Lecture 12 : Identity and Primality testing

1. Identity testing
   - Matrix multiplication verification
   - Polynomial Identity Testing (PIT) via Schwartz-Zippel
   - perfect matching identification

2. Primality testing
   - Fingerprinting
   - Basics of Number Theory, Group theory
   - Fermat test, Euler test, Miller-Rabin test

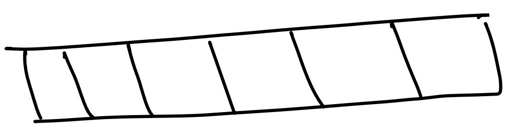
Is there an elephant in the room?

Deiter: Don't know, the lights are off.

Randy: [Runs around the room for a while] pretty sure there isn't one.

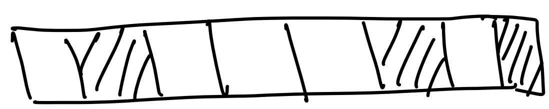
Empty list checking

Given a large list $A[1, \dots, n]$ that is either empty or has $n/2$ non empty elements. For any $1 \le i \le n$, we get to know if $A[i] = \phi$ or not.



Empty (YES)

or



Not empty (NO)

Any deterministic algorithm will have to probe $n/2$ positions in the worst case.

probe c random indices $i_1, \ldots, i_c$

- if $A[i_1] = \cdots = A[i_c] = \phi$, return "YES"
- else if $A[i_k] \neq \phi$, return "NO"

witness for non-emptyness

one sided error → False positives are possible but probability $\leq 2^{-c}$.

→ NO false negatives

This is pretty much the core principle in what follows. Smarts in choosing the list and elements carefully.

Freivalds' Algorithm: Matrix Multiplication verification

Given $n \times n$ matrices A, B, C.

Output "Yes" if $AB = C$. "NO" otherwise.

Trivial deterministic algorithm is to compute AB and entrywise compare with C. Takes n^{ω} time

$\omega \to 2.371552$

1. Pick a random vector $x = (x_1, x_2, \ldots, x_n)$ such that x_i is i.i.d uniform from some finite set S, $|S| \geq 2$.

2. if $(AB)x \neq Cx$, return "NO"

else return "YES"

each run of this algorithm takes $O(n^2)$ time as $(AB)x = A(Bx)$. Checking $(AB)x \stackrel{?}{=} Cx$ requires three matrix-vector multiplications.

Lemma: If $P \in \mathbb{R}^{n \times n}$, $P \neq 0$, then $\Pr_{x \sim S^n}[Px = 0] \leq \frac{1}{|S|}$.

proof:- Without loss of generality, assume $P_{11} \neq 0$.

If $Px = 0$, $\sum_{j=1}^{m} P_{1j} x_j = 0 \Rightarrow x_1 = -\frac{1}{P_{11}} \left[\sum_{j \geq 2} P_{1j} x_j\right]$

For any fixed $x_2, \dots, x_n$, there is exactly one choice for x_1 that satisfies the above condition.

So $\Pr[Px = 0] \leq \frac{1}{|S|}$. □

Corollary: If $AB \neq C$, $\Pr[(AB)x = Cx] \leq \frac{1}{|S|}$.

[set $P = AB - C$ in lemma]

Intuitively, the null space $\{x : Px = 0\}$ of a non-zero matrix P is "sparse". There is abundance of witness $\{x : Px \neq 0\}$.

Px is a multivariate polynomial of degree 1. This property can be generalized to polynomials of degree d. The null space becomes roots/zeros of P.

A degree-d polynomial is of the form:

$$p(x_1, \dots, x_n) = \sum_{\substack{\sum_{i=1}^{n} \alpha_i \le d \\ \alpha_i \in \mathbb{Z}_{\ge 0}}} c_\alpha \prod_{1 \le i \le n} x_i^{\alpha_i}.$$

p is a polynomial over field $\mathbb{F}$ if $c_\alpha \in \mathbb{F}$.

The polynomial p should not be confused with the function it computes. [x^2, x compute the same function in $\mathbb{F}_2$. This happens if degree $\ge$ size of field.]

If $\deg(p) \le |\mathbb{F}|-1$, then the polynomial is uniquely determined by the function it computes.

A zero polynomial is the polynomial with all coefficiets $c_\alpha = 0$. So $x^2 - x$ over $\mathbb{F}_2$ is not a zero polynomial even though it computes zero everywhere.

<u>Fact [degree mantra]</u>

A univariate polynomial $p(x)$ over field $\mathbb{F}$ has at most $\deg(p)$ roots, unless $p(x)$ is the zero polynomial.

A corollary of the Fact above is that

$$\Pr_{x \sim S}[P(x) = 0] \leq \frac{d}{|S|}.$$

when x is picked uniformly randomly from a set $S \subseteq \mathbb{F}$.

<u>Schwartz-Zippel</u>

For $P \neq 0$, $\Pr_{\substack{x_i \sim S \\ \text{for } 1 \leq i \leq n}}[P(x_1, x_2, \ldots, x_n) = 0] \leq \frac{d}{|S|}$.

<u>Proof</u>:- proof is by induction on n. For the base case $n = 1$, $P(x)$ has at most d roots. So

$$\Pr_{x_1 \sim S}[P(x_1) = 0] \leq \frac{d}{|S|}.$$

For the inductive step, let k be the largest degree of x_1 in P and write

$$P(x_1, \ldots, x_n) = \underbrace{M(x_2, \ldots, x_n)}_{\text{degree } \leq d-k} x_1^k + \underbrace{N(x_1, \ldots, x_n)}_{\text{degree of } x_1 < k}$$

Let $\mathcal{E}$ be the event that $M(x_2, \ldots, x_n) = 0$.

(M is not the zero polynomial by definition)

case 1: $\mathcal{E}$ happens. From inductive hypothesis,

$$\Pr[\mathcal{E}] \leq \frac{d-k}{|S|}$$

case 2: $\neg\mathcal{E}$ happens. Let $P'(x) = P(x, x_2, \ldots, x_n)$ be the univariate polynomial obtained by fixing $x_2, \ldots, x_n$ conditioned on $\mathcal{E}$ not happening

$$\Pr_{x \sim S}\left[p'(x) = 0 \mid \neg \mathcal{E} \right] \le \frac{k}{|S|}$$

$$\Pr\left[P(x_1, \dots, x_n) = 0 \right] = \Pr\left[P(x_1, \dots, x_n) = 0 \mid \mathcal{E} \right] \cdot \Pr(\mathcal{E})$$

$$+ \Pr\left[P(x_1, \dots, x_n) = 0 \mid \neg \mathcal{E} \right] \cdot \Pr(\neg \mathcal{E})$$

$$\le 1 \cdot \frac{d-k}{|S|} + \frac{k}{|S|} \cdot 1 = \frac{d}{|S|} \quad \square$$

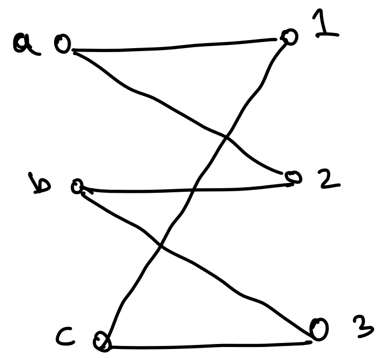
Schwartz-Zippel naturally gives an algorithm for polynomial identity testing. To check if $Q \equiv R$, we check if $P = Q - R$ is zero by evaluating at random points.

$$\det \begin{bmatrix} x+y & x^2 - y^2 & 0 \\ 1 & x & 1 \\ 0 & y & 1 \end{bmatrix} \to \text{zero?}$$

Detecting Perfect Matchings by Computing a Determinant

$$\begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{array}$$

bi-adjacency matrix



G

$$E = \begin{bmatrix} x_{11} & x_{12} & 0 \\ 0 & x_{22} & x_{23} \\ x_{31} & 0 & x_{33} \end{bmatrix}$$

Edmonds matrix

$\det(E) = x_{11}x_{22}x_{33} + x_{12}x_{23}x_{31}$

Fact:- every monomial in $\det(E)$ corresponds to a perfect matching in G.

$\Downarrow$

over any field $\mathbb{F}$

$\det(E)$ is a non-zero polynomial iff G contains a perfect matching.

PM-tester (bipartite graph G, $S \subseteq \mathbb{F}$)

1. $E \leftarrow$ edmonds matrix of graph G
2. Sample each non-zero entry $x_{i,j} \sim S$ uniformly and independently at random.
3. $\tilde{E} \leftarrow$ matrix with sampled values substituted

 if $\det(\tilde{E}) = 0$ then

  return G does not have a PM (NO)

 else

  return G contains a PM (Yes)

No false positives, $\Pr(\text{false negative}) \le \frac{n}{|S|}$ (choose $|S| \ge n^3$)

Computing the determinant takes $O(n^3)$ time by gaussian elimination. But $O(n^\omega)$ time algorithms known [Bunch, Hopcroft]

→ parallel algorithms using $O(\log^2 n)$ time, $O(n^{3.5})$ processors known [Berkowitz]

The PM detection algorithm can be converted to a PM finding algorithm.

Find-PM (bipartite graph G, $S \subseteq \mathbb{F}$)

1. Assume G has a perfect matching let $e = uv$ be an edge in G.

   if PM-tester($G[E-e]$ == YES) then

   return Find-PM ($G[E-e]$, S)

   else

   $M' \leftarrow$ Find-PM($G[V-\{u,v\}]$, S)

   return $M' \cup \{e\}$

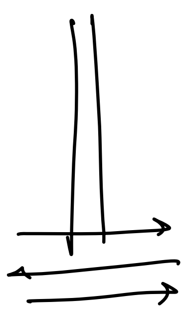
$O(mn^{\omega})$ time

we conclude the identity testing part with another nice application of the monovariate case of Schwartz-Zippel (degree mantra)

Communication Complexity of Equality

Alice | Bob

$a \in \{0,1\}^n$ | $b \in \{0,1\}^n$



Can communicate
back & forth

Goal: Test if $a=b$ using min # bits communicated.

Any deterministic algorithm needs $\geq n$ bits

(essentially, Alice sends Bob her message)

The following protocol uses $O(\log n)$ bits of communication with $\Pr[\text{error}] \leq \frac{1}{\text{poly}(n)}$.

over coin flips of protocol

Polynomial - Protocol

1. Alice sends Bob an arbitrary prime

$n^2 \leq q \leq 2n^2$ $\qquad \log q \to O(\log n)$ bits

2. Alice forms $A(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x$

Bob forms $B(x) = b_n x^n + b_{n-1} x^{n-1} + \dots + b_1 x$

Goal is to decide if $A \equiv B$.

3. Alice picks a random $\alpha \in \mathbb{F}_q$ and sends

$A(\alpha) \in \mathbb{F}_q$ to Bob $\qquad O(\log n)$ bits

Bob computes $B(\alpha)$, says "Yes" if $A(\alpha) = B(\alpha)$

"No" otherwise.

if $a=b$, $\Pr[\text{"YES"}] = 1$

if $a \neq b$, $\Pr[\text{"YES"}] = \Pr_{\alpha \sim \mathbb{F}_q}[A(\alpha) = B(\alpha)] \leq \frac{n}{q} \leq \frac{1}{n}$.

we can design a protocol that does not involve polynomials but uses the properties of prime numbers.

Prime - protocol

1. Alice picks a prime p u.a.r from $\{1, 2, 3, \ldots, T\}$
2. Alice sends p and $a \bmod p$ to Bob
3. Bob says "yes" if $b \bmod p = a \bmod p$
   "NO" otherwise

if $a = b$, $\Pr[\text{"yes"}] = 1$

if $a \neq b$, $\Pr[\text{"YES"}] = \Pr[b \equiv a \bmod p]$

$$\overset{(1)}{\Leftrightarrow} \quad p \mid |b-a|$$

$|b-a|$ is a n bit number so has $\leq n$ distinct prime factors.

so $\Pr[b \equiv a \bmod p] \leq \dfrac{n}{\#\text{primes} \leq T}$

$\pi(T) := \#$ primes $\leq T$

Theorem [Prime Number Theorem]

$$\frac{x}{\ln x} \leq \pi(x) \leq \frac{1.26\,x}{\ln x} \quad \forall\, x \geq 17$$

so $$\frac{n}{\pi(T)} \leq \frac{n \ln T}{T}$$

picking $T = cn\log n$ gives $\Pr[\text{error}] \leq \frac{1}{c} + o(1)$

Step 2 requires $O(\log T) = O(\log n)$ bits of communication

How is step 1 executed?

Sample a random number in $[2, T]$ until it is prime.

$$\mathbb{E}[\#\text{trails}] \sim \frac{T}{\pi(T)} = \ln T \sim \ln n.$$

But how do we verify that a number is prime?

Primality Testing

Fingerprinting, RSA cryptography, .. etc require a supply of primes (with thousands of bits).

Given an integer n, we wish to determine if n is prime or composite.

The following naive algorithm is known since 2000 years ago:

for $a = 2, 3, \ldots, \lfloor \sqrt{n} \rfloor$,
  if $a | n$, output "composite" and halt
output "prime"

This takes $O(\sqrt{n})$ iterations which is exponential in the input size. We want $O(\text{poly}(\log n))$ run time.

Does choosing a randomly help?

NO. If $n = pq$ for two primes p, q, there are only two non-trivial divisors

P, Q .

Some preliminaries

1. Repeated exponentiation. For any $a, b \leq n$, we can compute $a^b \bmod n$ by repeated squaring using $O(\log n)$ multiplications [of $O(\log n)$ bit numbers]

2. Euclid's algorithm. For any a, b we can compute their gcd using $O(\log a + \log b)$ additions and divisons. Binary gcd algorithm uses $O(\log n)$ bit operations.

3. Group $(G, \circ)$ — G → set of elements, $\circ$ → binary operation

(i) $a \circ b \in G$ $\forall a, b \in G$ (closure)

(ii) $(a \circ b) \circ c = a \circ (b \circ c)$ (associativity)

(iii) $\exists e$ s.t $a \circ e = e \circ a = a$ (identity)

(iv) $\exists a^{-1}$ for all a s.t $a \circ a^{-1} = a^{-1} \circ a = e$ (inverse)

[most groups we consider are also commutative $a \circ b = b \circ a$]

examples :- 1. $(\mathbb{Z}, +)$ 2. $(\mathbb{Z}_n, +)$ — $\mathbb{Z}_n$ → $\{0, 1, 2, \ldots, n-1\}$ numbers mod n

3. $(\mathbb{Z}_n^*, *)$ multiplicative group of numbers co-prime to n.

$|\mathbb{Z}_n^*| = \varphi(n)$ (euler totient function)

when n is prime, the elements are $\{1, 2, \ldots, n-1\}$

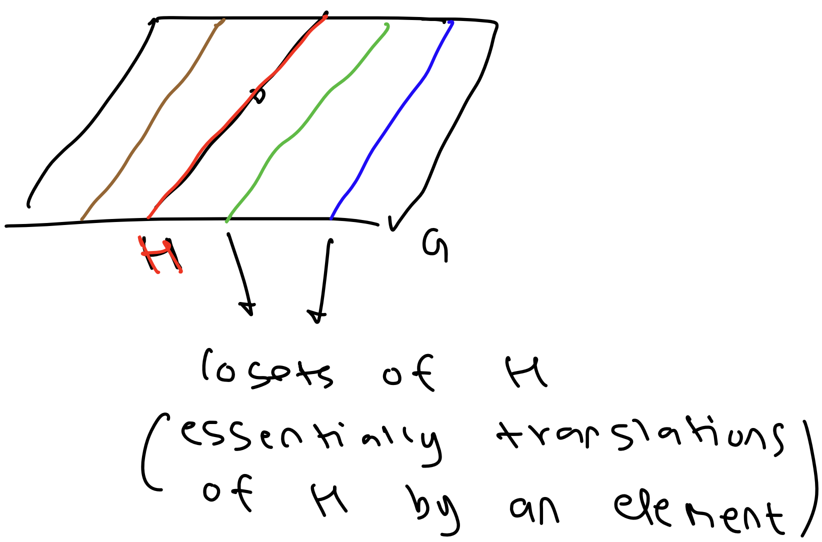
$H \subseteq G$ is a subgroup of G if H is a group.

Lagrange's Theorem

For any subgroup H of G, $|H|$ divides $|G|$.

In particular, if $H \neq G$, $|H| \leq |G|/2$.

rough proof idea:-



cosets of H partition G.

$gH = \{g \circ h : h \in H\}$

each coset has same size as H.

ex:- $\mathbb{Z}_6^+ = \{0, 1, 2, 3, 4, 5\}$

$H = \{0, 2, 4\}$ is a subgroup.

the group partitions into cosets $0+H$, $1+H$.

Fermat test

Fermat's little theorem

For any prime p, $a \in \mathbb{Z}_p^*$, one has $a^{p-1} \equiv 1 \mod p$. More generally for any (finite) group G, and $a \in G$, we have $a^{|G|} = e$.

$$\Big(a^x = \underbrace{a \circ a \circ \dots a}_{x \text{ times}} \Big)$$

<u>Proof</u>:- Let $H = \{a^k : k \in \mathbb{Z}\}$. H is a subgroup of G. Since G is finite, H is finite.

$H = \{a, a^2, a^3, \dots a^k = e\}$ as $e \in H$.

$k \mid |G|$ from Lagranges theorem.

$a^{|G|} = (a^k)^{|G|/k} = e$. $\square$

For general n, let $A_n = \{a : a^{n-1} \equiv 1 \mod n\}$. Fermat's little theorem says that $A_n = \mathbb{Z}_n^*$ when n is prime.

<u>Claim</u>:- For any n, A_n is a subgroup of $\mathbb{Z}_n^*$

If it happens that for composite n, A_n is always a proper subgroup of $\mathbb{Z}_n^*$, then $|A_n| \leq |\mathbb{Z}_n^*|/2$ by lagrange's theorem.

This would imply an abundance of witness a s.t $a^{n-1} \not\equiv 1 \mod n$ for composite numbers.

<u>Fermat test algorithm</u>

1. pick random $a \in \{1, 2, \ldots, n-1\}$
2. if $(a, n) \neq 1$, return "No" (n is composite)
3. if $a^{n-1} \bmod n \neq 1$ return "No" (n is composite)

   else return "Yes"

Turns out there are numbers n s.t $A_n = \mathbb{Z}_n^*$. That is, $a^{n-1} \equiv 1 \mod n$ for all a s.t $(a, n) = 1$. Such n are called Carmichael numbers (561, 1105, 1729, 2465, ...)

These numbers fool the Fermat test.

Euler's test

This slightly strengthens Fermat test by checking that $a^{(n-1)/2} \equiv \pm 1 \bmod n$.

[For $x = a^{(n-1/2)}$, $x^2 \equiv 1 \bmod n$ for prime n.

$(x-1)(x+1) \equiv 0 \bmod n$ "

$\Rightarrow x \equiv \pm 1 \bmod n$ "]

The last step need not follow for composite numbers.

1729, 2465 fool the euler test.

Miller - Rabin Algorithm

Euler test tries to find a non-trivial square root of 1 for just one step.

Miller Rabin test continues trying as long as possible.

Assume n is odd and not a prime power
(we can decide if $n = p^s$ quickly by searching for $1 \le s \le \lceil \log n \rceil$, binary search for n's)

So let $n-1 = 2^c d$ where d is odd.

Miller - Rabin Test. pick random $a \in \{1, 2, \dots, n-1\}$
if $\gcd(a, n) \neq 1$ return NO. So assume $a \in \mathbb{Z}_n^*$.
Consider $a^{n-1}, a^{(n-1)/2}, \dots, a^d$ (in this order). There are three possibilities.

1. Either all the numbers are 1. Output prime
2. The first entry that differs from 1 is not -1. Return composite (we found a non-trivial square root of 1)
3. The first entry that differs from 1 is -1. Output prime (we gave up on a being a witness, as we cannot proceed further once we see a -1)

example for Carmichael number $n = 561$, $n-1 = 560 = 2^4 \cdot 35$
For $a = 2$, $a^{560} = 1$, $a^{280} = 1$, $a^{140} = 67$, ... (mod 561)

Theorem

For any composite number $n > 2$ (not a prime power), the test returns composite for at least half the witnesses $a \in \mathbb{Z}_n^*$.

proof:- Let $t \in \{0, 1, 2, \dots, c\}$ be the largest power such that $x^{2^t d} \neq 1 \mod n$ for some x.

$\left[x^{2^{t+1} \cdot d} = 1 \quad \forall x \right]$

Such a t exists as $x = n-1$ satisfies $(n-1)^d = -1 \bmod n$ for $t = 0$.

$$\begin{array}{c|ccccccc} & a^{d} & a^{2d} & a^{4d} & \cdots & a^{2^t d} & a^{(n-1)/2} & a^{n-1} \\ \hline a & & & & & -1 & 1 & 1 \\ \vdots & & & & & & 1 & 1 \\ \vdots & & -1 & & & & 1 & 1 \\ -1 & -1 & 1 & 1 & & \cdot & 1 & 1 \end{array}$$

We will show at least half of the elements $a \in \mathbb{Z}_n^*$ satisfy $a^{2^t d} \not\equiv \pm 1 \bmod n$. [These a are witness for compositeness]

Let $S = \{a : a^{2^t d} = \pm 1\}$. S is a subgroup of $\mathbb{Z}_n^*$.

It suffices to show that S is a proper subgroup of $\mathbb{Z}_n^*$ (and hence $|S| \le |\mathbb{Z}_n^*|/2$)

Suppose for contradiction that $S = \mathbb{Z}_n^*$. We know there is some x s.t. $x^{2^t d} = -1 \bmod n$.

Since n is composite, $\exists\, r, s > 1$ s.t. $n = r \cdot s$ (n is not a prime power is used here).

Let y be a number such that

$y \equiv x \bmod r$, $y \equiv 1 \bmod s$ [exists by Chinese remainder theorem]

$$y^{2^t d} \equiv x^{2^t d} \bmod r \quad , \quad y^{2^t d} \equiv 1 \bmod s$$

$$\downarrow$$

$$-1 \bmod r$$

If $y^{2^t d} \equiv 1 \bmod n \Rightarrow y^{2^t d} \equiv 1 \bmod r \Rightarrow 2 \equiv 0 \bmod r$ ✗

If $y^{2^t d} \equiv -1 \bmod n \Rightarrow y^{2^t d} \equiv -1 \bmod s \Rightarrow 2 \equiv 0 \bmod s$ ✗

$y \notin S$ so S has to be a proper subgroup.

$\square$

Miller observed that assuming the Generalized Riemann Hypothesis, a witness a exists in the first $O((\log n)^2)$ values of a.

This gives a deterministic primality testing algorithm conditioned on GRH.