Primes and How to Recognize them

Primality Testing Algorithms

Satwant Rana[1]

Advised by, Amitabha Tripathi[2]

[1] 2012 MT 50618
Mathematics Department
IIT Delhi

[2] Professor
Mathematics Department
IIT Delhi

MTP Presentation, 2016-17

Motivation

Prime numbers are central to Number Theory, acting as the atomic units around which all numbers are built. Therefore it is barely a surprise that *Primality Testing* is a problem with a rich history in Number Theory.

The motivation behind this project is to rediscover and implement the greatest and the latest in primality testing algorithms.

Primes and Composites

Primes are defined as natural numbers which are only divisible by 1 and themselves. Formally, given $p \in \mathbb{N}$ is a prime, if whenever $q \mid p$, then $q \in \{1, p\}$.

Any natural greater than 1 which is not a prime is called a *composite*.

A Naive Primality Test

Algorithm 1 Naive Primality Test

```
procedure NaivePrimalityTest(n)
    d ← 2
    while d ≤ n − 1 do
        r ← n mod d
        if r = 0 then
            return false                    ▷ n is composite
        d ← d + 1
    return true                             ▷ n is prime
```

Time Complexity - $O(n)$

An Optimization

If $n, a, b \in \mathbb{Z}$ such that $n = ab$, then $min(a, b) \leq \sqrt(n)$.

Algorithm 2 Optimized Naive Primality Test

```
procedure OptimizedNaivePrimalityTest(n)
    d ← 2
    while d ≤ min(n − 1, √n)) do
        r ← n  mod d
        if r = 0 then
            return false                          ▷ n is composite
        d ← d + 1
    return true                                   ▷ n is prime
```

Time Complexity - $O(\sqrt{n})$

Compositeness Tests

A successful *primality test* proves that a given number is prime, whereas a successful *compositeness test* proves that a given number is composite.

e.g. If $n > 2$ and $2 \mid n$, then n is composite.

If a compositeness test is not successful, then we can't comment on the primality of the given number.

Composite numbers which the compositeness test labels as primes are called the *pseudoprimes* for the test.

Fermat's (Little) Theorem

Theorem (Fermat's Theorem)

Given prime p, and $a \in \mathbb{Z}$, $(a, p) = 1$ we have,

$$a^{p-1} \equiv 1 \mod p$$

Corollary (1)

Given prime p, and $a \in \mathbb{Z}$ we have,

$$a^p \equiv a \mod p$$

Fermat's Theorem as a Compositeness Test

The following Corollary 2 is a simple compositeness test using *Fermat's Theorem*.

Corollary (2)

If $n \in \mathbb{N}$, $n \geq 2$ and $\exists a \in \mathbb{Z}$ such that,

$$a^n \not\equiv a \mod n$$

then n is not a prime.

For instance, for $n = 9$, $2^9 \equiv 8 \not\equiv 2 \mod 9$, indicating the compositeness of 9.

Fermat Pseudoprimes

There do exist combinations of a and composite n which satisfy the *Fermat's Theorem*.

For instance $n = 341 = 11.31$ gives $2^{341} \equiv 2 \mod 341$. This makes 341 a pseudoprime to the Fermat's Compositeness Test, or a *Fermat Pseudoprime*.

Although, in this case a change of base a from 2 to 3 yields $3^{341} \equiv 168 \not\equiv 3 \mod 341$ which indicates that 341 is not a prime.

Carmichael Numbers

Given $n \in \mathbb{Z}$ is a *Carmichael Number*, if $a^{n-1} \equiv 1 \mod n$, $\forall a \in \mathbb{Z}, (a, n) = 1$.

The smallest example of *Carmichael Numbers* is 561, and there exist infinitely many of them.

Carmichael Numbers are *Fermat Pseudoprimes* for each base a comprime to n.

Eucledian Algorithm for G.C.D.

For $a, b \in \mathbb{Z}$, we have $(a, 0) = a$, $(a, b) = (a, b - a)$ and therefore $(a, b) = (a, b \mod a)$.

Algorithm 3 Euclidean Algorithm

---

procedure $\textsc{EuclideanAlgorithm}(a, b)$
    $a \leftarrow \textsc{Abs}(a)$
    $b \leftarrow \textsc{Abs}(b)$ ▷ Eliminating negative signs
    if $a > b$ **then**
        $\textsc{Swap}(a, b)$
    while $a \neq 0$ **do**
        $c \leftarrow b \mod a$
        $b \leftarrow a$
        $a \leftarrow c$
    return b

Time Complexity - $O(\log \min(|a|, |b|))$

Logarithmic Exponentiation

$$a^n = \begin{cases} 1 & n = 0 \\ (a^{\frac{n}{2}})^2 & n \equiv 0 \mod 2 \\ a(a^{\frac{n-1}{2}})^2 & n \equiv 1 \mod 2 \end{cases}$$

Algorithm 4 Recursive Logarithmic Exponentiation

procedure $\textsc{LogarithmicExponentiation}(a, n, m)$
    $result \leftarrow 1 \mod m$ ▷ Calculates $a^n \mod m$
    if $n > 0$ & $n \equiv 0 \mod 2$ **then**
        $result \leftarrow \textsc{LogarithmicExponentiation}(a, \frac{n}{2}, m)$
        $result \leftarrow result * result \mod m$
    else if $n > 0$ & $n \equiv 1 \mod 2$ **then**
        $result \leftarrow \textsc{LogarithmicExponentiation}(a, \frac{n-1}{2}, m)$
        $result \leftarrow result * result \mod m$
        $result \leftarrow result * a \mod m$
    return $result$

Logarithmic Exponentiation

If $n = b_{d-1}b_{d-2}\dots b_0 = \sum_{i=0}^{d-1} b_i 2^i$, then

$$a^n = a^{\sum_{i=0}^{d-1} b_i 2^i} = \prod_{i=0}^{d-1} a^{b_i 2^i}$$

Algorithm 5 Iterative Logarithmic Exponentiation

procedure LogarithmicExponentiation(a, n, m)
  $result \leftarrow 1 \mod m$ ▷ Calculates $a^n \mod m$
  $b \leftarrow a$
  while $n > 0$ **do**
    if $n \mod 2 = 1$ **then** ▷ If rightmost bit is 1
      $result \leftarrow result * b \mod m$ ▷ Multiply by b
    $b \leftarrow b * b \mod m$ ▷ b stores a^{2^i} on i^{th} step
    $n \leftarrow \frac{n}{2}$ ▷ Remove rightmost bit
  return result

Time Complexity - $O(\log n)$

Fermat's Compositeness Test

Using current discussion, *Fermat's Compositeness Test* can be implented as

Algorithm 6 Fermat's Compositeness Test

procedure $\textsc{FermatCompositenessTest}(a, n)$
    $gcd \leftarrow \textsc{EucledianAlgorithm}(a, n)$.
    if $gcd > 1$ & $gcd < n$ **then**
        return *false*
    $left \leftarrow \textsc{LogarithmicExponentiation}(a, n, n)$
    $right \leftarrow a \mod m$
    return $left \neq right$

Fermat's Probabilistic Primality Test

Every failed run of a compositness test reduces the probability of compositeness, and increases the probability of primality. So we have a *Probabilistic Primality Test*,

Algorithm 7 Fermat's Probabilistic Primality Test

```
procedure FermatProbabilisticPrimalityTest(n, iter)
    while iter > 0 do                              ▷ iter is number of iterations
        a ← Random(0, n − 1)                       ▷ Random number in [0, n − 1]
        check ← FermatCompositenessTest(a, n)
        if check then
            return false                           ▷ Composite found
        iter ← iter − 1
    return true                                    ▷ Probable prime found
```

Time Complexity - $O(\log n)$

Fermat's Primality Test

If we have a table of *pseudoprimes* then a simple check removes the flaw from *Fermat's Compositeness Test*.

D.H. Lehmer prepared a table of all Fermat pseudoprimes below 2.10^8 for the base 2 with no factor < 317. Thus a primality test to check primality for $n < 2.10^8$ can be formulated.

Fermat's Primality Test

Algorithm 8 Fermat's Primality Test

procedure $\textsc{FermatPrimalityTest}(n)$
  if $n \geq 2.10^8$ **then**
    return *false* ▷ Fail if out of range
  for $i = 2$, $i \leq \min(313, n-1)$, $i \leftarrow i+1$ **do**
    if $i \mid n$ **then**
      return *false* ▷ Factor ≤ 313
  if $\textsc{IterativeLogarithmicExponentiation}(2, p-1, p) \not\equiv 1 \mod 2$ **then**
    return *false* ▷ Composite by Fermat's Theorem
  return $!\textsc{IsLehmerPseudoprime}(n)$ ▷ Check Lehmer's Table

A Generalization of Fermat's Theorem

The reason why Fermat's Theorem can only be a used to create a Compositeness Test, is that it's only a necessary condition on primality. Here's a (generalized) necessary and sufficient generalization of Fermat's Little Theorem.

Theorem

Given $n \in \mathbb{N}$, $n \geq 2$ *and* $a \in \mathbb{Z}$, $(a, n) = 1$, *then* n *is prime if and only if*

$$(X + a)^n \equiv X^n + a \mod n$$

A Generalization of Fermat's Theorem

A simple primality test - Choose an apt a, and then test the congruence $(X + a)^n \equiv X^n + a \mod n$. But there are $O(n)$ terms in the polynomial, making our test $\Omega(n)$.

One way to reduce the number of terms in the polynomial is to consider the congruences modulo $X^r - 1$ additionally, for a small r.

Build-up to a Polynomial Time Primality Test

To make things concrete, we consider the congruence,

$$(X + a)^n \equiv X^n + a \mod (n, X^r - 1)$$

and build a primality test around it.

The above congruence follows for prime n as before, but depending upon how small r we chose, some composite values of n may also satisfy the congruence now.

AKS Primality Test uses a polynomial r, and a polynomial number of values of a to deduce whether n is a prime or not.

AKS Primality Test

Algorithm 9 AKS Primality Test

procedure $\text{AKSPRIMALITYTEST}(n)$
    if $n < 2$ or $n = a^b$ for $a, b \in \mathbb{N}$ and $b \geq 2$ **then**
        return *false* ▷ Step 1
    $r \leftarrow min\{i : i \in \mathbb{N},\ i \leq \max(3,\ \lceil \log^5 n \rceil),\ o_i(n) > \log^2 n\}$ ▷ Step 2
    for $a = 2,\ i \leq r,\ a \leftarrow a + 1$ **do**
        if $1 < (a, n) < n$ **then**
            return *false* ▷ Step 3
    if $n \leq r$ **then return** *true* ▷ Step 4
    for $a = 1,\ a \leq \lfloor \sqrt{\phi(r)} \log n \rfloor,\ a \leftarrow a + 1$ **do**
        if $(X + a)^n \not\equiv X^n + a \mod (n, X^r - 1)$ **then**
            return *false* ▷ Step 5
    return *true* ▷ Step 6

Time complexity - $\tilde{O}(r^{\frac{3}{2}} log^3 n) = \tilde{O}(log^{10.5} n)$

AKS Primality Test - Prime Input

Lemma

If n is a prime then Algorithm 9 returns true.

Proof.

If n is a prime, then Steps 1 and 3 can never return *false*. Also by previous discussion, Step 5 can't return *false*. So the algorithm returns *true* in either Step 4 or Step 6. $\square$

AKS Primality Test - Is n a perfect power?

Algorithm 10 Perfect Power Test

```
procedure PerfectPowerTest(n)
    if n = 1 then
        return true
    for b ← 2, b ≤ log n, b ← b + 1 do
        l ← 1, u ← n
        while l < u do
            m ← ⌊(l+u)/2⌋
            x ← m^b            ▷ Use Logarithmic Exponentiation here
            if x = n then  return true
            else if x < n then  l ← m + 1
            else r ← m − 1
    return false
```

Time Complexity - $O(\log^4 n)$

AKS Primality Test - Existence of a small r

Lemma

Let $LCM(m)$ be lcm of first m numbers. We have for $m \geq 7$,

$$LCM(m) \geq 2^m$$

Lemma

There exists an $r \leq \max(3, \lceil \log^5 n \rceil)$ and $o_r(n) > \log^2 n$.

Consider the smallest $r \in \mathbb{N}$ which doesn't divide the product P defined as,

$$P = n^{\log B} . \prod_{i=1}^{\lfloor \log^2 n \rfloor} n^i - 1$$

AKS Primality Test - Some definitions

Remaining case is of composite n. Let $p|n$, such that $o_r(p) > 1$.
So $(p, r) = (n, r) = 1$.

$$\forall\ 0 \leq a \leq l$$

$$(X+a)^n \equiv X^n + a \mod (n, X^r - 1)$$

$$(X+a)^p \equiv X^p + a \mod (p, X^r - 1)$$

$$\implies ((X+a)^p)^{\frac{n}{p}} \equiv (X^p)^{\frac{n}{p}} + a \mod (p, X^r - 1)$$

$$\implies (X^p+a)^{\frac{n}{p}} \equiv (X^p)^{\frac{n}{p}} + a \mod (p, X^r - 1)$$

$$\implies (X+a)^{\frac{n}{p}} \equiv X^{\frac{n}{p}} + a \mod (p, X^r - 1)$$

AKS Primality Test - Some definitions

Definition

For polynomial $f(X)$ and $m \in \mathbb{N}$, if m satisfies

$$f(X)^m \equiv f(X^m) \mod (p, X^r - 1)$$

then m is said to be introspective for $f(X)$.

Lemma

If m and m' are both introspective for a polynomial $f(X)$, then so is $m.m'$.

Lemma

If m is introspective for both the polynomials $f(X)$ and $g(X)$, then it is also introspective for $f(X).g(X)$.

AKS Primality Test - About two groups

Every number in the set $I = \{\frac{n}{p}^i.p^j \mid i,j \in \mathbb{N}_0\}$ is introspective for every polynomial in the set $P = \{\prod_{a=0}^{l}(X+a)^{e_a} \mid e_a \in \mathbb{N}_0\}$.

The first group is the set of residues of all elements of I modulo r. Let's denote this set as G, and it is a subset of $\mathbb{Z}_r^*$ as $(n,r) = (p,r) = 1$. Let $t = |G|$. Since $n^i \in G \; \forall \; i \in \mathbb{N}_0$ and $o_r(n) > \log^2(n)$, we have $t > \log^2(n)$.

Let $Q_r(X)$ be r^{th} cyclotomic polynomial over F_p. $Q_r(X)$ divides $X^r - 1$ and factors into irreducible factors of degree $o_r(p)$. Let $h(X)$ be one such irreducible factor. Since $o_r(p) > 1$, degree of $h(X)$ is greater than one. The second group is the set of all residues of polynomials in P modulo $h(X)$ and p. Let $\mathcal{G}$ be this group. This group is generated by elements $X, X+1, X+2, \ldots, X+l$ in the field $F = F_p[X]/(h(X))$ and is a subgroup of the multiplicative group of F.

AKS Primality Test - Estimating the size of $\mathcal{G}$

Lemma (Lenstra)

$|\mathcal{G}| \geq \binom{t+l}{t-1}$

Lemma

If n is not a power of p, then $|\mathcal{G}| \leq n^{\sqrt{t}}$

AKS Primality Test - Completing the proof

Theorem

If Algorithm 9 returns true then input n is prime.

$$|\mathcal{G}| \geq \binom{t+l}{t-1}$$

$$|\mathcal{G}| \geq \binom{\lfloor\sqrt{t}\log n\rfloor + 1 + l}{\lfloor\sqrt{t}\log n\rfloor}$$

$$|\mathcal{G}| \geq \binom{2\lfloor\sqrt{t}\log n\rfloor + 1}{\lfloor\sqrt{t}\log n\rfloor}$$

$$|\mathcal{G}| \geq 2^{\lfloor\sqrt{t}\log n\rfloor + 1} > 2^{\sqrt{t}\log n} = n^{\sqrt{t}}$$

So n is a power of p, which contradicts the structure of Algorithm 9.

Implementation and Simulations

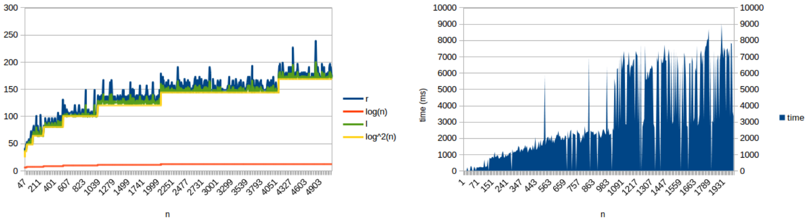

Figure: AKS simulations for various values of input n.

Implementation exists at
`https://github.com/satwantrana/primality-testing`.

A generalisation of AKS

Theorem

Given $n, r \in \mathbb{N}$ and $S \subseteq \mathbb{N}$ such that,

1. *n can be written as a product of elements from S.*
2. *S is not generated, under multiplication, by a proper subset of S.*
3. *$(n, r) = 1$ and $o_r(n) > \log^{1+\frac{1}{k}} n$.*
4. *$(X + a)^m \equiv X^m + a \mod (X^r - 1, n)$ $\forall$ $m \in S$ and $1 \leq a \leq l_k$.*

where $k = |S|$, $N = \prod_{m \in S} m$ and $l_k = \lfloor \phi(r) 2^{log_n N} \rfloor$; then n is power of a prime.

Further, such an r exists which is prime with $r \leq \log^{3+\frac{2}{k}} n$. Also, if $o_r(N) > log^{1+\frac{1}{k}} N$, then l_k gets improved to $\lfloor \phi(r)^{\frac{1}{k+1}} log N \rfloor$ with such an existing with $r \leq \log^{3+\frac{2}{k}} N$.

Time complexity - $\tilde{O}(l_k \cdot \log N \cdot r \cdot \log n)$

More introspective numbers

Lemma

Given distinct primes n, $r \in \mathbb{N}$, then $\forall f(X) \in \mathbb{Z}[X]$,

$$f(X)^{n^{\phi(r)}-1} \equiv 1 \mod (n, X^r - 1)$$

Corollary

If n and r are primes, then $\{n + \lambda(n^{\phi(r)} - 1) : (\lambda, n) = 1\}$ *is an infinite set of introspective numbers, all of which are coprime to n, for all polynomials* $f(X) \in \mathbb{Z}[X]$.

Conclusion

... and Thank You!