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 even
     \[ = -1 \] if \( \pi \in S_n \) is odd
  2. \( \text{Det}(M) = \sum_{\pi \in S_n} (-1)^{\text{Sign}(\pi)} \prod_{i=1}^{n} M_{i\pi(i)} \)

- \( M \in \mathbb{F}^{n \times n} \) is an n x n matrix over field \( \mathbb{F} \).

- Definition involves \( n! \approx (\frac{n}{e})^n \) 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
  \( \uparrow \)
  \( \uparrow \)
  No signs!

- Belief: \( \text{Perm}(M) \) requires \( \exp(n) \) time to compute
  \( \mathbb{P} \neq \mathbb{NP} \Rightarrow \text{Belief} \)
Polynomials + Algorithms

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

2. Multiplication: Naively $O(n^2)$.
   - But slightly can do better: Karatsuba $O(n^{1.5})$.
   - Even better: $O(n \log n \log \log n)$ field operations over any field.

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

Let $w$ be $2^{2^n}$ root of unity.

Given $f, g \in \mathbb{F}[x] \Rightarrow \left< f(w^i) \right>_{i=1}^{2^n}$

$\text{FFT} \Rightarrow \left< g(w^i) \right>_{i=1}^{2^n}$

$O(n \log n)$ time $\downarrow \leftarrow O(n)$ time

$(f \cdot g) \in \mathbb{F}[x] \Leftarrow \left< (f \cdot g)(w^i) \right>_{i=1}^{2^n}$

$\text{FFT}$
Polynomials + Algorithms - 2

3) Interpolation: Given $a_0, \ldots, a_n$ compute coeff. of
$f(a_0), \ldots, f(a_n)$ (deg $f < n$)

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

4) Multipoint Evaluation: Given coeff. of $f$ & $a_0, \ldots, a_n$
compute $f(a_0), \ldots, f(a_n)$

$O(n \log^c n)$ 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.

(\# Even polytime is non-trivial!)

(Field specific).
**Factorization Over Finite Fields**

**Key Idea:**

\[ X^2 - X = \prod_{\alpha \in \mathbb{F}_2} (X - \alpha) \]

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

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

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

- **Hopefully:**

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

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

\( a \neq \frac{1}{2} (x - \beta) \) does not.

- To get all poly, factor

\( (x - y)^2 + a(x - \delta) + b \) for random \( y \) of our choice

\[ x^2 + a'x + b' \] roots \( \delta + \alpha, \delta + \beta \)

w.p. \( \geq \frac{1}{2} \) \( (x - (\delta + \xi)) \mid x^{\frac{q-1}{2}} - 1 \) over \( \mathbb{F}_2 \)
Basis of factoring all kinds of polynomials

- Over rationals,
- Multivariate,
- Function fields.

One key notion:

\[ F \xrightarrow{\text{field}} F[x]/\langle p(x) \rangle \text{ field} \]

\[ \text{field} \xrightarrow{\text{field}} \left( \frac{F[x]}{\langle p(x) \rangle} \right)_{\text{field}} \]

Given factorization alg. for \( F[x] \) => Can get factorization alg. for \( K[x] \)

for either way to get \( K \)
Primality Testing

- Given $0 \leq N \leq 2^n$, compute determine if $N$ is prime.

- 70's: [Rubin, 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:

  \[
  \text{for } a \neq 0 \quad (x + a)^N \equiv x^N + a \pmod{N}
  \]

  \[
  \implies N \text{ is prime.}
  \]
* But how to check identity? taken \( \sim N = \exp(n) \) time

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

\[
(X + a)^N \equiv X^N + a \pmod{N, Q(x)}
\]

[Still randomized. No proofs of primality.]

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

**Final Algorithm:**

```plaintext
for \( a = 1 \ldots (\log N)^2 \) do
    for \( y = 1 \ldots (\log N)^5 \) do
        Verify \( Q(x)\pmod{x^N + a} = (x+a)^N \pmod{N, x^r - 1} \)
    end
end

Accept if all tests accept.
```

* Key ingredient in analysis: \( Q[x] / (N, x^r - 1) \)
Graph Isomorphism

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

\( H = (W, F) \quad F \subseteq W \times W \)

are \( G \) and \( H \) isomorphic?

i.e. \( \exists \phi : V \rightarrow W \) 1-1

\( \text{s.t.} \quad (\phi(v), \phi(u)) \in F \iff (u,v) \in E \)

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]:\( O(n^{\text{log}^*n}) \) time algorithm
  - [Not poly but almost!]

Key Idea: Solve "String Isomorphism"
String Isomorphism:
- Given \( a, b \in \Sigma^n \) for finite \( \Sigma \),
  \( \exists \Pi_1, \ldots, \Pi_k \subseteq \Sigma_n \) generating group \( G \).
- Determine if \( \exists \Pi \subseteq G \) s.t.
  \[ a \cdot \Pi = b \quad \forall i \in n \quad a_{\Pi(i)} = b_i \]

Easy: Graph Isomorphism \( \leq \) String Isomorphism.

Hard part: String Isomorphism in time \( O(n^{100n}) \).

- Lots of Group Theory

- New Algorithms in Group Theory

\[ \text{Membership in Permutation Groups used heavily.} \]