PRIMES is in P

Manindra Agrawal
Neeraj Kayal
Nitin Saxena
Dept of CSE, IIT Kanpur

The Problem

- Given number n, test if it is prime efficiently.

Efficiently = in time a polynomial in number of digits

= $(\log n)^c$ for some constant c

PRIMES = set of all prime numbers

The Trial Division Method

Try dividing by all numbers up to $n^{1/2}$.

- Already known since ~230 BC (Sieve of Eratosthenes)
- takes exponential time: $\Omega(n^{1/2})$.
- Also produces a factor of n when it is composite.

Fermat's Little Theorem

if n is prime then for any a:

$a^n = a \pmod{n}$.

- It is easy to check:
  - Compute a^2, square it to a^4, square it to a^8, …
  - Needs only $O(\log n)$ multiplications.

A Potential Test

- For a “few” a’s test if $a^n = a \pmod{n}$;
- if yes, output PRIME else output COMPOSITE.

  - This fails!
  - For n = 561 = 3 * 11 * 17, all a’s satisfy the equation!!

PRIMES in NP $\cap$ coNP

- A trivial algorithm shows that the problem is in coNP: *guess* a factor of n and verify it.
- In 1974, Vaughan Pratt designed an NP algorithm for testing primality.

PRIMES in P (conditionally)

- In 1973, Miller designed a test based on Fermat's Little Theorem:
  - It was efficient: $O(\log^4 n)$ steps
  - It was correct assuming Extended Riemann Hypothesis.

PRIMES in coRP

- Soon after, Rabin modified Miller's algorithm to obtain an unconditional but randomized polynomial time algorithm.
  - This algorithm might give a wrong answer with a small probability when n is composite.
- Solovay-Strassen gave another algorithm with similar properties.

PRIMES in P (almost)

- In 1983, Adleman, Pomerance, and Rumely gave a deterministic algorithm running in time $(\log n)^{c \log \log \log n}$.

PRIMES in RP

- In 1986, Goldwasser and Kilian gave a randomized algorithm that
  - works almost always in polynomial time
  - errs only on primes.
- In 1992, Adleman and Huang improved this to an algorithm that is always polynomial time.

Our Contribution

We provide the first deterministic and unconditional polynomial-time algorithm for primality testing.

Main Idea

- Generalize Fermat's Little Theorem:
  - Ring Z/nZ does not seem to have nice structure to exploit.
  - So extend the ring to a larger ring in the hope for more structure.
- Consider polynomials modulo n and $X^r - 1$, or the ring $Z/nZ[X]/(X^r-1)$.

Generalized FLT

If n is prime
then for any a:
$$(X + a)^n = X^n + a \ (\text{mod } n, X^r - 1).$$

- Potential test: for a "small" r and a "few" a's, test the above equation.

It Works (Almost)!

- We prove:

If

$(X + a)^n = X^n + a \pmod{n, X^r - 1}$

for every $0 < a < 2 \sqrt{r} \log n$

and for suitably chosen "small" r

then

either n is a prime power or has a prime divisor less than r

The Algorithm

- Input n.

1. Output COMPOSITE if $n = m^k$, $k > 1$.
2. Find the smallest number r such that $O_r(n) > 4 (\log n)^2$.

   $O_r(n)$ = order of n modulo r.
3. If any number < r divides n, output PRIME/COMPOSITE appropriately.
4. For every $a \leq 2 \sqrt{r} \log n$:
   - If $(X+a)^n \neq X^n + a \pmod{n, X^r - 1}$ then output COMPOSITE.
5. Output PRIME.

Correctness

- If the algorithm outputs COMPOSITE, n must be composite:
  - COMPOSITE in step 1 $\Rightarrow$ $n = m^k$, $k > 1$.
  - COMPOSITE in step 3 $\Rightarrow$ a number $< r$ divides n.
  - COMPOSITE in step 4 $\Rightarrow$ $(X+a)^n \neq X^n + a \pmod{n, X^r-1}$ for some a.
- If the algorithm outputs PRIME in step 3, n is a prime number $< r$.

When Algorithm Outputs PRIME in Step 5

- Then $(X+a)^n = X^n + a \pmod{n, X^r-1}$ for $0 < a \leq 2\sqrt{r} \log n$.
- Let prime $p \mid n$.
- Clearly, $(X+a)^n = X^n + a \pmod{p, X^r-1}$ too for $0 < a \leq 2\sqrt{r} \log n$.
- And of course, $(X+a)^p = X^p + a \pmod{p, X^r-1}$ (according to generalized FLT)

Introspective Numbers

- We call any number m such that $g(X)^m = g(X^m) \pmod{p, X^r-1}$ an introspective number for $g(X)$.

- So, 1, p and n are introspective numbers for $X+a$ for $0 < a \leq 2\sqrt{r}\log n$.

Introspective Numbers Are Closed Under *

Lemma: If s and t are introspective for $g(X)$, so is $s * t$.

Proof:

$$g(X)^{st} = g(X^s)^t \pmod{p, X^r - 1}, \text{ and}$$

$$g(X^s)^t = g(X^{st}) \pmod{p, X^{sr} - 1}$$

$$= g(X^{st}) \pmod{p, X^r - 1}.$$

So There Are Lots of Them!

- Let $I = \{ n^i * p^j \mid i, j \geq 0\}$.
- Every m in I is introspective for $X+a$ for $0 < a \leq 2 \sqrt{r} \log n$.

Introspective Numbers Are Also For Products

Lemma: If m is introspective for both $g(X)$ and $h(X)$, then it is also for $g(X) * h(X)$.

Proof:

$$(g(X) * h(X))^m = g(X)^m * h(X)^m$$
$$= g(X^m) * h(X^m) \pmod{p, X^r-1}$$

So Introspective Numbers Are For Lots of Products!

- Let $Q = \{ \prod_{a=1,\, 2\sqrt{r}\log n} (X + a)^{e_a} \mid e_a \geq 0\}$.
- Every m in I is introspective for every g(X) in Q!
- So there are lots of introspective numbers for lots of polynomials.

Low Degree Polynomials in Q

- Let $t = O_r(n,p)$.
- Let Q_{low} be set of all polynomials in Q of degree < t.
- There are > $n^{2\sqrt{t}}$ distinct polynomials in Q_{low}:
  - Consider all products of X+a's of degee < t.
  - There are $\binom{t-1+2\sqrt{r}\log n}{2\sqrt{r}\log n-1}$ > $n^{2\sqrt{t}}$ of these (since $p \geq r$ and $\sqrt{t} > 2 \log n$).

Finite Fields Facts

- Let $h(X)$ be an irreducible divisor of r^{th} cyclotomic polynomial $C_r(X)$ in the ring $F_p[X]$:
  - $C_r(X)$ divides X^r-1.
  - Polynomials modulo p and $h(X)$ form a field, say F.
  - $X^i \neq X^j$ in F for $0 \leq i \neq j < r$.

Moving to Field F

- Since $h(X)$ divides X^r-1, equations for introspective numbers continue to hold in F.
- $|| \{X^m \mid m \in I\} || = t$ since $O_r(n,p) = t$.

- We now argue over F.

Q_{low} injects into F

- Let $f(X)$, $g(X)$ in Q_{low}, $f(X) \neq g(X)$.
- If $f(X) = g(X)$ in the field F then
  - For every m in I, $f(X^m) = f(X)^m = g(X)^m = g(X^m)$ in F.
  - So polynomial $P(Y) = f(Y) - g(Y)$ has t roots in F.
  - Contradiction since degree of $P(Y)$ is $< t$.

Completing the Proof

- There must be a, b, c, d ≤ √t such that:

  $$(a,b) \neq (c,d) \text{ and}$$

  $$n^a * p^b \ (= s) = n^c * p^d \ (= s') \ (\text{mod } r)$$

  – Since $O_r(n,p) = t$.
- Let $g(X)$ be any polynomial in Q_{low}.
- Then modulo (p, X^r-1):

  $g(X)^s = g(X^s)$ [since s is introspective]

  $= g(X^{s'})$ [since s = s' (mod r)]

  $= g(X)^{s'}$ [since s' is introspective]

Proof Contd.

- Therefore, $g(X)$ is a root of the polynomial $P(Y) = Y^s - Y^{s'}$ in the field F.
- Since Q_{low} has more than $n^{2\sqrt{t}}$ polynomials in F, $P(Y)$ has more than $n^{2\sqrt{t}}$ roots in F.
- However, $\max\{s,s'\} \leq n^{2\sqrt{t}}$.
- Therefore, $s = s'$ implying that $n = p^e$ for some e.

The Choice of r

- We need r such that $O_r(n) > 4 (\log n)^2$.
- Any r such that $O_r(n) \leq 4 (\log n)^2$ must divide

$$\prod_{k=1,\, 4 \log^2 n} (n^k - 1) < n^{16 \log^4 n} = 2^{16 \log^5 n}.$$

- LCM of first m numbers is at least 2^m (for $m > 7$).
- Therefore, there must exist an r that we desire $\leq 16 (\log n)^5 + 1$.

Remarks

- Our algorithm is impractical – its running time is $O^{\sim}(\log^{10.5} n)$ provably and $O^{\sim}(\log^{6} n)$ heuristically.
- To make it practical, one needs to bring the exponent down to 4 or less.
- As of now, best known running time is $O^{\sim}(\log^{6} n)$ [Lenstra & Pomerance].

Further Improvement?

- Conjecture: If $n \neq 1 \pmod r$ for some $r > \log\log n$ and $(X-1)^n = X^n - 1 \pmod{n, X^r - 1}$ then n must be a prime power.

- Yields a $O^{\sim}(\log^3 n)$ time algorithm.