# Algebra And Algorithms

- FACTORIZATION OF POLYNOMIALS
- PRIMALITY TESTING
- GRAPH ISOMORPHISM
- ~~MATRIX MULTIPLICATION~~
- Course Wrap-up..

---

1. History: "Algebra" (Al Jabr) comes from a text by Al Khwarizmi ("Algorithm")
   - Names Intertwined !!

   - Remarkable Algorithms (to me most surprising) are algebraic
     - Greatest Common Divisor (EUCLID)
     - Determinant (Gauss).

## Aside: Determinant

- formal Definition:

① $\text{Sign}(\pi) = +1$ if $\pi \in S_n$ is <u>even</u>

$= -1$ if $\pi \in S_n$ is <u>odd</u>

② $$\text{Det}(M) = \sum_{\pi \in S_n} (-1)^{\text{sign}(\pi)} \cdot \prod_{i=1}^{n} M_{i\,\pi(i)}$$

$M \in \mathbb{F}^{n \times n}$ is $n \times n$ matrix over field $\mathbb{F}$.

- Definition involves $n! \approx \left(\frac{n}{e}\right)^n$ summands.

- But can be computed in polynomial time.

- Contrast

$$\text{Perm}(M) = \sum_{\pi \in S_n} \prod_{i=1}^{n} M_{i,\pi(i)}$$

for permanent

No signs!

- Belief: Perm(M) requires exp(n) time to compute

$P \neq NP \implies$ Belief

# Polynomials + Algorithms

① Addition: Takes $\Theta(n)$ time for adding two deg $n$ polynomials

② Multiplication: Naively $O(n^2)$.

- But ~~slight~~ can do better. - Karatsuba $O(n^{1.8})$
- Even better: $O(n \log n \log\log n)$ field operations [over any field]

Essence of idea (over $\mathbb{C}$)

Let $\omega$ be $2n^{th}$ root of unity

Given

$$\begin{array}{ccc} f, g \in \mathbb{C}[x] & \xRightarrow{\text{FFT}} & \langle f(\omega^i)\rangle_{i=1}^{2n} \\ & & \langle g(\omega^i)\rangle_{i=1}^{2n} \\ & O(n\log n)\text{ time} & \Big\Downarrow \leftarrow O(n)\text{ time} \\ (f\cdot g) \in \mathbb{C}[x] & \xLeftarrow[\text{FFT}^{-1}]{} & \langle (f\cdot g)(\omega^i)\rangle_{i=1}^{2n} \end{array}$$

# Polynomials + Algorithms - 2

(3) Interpolation: Given $\alpha_1 \dots \alpha_n$, $f(\alpha_1) \dots f(\alpha_n)$ compute coeff. of $f$ ($\deg f < n$)

$O(n \log^c n)$ for some $c \leq 3$.

(4) Multipoint Evaluation: Given coeff of $f$ & $\alpha_1 \dots \alpha_n$ compute $f(\alpha_1), \dots f(\alpha_n)$

$O(n \log^c n)$ for some $c \leq 3$

(5) Division with Remainder: Given $f, g$ compute $q, r$ with $\deg r < \deg g$ s.t.

$$f(x) = q(x) \cdot g(x) + r(x)$$

$O(n \log^c n)$

(6) GCD: $O(n \log^c n)$

(7) FACTORIZATION: $O(n^{1.5})$ roughly.

(~~Eg~~ Even polytime is non-trivial!).
(Field specific).

# FACTORIZATION OVER FINITE FIELDS [$\mathbb{F}_q$ - field of size $q$]

Key Idea:

$$X^q - X = \prod_{\alpha \in \mathbb{F}_q} (x-\alpha)$$

- So if $x^2 - ax + b = (x-\alpha)\cdot(x-\beta)$

then $\gcd(x^2-ax+b,\ x^q - x) = x^2-ax+b$ !

- But $(x^q - x) = x(x^{\frac{q-1}{2}}-1)(x^{\frac{q-1}{2}}+1)$ [$q$ odd]

- Hopefully:

$$\gcd(x^2-ax+b,\ x^{\frac{q-1}{2}}-1) \neq 1,\ x^2-ax+b$$

- for typical polynomials .... w.p. $\frac{1}{2}$ $(x-\alpha) \mid x^{\frac{q-1}{2}}-1$

& $\frac{1}{2}$ $(x-\beta)$ does not.

- to get all poly, factor

$(x-\gamma)^2 + a(x-\gamma) + b$ for random $\gamma$ of our choice.

$\leadsto$

$x^2 + a'x + b'$ ; roots $\gamma+\alpha$, $\gamma+\beta$

w.p. $\geq \frac{1}{2}$ over $\gamma$ $(x-(\gamma+\alpha)) \mid x^{\frac{q-1}{2}}-1$

## Basis of factoring all kinds of polynomials

- Over rationals,
- Multivariate,
- Function fields.

### One Key Notion

$$\begin{array}{ccc} \mathbb{F} & \longrightarrow & \mathbb{F}[x]/\langle p(x)\rangle \quad \text{field} \\ \text{field} & \longrightarrow & \mathbb{F}(x) \quad \text{field} \end{array}$$

$\mathbb{K}$

Given factorization alg for $\mathbb{F}[z]$ $\Longrightarrow$ Can get factorization alg. for $\mathbb{K}[z]$ for either way to get $\mathbb{K}$

# PRIMALITY TESTING

- Given $0 \le N \le 2^n$, ~~compute~~ determine if $N$ is prime.
- 70's [Rabin, Miller, Solovay-Strassen]:
  - Randomized poly(n) time algorithm to test primality.
  - But no "proof" of primality
- 2003: [Agarwal, Kayal, Saxena]
  - Deterministic algorithm for primality
  - via Algebra!!
- Key Idea.

for $a \neq 0$
$$(X+a)^N = X^N + a \pmod N$$
$$\iff N \text{ is prime.}$$

- But how to check identity? takes $\sim N = \exp(n)$ time

- Idea 1: Pick $Q(x)$ of degree $\sim (\log N)^2$ at random. Verify

$$(X+a)^N = X^N + a \pmod{N, Q(x)}$$

[Still randomized. No proofs of primality.]

- Idea 2: Pick $Q(x) = X^r - 1$ for small $r \approx (\log N)^3$

Final Algorithm:

```
for a = 1 ... (log N)^2 do
    for r = 1 ... (log N)^3 do
        Verify (X^N + a) = (X+a)^N  (mod N, X^r - 1)
    end
end
Accept if all tests accept.
```

- Key ingredient in analysis. $Q[X]/(N, X^r-1)$

# Graph Isomorphism

Given: $G = (V, E)$ $\qquad E \subseteq V \times V$

& $H = (W, F)$ $\qquad F \subseteq W \times W$

are $G$ & $H$ isomorphic?

i.e. $\exists\ \phi : V \to W$ $\quad$ 1-1

s.t. $(\phi(u), \phi(v)) \in F \iff (u,v) \in E$ ?

---

History: Long known to be in "NP"

: Not believed to be "NP-hard"

: But no polytime algorithms known.

Best till 2015: $\sim 2^{\sqrt{n}}$ ....

: [Babai 2015]: $O(n^{\log n})$ time algorithm.

[Not poly but almost!]

Key Idea: Solve "String Isomorphism" . . .

String Isomorphism :

- Given • $a, b \in \Sigma^n$ for finite $\Sigma$,
  & • $\{\pi_1 .. \pi_k\} \subseteq S_n$ generating group $G$.

- Determine if $\exists\, \pi \in G$ s.t.

$$a \circ \pi = b \quad \left[ \forall i \in n \quad a_{\pi(i)} = b_i \right]$$

———×———

• Easy: Graph Isomorphism $\leq$ String Isomorphism.

• Hard part: String Isomorphism in time $O(n^{\log n})$.

  - Lots of Group Theory
  - New Algorithms in Group Theory
    [Membership in Permutation Groups used heavily.]