Distinguishing prime numbers from composite numbers: the state of the art

D. J. Bernstein
University of Illinois at Chicago

Is it easy to determine whether a given integer is prime?

If “easy” means “computable”: Yes, of course.

If “easy” means “computable in polynomial time”: Yes. (2002 Agrawal/Kayal/Saxena)

If “easy” means “computable in essentially cubic time”: Conjecturally yes! See Williams talk tomorrow.

What about quadratic time?

What about linear time?

What if we want to determine *with proof* whether a given integer is prime?

Can results be verified faster than they're computed?

What if we want *proven* bounds on time?

Does randomness help?

Cost measure for this talk:
time on a serial computer.
Beyond scope of this talk:
use "AT" cost measure
to see communication, parallelism.

Helpful subroutines:
Can compute B-bit product,
quotient, gcd in time $\le B^{1+o(1)}$.
(1963 Toom; 1966 Cook;
1971 Knuth)

Beyond scope of this talk:
time analyses more precise
than "$\le B^{\text{constant}+o(1)}$."

Compositeness proofs

If n is prime and $w \in \mathbf{Z}$
then $w^n - w \in n\mathbf{Z}$
so n is "w-sprp":
the easy difference-of-squares
factorization of $w^n - w$,
depending on $\text{ord}_2(n-1)$,
has at least one factor in $n\mathbf{Z}$.

e.g.: If $n \in 5 + 8\mathbf{Z}$ is prime
and $w \in \mathbf{Z}$ then $w \in n\mathbf{Z}$ or
$w^{(n-1)/2} + 1 \in n\mathbf{Z}$ or
$w^{(n-1)/4} + 1 \in n\mathbf{Z}$ or
$w^{(n-1)/4} - 1 \in n\mathbf{Z}$.

Given $n \geq 2$: Try random w.
If n is not w-sprp, have proven n composite. Otherwise keep trying.

Given composite n,
this algorithm eventually finds compositeness certificate w.
Each w has $\geq 75\%$ chance.

Random time $\leq B^{2+o(1)}$
to find certificate if $n < 2^B$.
Deterministic time $\leq B^{2+o(1)}$
to verify certificate.

Open: Is there a compositeness certificate findable in time $B^{O(1)}$,
verifiable in time $\leq B^{1+o(1)}$?

Given prime n,
this algorithm loops forever.
After many w's we are
confident that n is prime . . .
but we don't have a proof.

Challenge to number theorists:
Prove n prime!

Side issue: Do users care?

Paranoid bankers: "Yes,
we demand primality proofs."

Competent cryptographers: "No,
but we have other uses for
the underlying tools."

Combinatorial primality proofs

If there are many elements of a particular subgroup of a prime cyclotomic extension of $\mathbf{Z}/n$ then n is a power of a prime. (2002 Agrawal/Kayal/Saxena)

Many primes r have prime divisors of $r-1$ above $r^{2/3}$ (1985 Fouvry). Deduce that AKS algorithm takes time $\le B^{12+o(1)}$ to prove primality of n.

Algorithm is *conjectured* to take time $\le B^{6+o(1)}$.

Variant using *arbitrary* cyclotomic extensions takes time $\leq B^{8+o(1)}$. (2002 Lenstra)

Variant with better bound on group structure takes time $\leq B^{7.5+o(1)}$. (2002 Macaj; same idea without credit in 2003 revision of AKS paper)

These variants are conjectured to take time $\leq B^{6+o(1)}$.

Variant using Gaussian periods is *proven* to take time $\leq B^{6+o(1)}$. (2004 Lenstra/Pomerance)

What if n is composite?
Output of these algorithms is a compositeness proof.

Time $\le B^{4+o(1)}$ to verify proof.
Time $\le B^{6+o(1)}$ to find proof.

For comparison, traditional sprp compositeness proofs:
verify proof, $\le B^{2+o(1)}$;
find proof, random $\le B^{2+o(1)}$.

For comparison, factorization:
verify proof, $\le B^{1+o(1)}$;
find proof, conjectured
$\le B^{(1.901\ldots+o(1))(B/\lg B)^{1/3}}$.

Benefit from randomness?

Use random Kummer extensions; twist. (2003.01 Bernstein, and independently 2003.03 Mihăilescu/Avanzi; 2-power-degree case: 2002.12 Berrizbeitia; prime-degree case: 2003.01 Cheng)

Many divisors of $n^{\cdots} - 1$ (overkill: 1983 Odlyzko/Pomerance). Deduce: time $\le B^{4+o(1)}$ to verify primality certificate.

Random time $\le B^{2+o(1)}$ to find certificate.

Open: Primality proof with proven deterministic time $\leq B^{5+o(1)}$ to find, verify?

Open: Primality proof with proven random time $\leq B^{3+o(1)}$ to find, verify?

Open: Primality proof with reasonably *conjectured* time $\leq B^{3+o(1)}$ to find, verify?

Prime-order primality proofs

If $w^{n-1} = 1$ in $\mathbf{Z}/n$, and $n - 1$ has a prime divisor $q \geq \sqrt{n}$ with $w^{(n-1)/q} - 1$ in $(\mathbf{Z}/n)^*$, then n is prime. (1876 Lucas, 1914 Pocklington, 1927 Lehmer)

Many generalizations.

Can extend $\mathbf{Z}/n$. (1876 Lucas, 1930 Lehmer, 1975 Morrison, 1975 Selfridge/Wunderlich, 1975 Brillhart/Lehmer/Selfridge, 1976 Williams/Judd, 1983 Adleman/Pomerance/Rumely)

Can prove arbitrary primes.
Proofs are fast to verify
but often very slow to find.

Replace unit group by
random elliptic-curve group.
(1986 Goldwasser/Kilian;
point counting: 1985 Schoof)

Use complex-multiplication
curves; faster point counting.
(1988 Atkin; special cases:
1985 Bosma, 1986
Chudnovsky/Chudnovsky)

Merge square-root computations.
(1990 Shallit)

Culmination of these ideas
is “fast elliptic-curve
primality proving” (FastECPP):

Conjectured time $\le B^{4+o(1)}$
to find certificate
proving primality of n.

Proven deterministic time
$\le B^{3+o(1)}$ to verify certificate.

For comparison, combinatorics:
proven random $\le B^{2+o(1)}$ to find,
$\le B^{4+o(1)}$ to verify.

Variant using
genus-2 hyperelliptic curves:

Proven random time $B^{O(1)}$
to find certificate
proving primality of n.
(1992 Adleman/Huang)

Tools in proof: bounds on size
of Jacobian (1948 Weil); many
primes in interval of width $x^{3/4}$
around x (1979 Iwaniec/Jutila).

Proven deterministic time
$\le B^{3+o(1)}$ to verify certificate.

Variant using elliptic curves with large power-of-2 factors (1987 Pomerance):

Proven existence of certificate proving primality of n.
Proven deterministic time $\leq B^{2+o(1)}$ to verify certificate.

Open: Is there a primality certificate findable in time $B^{O(1)}$, verifiable in time $\leq B^{2+o(1)}$?

Open: Is there a primality certificate verifiable in time $\leq B^{1+o(1)}$?

Verifying elliptic-curve proofs

Main theorem in a nutshell:
If an elliptic curve
$E(\mathbf{Z}/n)$ has a point
of prime order $q > (\lceil n^{1/4} \rceil + 1)^2$
then n is prime.

Proof in a nutshell:
If p is a prime divisor of n
then the same point mod p
has order q in $E(\mathbf{F}_p)$,
but $\#E(\mathbf{F}_p) \le (\sqrt{p} + 1)^2$
(Hasse 1936), so $n^{1/2} < p$.

More concretely:

Given odd integer $n \geq 2$,
$a \in \{6, 10, 14, 18, \ldots$, integer c,
$\gcd\{n, c^3 + ac^2 + c\} = 1$,
$\gcd\{n, a^2 - 4\} = 1$,
prime $q > (\lceil n^{1/4} \rceil + 1)^2$:

Define $x_1 = c$, $z_1 = 1$,
$x_{2i} = (x_i^2 - z_i^2)^2$,
$z_{2i} = 4x_i z_i (x_i^2 + a x_i z_i + z_i^2)$,
$x_{2i+1} = 4(x_i x_{i+1} - z_i z_{i+1})^2$,
$z_{2i+1} = 4c(x_i z_{i+1} - z_i x_{i+1})^2$.

If $z_q \in n\mathbf{Z}$ then n is prime.

For each prime p dividing n:
$(a^2 - 4)(c^3 + ac^2 + c) \neq 0$ in $\mathbf{F}_p$,
so $(c^3 + ac^2 + c)y^2 = x^3 + ax^2 + x$
is an elliptic curve over $\mathbf{F}_p$;
$(c, 1)$ is a point on curve.

On curve: $i(c, 1) = (x_i/z_i, \ldots)$
generically. (1987 Montgomery)
Analyze exceptional cases, show
$q(c, 1) = \infty$. (2006 Bernstein)

Many previous ECPP variants.
Trickier recursions,
typically testing coprimality.

Finding elliptic-curve proofs

To prove primality of n: Choose random E. Compute $\#E(\mathbf{Z}/n)$ by Schoof's algorithm.

Compute $q = \#E(\mathbf{Z}/n)/2$. If q doesn't seem prime, try new E.

If $q \geq n$ or $q \leq (\lceil n^{1/4} \rceil + 1)^2$: n is small; easy base case.

Otherwise:
Recursively prove primality of q.
Choose random point P on E.
If $2P = \infty$, try another P.
Now $2P$ has prime order q.

Schoof's algorithm:
time $B^{5+o(1)}$.

Conjecturally find prime q after $B^{1+o(1)}$ curves on average.
Reduce number of curves
by allowing
smaller ratios $q/\#E(\mathbf{Z}/n)$.

Recursion involves $B^{1+o(1)}$ levels.
Reduce number of levels
by allowing and demanding
smaller ratios $q/\#E(\mathbf{Z}/n)$.

Overall time $B^{7+o(1)}$.

Faster way to generate curves
with known number of points:
generate curves with
small-discriminant
complex multiplication (CM).
Reduces conjectured time
to $B^{5+o(1)}$.
With more work: $B^{4+o(1)}$.

CM has applications
beyond primality proofs:
e.g., can generate CM curves
with low embedding degree
for pairing-based cryptography.

Complex multiplication

Consider positive squarefree integers $D \in 3 + 4\mathbf{Z}$.
(Can allow some other D's too.)

If prime n equals $(u^2 + Dv^2)/4$ then "CM with discriminant $-D$" produces curves over $\mathbf{Z}/n$ with $n + 1 \pm u$ points.

Assuming $D \leq B^{2+o(1)}$:
Time $B^{2.5+o(1)}$.
Fancier algorithms: $B^{2+o(1)}$.

First step: Find all vectors $(a, b, c) \in \mathbf{Z}^3$ with $\gcd\{a, b, c\} = 1$, $-D = b^2 - 4ac$, $|b| \le a \le c$, and $b \le 0 \Rightarrow |b| < a < c$.

How?

Try each integer b between $-\lfloor\sqrt{D/3}\rfloor$ and $\lfloor\sqrt{D/3}\rfloor$.
Find all small factors of $b^2 + D$.
Find all factors $a \le \lfloor\sqrt{D/3}\rfloor$.
For each (a, b), find c and check conditions.

Second step: For each (a, b, c) compute to high precision $j(-b/2a + \sqrt{-D}/2a) \in \mathbf{C}$.

Some wacky standard notations:

$q(z) = \exp(2\pi i z)$.

$$\eta^{24} = q\Big(1 + \sum_{k \ge 1} (-1)^k q^{k(3k-1)/2} + \sum_{k \ge 1} (-1)^k q^{k(3k+1)/2}\Big)^{24}.$$

$f_1^{24}(z) = \eta^{24}(z/2)/\eta^{24}(z)$.

$j = (f_1^{24} + 16)^3 / f_1^{24}$.

How much precision is needed?

Answer: $\leq B^{1+o(1)}$ bits;
$\leq B^{0.5+o(1)}$ terms in sum;
$\leq B^{1+o(1)}$ inputs (a, b, c);
total time $\leq B^{2.5+o(1)}$.

Don't need explicit
upper bound on error.
Start with low precision;
obtain interval around answer;
if precision is too small,
later steps will notice
that interval is too large,
so retry with double precision.

Third step: Compute product
$H_{-D} \in \mathbf{C}[x]$
of $x - j(-b/2a + \sqrt{-D}/2a)$
over all (a, b, c).

Amazing fact: $H_{-D} \in \mathbf{Z}[x]$.
The j values are
algebraic integers
generating a class field.

$\le B^{1+o(1)}$ factors.
Time $\le B^{2+o(1)}$.

Fourth step: Find a root r of H_{-D} in $\mathbf{Z}/n$.

Easy since n is prime.

Amazing fact: the curve $y^2 = x^3 + (3x+2)r/(1728-r)$ has $n+1+u$ points for some (u, v) with $4n = u^2 + Dv^2$.

FastECPP using CM

To prove primality of n:

Choose $y \in B^{1+o(1)}$.

For each odd prime $p \le y$,
compute square root of p
in quadratic extension of $\mathbf{Z}/n$.
Also square root of -1.

Each square root
costs $B^{2+o(1)}$.
Total time $B^{3+o(1)}$.

For each positive squarefree
y-smooth $D \in 3 + 4\mathbf{Z}$
below $B^{2+o(1)}$,
compute square root of $-D$
in quadratic extension of $\mathbf{Z}/n$.

Each square root
costs $B^{1+o(1)}$:
multiply square roots of primes.

Total time $B^{3+o(1)}$.

For each D
having $\sqrt{-D} \in \mathbf{Z}/n$,
find u, v with $4n = u^2 + Dv^2$,
if possible.

This can be done by
a half-gcd computation.
Each D costs $B^{1+o(1)}$.

Total time $B^{3+o(1)}$.

Conjecturally there are
$B^{1+o(1)}$ choices of (D, u, v).

Look for $n + 1 \pm u$
having form $2q$ where q is prime.
More generally:
remove small factors
from $n + 1 \pm u$;
then look for primes.

Each compositeness proof
costs $B^{2+o(1)}$.
Total time $B^{3+o(1)}$.

Conjecturally have
several choices of (D, u, v, q),
when $o(1)$'s are large enough.

Use CM to construct curve
with order divisible by q.
Time $\le B^{2.5+o(1)}$; negligible.

Problems can occur.
Might have $n+1+u$
when $n+1-u$ was desired,
or vice versa. Curve might not
be isomorphic to curve of desired
form $y^2 = x^3 + ax^2 + x$.
Can work around problems,
or simply try next curve.

Recursively prove q prime.

Deduce that n is prime.

$\le B^{1+o(1)}$ levels of recursion.

Total time $\le B^{4+o(1)}$.

Verification time $\le B^{3+o(1)}$.

Open: Can we quickly find (E, q) with E an elliptic curve (or another group scheme), q prime, $q \in [n^{0.6}, n^{0.9}]$, and $\#E(\mathbf{Z}/n) \in q\mathbf{Z}$?