test from the algorithm we have discussed above. Note that the steps 1 and 5 of the algorithm merely invoves the checking of congruences and that the main steps involved are the steps 2,3 and 4. We know that the gcd of two numbers can be calculated using the Euclidean algorithm having a complexity of $O(\log^2 n)$ and if we look at the algorithm we proposed for calculating the value of the Jacobi symbol, it's not difficult to realize that this too has the same complexity as the Euclidean or the extended Euclidean algorithm, involving as it does a recursive process to get the answer. So, it can be said that both steps 2 and 4 of the algorithm involves $O(\log^2 n)$ time-complexity. However, as we saw earlier, step 3, involving the raising of a to power $\frac{n-1}{2}$, i.e., to a power $O(n)$ modulo n, can be accomplished using the method of modular exponentiation by repeated squaring and essentially involves $O(\log\ n)$ multiplications of two $O(\log\ n)$-bit numbers and hence requires $O(\log^3 n)$ time. So for each value of a we test, we must spend $O(\log^3 n)$ time and testing out k bases involves a time-complexity of $O(k\log^3 n)$. Since this complexity in step 3 predominates over those of the other steps, it essentially puts the overall time-complexity of the algorithm at $O(k\log^3 n)$.

Thus we see that complexity-wise, the Solovay-Strassen algorithm does almost as well as the Miller-Rabin; however, the latter, having a lower failure probability, has largely superceded the former as far as fast probabilistic primality testing is considered. The Solovay-Strassen algorithm, however, is historically important for first having shown the practical utility of the RSA cryptosystem. Both these tests are, however, probabilistic since they involve the choice of random bases, i.e., values of the parameter a. Recently, Adleman, Pomerance and Rumely have proposed an algorithm that is deterministic but runs in $(\log\ n)^{O(loglog\log n)}$ time, as was shown by Cohen and Lenstra. However, none of these tests achieves the quality of being both deterministic and polynomial in time-complexity at the same time. We now present an algorithm that simultaneously satisfies both properties, viz. the AKS algorithm.

11 The AKS Algorithm – Some Introductory Ideas

The basic congruence on which the deterministic AKS algorithm is based is basically a generalization of Fermat's little theorem to polynomials with coefficients in $\mathbf{Z}/n\mathbf{Z}$, n being any prime number, and it's this result that we state and prove now.

Theorem : Let n be any natural number. Then the congruence $(x + a)^n \equiv$

$(x^n + a) \pmod n$ holds if and only if n is a prime. Here x is an arbitrary variable and a satisfies gcd(a, n)=1.

Proof : Firstly, we show that n is a prime if and only if $n|\binom{n}{k} \forall k < n$. To see this, note first that $\binom{n}{k} = \frac{n(n-1)(n-2)...(n-k+1)}{k(k-1)(k-2)...2.1}$ and that n clearly divides the numerator of this expression. If n is prime, it cannot have any common factor with any term in the denominator as all the factors there are less than n. So, if n is a prime, then it divides the numerator of the expression but is coprime to the denominator, which forces that $n|\binom{n}{k} \forall k \in (1, 2...n-1)$. Conversely, if n is composite, assume p to be a prime factor of n with $n = pk$. Then $\binom{n}{p} = \frac{n(n-1)(n-2)...(n-p+1)}{p(p-1)(p-1)...1} = \frac{k(n-1)(n-2)...(n-p+1)}{(p-1)(p-2)...1}$ as $n = pk$. If $n|\binom{n}{p}$, it must divide the numerator, i.e., $n = pk$ must divide $k(n-1)(n-2)...(n-p+1) \Longrightarrow p|(n-1)(n-2)...(n-p+1)$. But $p|n$ and so cannot divide any of these factors and, being a prime, cannot divide their product as well. So, if n is composite, $n \not| \binom{n}{p}$ for any prime factor p of n. So we have the result that n is prime if and only if $n|\binom{n}{k} \forall k < n$.

Now, consider $(x + a)^n - (x^n + a) = \sum_{i=1}^{n-1} \binom{n}{i} a^{n-i} x^i$. If n is prime, we just saw that $n|\binom{n}{i} \forall i < n$ and hence, $n|(x + a)^n - (x^n + a)$ so that $(x + a)^n \equiv (x^n + a) \pmod n$ holds. Conversely, if n is composite and p a prime factor of it, as above, $n \not| \binom{n}{p}$ and since gcd$(a, n) = 1 \Longrightarrow$gcd$(a, p) = 1$, we have that n is also coprime to a^{n-p}. So the coefficient of x^p in $(x + a)^n - (x^n + a) = \binom{n}{p} a^{n-p} \not\equiv 0 \pmod n$, implying that $(x+a)^n \not\equiv (x^n + a) \pmod n$. This proves our theorem.

The above theorem may itself provide us with a way of testing the primality of n, but to test the congruence $(x + a)^n \equiv (x^n + a) \pmod n$ would require us to compare n coefficients and so the algorithm will be O(n), i.e., exponential in complexity, which is not what we want. To get around this difficulty, the AKS algorithm compares the coefficients modulo another polynomial of the form $x^r - 1$, for some chosen value of r. Even then, we cannot test the congruence for all values of $a \in (1, 2...n)$ if we want to have a polynomial complexity. The AKS algorithm, however, says that if we test the congruence for all $a \in (1, 2...s)$ for some s determined by the value of r, we can pronounce a number to be prime or composite with 100% accuracy. The algorithm also says that the value of r is bound by some polynomial in log n, implying that we do have a deterministic polynomial-time algorithm for primality testing. Before we speak about the choice of this r and s in the next section, we first give a definition.

Definition : Given any integer r which satisfies gcd(r, n)=1, the order of n modulo r, denoted by $O_r(n)$ is the smallest positive integer k for which the congruence $n^k \equiv 1 \pmod r$ holds.

We know fom Euler's theorem that if gcd$(r, n) = 1$, $n^{\phi(r)} \equiv 1 \pmod r$, where ϕ is the Euler's totient function. Let $\phi(r) = qk + s$, q,$s \in \mathbf{Z}$, by

the division algorithm, where $0 \leq s < k$ is the remainder. Then $n^{qk+s} \equiv 1 \pmod{r} \Longrightarrow (n^k)^q.n^s \equiv 1 \pmod{r} \Longrightarrow n^s \equiv 1 \pmod{r}$ as $n^k \equiv 1 \pmod{r}$, k being $O_r(n)$. But, by definition, k is the smallest natural number for which the congruence $n^k \equiv 1 \pmod{r}$ holds. So, $n^s \equiv 1 \pmod{r}$, together with $0 \leq s < k$, implies that $s = 0$. So, $\phi(r) = qk$ for some $q \in \mathbf{Z}$, implying that $O_r(n)|\phi(r)$.

12 The Pseudocode of the AKS Algorithm

We give below the pseudocode of the AKS algorithm. Here, log denotes logarithm to base 2.

1. Input $n \in \mathbf{N}$,$n \neq 1$.
2. Check if $n = a^b$ with $a \in \mathbf{N}$ and $b > 1$. If yes, output COMPOSITE.
3. Find smallest r satisfying $O_r(n) > \log^2 n$.
4. If $\gcd(a, n) > 1$ for some $a \leq r$, output COMPOSITE.
5. If $n \leq r$, output PRIME.
6. For a in range $(1, \lfloor \sqrt{\phi(r)} \log n \rfloor)$, if $(x+a)^n \not\equiv (x^n + a) \pmod{x^r - 1, n}$, output COMPOSITE.
7. Else output PRIME.

Note that this algorithm can output composite only in steps 2,4 and 6. If n is prime, it can neither be a perfect power, nor can it satisfy $\gcd(a, n) \neq 1$ for any $a \leq r$ since $r < n \Longrightarrow a < n$ in this case, implying that steps 2 and 4 never return composite for a prime n. Again, by the result in the last section, all primes n do satisfy $(x+a)^n \equiv (x^n + a) \pmod{x^r - 1, n}$, so that step 6 returns composite if and only if n is coposite. So, we conclude that the AKS algorithm can output composite only if the input is indeed so. But, is the converse true? In other words, if the algorithm outputs prime, does it nessesarily mean that the input n is prime? Yes, it does; the proof of this fact can be found in the original paper "Primes is in P" by AKS. We do not give that proof here, but we note that the above algorithm can output prime in steps 5 and 7 only, and its outputting prime in step 5 only becomes important when $n \leq 5690034$ because of the lemma we state below.

Lemma : The value of r found in step 3 of the algorithm satisfies $r \leq \lceil \log^5 n \rceil$ for $n \geq 3$ and for $n = 2$, $r \leq 3$.

So, step 5, which returns prime if $n \leq r$, only becomes important if $n \leq r \leq \lceil \log^5 n \rceil$ and 5690034 is the greatest number satisfying this inequality. Since we already know all the primes below this, we need not particularly worry about this step. But for the proof of the fact that step 7 of the algorithm returns prime iff the input is prime, we can always refer to the paper by AKS.

13 The Complexity of the AKS Algorithm - A Discussion

In this section, we examine the complexity of each and every step of the AKS algorithm and deduce its overall time-complexity.

In the third step of the AKS algorithm, we need to find the smallest r satisfying $O_r(n) > \log^2 n$. The algorithm for doing so can be presented in pseudocode as follows.

1. Input n.
2. For r in range $(1, \lceil \log^5 n \rceil)$, starting with $r = 1$, and k in range $(1, \lceil \log^2 n \rceil)$ check if $n^k \not\equiv 1 \pmod{r}$ for all k in this range. If yes, output r.
3. Else, $r = r + 1$.

Note that in this algorithm to calculate the value of r in the AKS test, step 2 requires us to check the congruence $n^k \not\equiv 1 \pmod{r}$ for $O(\log^2 n)$ values of the parameter j, i.e., for $O(\log^2 n)$ values of j as $k = \lceil \log^2 n \rceil$. But, looking at step 2 of this pseudocode, it's evident that we must check in this way for $\lceil \log^5 n \rceil$ values of r in the worst case, because the lemma in the last section assures us that we will always find an $r \leq \lceil \log^5 n \rceil$ satisfying $O_r(n) > \log^2 n$. So, we have to spend $O(\log^2 n)$ time for each of the $O(\log^5 n)$ values of r we have to test out in the worst case. This puts the time-complexity for finding the r of the AKS test, i.e., the time-complexity of step 2 of the algorithm at $O(\log^2 n \times \log^5 n) = O(\log^7 n)$.

The fourth step of the AKS algorithm requires the computation of $\gcd(a, n)$ for all $a \leq r$. Since $r \leq \lceil \log^5 n \rceil$, this essentially involves gcd computations for $O(\log^5 n)$ values of a. But, computation of $\gcd(a, n)$ for each value of a by the Euclidean algorithm takes $O(\log n)$ time, as can be easily shown, and hence, the complexity of this step of the algorithm comes down to $O(\log n \times \log^5 n) = O(\log^6 n)$. The inequality in step 5 of the AKS algorithm can be trivially tested in $O(\log n)$ time.

14 The Future of Primality Testing

We saw that the AKS test has indeed provided us with an unconditional, general, deterministic and polynomial-time algorithm for primality-testing. However, despite being polynomial in complexity and deterministic, other probabilistic tests we have discussed before have a lower time-complexity and can provide us with larger primes neede for, say, cryptography with high enough accuracy. After AKS published their "Primes is in P" paper, several efforts were made to reduce the complexity of the algorithm and several conjectures, if found to be true, can significantly aid our purpose. The Sophie-Germain prime density conjecture, which states that the number of primes $p \leq m$ with $2p + 1$ also a prime is asymptotically equal to $\frac{2Cm}{ln^2 m}$ where $C \approx 0.66$ is a constant, was widely

believed to be able to bring down the AKS complexity to $O(\log^6 n)$. However, a 2006 paper by Pomerance and Lenstra has given us an unconditional variant of the AKS algorithm which runs in $O(\log^6 n)$ time. But, probabilistic tests such as the Miller-rabin and the Solovay-Strassen, having a complexity of $O(\log^3 n)$ work better still and further improvements of the AKS algorithm, ideally achieving a complexity of $O(\log^3 n)$ unconditionally and deterministically remains a work or the future.

References

[1] M. Agarwal, P. Saxena Primes in P. International Journal of Maths, pp. 200–210, 2009.