Recall from last day:
- randomized Monte Carlo algorithm to test if a number \( n \) is prime
  - poly. time
  - one sided error:
    if alg. says \( n \) is composite, it's correct
    if alg. says \( n \) is prime, \( \text{Prob}[\text{error}] \leq \frac{1}{2} \)
    - can improve w/ repeated trials.

There is a deterministic (non-randomized) poly. time primality test, but above is more practical.

Follow ups:
- how to generate a large random \( t \)-bit prime
  - generate random number between \( 2^t \), \( 2^{t+1} - 1 \)
  - test if prime
  - if not, repeat (generate new number)

To show that this is practical, we need distribution of primes — see Prime Number Theorem, next page.

- RSA cryptosystem needs large primes and relies on hardness of factoring
  \( n = p, q \) \( p, q \) prime.

- factoring — no poly. time alg. known.
  decision version: given numbers \( n, m \) does \( n \) have a factor \( \leq m \)?
  e.g. \( m = \sqrt{n} \) gives primality testing
  — for general \( m \), no good randomized alg. known
  — but problem not known to be NP-complete.
Fingerprinting - continues idea of hashing.

Example: Testing equality of strings
- too expensive to send/compare strings.
  e.g. two databases maintained in separate locations.
  solution: send & compare a smaller fingerprint.

$x - n$-bit binary = number $< 2^n$

$$H_p(x) = x \mod p \quad p \text{ a prime chosen at random}$$

size of $H_p(x)$ is $\leq \log(M)$ from $1 \leq M \leq$ chosen later.

# bits

$$x = y \Rightarrow H_p(x) = H_p(y)$$

But can have $H_p(x) = H_p(y)$ but $x \neq y$ \(\exists \) "failure".

iff $p$ divides $|x - y|$.

What is $\text{Prob} \exists \text{failure}$?

Let $\pi(N) = \# \text{primes } < N \sim \frac{N}{\ln N}$, Prime Number Theorem

Another result from number theory

$\# \text{primes divide } A < 2^n$ is $\pi(n)$

$$\text{Prob} \exists \text{failure} = \frac{\# \text{primes } p < M \text{ and } p \text{ divides } |x - y| < 2^n}{\pi(M)} = \frac{\pi(n)}{\pi(M)}$$

Pick $M = n^2$ then

$$\text{Prob} \exists \text{failure} = \frac{n}{\ln n} \cdot \frac{\ln n^2}{n^2} = \frac{2}{n}$$

and size of fingerprint is $O(\log M) = O(\log n)$
Verifying polynomial identities.

Ex. Vandermonde matrix

\[ M = \begin{bmatrix}
1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\
1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
1 & x_n & x_n^2 & \cdots & x_n^{n-1}
\end{bmatrix} \text{ n \_ terms}
\]

Vandermonde identity \( \det(M) = \prod_{i<j} (x_i - x_j) \)

We can verify this by substituting random values for variables.

Note: can compute determinants efficiently (for matrix of numbers).

In fact, can compute \( \det(M_{n \times n}) \) as fast as matrix multiplication \( \mathcal{O}(n^w) \) where \( w \approx 2.373 \)

- can compute modulo prime to avoid large numbers overflow

For this example there is a proof of identity, but the same idea can be used in symbolic math programs, theorem proving, etc.

Thm [Schwartz-Zippel]

Let \( f(x_1, \ldots, x_n) \) be a multivariate polynomial of total degree \( d \) (e.g. \( x_1 x_2^3 + x_2^2 + x_1 x_2 \) has total degree 4).

If \( f \) is not identically 0 and we choose values \( a_1, \ldots, a_n \) for \( x_1, \ldots, x_n \) independently and uniformly from a finite set \( S \) then (\( S \subseteq F \))

\[ \Pr \exists \ f(a_1, \ldots, a_n) = 0 \leq \frac{d}{|S|} \]

e.g. \( S = \{-1, 1, \ldots, d\} \Rightarrow \Pr \exists f(a_1, \ldots, a_n) = 0 \leq \frac{1}{2} \)

We can do computation of \( f(a_1, \ldots, a_n) \mod \) prime \( p \).
Result. Testing polynomial identity $f(x_1, \ldots, x_n) \equiv 0$?

Monte Carlo alg. - poly. runtime
- one-sided error: if alg. says $\neq 0$ - correct
- $\equiv 0$ - Pr(error) small.
OPEN - poly. time non-randomized alg.

Proof. by induction on $n$

basis: $n=1$. A polynomial of deg. $d$ has $\leq d$ roots, so there are $\leq d$ choices out of $|S|$ to get 0.

induction:
Write $f(x_1, \ldots, x_n) = \sum_{i=0}^{d} x_i^d f_i(x_2 \ldots x_n)$

since $f$ is not identically 0, at least one $f_i \neq 0$

if $f \neq 0$ let $k$ be largest such $i$

$f_k(x_2 \ldots x_n) \neq 0$ and has total degree $\leq d-k < d$

By induction $\Pr \{ f_k(a_2, \ldots, a_n) \neq 0 \} \leq \frac{d-k}{|S|}$

if $a_i$'s chosen ind. from $S$.

Also, if $f_k(a_2, \ldots, a_n) \neq 0$ then

$f(x_1, a_2, \ldots, a_n)$ is a deg. $k$ poly. in $x_1$

and we can apply basis case, with conditional prob.

$\Pr \{ f(a_1, \ldots, a_n) = 0 \mid f_k(a_2, \ldots, a_n) \neq 0 \} \leq \frac{k}{|S|}$

Finally, we use

$\Pr \{ A \} \leq \Pr \{ B \} + \Pr \{ A \mid B \}$

$\Pr \{ f(a_1, \ldots, a_n) = 0 \} \leq \Pr \{ f_k(a_2, \ldots, a_n) = 0 \}$

$+ \Pr \{ f(a_1, \ldots, a_n) = 0 \mid f_k(a_2, \ldots, a_n) \neq 0 \} \leq \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}$

$\Box$
In case you worry about this, here is a formal proof.

We use the definition $\Pr[\exists x | Y] \leq \Pr[\exists x \cap \neg Y]$.

\[
\Pr[\exists A] = \Pr[\exists A \cap \neg B] + \Pr[\exists A \cap B]\n = \Pr[\exists B \Pr[\exists A | \neg B] + \Pr[\exists B] \Pr[\exists A | B]\n\leq \Pr[\exists B] + \Pr[\exists A | \neg B]
\]

Application verifying matrix multiplication, Freivald's technique

Matrices $A$, $B$, $C$ $n \times n$ verify $AB = C$

Naive matrix mult. $O(n^3)$

Fastest matrix mult. alg. $O(n^{2.373})$, Coppersmith-Winograd

Slight improvements of ($\text{Complicated!}$)

You know Strassen, $O(n^{2.8})$

Chance of bug in program is high.

Verification would be good (if fast)

$ABx = Cx$ $x = (x_1, \ldots, x_n)$ — poly. identity problem.

Degree 1 multivariate polynomial so we can use $S = \{0, 1\}$

Algorithm:

- Choose each $x_i = \text{rand} [0, 1]$.
- Compute $A(Bx)$ and $Cx$.
- If $ABx = Cx$ output YES (maybe)
- Else output NO — for sure.

\[
\Pr[\text{error}] \leq \frac{d}{|S|} = \frac{1}{2}
\]

$\{0, 1\}$

$O(n^2)$ time and we can repeat to reduce error.