Oct13

Monday, October 12, 2020 9:00 PM

Idea

- Miller-Rabin checks (probable) primality quickly
- Randomly guessing #s & checking primality seems to work... why?

The Distribution of Primes

Question

> If I pick a random # how likely is it to be prime?

Def^n Let $N \in \mathbb{N}$ we define

$$\pi(N) := \#\{\text{primes } p \mid 2 \leq p \leq N\}$$

Example

$\pi(10) = 4$ $\qquad 2 \leq \underbrace{2, 3, 5, 7}_{4} \leq 10$

$\pi(11) = 5$

$\pi(12) = 5$

$\pi(13) = 6$

Property

It is an increasing function.

Theorem (Prime number theorem)

* $$\lim_{n\to\infty} \frac{\pi(n)}{n/\ln(n)} = 1$$

* $$\pi(n) = O\left(\frac{n}{\ln(n)}\right)$$

Proof is hard, uses complex analytic methods.

Consequence

If I look at all #s between 1 and n.

The proportion of primes is $\approx \frac{1}{\ln(n)}$

Slogan

A randomly chosen N has about a 1/N chance of being prime

Examples

How many primes btwn 900,000 & 1,000,000

$$\pi(1{,}000{,}000) - \pi(900000)$$

$$\approx \frac{1000000}{\ln(1000000)} - \frac{900000}{\ln(900000)}$$

$$\approx \underline{6736.62}$$ (# exactly 7224)

$$\text{percentage} = \frac{6736.62}{100000} \approx 6 \text{ or } 7\%$$

Example

What about 1024 bit primes?

primes $2^{1023} < p < 2^{1024}$

$$\pi(2^{1024}) - \pi(2^{1023}) \approx \frac{2^{1024}}{1024 \ln 2} - \frac{2^{1023}}{1023 \ln 2}$$

$$\approx 2^{1013.52}$$

So approx 0.14% of 1024 bit #s are prime.

$$(1-.0014)^n \xrightarrow{n\to\infty} 0 \quad \text{pre}$$

So expect it to take ≈ 700 random checks to find a prime!

(related to $(1-.0014)^{700} < 0.5$.)

Remark

Know evens, mults of 3, 5, 7, 11 not prime

e.g. $2\cdot 3\cdot 5\cdot 7\cdot 11 = 2310$.

Find m w/ $\gcd(m, 2310) = 1$

e.g. $1139 = m$

#s of form $2310 + 1139k$ ← are never mults of 2, 3, 5, 7, 11

more likely than a random # to be prime.

Restrict our search for primes to #s of form $2310+1139k$. go faster.

Remark

Probabilistic vs Provable primality

- Probably Prime runs in $O(\log(n))$ But it doesn't Prove what it returns is prime.
- If we assume Riemann-Hypothesis (not proven) can prove $\exists$ miller rabin witness for n that is $< \ln(n)^2$. Use this to write an $O(\log(n)^3)$ algorithm for primality testing.
- $\exists$ AKS primality test (2002). Proves n is prime (or not) in

$$O(\log(n)^{6+\epsilon})$$

Integer Factorization

RSA relies on the difficulty of factoring $N=p\cdot q$ into p & q.

The Naive Algorithm

Does $2 | N$? no

Does $3 | N$? no

Does $4 | N$? no . . . . . .

⋮

Does $p | N$ ← yes! Done

↳ Let $q = N/p$ & $N = pq$

This guaranteed to terminate... in $\sqrt{N}$ steps.

If $N = O(2^k)$ (k-bits)

This takes $\sqrt{N} = O(2^{k/2})$ steps

Exponential Bad.

Slightly Better

Restrict to checking primes

$2 | N$? no

$3 | N$?

$5 | N$?

$7 | N$?

⋮

How many divisions?

= How many primes $< \sqrt{N}$

$= \pi(\sqrt{N}) \approx \frac{\sqrt{N}}{\ln(\sqrt{N})}$ ← prime # thm.

$N \approx 2^k$

Takes $O\left(\frac{2^{k/2}}{\ln(2^{k/2})}\right) = O\left(\frac{2^{k/2}}{k/2 \ln 2}\right)$

$= O\left(\frac{2^{k/2}}{k}\right)$

↑ still exponential.

Moral: Not that much better b/c there are LOTS of primes.

Pollard's P−1 Method

Idea

*Presented w/ $N = pq$. Want p, q.

* Suppose we find some L satisfying:

$$\underline{*(p-1) \mid L} \qquad\qquad * \ (q-1) \nmid L.$$

$$\Updownarrow$$

$$L = (p-1)i \qquad\qquad L = (q-1)j + k$$

$$k \neq 0$$

* Exploit this in the following way: we fix some a, & consider a^L

① $a^L = a^{i(p-1)} \equiv 1 \bmod p.$

② $a^L = a^{j(p-1)+k} \equiv a^k \bmod q.$

↑ probably not 1.

So $p \mid a^L - 1$ & $q \nmid a^L - 1$

But $(\text{Factors of } N) = \{1, p, q, pq\}$

$$\Longrightarrow \gcd(\underline{a^L - 1, N}) = p$$

↑ Found a factor!

Note We've reduced factoring N to finding an L s.t.

$$p-1 \mid L \quad \& \quad q-1 \nmid L.$$

Rmk Don't even know p, q ($p-1$ or $q-1$).

Insight

If $p-1$ is a product of small primes then $p-1 \mid n!$ for n (relatively) small.

Example $2310 = 2\cdot 3\cdot 5\cdot 7\cdot 11$

so $2310 \mid 11!$

In fact If $p-1 = p_1^{\alpha_1} \cdots p_t^{\alpha_t}$

$n! = \underline{(\alpha_i p_i)!}$ for largest $\alpha_i p_i$

$$p^{\alpha} \mid (\underline{\alpha \cdot p})!$$

$\geq \alpha$ many p's in it.

So I just try $n! = L$ as n increases

Compute $GCD(a^{n!}-1, N) = ?$ (only be $1, p, q, pq$)

If I get 1, increase n.

If I get some m w/ $1 < m < N$ I factored N ($p = m$, $q = N/m$)

Rmk if $GCD(b, N) = d$ any $d \neq 1, N$ I'm done. Pollard's $p-1$ method just gives me reasonable candidates for b.

In Summary $N = pq$

① If $p-1 \mid L$ Then good chance $\gcd(a^L - 1, N) = 1$

② If $p-1$ has small prime factors, good chance that $(p-1) \mid n!$ for small n.

100 is small

Implementation Concerns

① $a^{n!}$ gets HUGE fast.

Ex $2^{100!} \approx 10^{10^{157}} >$ # particles in known universe.

Fix $\gcd(a^{n!} - 1, N)$ only depends on residue of $a^{n!} - 1$ mod N.

$$\left(a = bq + r \Rightarrow \gcd(a,b) = \gcd(b,r) \right)$$

$0 \leq r < b$

Fix do FastPower$(2, n!, N)$
Subtract 1.

② Even computing $n!$ is unwieldy.

Ex
square cube fourth fifth
compute $a, a^{2!}, a^{3!}, a^{4!}, a^{5!}, \ldots a^{n!}$

Then compute $\gcd(a^{n!} - 1, N)$.

Notice $a^{(i+1)!} = \left(a^{i!}\right)^{i+1}$

Algorithm (Pollard's p-1 factorization alg).

Input: $N = pq$ to be factored.

① Set $a = 2$ (or some other convenient #

② Loop $j = 2, 3, 4, \dots$ some upper bound.

③ Let $\underline{a = a^j}$ $(\equiv 2^{j!} \bmod N)$

④ Compute $GCD(a-1, N) = d$

⑤ If $1 < d < N$.

Done. $p = d$ $q = N/D$.

⑥ If $d = N$ Break or Change a.

Analyze Time

Suppose Finishes in n loops ($p-1 \mid n!$, $n \approx \ell$ for max'l such prime divisor ℓ of $p-1$

ith loop $O(\log N)$

So $O(n \log N)$

Prime factors of $p-1$ are big.

This could take a while.

Remark If you are generating primes p, q for RSA, check if prime divisors of $p-1$ and $q-1$ are big.

How?

Try factoring w/ small primes.

If you can't, an attacker probably can't either.

Do same for DLP w/ $\mathbb{F}_p^*$. Pohlig-Hellman.

Can utilize factors of $p-1$ as well.

Moral:

* What looks like a secure RSA key may be insecure based on values of $p-1$ & $q-1$.

* Surprising fact that factors of $p-1$ help w/ factoring pq.

One other Avenue

Factoring via difference of squares

$$a^2 - b^2 = (a-b)(a+b)$$ ← powerful for finding factorizations.

Given some N. ← trying to factor.

Compute $N+b^2$ see if it's a perfect square.

If yes $N + b^2 = a^2$

$$N = a^2 - b^2 = \underbrace{(a-b)(a+b)}$$

↖ Factored N!

Example 57

$57 + 1^2 = 58$ ← Not a square

$57 + 2^2 = 61$ ↙

⋮

$57 + 8^2 = 121 = 11^2$

$$57 = 11^2 - 8^2 = (11-8)(11+8)$$

$$= 3 \cdot 19$$

Notice Can try w/ kN.

If $\underline{k}N+b^2=a^2$

$kN=(a-b)(a+b)$ so

N & $(a-b)$ or N & $(a+b)$ prob share a factor.

Compute $\gcd(N, a-b)$, $\gcd(N, a+b)$.

Example Factor 21.

$63=21\cdot 3$

$63+1^2=8^2$

$\underline{3}\cdot 21=63=(8-1)(8+1)=\underline{7}\cdot 9$

$\gcd(21,7)=7$

$\gcd(9,21)=3$

---

i.e. Find a, b s.t. $a^2-b^2=kN$

Use this to factor kN & probably factor N.

Clear Problem

Find $a\neq b$ s.t. $a^2\equiv b^2 \bmod N$

Turned factorization problem into ↑