Lecture 1 (30-07-2021)

30 July 2021 17:31

Polynomials – $x^2 + 2x + 1$

$x^3 + 7x^2 + 9x + 3y + 2.1$

$x^3 + \sqrt{3}y$

$x^4 + 2.1i$ $\quad (i^2 = -1)$

There are formal variables above and co-efficients from some other "objects".

Fields – Reals $\mathbb{R}$, Rationals $\mathbb{Q}$, Complex $\mathbb{C}$. } set, with operations $(+, \cdot)$, operations have certain properties.

In general: $F \to$ a set

binary operations : $+, \cdot$

Properties:

① Associative : $a + (b + c) = (a + b) + c =: a + b + c$ } $\forall a, b, c \in F$

$(a \cdot b) \cdot c = a \cdot (b \cdot c)$

② Commutativity : $a + b = b + a, \quad a \cdot b = b \cdot a \quad \forall a, b \in F$

③ Distributivity : $a \cdot (b + c) = a \cdot b + a \cdot c \quad \forall a, b, c \in F$

④ Existence of identity : $\exists 0 \in F$ s.t. $0 + a = a \quad \forall a \in F$ (0 → additive identity)

$\exists 1 \in F$ s.t. $1 \cdot a = a \quad \forall a \in F$ (1 → multiplicative identity)

So far, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{C}$, $\mathbb{R}$ has the above properties.

(If such an identity exists, it is unique.)

⑤ Existence of : $\forall a \in F \ \exists b \in F$ s.t. $a + b = 0$.

(5) Existence of inverse: $\forall a \in F \ \exists b \in F$ s.t. $a + b = 0$.

$\forall a \in F \setminus \{0\} \ \exists c \in F$ s.t. $ac = 1$.

(Denoted $-a$ and a^{-1}, resp.)

Example. $\mathbb{Z}$ is not a field. 2 has no mult. inverse.

$\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ are fields, these are infinite.

There are finite fields as well: $\{0, 1\}$ → set
define + and · modulo 2.

This field is denoted $\mathbb{F}_2$.

Similarly, there is $\mathbb{F}_3 = \{0, 1, 2\}$ → sums and products modulo 3.

In general, given a prime p, we have

$\mathbb{F}_p = \{0, 1, \ldots, p-1\}$ → modulo p.

Moreover, this is "the unique" field with p elements.

But these are not all. For p prime and $k \in \mathbb{N}$,

$\exists$ a field with p^k elements, denoted $\mathbb{F}_{p^k}$.

(Again unique.)

Remark. $\mathbb{F}_4$ is NOT $\{0, 1, 2, 3\}$ with + and · modulo 4.

($2^{\neq 0}$ has no mult. inverse.)

[Alternate notation: $\mathbb{F}_{p^k} \equiv GF(p^k)$.]

Non-examples of fields (1) $\mathbb{Z}$ is not a field.

(2) $GL_n(\mathbb{F}) = \{ n \times n$ inv. matrices over $\mathbb{F}\}$

↳ not even closed under (usual) addition

(3) $M_n(\mathbb{F}) = \{$all $n \times n$ matrices over $\mathbb{F}\}$

↳ non-commutative for $n \geq 2$.

Subsequent lectures: - More about finite fields
- What $\mathbb{F}_{p^k}$ looks like.
- Other properties. - useful later.

Commutative Ring. Field properties apart from multiplicative inverse.

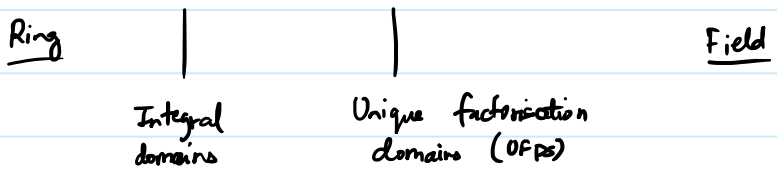
E.g. $\mathbb{Z}$ is a commutative ring.

$F[x]$ — where F is a field

$F[x_1, \dots, x_n]$ - ring of polynomials in variables $x_1, \dots, x_n$.

- We shall use "ring" to mean "commutative ring".

→ Now we define something in between rings and fields.



Def^n. A ring R is an integral domain if

$\forall a, b \in R: \quad ab = 0 \Rightarrow a = 0$ or $b = 0$.

(There are no nonzero zerodivisors.)

Examples. $\mathbb{Z}$ is an int domain, even $\mathbb{Z}[x]$ and $\mathbb{F}[x]$.

$\{0, 1, 2, 3\}$ with $+$ and $\cdot$ mod 4 is NOT.

Def^n. $a \in R$ is a unit if it has a multiplicative inverse (in R).

That is, $\exists b \in R$ s.t. $ab = 1$.

Example. $\mathbb{Z}$ → only two units $\{\pm 1\}$.

$\mathbb{F}[x]$ → the units are precisely $\mathbb{F}^* := \mathbb{F} \setminus \{0\}$.

Def^n. $a \in R$ is said to be **irreducible** if $\forall b, c \in R$:

$$a = bc \Rightarrow b \text{ is a unit or } c \text{ is a unit.}$$

Example. ① In $\mathbb{Z}$, the irreducibles are precisely the primes.

② $R = \mathbb{F}[x]$. Irreducible polynomials = pdy. which cannot be written as $b(x)\,c(x)$ where $\deg(b(x)) \geq 1$ and $\deg(c(x)) \geq 1$.

(since the units are precisely non-zero field constants)

Def^n. An integral domain R is said to be a UFD if

① every element can be written as a product of irreducible elements,

② decomposition in ① is unique up to multiplication by units of R.

Example. ① $\mathbb{Z}$ is a UFD (Fundamental Theorem of Arithmetic)

② $\mathbb{F}[x]$, $\mathbb{F}[x_1, \ldots, x_n]$ are also UFDs.

Aim: Design efficient algorithms for coming up with a polynomial decomposition into irreducibles.

Getting from an integral domain to a field.

Given a problem over $\mathbb{Z}$ → would be useful to view it over a field.

Example. $f(x) \in \mathbb{Z}[x]$ → want to factorise

will be helpful to think of it as $f \in \mathbb{Q}[x]$.

- Work over $\mathbb{Q}[x]$.
- Recover solution over $\mathbb{Q}[x]$

In general: $R \to$ integral domain

$\tilde{R} \to$ field of fractions of R.

Defining $\tilde{R}$: $\tilde{R} \equiv R \times (R \setminus \{0\}) / \sim$

where $(a,b) \sim (a',b')$ iff $ab' = a'b$.

$\sim$ is indeed an equivalence relation.

The following are well-defined:

$$[(a,b)] + [(c,d)] = [(ad+bc, bd)],$$

$$[(a,b)] \cdot [(c,d)] = [(ac, bd)].$$

Example $R = \mathbb{F}[x]$. $\tilde{R} = \left\{ \frac{p(x)}{q(x)} : q(x) \neq 0 \right\}$

$\downarrow$

field of rational functions.

Lecture 2 (03-08-2021)

03 August 2021 17:33

Recall : polynomial ring $R[x]$. Assume that R is an integral domain.

Given $f(x) = \sum a_i x^i$, define

$$\deg(f(x)) := \max\{i : a_i \neq 0\}$$

for $f(x) \neq 0$.

If $f(x) \in R[x]$ and $\deg(f(x)) = d \Rightarrow f(x) = f_0 + \cdots + f_d x^d$ with $f_d \neq 0$.

f_d is called the leading coefficient of f. $LC(f)$

$f_d x^d$ is called the leading monomial of f. $LM(f)$

(leading coefficient, leading monomial)

$f(x)\, g(x) = h(x)$.

$$f(x) = f_0 + \cdots + f_d x^d$$

$$g(x) = g_0 + \cdots + g_t x^t$$

$$h(x) = h_0 + \cdots + h_{d+t}\, x^{t+d}, \text{ where}$$

$$h_j = \sum_{i=0}^{j} f_i \cdot g_{j-i} \quad \text{for} \quad 0 \leq j \leq t+d.$$

Moreover since R is an I.D., we have

$$LC(h) = LC(f)\, LC(g).$$

Lemma. (Inheritance)

$R \to$ ring, $R[x] \to$ polynomial ring.

Then,

(i) R is commutative $\iff R[x]$ is commutative.

(ii) R is an int. dom. $\iff R[x]$ is an int. dom.

(iii) R is a UFD $\iff R[x]$ is a UFD.

Cor. $F[x]$ is a UFD if F is a field.

(Any field is a UFD.)

Thus, $\mathbb{Q}[x]$, $\mathbb{R}[x]$, $\mathbb{C}[x]$, $\mathbb{F}_{p^k}[x]$ are all UFDs.

(Even $\mathbb{Z}[x]$ is, as per the lemma.)

Thus, it makes sense to talk about factorizations in these rings.

- $R \to$ UFD

Then, $R[x_1, \dots, x_n]$ is a UFD.

$\cong R[x_1, \dots, x_{n-1}][x_n] \cong \dots$ induct.

$$f(x_1, \dots, x_n) = \sum_{\text{finite sum}} f_{i_1, \dots, i_n} x_1^{i_1} \cdots x_n^{i_n}$$

$$\text{total deg}\left(\prod_j x_j^{i_j}\right) = \sum_j i_j.$$

$$\deg(f) = \max\left\{\text{total deg}\left(\prod_j x_j^{i_j}\right) : f_{i_1, \dots, i_n} \neq 0\right\}.$$

Division of univariate polynomials.

$\mathbb{F}$ - field.

Lemma 1. $\forall g, h \in \mathbb{F}[x]$ with $h \neq 0$, $\exists$ unique q (quotient), r (remainder) $\in \mathbb{F}[x]$ s.t.

(i) $g = h \cdot q + r$

(ii) $r = 0$ or $\deg(r) < \deg(h)$.

Proof: **Uniqueness** Assume q_1, q_2, r_1, r_2 s.t.

$$g = hq_1 + r_1 = hq_2 + r_2.$$

Then, $h \cdot (q_1 - q_2) = r_2 - r_1$.

If $q_1 \neq q_2$, then $r_2 \neq r_1$ and

$$\deg(r_2 - r_1) < \deg(h) \leq \deg(h \cdot (q_1 - q_2)). \quad \to\leftarrow$$

Thus, $q_1 = q_2$ and so, $r_1 = r_2$ as well.

Existence. By induction on $\deg(g)$. If $\deg(g) < \deg(h)$,

use $q = 0$ and $r = g$.

Else define $g' = g - \frac{LM(g)}{LM(h)} h$

and proceed. ∎

GCD of polynomials.

Def^n. • Let $h, g \in \mathbb{F}[x]$.
f is said to be a factor/divisor of g, written $f | g$,
if $\exists\, q \in \mathbb{F}[x]$ s.t. $g = hq$.

• Given $h, g \in \mathbb{F}[x]$, $p \in \mathbb{F}[x]$ is a GCD of h, g if

(1) p is a common divisor of g and h, i.e., $p | g$ and $p | h$.

(2) $\forall$ common divisors of g and h, we have $u | p$.

Observations:

① GCD is not (necessarily) unique.
Indeed, if p is a GCD of g, h, then
αp is also a GCD $\forall\, \alpha \in \mathbb{F}^\times$.

② Existence?
Follows from the fact that $\mathbb{F}[x]$ is a UFD

Lemma. Let $g, h \in \mathbb{F}[x] \setminus \{0\}$. Consider

$$I(g, h) = \{ ug + vh : u, v \in \mathbb{F}[x] \}.$$

Let p be a non-zero polynomial of lowest degree in $I(g, h)$.

Then,

(i) p is a common divisor of g and h.

(ii) p is a GCD of g and h.

(iii) if q is a GCD(g, h), then $\exists \alpha \in \mathbb{F}$ s.t.

$$q = \alpha p.$$

Obs. Lemma $\Rightarrow$ GCD is defined up to a nonzero scalar multiple.

Thus, we can loosely talk about "the GCD".

(Possible convention: assume GCD to be monic, i.e., LC $= 1$.)

Proof (of lemma):

① Write $g = pq + r$ and $p = ug + vh$.

Then, $r = (\ )g + (\ )h$.

But $\deg(r) < \deg(p)$. Thus, $r = 0$. $\therefore p \mid g$.

$\Rightarrow p \mid h$.

② Suppose $q \mid g$ and $q \mid h$.

Since $p = ug + vh$ and $q \mid$ RHS, we get

$q \mid$ LHS.

③ If q is a GCD, then $p \mid q$ as well.

$\therefore p \mid q$ & $q \mid p$.

$\Rightarrow q = \alpha p$ for some $\alpha \in \mathbb{F}^*$. ∎

Lecture 3 (06-08-2021)

06 August 2021 17:31

Agenda for the upcoming lectures:

1. Fields of size p^k for p prime and $k \in \mathbb{N}$.
2. Algorithms for operations on polynomials.
   - Addition / multiplication of poly
   - Division with remainder
   - Multipoint evaluation of polynomials
   - GCD of two polynomials

Takeaway: Fast Fourier Transform by Cooley-Tukey ("one of the most useful algorithms")

- $\mathbb{F}_{p^k}$.

$k = 1$: $\mathbb{F}_p = \{0, 1, \dots, p-1\}$ Arithmetic is modulo p

Theorem. $\forall$ p primes $\forall k \in \mathbb{N}$, $k \geq 1$, there is a (finite) field with p^k elements.

Moreover, (1) these are essentially unique,
(2) these are the only finite fields.
(Not proving this.)

Today: A description of $\mathbb{F}_{p^k}$.

Irreducible polynomial: $f \in \mathbb{F}[x]$ is irred. (over $\mathbb{F}$) if f cannot be written as a product of two nonconstant polynomials (in $\mathbb{F}[x]$).

Examples: (1) $x^2 + 3x + 2$ → reducible over $\mathbb{R}$

② $x^2+1 = (x+i)(x-i)$ → irred. over $\mathbb{R}$; ↳ red. over $\mathbb{C}$

Fact. $\forall$ prime p and $d \in \mathbb{N}$, $\exists$ irred. polynomial of deg equal to d in $\mathbb{F}_p[x]$

polynomial of deg $\leq d$ in $\mathbb{F}_p[x] = p^{d+1}$

Roughly $1/d$ fraction of these polynomials are irreducible.

Now, let $g \in \mathbb{F}_p[x]$, $\deg(g) = k$, with g irred.

$\mathbb{F}_{p^k}$

Elements: $\{ f \in \mathbb{F}_p[x] : \deg(f) < k \}$

↳ p^k many elements

Arithmetic: happens in $\mathbb{F}_p[x]$ and then we go mod g.

Does this satisfy the field axioms?

① Commutativity, assoc., distributivity → carry over from $\mathbb{F}_p[x]$.

② 0, 1, existence of additive inverse → again easily follows from $\mathbb{F}_p[x]$

③ Non-trivial: existence of multiplicative inverse

↳ Proof. Let $f \neq 0$ be with $\deg(f) < k$.

Let $d := \gcd(f, g)$.

Then, $\deg(d) \leq \deg(f) < k = \deg(g)$.

Since g is irred, $d = 1$.

Then, by last class' result, $\exists a, b \in \mathbb{F}_p[x]$ s.t.

$af + bg = 1$.

Going modulo g, we have

$af = 1$.

Thus, $\bar{a} := \underbrace{a \bmod g}_{\text{remainder obtained upon diving } a \text{ by } g}$ has $\bar{a} f = 1$ in $\mathbb{F}_{p^k}$. ☑

This is how we shall imagine fields of size p^k.

Properties.

① $\mathbb{F}_{p^k}$ is a vector space over $\mathbb{F}_p$.

② $\dim_{\mathbb{F}_p}(\mathbb{F}_{p^k}) = k$.

③ One natural basis $= \{1, x, \ldots, x^{k-1}\}$.

④ $\mathbb{F}_p \subseteq \mathbb{F}_{p^k}$

↳ prime subfield of $\mathbb{F}_{p^k}$

⑤ $\mathbb{F}_{p^k}$ is called a degree k extension of $\mathbb{F}_p$.

⑥ $\forall \alpha \in \mathbb{F}_{p^k}$, $\underbrace{p \cdot \alpha}_{\alpha + \cdots + \alpha \text{ } (p \text{ times})} = 0$.

} same is true in $\mathbb{F}_p[x]$ as well

Thus, $\mathbb{F}_{p^k}$ has characteristic p.

($\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, ... have char $= 0$)

X X

Arithmetic with polynomials.

$f(x) \in \mathbb{F}[x]$, deg $= d$.

This defines a natural function

$$\text{Eval}_{f(x)} : \mathbb{F} \longrightarrow \mathbb{F}, \quad a \longmapsto f(a).$$

That is, if $f(x) = f_0 + f_1 x + \cdots + f_d x^d$,

then $f(a) = f_0 + f_1 a + \cdots + f_d a^d$.

Q. Does the evaluation of a polynomial tell us anything about the co-efficients.

Ans. Given sufficiently many evaluations and hypothesis on degree, YES.

Special case: $f(x) \in \mathbb{F}[x]$, deg $\leq d$.

$f(a) = 0$ for some $a \in \mathbb{F}$.

What can we say about f?

Lemma $f(x) = 0 \Rightarrow (x - a) \mid f(x)$.

Proof. Use the division lemma to write

$$f(x) = q(x)(x - a) + r(x)$$

with $r(x) = 0$ or $\deg(r(x)) < 1$.

$\therefore r$ is a constant. Putting $x = a$ above gives $r = 0$.

$$\therefore f(x) = q(x)(x - a). \qquad \square$$

Cor. Let $0 \neq f \in \mathbb{F}[x]$, and $d := \deg(f)$.

Then, # zeroes of f in $\mathbb{F} \leq d$.

Proof. Induct on d. $d = 1$ is clear.

$d \geq 2$: If f has no root in $\mathbb{F}$, then done.

Else, it has a root $a \in \mathbb{F}$ and then, we

can write $f(x) = (x-\alpha)\,g(x)$.

Now,

$\{\text{roots of } f(x)\} \subseteq \{a\} \cup \{\text{roots of } g(x)\}$. ∎

$\#\{\text{roots of } g(x)\} \leq d-1$ by ind.

A similar inductive argument also gives the following:

If $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ are distinct roots of $f(x)$, then

$(x-\alpha_1)\cdots(x-\alpha_n) \mid f(x)$.

Theorem. (Interpolation)

$\forall \alpha_1, \ldots, \alpha_t \in \mathbb{F}$ distinct and

$\forall \beta_1, \ldots, \beta_t \in \mathbb{F}$ (not nec. distinct),

$\exists f(x) \in \mathbb{F}[x]$ s.t. $f(\alpha_i) = \beta_i$ for $i = 1, \ldots, t$, and

$\deg(f(x)) \leq t-1$.

Moreover, such an f is unique.

Proof is part of HW-1.

Thus, the above gives a way of going from evaluations to coefficients.

Question to think about:

Q. $f = f_0 + f_1 x + \cdots + f_d x^d$ and $\alpha \in \mathbb{F}$.

Design algo to compute $f(\alpha) \in \mathbb{F}$.

(How costly is it?)

Lecture 4 (10-08-2021)

10 August 2021 17:32

Input: $f, g \in \mathbb{F}[x]$, $f = \sum_{i=0}^{d} f_i x^i$, $g = \sum_{i=0}^{d} g_i x^i$.

Goal: Output $f \cdot g$, as a list of co-efficients.

Naive algorithm:

$$(fg)_i = \sum_{j=0}^{i} f_j g_{j-i}. \quad \text{—— (1)}$$

Algo 1: For $i = 0$ to $2d$,
use (1) to compute deg i co-efficient in fg.

Running time: for i^{th} iteration, we have i mult. and $i+1$ additions. $\sim 2i$

$\therefore$ Total $= \sum_{i=1}^{2d} (2i) = O(d^2)$

Input size $= O(d)$

quadratic complexity

(Note: To compute running time, we will just count field operations for "algebraic algorithms".)

TODAY: Algorithm for polynomial multiplication in time $O(d \log d)$

① Cooley-Tukey - $O(d \log d)$ over $\mathbb{C}$, some more fields.

② Schonage-Strassen → $O(d \log d \log\log d)$ any field.

⋮

③ 2019: [Harvey, ——] → $O(d \log d \, 2^{\log^* d})$ → (all finite fields?)

④ last month:
Ben-Sasson, Carmon, → $O(d \log d)$ → all finite field.

Kopparty, Levit (2021)

Let us look at ①.

Key idea. Two ways of representing polynomials:

① Coefficients representation → use of coeffs
$f \equiv (f_0, f_1, \ldots, f_d)$.

② Evaluation representation → evaluation on $d+1$ distinct points.

$\alpha_1, \ldots, \alpha_{d+1}$: $f \equiv (f(\alpha_1), \ldots, f(\alpha_{d+1}))$

$g \equiv (g(\alpha_1), \ldots, g(\alpha_{d+1}))$.

Obs : The naive algo in coeff rep runs in $O(d^2)$ time.

Eval representation?

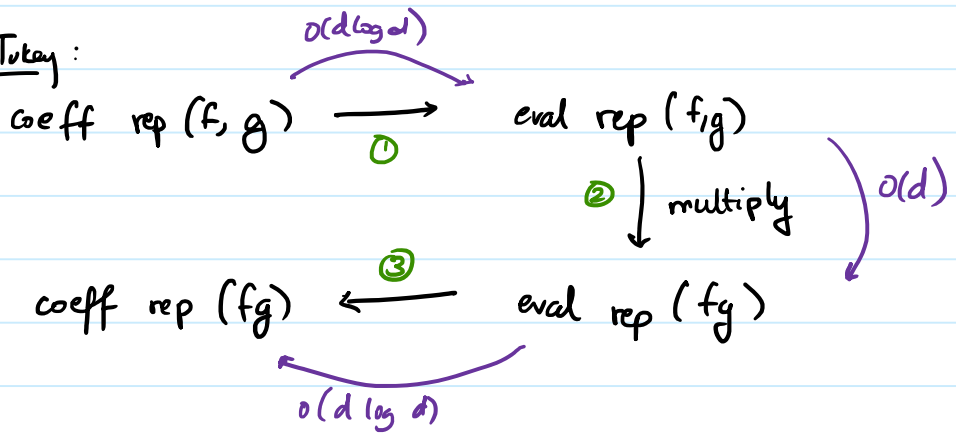
Eval. repr. of product is simply

$((fg)(\alpha_1), \ldots, (fg)(\alpha_{n})) \equiv (f(\alpha_1)g(\alpha_1), \ldots, f(\alpha_n)g(\alpha_n))$.

Can do in $O(d)$.

Moral : Multiplication looks easy in eval. repr.

Cooley - Tukey :



Note : $\deg(fg) \simeq 2d$.

Thus, we need $(2d+1)$ evaluations.

Hence, we must start with $2d+1$ evaluations even though $\deg(f), \deg(d) \leq d$.

Fast Multipoint Evaluation

First, we see how to do ① fast.

$$f = f_0 + f_1 x + \cdots + f_d x^d.$$

Want: evaluate f on $\sim 2d$ distinct points FAST.

$\alpha_1, \ldots, \alpha_t \longrightarrow$ distinct. $\quad (t = 2d+1)$

↳ freedom to choose

$$\begin{matrix} \\ i^{th} \text{ row} \rightarrow \end{matrix} \begin{bmatrix} 1 & \alpha_i & \cdots & \alpha_i^d \end{bmatrix}_{t \times (d+1)} \begin{bmatrix} f_0 \\ \vdots \\ f_d \end{bmatrix} = \begin{bmatrix} f(\alpha_1) \\ \vdots \\ f(\alpha_t) \end{bmatrix}$$

↳ even writing down the matrix will take too much time since $t(d+1) \sim 2d^2 \gg$ deg$f$.

CRUCIAL POINT: if $\alpha_1, \ldots, \alpha_t$ are chosen carefully, can compute $f(\alpha_1), \ldots, f(\alpha_t)$ in $O(d \log d)$ time.

Choice of $\alpha_1, \ldots, \alpha_t$. [Field plays a role.]

Primitive m^{th} root of 1:

$\omega \in \mathbb{C}$ is said to be a primitive m^{th} root of 1 if

① $\omega^m = 1$,

② $\omega^k \neq 1 \quad \forall\, k \in \{1, \ldots, m-1\}$.

Example. When working over $\mathbb{C}$, we can take

$\omega = \exp\left(\frac{2\pi i}{m}\right)$ as a primitive m^{th} root of 1.

Properties of primitive m^{th} root:

ω is a primitive m^{th} root of 1

① Then: $1, \omega, \ldots, \omega^{m-1}$ are all the m^{th} roots of 1.

② $1 + \omega + \cdots + \omega^{m-1} = 0$.

③ If m is even, then ω^2 is a primitive $(m/2)^{th}$ root of 1.

Pf. ① $(\omega^t)^{M/2} = 1$.

② If $1 \le k \le \frac{m}{2} - 1$, then $(\omega^2)^k = \omega^{2k} \neq 1$ since $2 \le 2k \le m-2$.

Back to algorithm:

ω = primitive t^{th} root of 1.

Points of evaluation = $\{1, \omega, \ldots, \omega^{t-1}\}$.

Assume: t is a power of 2.

That is, we come up with an efficient way of computing at t points when $t = 2^n$.

We can do this since we can always pick the next power of 2. Moreover, this power is at most twice the given number. (That is, $\exists n \exists d$ s.t. $n \le 2^d \le 2n$.)

Recursive algorithm: Evaluate a deg $t/2$ poly on all t^{th} roots of 1

↓ reduction

$2 \times$ [Evaluate a deg $t/4$ poly on all $(t/2)^{th}$ roots of 1]

$+ O(t)$ processing time.

Running time: $R(t) \leq 2R(t/2) + O(t)$

$\Rightarrow O(t \log t)$.

$$f(x) = f_0 + f_1 x + \cdots + f_d x^d$$
$$= (f_0 + f_2 x^2 + \cdots) + (f_1 x + f_3 x^3 + \cdots).$$

$$f_{even}(y) := f_0 + f_2 y + f_4 y^2 + \cdots + f_{2i} y^i + \cdots$$
$$f_{odd}(y) := f_1 + f_3 y + f_5 y^2 + \cdots + f_{2i+1} y^i + \cdots$$

Note. $\deg(f_{even}) \leq d/2$, $\deg(f_{odd}) \leq d/2$.

Also, $f(x) = f_{even}(x^2) + x f_{odd}(x^2)$.

$$f(\omega^i) = f_{even}(\omega^{2i}) + \omega^i f_{odd}(\omega^{2i}).$$

Since ω is a primitive t^{th} root, we know that ω^2 is a primitive $(t/2)^{th}$ root.

Suffices to have evaluations of f_{even}, f_{odd} (deg $\leq t/2$) on $\{(\omega^2)^i : i \in \{0, \ldots t/2\}\}$.

Thus, we have done ① in $O(d \log d)$ time.

Now, we need to do ③:

Eval rep (deg $2d$ poly, $2d+1$ points) $\longrightarrow$ Coefficients.

Linear system:

Earlier

known $\rightarrow$ unknown $\rightarrow$

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^d \\ & & \vdots & \end{bmatrix} \begin{bmatrix} f_0 \\ \vdots \end{bmatrix} = \begin{bmatrix} f(1) \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} 1 & \omega & \cdots & \omega^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{t-1} & & (\omega^{t-1})^d \end{bmatrix} \begin{pmatrix} \\ \vdots \\ f_d \end{pmatrix} = \begin{bmatrix} \vdots \\ f(\omega^{t-1}) \end{bmatrix}$$

Now: $h = fg$

$h \rightarrow h(1), h(\omega), \ldots, h(\omega^{t-1})$.

$\deg(h) \leq t-1$.

want: coeff of h $\quad h \equiv h_0 + h_1 x + \cdots + h_t x^{t-1}$.

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{t-1} & \cdots & \omega^{(t-1)(t-1)} \end{bmatrix}_{t \times t} \begin{bmatrix} h_0 \\ \vdots \\ h_t \end{bmatrix} = \begin{bmatrix} h(1) \\ \vdots \\ h(\omega^{t-1}) \end{bmatrix}$$

V_ω — known

V_ω is full rank. Then,

$$\text{coeff}(h) = (V_\omega)^{-1} \cdot \text{eval}(h).$$

Define $\quad N_\omega := \frac{1}{t} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & (\omega^{-1}) & (\omega^{-1})^2 & \cdots & (\omega^{-1})^{t-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (\omega^{-1})^{t-1} & (\omega^{-1})^2 & & (\omega^{-1})^{(t-1)^2} \end{bmatrix}$.

Claim. $N_\omega = V_\omega^{-1}$.

Note that by Claim, we are done since we are in the same position as in (1).

($\because \omega^{-1}$ is also a primitive t^{th} root of 1.)

Proof. Simply compute. If $\omega^i \neq 1$, then $\sum_{i=0}^{t} (\omega^i)^i = 0$.

Rmk: Simply compute. If $\omega \neq 1, t$ then $\sum_{i=0}^{t} (\omega^i)^i = 0$.

This algorithm is called the FFT (Fast Fourier Transform).

- Where did we use the field?
  1. Existence of primitive roots.
  2. Existence of $\frac{1}{t}$. (t should not be divisible by $\mathrm{char}(\mathbb{F})$.)

Next: 1. Division with remainder.
2. Fast multipoint evaluation for any $\alpha_1, \ldots, \alpha_t$.
3. (May be) multipoint evaluation for multivariate polynomials.

Lecture 5 (13-08-2021)

13 August 2021 17:35

Input: $f, g \in \mathbb{F}[x]$, $d \geq \deg(f) \geq \deg(g)$ with $g \neq 0$.

Output: $q, r \in \mathbb{F}[x]$ s.t. $f = g \cdot q + r$, $\deg(r) < \deg(g)$.

Naive: Long division
- Running time: $O(d^2)$

Today: An algorithm which runs in $O(d \log^2 d)$.

More precisely: $O(M(d) \log(d))$, where $M(d)$ is time complexity of multiplying two polynomials in $\mathbb{F}[x]$.

we had seen $M(d) = O(d \log(d))$ for "nice" fields

New ideas:
1) Reversal of a polynomial
2) Newton iteration / Hensel lifting ← very useful

Reversal

$$f = f_0 + f_1 x + \cdots + f_d x^d \quad ; \quad f_d \neq 0.$$

$$\begin{aligned} \mathrm{Rev}(f) &= f_d + f_{d-1} x + \cdots + f_1 x^{d-1} + f_0 x^d \\ &= x^d \cdot f(1/x) \\ &= x^{\deg(f)} \cdot f(1/x). \end{aligned}$$

↳ note that not linear, i.e., $\mathrm{Rev}(f+g) = \mathrm{Rev}(f) + \mathrm{Rev}(g)$ MAY NOT hold!

Reversal and division

$$f = g \cdot q + r$$

(Assume $r \neq 0$ for the time being.)

$\Rightarrow \mathrm{Rev}(f) = x^d f(1/x)$

$= x^d \left(g(1/x)\, q(1/x) + r(1/x) \right)$

$= x^d g(1/x)\, q(1/x) + x^d r(1/x)$

① $\mathrm{rev}(g) = x^{\deg(g)} g(1/x)$, $\mathrm{rev}(q) = x^{\deg(q)} q(1/x)$, $\mathrm{rev}(r) = x^{\deg(r)} r(1/x)$.

② $\deg(g) + \deg(q) = \deg(f) = d.$

$\therefore x^d g(1/x) q(1/x) = \mathrm{rev}(g)\,\mathrm{rev}(q)$

③ $\deg(r) < \deg(g) \le d.$

$\therefore \mathrm{rev}(f) = \mathrm{rev}(g)\,\mathrm{rev}(q) + x^{d-\deg(r)}\,\mathrm{rev}(r)$

Note: $d - \deg(r) > d - \deg(g) = \deg(q).$ ($d - \deg(r) = \ell$)

$\therefore$ $\boxed{\mathrm{rev}(f) = \mathrm{rev}(g)\,\mathrm{rev}(q) + x^{\ell}\,\mathrm{rev}(r).}$ — (2)

$\ell > \deg(q) \geq \deg(\mathrm{rev}(q)).$

Obs. Degree of every monomial in $x^{\ell}.\mathrm{rev}(r)$ is $\geq \ell > \deg(q).$

(2) + Obs. $\Rightarrow$ $\boxed{\mathrm{rev}(f) \equiv \mathrm{rev}(g)\,\mathrm{rev}(q) \mod x^{\ell}.}$ — (3)

(Known: $\mathrm{rev}(f)$, $\mathrm{rev}(g)$)

Plan: 1. Use (3) to solve for $\mathrm{rev}(q) \mod x^{\ell}$.

However, $\ell > \deg(q) \geq \deg(\mathrm{rev}(q)).$

Thus, solving it modulo x^{ℓ} is sufficient.

(That is: $\mathrm{rev}(q) = \mathrm{rev}(q) \mod x^{\ell}.$)

(the canonical remainder of deg $< \ell$)

② $rev(\ ) \longrightarrow q$ (EASY)

③ $r := f - gq$ (EASY)

Then, it reduces to solving (3).

Note: we don't know ℓ but taking $\ell = 1 + \deg(f) - \deg(g)$ works. (Even if $r = 0$)

Want: A fast algorithm for solving (3).

Idea: Suppose we have $h(x) \in \mathbb{F}[x]$ such that

$$h(x) \cdot rev(g) \equiv 1 \mod x^\ell.$$

That is, $h(x)$ = inverse of $rev(g)$ in $\mathbb{F}[x]/\langle x^\ell \rangle$.

Multiply (3) with $h(x)$ to get:

Then, $rev(q) \equiv h \cdot rev(f) \mod x^\ell$.

Reduces to: Inverse computation for $rev(g)$ in $\mathbb{F}[x]/\langle x^\ell \rangle$.

Existence of such an h:

$g = g_0 + \cdots + g_m x^m$ with $g_m \neq 0$. ($m = \deg(g)$)

$rev(g) = g_0 x^m + \cdots + g_{m-1} x + g_m$.

Note: $g_m \neq 0$ and thus $x \nmid rev(g)$.

In turn, $\gcd(rev(g), x^\ell) = 1$ and so, such an h exists.

Proof. Let $h(x) = h_0 + h_1 x + \cdots + h_n x^n$ be an arbitrary poly.

Want: $h_0, \ldots, h_n$ s.t.

$$(h_0 + h_1 x + \cdots + h_n x^n)(g_m + g_{m-1} x + \cdots + g_0 x^m) \equiv 1 \mod x^\ell.$$

$\Leftrightarrow \forall i < \ell$, coeff of x^i in $h \cdot rev(g)$

$= \text{coeff of } x^i \text{ in } 1$

$= \begin{cases} 0 ; & i \neq 0, \\ 1 ; & i = 0. \end{cases}$

$\Leftrightarrow$ ① coeff of x^0 in $h_{rev}(g) = 1$,
② coeff of x^i in $h \cdot rev(g) = 0 \quad \forall\, 1 \le i \le l$.

$h_0 \cdot g_m = 1$
$h_0 g_{m-1} + h_1 g_m = 0$
$\vdots$

} linear system in h_i

WLOG $n = l-1$.

rewrite

$$\begin{bmatrix} g_m & & 0 \\ & g_m & \\ * & & g_m \end{bmatrix}_{l \times l} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{l-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Since $g_m \neq 0$, the left matrix is full-rank.
Thus, unique sol^n exists.

Solve the linear system to get h.
↳ Still quadratic :/

Want : fast!

Enter: Newton Iteration / Hensel lifting.

Want: $h \cdot rev(g) \equiv 1 \mod x^l$.

Notation : $b := rev(g)$
$= b_0 + \cdots + b_m x^m$ with $b_0 \neq 0$.
($b_i = g_{m-i} \quad \forall i \in [0, m]$)

Want: $h \in \mathbb{F}[x]$ s.t. $h \cdot b \equiv 1 \mod x^l$.

Induction: At the end of i^{th} iteration, we have,

Induction: At the end of i^{th} iteration, we have,

$H_i \in \mathbb{F}[x]$ s.t. $H_i \cdot b \equiv 1 \mod x^{2^i}$

and $\deg(H_i) < 2^i$.

Base case: 0

Stop when: $2^i > \ell$; $i = \lceil \log \ell \rceil$

(If we do this fast, we are done.)

Runtime: $\log \ell \leq O(\log d)$ iterations.

(As we shall see: constant # of polynomial multiplications $M(d)$ in each iteration.)

Base: $H_0 b = 1 \mod x$

Take $H_0 = b_0^{-1}$.

Induction: Suppose we have H_k with $\deg < 2^k$ s.t.

$H_k b \equiv 1 \mod x^{2^k}$.

Will find an $A(x)$, $\deg(A) < 2^k$ and set

$H_{k+1} := H_k + x^{2^k} \cdot A$.

Want: $(H_k + x^{2^k} \cdot A) b \equiv 1 \mod x^{2^{k+1}}$

Sketch: $b = b_0 + b_1 x + \cdots + b_m x^m$

$= (b_0 + \cdots + b_{2^k - 1} x^{2^k - 1}) + x^{2^k} (b_{2^k} + \cdots + b_m x^{m - 2^k})$.

$= B_0 + x^{2^k} B_1$

Want: A s.t.

$(H_k + x^{2^k} A)(B_0 + x^{2^k} B_1) \equiv 1 \mod x^{2^{k+1}}$.

$$(H_k + x^{2^k} A)(B_0 + x^{2^k} B_1) \equiv 1 \bmod x^{2^{k+1}}.$$

$$\Leftrightarrow H_k B_0 + x^{2^k}(AB_0 + H_k B_1) + x^{2^{k+1}} AB_1 \equiv 1 \bmod x^{2^{k+1}}$$

($x^{2^{k+1}} \to 0$)

$$\Leftrightarrow H_k B_0 + x^{2^k}(AB_0 + H_k B_1) \equiv 1 \bmod x^{2^{k+1}}.$$

Induction hyp: $H_k b \equiv 1 \bmod x^{2^k}$

$\Leftrightarrow \exists C \in \mathbb{F}[X]$ s.t. $H_k B_0 = 1 + C \cdot x^{2^k}$

$$1 + x^{2^k}(C + AB_0 + H_k B_1) \equiv 1 \bmod x^{2^{k+1}}$$

$$\Leftrightarrow C + AB_0 + H_k B_1 \equiv 0 \bmod x^{2^k}$$

$$\Leftrightarrow AB_0 \equiv -(C + H_k B_1) \bmod x^{2^k}$$

note $H_k B_0 \equiv 1 \bmod x^{2^k}$

$$\Leftrightarrow A \equiv -H_k(C + H_k B_0) \bmod x^{2^k}$$

Remains: Getting C. $x^{2^k} C = (H_k B_0 - 1)$.

$$H_{k+1} := H_k + x^{2^k} A = H_k - H_k C x^{2^k} - H_k^2 B_1 x^{2^k}$$

$$= H_k - H_k(H_k B_0 - 1) - H_k^2 B_1 x^{2^k}. \quad \square$$

Lecture 6 (17-08-2021)

17 August 2021 17:32

Univariate multipoint evaluation.

Input: $f(x) \in \mathbb{F}[x]$, $\deg = d$. $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$.

Output: $f(\alpha_1), \ldots, f(\alpha_n)$.

Naive. Iteratively (from $i = 1$ to n), evaluate $f(\alpha_i)$.

$f(x) = f_0 + f_1 x + \cdots + f_d x^d$.

Time complexity : $O(nd)$.

↳ Calculating $f(\alpha)$.

↳ Digression: A#1 ① Compute $\alpha^0, \alpha^1, \ldots, \alpha^d$ → d products

② Compute $f_0\alpha^0, \ldots, f_d\alpha^d$ → d products

③ Sum them up → $d+1$ sums

A#2 Horner's Method.

$$f(x) = f_0 + x(f_1 + x(f_2 + x(f_3 + \cdots + f_d))). \quad \text{—(1)}$$

Work with (1) (instead of monomial by monomial).

Here, we use $\sim d$ products and $\sim d+1$ summations.

Today:

Thm There is a deterministic algo. for univariate multipoint evaluations, which runs in time

$$O((n+d)\log^3(nd)).$$

(If $\mathbb{F}$ is "FFT friendly".)

(If $\mathbb{F}$ is "FFT friendly".)

↳ $O((n+d)\log^3 nd \log\log nd)$ elsewise

Open question: No simpler algorithm for multivariates.

Partial results: ① Nearly linear time algo over finite fields, if # vars is small.

- ↳ does weird bit arithmetic
- ↳ not algebraic, in particular
- ↳ Kedlaya-Umans (2008)

② # vars $\leq d^{O(1)}$, char($\mathbb{F}$) small $d^{O(1)}$

Nearly linear time algebraic algorithm.

Remainder-Tree based algorithm (Borodin et al. '70s)

I/P. f; $\alpha_1, \ldots, \alpha_n$.

① $$g_1(x) := \prod_{i=1}^{\lfloor n/2 \rfloor} (x - \alpha_i),$$

$$g_2(x) := \prod_{i=\lfloor n/2 \rfloor + 1}^{n} (x - \alpha_i).$$

② $$f_1 := f(x) \mod g_1(x)$$
$$f_2 := f(x) \mod g_2(x).$$

③ Recursion
- ① Eval f_1 on $\alpha_1, \ldots, \alpha_{\lfloor n/2 \rfloor}$,
- ② Eval f_2 on $\alpha_{\lfloor n/2 \rfloor + 1}, \ldots, \alpha_n$.

Return the values. END.

Proof (of correctness).

Proof (of correctness)

TS: ① $f_1(\alpha_i) = f(\alpha_i) \quad \forall i \in \{1, \ldots, \lfloor n/2 \rfloor\}$

② $f_2(\alpha_i) = f(\alpha_i) \quad \forall i \in \{\lfloor n/2 \rfloor + 1, \ldots, n\}$.

We prove ①. By $n/2$, we shall mean $\lfloor n/2 \rfloor$.

$$g_1 = \prod_{i=1}^{n/2} (x - \alpha_i).$$

$$f_1 = f \bmod g_1.$$

$$\Rightarrow \exists q_1 \in \mathbb{F}[x] \text{ s.t.}$$

$$f = q_1 g_1 + f_1.$$

$$\therefore f(x) = q_1 \cdot \prod_{i=1}^{n/2} (x - \alpha_i) + f_1.$$

Plug $x = \alpha_i$ for $i \in \{1, \ldots, n/2\}$ and conclude. ∎

Time complexity: n, d ; $m := \max\{n, d\}$

$$T(m, m) \leq 2T(m/2, m/2) + \Theta(\text{steps } 1 \& 2)$$

↳ recurse

Step 2: Division with remainder (Assume nice field.)

$O(m \log^2 m)$

Step 1: $g_1 = (x - \alpha_1) \cdots (x - \alpha_{n/2})$.

Naively: $n/2$ multiplications of deg $\leq n/2$ each.

$= \Theta(n^2 \log n)$ → Too long!

(Even worse than original $\Theta(nd)$.)

But we can do multiplication itself recursively.

Comes down to $R(n) \leq 2R(n/2) + c \cdot n/2 \log n$.

Thus, $R(n) = O(n \log^2 n)$.

Plug back in to get

$$T(m, m) \leq 2T(m, m) + O(m \log^2 m).$$

Thus, $T(m, m) \leq O(m \log^3 m)$.

(If $\mathbb{F}$ is not FFT friendly, then we just have an extra $\log\log m$ factor.)

Exercise I/P : $b, a_1, \ldots, a_n \in \mathbb{N}$.

O/P : $b \bmod a_1, \ldots, b \bmod a_n$

Design a fast algorithm for this. [count bit operations]

Some other similar algorithms:

1. Fast interpolation. → maybe in homework
2. Fast GCD computation. → Project topics
3. Fast CRT.

Next up:

1. Algorithm for GCD, Extended Euclid's Algorithm,
2. Chinese Remainder Theorem,
3. Resultants and Connections to GCD,
4. Gauss' Lemma.

Note: many of these 'fast' poly arithmetic algos also work over rings (under some mild conditions).

H/W: Check this + figure out missing details.

GCD computation.

I/P: $f, g \in \mathbb{F}[x]$, $\deg \leq d$, (Not both zero.

O/P: $\gcd(f, g)$.

Euclid's algorithm:

1) If $g = 0$, output f.
2) Else, $q, r \to$ quotient, remainder of $f \div g$.
   Output $\gcd(g, r)$.

Runtime (naively): Analyse two steps at a time.

$$\gcd(\underset{\leq d}{f}, \underset{\leq d}{g}) \longrightarrow \gcd(\underset{\leq d}{g}, \underset{\leq d-1}{r}) \longrightarrow \gcd(\underset{\leq d-1}{r}, \underset{\leq d-2}{r-1})$$

$$T(d) \leq T(d-1) + O(d \log^2 d)$$

$\mathbb{F}_2$ (FFT friendly)

$$\Rightarrow T(d) \leq \theta(d^2 \log^2 d)$$

→ Faster algorithms are known.

Extended GCD

Recall: $\gcd(f, g) = a \cdot f + b \cdot g$ for some $a, b \in \mathbb{F}[x]$.

Q. Want an algo to compute a, b given f, g

Exercise: Extend Euclid's algorithm to solve this question.
(That is called Extended Euclid's Algorithm.) (EEA)

(That is called Extended Euclid's Algorithm (EEA).)

Lemma Let $f, g \in \mathbb{F}[x]$ be nonconstant. $h := \gcd(f, g)$. Then, $\exists\, a, b \in \mathbb{F}[x]$ s.t.

① $h = af + bg$,

② $\deg(a) < \deg(g)$ and $\deg(b) \geq \deg(f)$,

③ there is an efficient algorithm to compute a, b given f, g.

Proof. Take any a, b s.t. $h = af + bg$.

(Can use EEA to get, for example.)

Apply division lemma to get

$$a = q \cdot g + a^*.$$

$\deg(a^*) < \deg(g)$.

$$\Rightarrow \quad h = (qg + a^*) f + bg$$

$$= \underline{a^* f} + g(b + qf)$$

satisfies degree condition

Claim. $\deg(b + qf) < \deg(f)$.

Note that proving this claim finishes the proof of ② & ③.

Proof. We have

$$g \cdot (b + qf) = \underbrace{h - a^* f}_{\deg \leq \deg(f) + \deg(g) - 1}.$$

Thus, $\deg(b + qf) \leq \deg(f) - 1$. ∎

Lecture 7 (20-08-2021)

20 August 2021 17:31

Chinese Remainder Theorem

Theorem. $f_1, \dots, f_k \in \mathbb{F}[x]$ are pairwise coprime, i.e., $\forall i \neq j : \gcd(f_i, f_j) = 1$.
Then, $\forall a_1, \dots, a_k \in \mathbb{F}[x]$ $\exists!$ $g \in \mathbb{F}[x]$ satisfying:

(i) $g \equiv a_i \mod f_i \quad \forall i \in \{1, \dots, k\}$, and

(ii) $\deg(g) < \sum_{i=1}^{k} \deg(f_i)$.

Proof. ① Existence of a solution of 'low' degree.

② Uniqueness

We prove it for $k = 2$. The general case is left as an exercise.

Want to find g s.t.

$$g \equiv a_1 \mod f_2,$$
$$g \equiv a_2 \mod f_1.$$

Since $\gcd(f_1, f_2) = 1$, the EEA gives us u and v s.t.

$$uf_1 + vf_2 = 1.$$

Thus, $a_1(uf_1 + vf_2) = a_1$, and

$a_2(uf_1 + vf_2) = a_2$.

Then, $g = a_1 v f_2 + a_2 u f_1$ satisfies the modular congruences.

Replace g with $g \mod f_1 f_2$.

Still satisfies the congruences and

$$\deg(g) < \deg(f_1) + \deg(f_2).$$

Uniqueness. If g' is another such, then

f_1 and f_2 divide $g - g'$.

Uniqueness: g ... f_1 and f_2 divide $g - g'$.

$\therefore f_1 f_2 \mid (g - g')$. ($\because \gcd(f_1, f_2) = 1$)

By degree hypothesis, $g = g'$. ∎

Applications we see in this course:

① Multivariate multipoint evaluation (Kedlaya - Umans)

② Primality testing (Agrawal - Biswas, Agrawal - Kayal - Saxena)

Applications seen so far (in disguise):

(In HW) ① Interpolation: $(a_1, b_1), \ldots, (a_n, b_n)$, $a_i \neq a_j$.

Wanted: $f(x)$ of deg $< n$ s.t. $f(a_i) = b_i$.

Note: $f(a_i) = f(x) \mod (x - a_i)$.

Thus, CRT gives existence and uniqueness of such an f.

(In HW) ② The last question of HW as well.

Resultant

$f, g \in \mathbb{F}[x]$; $\deg(f) = m \geq 1$, $\deg(g) = n \geq 1$.

$I(f, g) = \{uf + vg : u, v \in \mathbb{F}[x]\}$.

(polynomials of deg $< n$)

$\Gamma_{f,g} : \mathbb{F}_{<n}[x] \times \mathbb{F}_{<m}[x] \longrightarrow \mathbb{F}_{<m+n}[x]$.

$\Gamma_{f,g}(u, v) = uf + vg$.

Observations:

Observations:

① $\Gamma_{f,g}$ is a linear map over $\mathbb{F}$.

② $\dim_{\mathbb{F}}(\text{domain}) = m+n = \dim_{\mathbb{F}}(\text{codomain})$.

Q. What is the kernel?
When is the kernel trivial, i.e., when is $\Gamma_{f,g}$ invertible?

Fix f, g and write $\Gamma_{f,g}$ as Γ for simplicity.

$$\ker(\Gamma) = \{(u, v) : uf + vg = 0\}$$

Example: $\gcd(f, g) = h$, $\deg(h) \geq 1$.
$\rightarrow$ Kernel is non trivial.

$\left(\frac{g}{h}, -\frac{f}{h}\right)$ is in the kernel.

↳ this is a valid input since $\deg(h) \geq 1$.

Thm. $\ker(\Gamma_{f,g})$ is non-trivial $\Leftrightarrow$ $\gcd(f, g) \neq 1$.

Proof. ($\Leftarrow$) done above

($\Rightarrow$) By contrapositive. That is, $\gcd(f, g) = 1 \Rightarrow$ trivial kernel.
Pick $(u, v) \in \ker(\Gamma)$.

Then, $uf = -vg$.

$\therefore f \mid (-v) \Rightarrow v = 0$. ($\deg v < \deg f$)

Similarly $u = 0$. ∎

Matrix corresponding to Γ:

Pick the "obvious" bases. $u = \sum_{i=0}^{n-1} u_i x^i$,

$m-1$

Pick the obvious bases. $u = \sum_{i=0}^{n-1} u_i x^i$,

$$v = \sum_{j=0}^{m-1} v_j x^j.$$

$$uf + vg = h = \sum_{\ell=0}^{m+n-1} h_\ell x^\ell.$$

$h_0 = f_0 u_0 + g_0 v_0$

$h_1 = u_0 f_1 + u_1 f_0 + v_0 g_1 + v_1 g_0$

$$\begin{bmatrix} f_0 & \cdots & 0 & g_0 & 0 & \cdots & 0 \\ f_1 & f_0 & \cdots & 0 & g_1 & g_0 & \cdots \\ f_2 & f_1 & f_0 & & g_2 & g_1 & g_0 \\ f_3 & f_2 & f_1 & \ddots & & g_2 & g_1 \\ \vdots & \vdots & \vdots & & & & \ddots \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \\ v_0 \\ \vdots \\ v_{m-1} \end{bmatrix} = \begin{bmatrix} h_0 \\ \vdots \\ h_{m+n} \end{bmatrix}.$$

(n columns for the u's, m columns for the v's)

$(m+n) \times (m+n)$

Exercise: Write Γ when $\deg(f) = 2$, $\deg(g) = 3$.

This matrix is called the Sylvester matrix $S(f,g)$.

$$\text{Resultant}(f, g) := \det(S(f,g)).$$

Thus, we have:

Thm $\text{GCD}(f, g) = 1 \iff \ker(\Gamma) = 0 \iff \text{resultant}(f,g) \neq 0.$

Note that the resultant is simply a polynomial in coefficients of f and g!

A useful property:

Lemma $R(f, g) \in I(f, g)$

Proof. $R: \det \begin{bmatrix} f_0 & & g_0 & \\ f_1 & f_0 & g_1 & g_0 \\ f_2 & f_1 & g_2 & g_1 \end{bmatrix}$

Proof.

$R = \det \begin{bmatrix} f_1 & f_0 & & g_1 & g_0 & \\ f_2 & f_1 & \ddots & g_2 & g_1 & \ddots \\ \vdots & f_2 & & \vdots & \vdots & \\ & \vdots & & & & \end{bmatrix}$.

View the above in $\mathbb{F}(x) \supseteq \mathbb{F}$.

Apply the following transformation:

$$R_1 \mapsto R_1 + xR_2 + \cdots + x^{m+n-1} R_{m+n}.$$

$$R = \det \begin{bmatrix} f & xf & \cdots & x^{n-1}f & g & xg & \cdots & x^{m-1}g \\ & & & \text{Same} & & & & \end{bmatrix}.$$

Now, expand along first row to get

$$f(\quad) + xf(\quad) + \cdots x^{n-1}f(\quad) + g(\quad) + \cdots x^{m-1}g(\quad).$$

f_u g^v $\blacksquare$

Lecture 8 (24-08-2021)

24 August 2021 17:32

Recall $R(f, g) = \det(S(f, g))$

Sylvester matrix

$R(f, g) \neq 0 \iff GCD(f, g) = 1.$

$R(f, g) = uf + vg$ for some $u, v \in \mathbb{F}[x]$.

Discriminant

Square free polynomial

$f \in \mathbb{F}[x]$ is square free if there does not exist $g \in \mathbb{F}[x]$ s.t. $\deg(g) \geq 1$ and $g^2 \mid f$.

Q. How do we detect square freeness?

For a quadratic polynomial $p(x) = ax^2 + bx + c$, $(a \neq 0)$ we see that $p(x)$ is square free iff $b^2 - 4ac \neq 0$.

In general, the answer is: discriminant.

Defn. (Formal derivatives)

$$\frac{d}{dx} : \mathbb{F}[x] \longrightarrow \mathbb{F}[x],$$

$\forall i > 0$, we have

$$\frac{d}{dx}(x^i) = i \cdot x^{i-1}.$$

For $i = 0$: $\frac{d}{dx}(1) = 0.$

Extend this linearly. This is called the formal derivative.

Check. $\frac{d}{dx}$ satisfies the product rule.

Obs. f is squarefree iff $\gcd(f, \frac{d}{dx} f) = 1$

$\Updownarrow$

$R(f, \frac{d}{dx} f) \neq 0.$

Def. The discriminant of f is defined as

$$D(f) = R(f, \tfrac{d}{dx} f).$$

Example. $f(x) = ax^2 + bx + c,$

$$\frac{d}{dx} f(x) = 2ax + b.$$

$$R(f, \tfrac{d}{dx} f) = \det \begin{bmatrix} c & b & 0 \\ b & 2a & b \\ a & 0 & 2a \end{bmatrix}$$

$$= 4a^2c - b(2ab - ab)$$

$$= 4a^2c - ab^2$$

$$= -a(b^2 - 4ac).$$

Proof of Obs:

- f is not square $\Rightarrow \gcd(f, \frac{d}{dx} f) \neq 1.$

Proof. Let $g^2 \mid f$ where $\deg(g) \geq 1.$

Then, $f = g^2 h$ for some $h \in \mathbb{F}[x].$

$\Rightarrow f' = g^2 h' + 2gg'h.$

Thus, $g \mid f'$ as well. $\therefore g \mid \gcd(f, \frac{d}{dx} f).$

Conversely, suppose $\gcd(f, \frac{df}{dx}) = 1.$

Pick an irreducible (common) factor $p.$

$f = ph$

$\Rightarrow f' = ph' + p'h$

$\Rightarrow p \mid p' h$ ⟩ $\deg(p') < \deg(p)$ and p is irred (assume $p' \neq 0$)

$\Rightarrow p \mid h$

$\Rightarrow p^2 \mid f.$

均

- Vanishing of $\frac{d}{dx} f$.

Let $\mathbb{F} = \mathbb{F}_{p^k}$. Then, $\frac{d}{dx} f(x) = 0$

$\Updownarrow$

$$f(x) = f_0 + f_p x^p + \cdots + f_{kp} x^{kp}.$$

In such a case, define

$$g(x) = (f_0)^{1/p} + (f_p)^{1/p} x + \cdots + (f_{kp})^{1/p} x^k.$$

To show: (i) $g \in \mathbb{F}_{p^k}[x]$, i.e., all those p^{th} roots exist,

(ii) $g^p = f$.

↓

this follows since char $= p$.

Lemma. Let $\mathbb{F} = \mathbb{F}_{p^t}$. Then,

$$\forall \alpha \in \mathbb{F} \quad \exists \beta \in \mathbb{F} \text{ s.t. } \beta^p = \alpha.$$

(Every element has a p^{th} root.)

Proof. Define $\psi : \mathbb{F} \to \mathbb{F}$ by

$\psi(a) = a^p$.

Note that $\psi(a+b) = \psi(a) + \psi(b)$.

In fact, ψ is $\mathbb{F}_p$-linear since $a^p = a \ \forall a \in \bar{\mathbb{F}}_p$.

Also, $\psi(a) = 0 \Rightarrow a^p = 0 \Rightarrow a = 0$.

Thus, $\ker \psi = 0$. Since $\mathbb{F}$ is a f.d.v.s. over $\mathbb{F}_p$, it follows that ψ is onto. ∎

Lemma Gauss' Lemma

Let $f(x) \in \mathbb{Z}[x]$. $\subseteq \mathbb{Q}[x]$

$f(x)$ has a nontrivial factor over $\mathbb{Z}$

iff

$f(x)$ has a nontrivial factor over $\mathbb{Q}$.

(Moreover, given a factorisation over $\mathbb{Q}$ efficiently gives one over $\mathbb{Z}$.)

More generally: We can replace $\mathbb{Z}$ with a UFD R and $\mathbb{Q}$ with $\mathrm{Frac}(R)$.

Examples. ① $\mathbb{Z} \to \mathbb{Q}$

② $\mathbb{F}[x,y] \to \mathbb{F}(x)[y]$

Proof. ($\Downarrow$) is trivial.

($\Uparrow$) Given: $f = gh$ with $\deg g, h \geq 1$, $g, h \in \mathbb{Q}[x]$.

Want: $f = \tilde{g}\tilde{h}$ with $\deg \tilde{g}, \tilde{h} \geq 1$, $\tilde{g}, \tilde{h} \in \mathbb{Z}[x]$.

Lecture 9 (27-08-2021)

27 August 2021 17:31

Multivariate midpoint evaluation (MME)

I/p: $f(x_1, \ldots, x_m) \in \mathbb{F}[x_1, \ldots, x_m]$, $\deg_{x_i}(f) < d$

$a_1, \ldots, a_N \in \mathbb{F}^m$

O/p: $f(a_1), \ldots, f(a_N)$

Seen so far: $m = 1 \rightsquigarrow O(d \cdot \text{poly}\log(d))$ ↳ quasilinear Trivial: $O(d^2)$

Input was: d var, d points $\sim 2d$ field inputs

Solved in '71.

$m = 2$. "Open" until recently. Trivial: $O(d^2 \cdot N)$

2008

Thm. [Kedlaya-Umans '08]

There is an $(\text{input-size})^{1+o(1)}$ time algorithm for multivariate multipoint evaluation if

① $m \leq d^{o(1)}$,

② $\mathbb{F}$ - finite field.

Not algebraic

Algebraic: Another algorithm which works over field of small characteristic.

Open:
- Algorithms for infinite fields (#arithmetic operations)
- Algebraic algorithm over all finite fields
- Large m

- Large m

Simplifying assumption:

$\mathbb{F} = \mathbb{F}_p$ for a prime p (Not too tough to extend to $\mathbb{F}_{p^d}$.)

Input size: $d^m + Nm$

Desirable runtime: $\Theta((d^m + Nm)\,\mathrm{poly}(d, m))$.

Toy (but useful) example:

Set of evaluation points is nice.

- $N = p^m$
- Set of eval points $= \mathbb{F}_p^m$

$m = 2$: $f(x, y) = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} f_{ij}\, x^i y^j \in \mathbb{F}[x, y]$

$= \sum_{j=0}^{d-1} y^j \left(\sum_{i=0}^{d-1} f_{ij}\, x^i \right) \in \mathbb{F}[x][y]$

$= \sum_{j=0}^{d-1} y^j \cdot f_j(x)$, $\quad f_j(x) = \sum_{i=0}^{d-1} f_{ij}\, x^i$.

- Algorithm.
  1. Evaluate $f_j(x)$, $j = 0, \ldots, d-1$ on $\mathbb{F}$.
  2. for each $\alpha \in \mathbb{F}$, evaluate $f(\alpha, y) = \sum_{j=1}^{d-1} f_j(\alpha)\, y^j$ on $\mathbb{F}$.

→ d univariate multipoint eval — deg $d-1$

↳ d univariate multipoint eval — deg $d-1$, p points

↳ p-iter → univariate multipoint eval of deg $d-1$, on p points.

Runtime: $d\,\tilde{O}(d+p) + p\,\tilde{O}(d+p)$

$= \tilde{O}((d+p)^2)$

$\leq \tilde{O}(d^2+p^2)$ | $(d+p)^2 \leq 2(d^2+p^2)$

$\tilde{O}(f(n)) = O(f(n)\,\mathrm{poly}(\log(f(n))))$

Generalisation to $m>2$ is clear.

$$f(x_1,\dots,x_m) = \sum_{i=0}^{d-1} x_m^i\, f_i(x_1,\dots,x_{m-1})$$

Step 1. Eval $f_0,\dots,f_{d-1}$ on $\mathbb{F}^{m-1}$.

Step 2. $\forall a \in \mathbb{F}^{m-1}$, evaluate $f(a,x_m)$ on $\mathbb{F}$.

Runtime: $T(m) \leq d\cdot T(m-1) + p^{m-1}\cdot \tilde{O}(d+p)$

solve ↳ $T(m) \leq \tilde{O}(d^m + p^m)$

High level idea: Want to go from evaluating f on arbitrary sets of points in $\mathbb{F}_p^m$ to something like $\mathbb{F}_{p'}^m$.

Step 1. Go from $\mathbb{F}_p$ to $\mathbb{Z}$. (Think of $\mathbb{F}_p = \{0,1,\dots,p-1\}$.)

$f \in \mathbb{F}_p[\bar{x}] \rightsquigarrow \tilde{f} \in \mathbb{Z}[\bar{x}]$

$a_1,\dots,a_N \in \mathbb{F}_p^N \rightsquigarrow \tilde{a}_1,\dots,\tilde{a}_m \in \mathbb{Z}^m$

get fast algo to do $\tilde{f}(\tilde{a}_1),\dots,\tilde{f}(\tilde{a}_m)$

and then reduce mod p. (Assume ↳ this is efficient.)

Step 2. Pick 'many' small primes $p_1, \ldots, p_t$.

$\in \mathbb{F}_{p_i}[\bar{x}]$

$\forall i \in [t]: \quad Q_i(\bar{x}) := \tilde{f}(\bar{x}) \mod p_i.$

$b_{ij} := \tilde{a}_j \mod p_i, \quad j = 1, \ldots, N.$

$\in \mathbb{F}_{p_i}^m$

Step 3 $\forall i \in [t]$: Solve $MME(Q_i, \mathbb{F}_{p_i}^m)$.

Step 4. $\forall i \in [t]$, $\forall j \in [N]$: read off $Q_i(b_{ij})$.

Step 5. (Recovery)

$\forall i \in [N]$, recover $f(a_j)$ from $\{Q_i(b_{ij}) : i \in [t]\}$.

via CRT

CRT: $p_1, \ldots, p_t$ primes

Then, the map

$\psi: \mathbb{Z}/p_1 \cdots p_t \longrightarrow \mathbb{Z}/p_1 \times \ldots \times \mathbb{Z}/p_t,$

$a \mod p_1 \cdots p_t \mapsto (a \mod p_1, \ldots, a \mod p_t)$

is a bijection.

Correctness

Claim: The algorithm is correct if

$$|\tilde{f}(\tilde{a}_j)| < \prod_{i \in [t]} p_i.$$

$\in \mathbb{Z}$

How many primes do we need?

Well, max out all coeffs and entries to see that

$$|\tilde{f}(\tilde{a}_j)| \leq d^m (p-1)^{md} \leq d^m p^{md}.$$

Suffices to take t large enough so that the product of the smallest t primes is $> d^m p^{md}$.

Clearly, we have $p_1 \cdots p_t > 2^t$ for $t > 1$.

Taking

$$t \sim \log_2 (d^m p^{md})$$
$$\sim m \log d + md \log p$$

does the job.

Note: $p_t \leq O(t \cdot \log t)$

FACT: # primes $< n$ is $\sim \frac{n}{\log n}$. Conclude

Running time:

Step 1. $O(N + d^m)$

Step 2. $\text{poly}(t)$ → getting the primes
$t \tilde{O}(N + d^m)$ → going modulo

Step 3. $t \cdot \tilde{O}(d^m + p_t^m)$ $\quad (p_1 < \cdots < p_t)$

Step 4. $t \cdot O(N + d^m)$

Step 5. $N \cdot \text{Time}(CRT)$

Fact: can do CRT in poly(input-size)

Here, input-size: $t \cdot \log(p_t)$

element of $\mathbb{F}_{p_i}$ has $\leq \log(p_i)$ bits
$\leq \log(p_t)$

Thus, the time is $O(N \cdot \text{poly}(t \log p_t))$.

Step 3 is the slowest. Thus,

Running Time $\leq t^5 \log^5(p_t)\, \tilde{O}(d^m + p_t^m)$

Lecture 10 (31-08-2021)

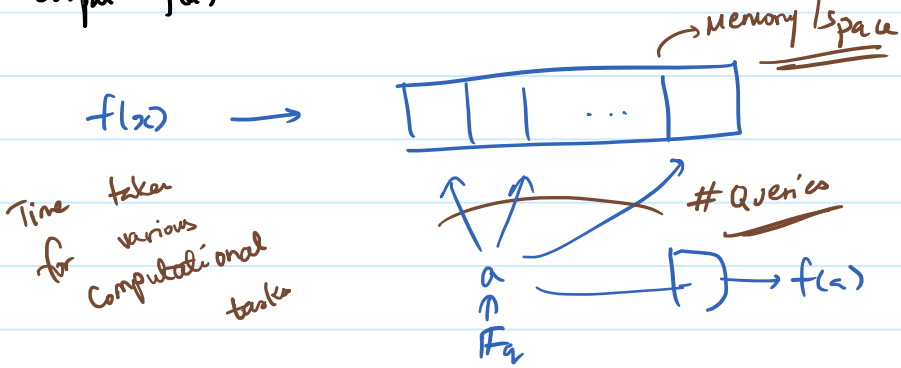
31 August 2021 17:21

Data Structure for polynomial evaluation

$\mathbb{F}_q$, n (think of q as n^3 or so.)

Data. $f(x) \in \mathbb{F}_q[x]$, deg $\leq n$.

Queries. $a \in \mathbb{F}_q$; output $f(a)$.



Think of memory as a collection of field elements.

(Think of each cell as $\log(q)$ bits.)

Want: small space + few queries for every evaluation

Examples.

① Storing coefficients. $f(x) = f_0 + \cdots + f_{n-1}x^{n-1}$.

Memory:

Queries: To find $f(a)$ → Read all cells and compute.

cells : n ; Space = $n \log(q)$

queries: n ; $n \log(q)$ → is this optimal? (we want output of one field element.)

→ optimal (WHY?!)

↖ Feels too much. ↗

② Store evaluations of $f(x)$ on all of $\mathbb{F}_q$.

Memory : | $f(x_1)$ | $f(x_2)$ | ... | $f(x_q)$ |

Query : Read the appropriate cell.

Space : q - cells $q \log q$ bits ↗ $\gg n \log q$ not so good

Query : 1 - cell $\log q$ bits ↘ optimal

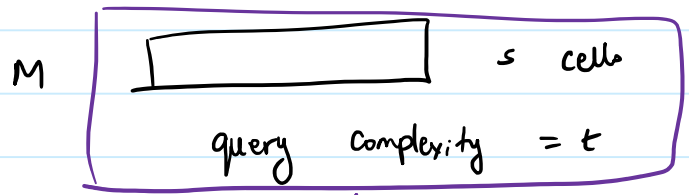
Q. Can we get the "best of both worlds"?
(*) $O(n)$ cells and $O(1)$ queries?

Thm (Larsen '12)
If # cells is $O(n)$, then # queries $\geq \Omega(\log n)$.

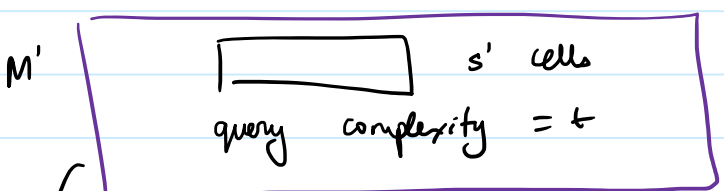
Thus, (*) cannot be achieved.

Thm (Kedlaya - Umans '08) (little o)
There exists a data structure which needs $O(n^{1+o(1)})$ space and $O(\text{poly}(\log n, \log q))$ queries.

Proof (Larsen) Suppose we have a structure with



use this to get



$s' < s$

query complexity = t

Catch: Not all answers are correct.

BUT evaluates $f(a)$ correctly for at least n distinct $a \in \mathbb{F}_q$

Claim: Can recover f correctly from M'.

Thus, in particular, $s' > n$. (We'll also show $s >> s'$ and conclude.)

More concretely:

① Going from M to M'.

$p \in [0, 1]$ — parameter

For every cell in M, delete the content with probability $1-p$. (Independently across cells.)

- Expected # non-empty cells in M' = $p \cdot s$.

- Expected # queries that M' answers correctly?

$\geq p^t \cdot q$.

(Suppose $a \in \mathbb{F}_q$ has $i_1, \ldots, i_t$ in M which are required. Then, it is answered correctly if all t are saved.)

Suppose we have an M' which has the above properties.

If $p^t q \geq n$, then the description of M' should be at least $n \log q$ bits long.

Description of M': cell numbers + content for those cells

$\log s$ bits, $\log q$ bits

length of desc: $ps \log s + ps \log q$

If $p^t q \geq n$, then $ps \log(qs) \geq n \log(q)$.

$q = n^3$ { pick $p \geq n^{2/t}$.

then $p^t q \geq n$ and hence

$$s \geq \frac{n \cdot \log q}{p \cdot \log(qs)} = \frac{n \log(n^3)}{p \log(n^4)} \geq \Omega\left(n \cdot n^{2/t}\right)$$

can replace n^3 with n^c and get similarly.

But we hadn't actually shown that such an M' exists.

Want to show: Existence of M' s.t. # cells $= ps$

correctly answered queries $\sim p^t q$.

Will show: If we pick a random subset of cells in M of size ps, then with prob > 0, it answers $p^t q$ queries correctly.

Random process: Pick $U \subseteq [s]$ of size ps uniformly at random from all subsets of $[s]$ of size ps.

For $a \in \mathbb{F}_q$, define the rv

$$Y_a = \begin{cases} 1 & \text{if } f(a) \text{ can be answered correctly on } U, \\ 0 & \text{else.} \end{cases}$$

Define $Y = \sum_{a \in \mathbb{F}_q} Y_a$.

Will show: $\mathbb{E}[Y] \geq p^t q (1 - \text{tiny})$.

$\Rightarrow \mathbb{P}[Y \geq p^t q (1 - \text{tiny})] > 0$.

$$\mathbb{E}[Y] = \sum_{a \in \mathbb{F}_q} \mathbb{E}[Y_a]$$

$$= \sum_{a \in \mathbb{F}_q} \mathbb{P}[Y_a = 1]$$

to answer $f(a)$, we queried $i_1, \ldots, i_t$ in M.

$$= \sum_{a \in \mathbb{F}_q} \Pr[Y_a = 1]$$

queried in ..., t in M.

$$= \sum_{a \in \mathbb{F}_q} \frac{\binom{s-t}{ps-t}}{\binom{s}{ps}}$$

$$= \sum_{a \in \mathbb{F}_q} \frac{(s-t)!}{(ps-t)!\,(s-ps)!} \cdot \frac{(ps)!\,(s-ps)!}{s!}$$

$$= \sum_{a \in \mathbb{F}_q} \frac{(s-t)!}{s!} \frac{(ps)!}{(ps-t)!}$$

$$= \sum_{a \in \mathbb{F}_q} \frac{(ps)^t}{s^t} \cdot \frac{(1 - 1/ps)(1 - 2/ps) \cdots (1 - \frac{t-1}{ps})}{(1 - 1/s)(1 - 2/s) \cdots (1 - \frac{t-1}{s})}$$

Homework

$$\geq q \cdot p^t \left(1 - \text{tiny}\right)$$

Lecture 11 (03-09-2021)

03 September 2021 17:30

Thm. (Kedlaya Umans) (KU)

There exists a data structure for poly eval for deg n polynomials in $\mathbb{F}_q[x]$ with

$$\text{Memory} : n^{1+o(1)}$$
$$\text{Query complexity} : \text{poly}\log(n, q).$$

(Think of q as poly(n).)

Recall: Multipoint Multivariate Evaluation. (MME) (q = prime)

I/P : $g(x_1, \ldots, x_m) \in \mathbb{F}_q[x]$, $\deg_{x_i}(g) \le d$,
$a_1, \ldots, a_N \in \mathbb{F}_q^m$

O/P : $g(a_1), \ldots, g(a_N)$.

KU: An algo for MME in time $\tilde{O}((N+d^m)\log q)$.

Outline :

① $g \in \mathbb{F}_q[\bar{x}] \longrightarrow \tilde{g} \in \mathbb{Z}[\bar{x}]$

$a_i \in \mathbb{F}_q^m \longrightarrow \tilde{a_i} \in \mathbb{Z}^m$

② Pick primes $p_1 < p_2 < \cdots < p_t$ (smallest) s.t.

$$\prod p_i > d^m q^{dm}.$$

$$t \sim dm \cdot \log(q) + m \log(d).$$

③ $\forall i \in [t]$

$\mathbb{F}_{p_i}[x] \ni h_i = \tilde{g} \mod p_i$

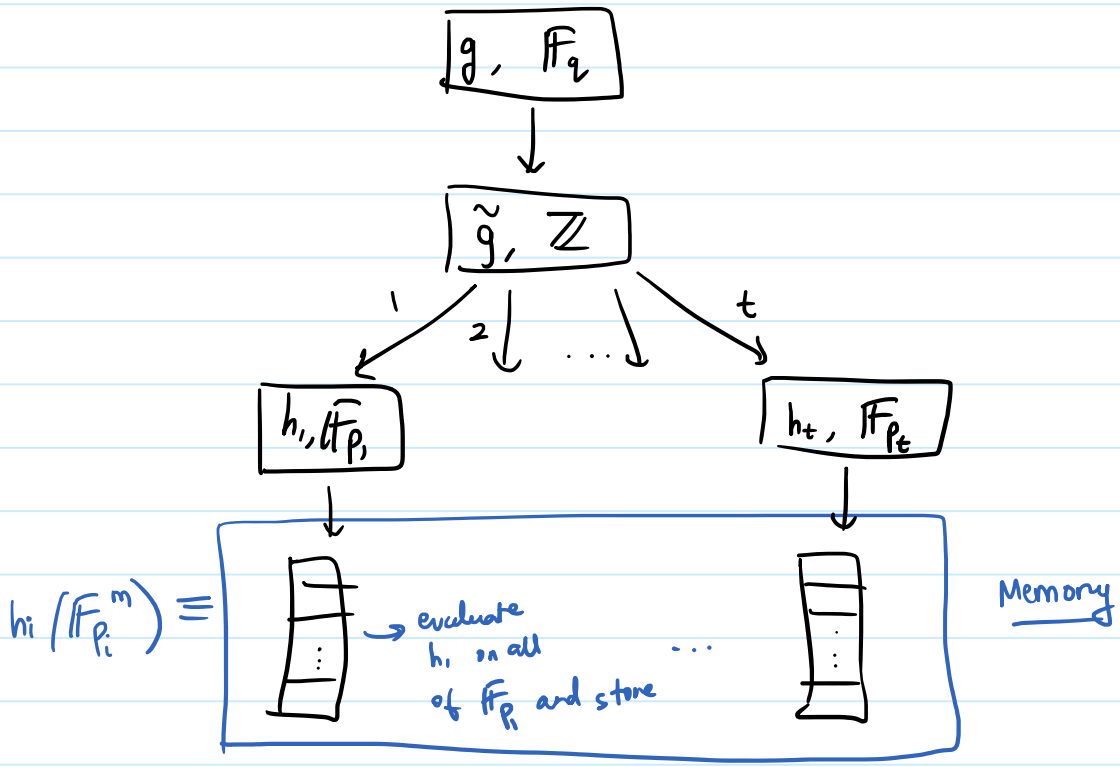
$\mathbb{F}_{p_i}^m \ni \beta_{ij} = \tilde{\alpha}_j \mod p_i$

[④ $\forall i \in t$: eval h_i on $\mathbb{F}_{p_i}^m$
Read off $h_i(\beta_{ij})$.] → Can recurse here!

⑤ Use CRT to recover $g(a_i)$ by solving

$\tilde{g}(\tilde{a}_j) \equiv h_i(\beta_{ij}) \mod p_i$ $\forall i$

Data structure from above:



Query: ① $a_i \in \mathbb{F}_q^m$, $a \longrightarrow \tilde{a} \in \mathbb{Z}^m$,

② $b_i = \tilde{a} \mod p_i$.

③ Read off $h_i(b_i)$ → only step accessing memory!

④ CRT

Resources: For each prime, we look at one cell from $h_i(\mathbb{F}_{p_i}^m)$.

cells used $= t \sim md \log q$

Memory $\leq t \cdot p_t^m$

$= (md \log q) \cdot (md \log q \cdot \log(md \log q))^m$

In principle: Optimal memory: d^m

Query: $O(1)$

Data structure for $m=1$: Aim: $n^{1+o(1)}$, $\text{poly}(\log n)$

cells $\leq n \log q$ ✓

Query $\leq \text{poly}(n, \log q)$ → not so good

Univariate to multivariate reduction:

Claim 1. $\forall g(x)$, $\deg(n)$, $\exists R(y_1, \ldots, y_{\log n})$ s.t.

① $\deg_i(R) \leq 1$

② $R(x, x^{2^1}, x^{2^2}, \ldots, x^{2^{\log n}}) = g(x)$.

Proof. Consider x^i for $i \leq n$.

Then write i in binary and conclude. □

Claim 2 $\forall g(x)$, $\deg(n)$ $\forall d, m \in \mathbb{N}$ s.t. $d^m > n$

$\exists R_{d,m}(y_1, \ldots, y_m)$ s.t.

① $\deg_i(R) \leq d-1$,

② $R(x, x^d, \ldots, x^{d^{m-1}}) = g(x)$.

(Choose deg param = d.
$m \geq \frac{\log n}{\log d}$ suffices.)

Back to data structure.

$g(x) \longrightarrow R_{d,m}(y_1, \ldots, y_m) \xrightarrow{KU}$ Data structure.

(Will need to set d appropriately
Then set $m = \frac{\log n}{\log d}$.)

Set $d = \log^{100} n$.

Analysis:

$$\#\text{primes} \leq md \log q = \frac{\log n}{\log d} \log^{100} n \log q$$
$$= \text{poly}(\log n).$$

$$\text{Memory} \sim (md \log q)^{m+1}$$
$$\sim \left(\frac{\log n}{100 \log\log n} \cdot \log^{100} n \log n\right)^{\frac{\log n}{100 \log\log n}}$$
$$\sim O\left((\log^{102} n)^{\frac{\log n}{100 \log\log n}}\right)$$
$$\sim O\left(2^{\frac{102}{100} \log\log n \cdot \frac{\log n}{\log\log n}}\right)$$
$$= O(n^{1+0.02}).$$

Lemma. (Schwartz – Zippel – DeMello – Lipton – Ore)

Let $\mathbb{F}$ be any field.

Let $g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial of total-degree $\leq d$.

Let $S \subseteq \mathbb{F}$.

If $g \neq 0$, then

$$\#\text{ zeros of } g \text{ on } \underbrace{S \times \cdots \times S}_{n \text{ times}} \leq d \cdot |S|^{n-1}.$$

Remarks.

① Holds for any $\mathbb{F}$ but interesting when S large.

② Interesting when $|S| > d$.

Take $|S| = 100d$, then

$$\#\text{ zeroes} \leq \frac{|S|^n}{100}.$$

③ Note that polynomials can have inf. many roots.

④ $n = 1$ gives the usual result.

Proof. Induction on n.

$n = 1$ is clear. ✓

Assume for $< n$.

$$g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_{n-1}][x_n].$$

$$\sum_{i=0}^{\ell} g_i(x_1, \ldots, x_{n-1}) \, x_n^i,$$

where $\ell = \deg_n(g) \leq d$,

and $g_\ell \neq 0$.

$$\#\text{zeroes}_{S^n}(g) = \sum_{(a_1, \ldots, a_{n-1}) \in S^{n-1}} \#\text{zeroes}_S\left(g(a_1, \ldots, a_{n-1}, x_n)\right)$$

① If $g_\ell(a_1, \ldots, a_{n-1}) \neq 0$, then

$$\#\text{zeroes}_S\left(g(a_1, \ldots, a_{n-1}, x_n)\right) \leq \ell.$$

② If $g_\ell(a_1, \ldots, a_{n-1}) = 0$, then

$$\#\text{zeroes}_S\left(g(a_1, \ldots, a_{n-1}, x_n)\right) \leq |S|.$$

Let $A := \{(a_1, \ldots, a_{n-1}) \in S^{n-1} : g_\ell(a_1, \ldots, a_{n-1}) = 0\}$.

$$\begin{aligned}\#\text{zeroes}_{S^n}(g) &\leq \ell \cdot \left(|S|^{n-1} - |A|\right) + |S| \cdot |A| \\ &= \ell \cdot |S|^{n-1} + \left(|S| - \ell\right)|A|.\end{aligned}$$

By induction, $|A| \leq (d-\ell)\,|S|^{n-2}$.

Note that it suffices to show it when $d < |S|$.

$$\begin{aligned}\therefore \#\text{zeroes}_{S^n}(g) &\leq \ell \cdot |S|^{n-1} + \left(|S| - \ell\right)(d-\ell)\,|S|^{n-2} \\ &= \ell \cdot |S|^{n-1} + (d-\ell)\,|S|^{n-1} - \ell(d-\ell)|S|^{n-2} \\ &\leq d \cdot |S|^{n-1}.\end{aligned}$$

∎

Lecture 12 (07-09-2021)

07 September 2021 17:30

Univariate polynomial factorisation

- Algorithms over finite fields
  - Cantor Zassenhaus poly(deg, log q) Randomised
  - Berlekamp poly(deg, log q, p) Deterministic

① Open: Deterministic poly(deg, log |F|) is not known.

② Design an algo (possibly randomised) which runs in time $\tilde{O}(d \cdot \log(q))$. ~ fastest known $(d \log q)^{1+o(1)}$ (Kedlaya-Umans)

- Algorithms over rationals
  - Lenstra-Lenstra-Lovasz (LLL)

Talk about lattices, shortest vectors on lattices

Baby question: Finding square roots in a finite field. (char ≠ 2)

Input: $a \in \mathbb{F}_q$.

Output: Does there exist $b \in \mathbb{F}_q$ s.t $b^2 = a$?
If yes, output such a b.

Consider: $x^2 - a \in \mathbb{F}_q[x]$

Claim: Suffices to factor $x^2 - a$. (To solve)

($x^2 - a$ factors iff a is a square. The monic linear factors give us the square root.)

Factoring $x^2 - a$

High level: Both CZ and Berlekamp proceed via reduction from

Baby Case : $q \equiv 3 \bmod 4$ — factorisation to GCD.

Let $G(x) := \prod_{i=0}^{q-1} (x - i) = x^q - x$.

Obs 1. If $a \in \mathbb{F}_q^2$, then
$(x^2 - a) \mid G(x)$.
In this case, $GCD(x^2 - a, G(x)) = x^2 - a$.

Strategy : Come up with $H(x)$ s.t. $GCD(x^2 - a, H)$ has deg $= 1$.

$$G(x) = x(x^{q-1} - 1)$$
$$= x\left(x^{\frac{q-1}{2}} - 1\right)\left(x^{\frac{q-1}{2}} + 1\right).$$

Take $H(x) = x^{\frac{q-1}{2}} - 1$.

- Now, we will claim that if a is a nonzero square, then $GCD(x^2 - a, H)$ has deg 1.
Algorithmically, we will be done since the GCD will give a non-trivial factor (and hence, a square root of a).
Moreover this algorithm is efficient :

Step 1. If $a = 0$, output 0.
Step 2. $R(x) = GCD(x^2 - a, H(x))$. → monic GCD
Step 3. If $\deg(R(x)) \neq 1$, then a has no square root.
Else, output the constant term.

→ Need to do this quick. Our method of $GCD(x^2 - a, H(x))$ gives $O(q)$ but not good enough.
Suffices to compute $\left(x^{\frac{q-1}{2}} - 1\right) \bmod (x^2 - a)$ in time poly$(\log q)$.
↳ Multipoint?

Solution: Repeated squaring. Compute $x^{\frac{q-1}{2}} \bmod x^2 - a$ via repeated squaring.

$x \bmod x^2 - a = x$, $x^2 \bmod x^2 - a = a$, $x^4 \bmod x^2 - a = a^2$

$\ldots$ $x^{2^i} \bmod x^2 - a = a^{2^{i-1}}$ $\forall i \geq 1$

↳ poly(log q) operations now

Proof of correctness relies on following claim:

Claim. If $a \in (\mathbb{F}_q^\times)^2$, then $\deg(\gcd(x^2 - a, H(x))) = 1$.

Proof. We have

$$\prod_{b \in \mathbb{F}_q^\times} (x - b) = \underbrace{\left(x^{\frac{q-1}{2}} - 1\right)}_{H(x)} \left(x^{\frac{q-1}{2}} + 1\right).$$

Suffices to show that $H(b) = 0 \Rightarrow H(-b) \neq 0$.

But this is true because $\frac{q-1}{2}$ is odd and $-1 \neq 1$. ☐

The above finishes computation of square root in $\mathbb{F}_q$ with $q \equiv 3 \bmod 4$.

Square roots over arbitrary $\mathbb{F}_q$.

High level: Come with a polynomial $H(x)$ s.t

These ensure that previous algo works

① With High Probability (w.h.p.) ($\sim 1/2$)
$\deg(\gcd(H(x), x^2 - a)) = 1$ if $a \in (\mathbb{F}_q^\times)^2$.

② $\gcd(H(x), x^2 - a)$ is computable in poly(log q) time.

$f(x) = x^2 - a$, $H(x) = x^{\frac{q-1}{2}} - 1$,

roots of $H(x) = (\mathbb{F}_q[x])^{\leq 2}$.

Define $\tilde{f}_{c,d}(x) := (x-c)^2 - a \cdot d^2$ for $c \in \mathbb{F}_q$, $d \in \mathbb{F}_q^\times$.

Note: Suffices to factor $\tilde{f}_{c,d}$ in order to factor f.

Lemma. With probability ≥ 0.48 over $c \sim \mathbb{F}_q$, $d \sim \mathbb{F}_q$.

(Choose c and d uniformly & independently from $\mathbb{F}_q$.)

$\mathrm{GCD}(\tilde{f}_{c,d}, x^{\frac{q-1}{2}} - 1)$ has deg $= 1$.

Proof. Suffices to show

$$\Pr_{c,d \in \mathbb{F}_q}\left(\begin{array}{l}\text{Exactly one of } \{c+db, c-db\} \text{ is} \\ \text{a root of } x^{\frac{q-1}{2}} - 1\end{array}\right) \geq 0.48.$$

Fix $\alpha, \beta \in \mathbb{F}_q$. The system $\begin{cases} c + db = \alpha \\ c - db = \beta \end{cases}$ has exactly one sol^n.

Thus,

$$\Pr_{c,d \sim \mathbb{F}_q}[c+db = \alpha \text{ AND } c-db = \beta] = \frac{1}{q^2}.$$

Thus, $c+db$ and $c-db$ are also independent.

Thus,

$$\Pr_{c,d \in \mathbb{F}_q}\left(\begin{array}{l}\text{Exactly one of } \{c+db, c-db\} \text{ is} \\ \text{a root of } x^{\frac{q-1}{2}} - 1\end{array}\right)$$

$$= 2 \cdot \frac{\left(\frac{q-1}{2}\right)}{q}\frac{\left(\frac{q+1}{2}\right)}{q}$$

$$= \frac{q^2-1}{2q^2} = \frac{1}{2}\left(1-\frac{1}{q^2}\right)$$

$$\geq \frac{1}{2}\left(1-\frac{1}{25}\right) = \frac{1}{2}\cdot\frac{24}{25}$$

$$= 12/25. \quad \blacksquare$$

Lecture 13 (21-09-2021)

21 September 2021 17:25

Input: $f \in \mathbb{F}_q[x]$, deg d. Given as list of coefficients.

As before, assume q odd.

Output: Irreducible polynomials $g_1, \dots, g_k \in \mathbb{F}_q[x]$ s.t.

$$f = g_1 \cdot \dots \cdot g_k.$$

Suffices to have an algorithm ② for a nontrivial factorisation, i.e., same input but output is $g \in \mathbb{F}_q[x]$ s.t. $g \mid f$ and $1 \leq \deg(g) < \deg(f)$.

Algorithm ② + Division + Recurse finishes original task.

Preprocessing phase:

① Ensure squarefreeness. (That is, $\nexists g \in \mathbb{F}_q[x]$ s.t. $\deg(g) \geq 1$ & $g^2 \mid f$.)

Compute f'.

If $f' \neq 0$, compute $\gcd(f, f')$:

- if deg ≥ 1, already done!
- if deg $= 0$, then we know f is squarefree.

Can do this quickly

If $f' = 0$, then $f = (\tilde{f})^p$ for some $\tilde{f} \in \mathbb{F}_q[x]$, where $p := \mathrm{char}(\mathbb{F}_q)$.

Repeat with $\tilde{f}$.

From now on, we assume f is square free.

② Distinct degree factorisation.

At the end of this step, we want the input to

be a product of distinct irreducible polynomials of the same degree.

At this point, we know

$f = g_1 \cdots g_k$ for distinct irreducibles g_i.

TOY EXAMPLE : $x^q - x = \prod_{\alpha \in \mathbb{F}_q} (x - \alpha)$.

$f \to$ square free.

Let $h = \gcd(f, x^q - x)$.

Then, h = product of all irreducible factors of deg 1.

deg 1 irred factors $\leftarrow h$, $f/h \hookrightarrow$ deg ≥ 2 irred factors

FACT : Let $k \in \mathbb{N}$.

Then, $x^{q^k} - x$ = product of all monic irred. $g \in \mathbb{F}_q[x]$ s.t. $\deg(g) \mid k$.

for $i = 1$ to $d = \deg(f)$:

$h_i = \gcd(f, x^{q^i} - x)$ → Can do this quickly as in earlier lec

$f \to f/h_i$

Output $h_1, h_2, \ldots, h_d$.

OBSERVE : For all $i \in \{1, \ldots, d\}$:

$$h_i = \prod_{\substack{g \in \mathbb{F}_q[x] \\ \text{irred} \\ g \mid f}} g.$$

If any of h_i is a proper factor, the done. Else exactly one is f and thus, f is in desired form. Moreover, we know the degree.

Going forward:

f is a product of distinct irreducibles of known degree.

(In particular, if $\deg(f) =$ this degree, then f is irred.)

Goal: Find a nontrivial factor.

Will use randomness.

For simplicity, we assume

$$f = g_1 \cdot g_2, \qquad \deg(f) = 2r.$$

Algorithm 2:

- Pick $A \in \mathbb{F}_q[x]$, $\deg(A) < 2r$, uniformly at random.
- Output $\gcd(f, A(x)^{\frac{q^r-1}{2}} - 1)$.

Running time: Fast as before, go by repeated squaring.

Correctness (Probabilistic):

Lemma If $f = g_1 \cdot g_2$, $\deg(g_1) = r$, $g_1 \neq g_2$ irreducibles.
Then,

$$\Pr_{\substack{A \in \mathbb{F}_q[x] \\ \deg(A) < 2r}} \left(\gcd(f, A^{\frac{q^r-1}{2}} - 1) \text{ is deg } r \right) \approx \tfrac{1}{2}.$$

Equivalent but cleaner phrasing:

Lemma $\forall g_1, g_2 \in \mathbb{F}[x]$, irred., distinct, deg r.
Then,

$$\mathbb{P}_{\substack{A \in \mathbb{F}_q[x] \\ \deg < 2r}} \left[\begin{array}{l} (A(x)^{\frac{q^r-1}{2}} - 1) \text{ is divisible} \\ \text{by exactly one of } g_1 \text{ or } g_2 \end{array} \right] \approx \tfrac{1}{2}.$$

Note: Lemma $\Rightarrow$ Correctness

Proof. Let us compute

$$\mathbb{P}_A\left[\, g_1 \mid \left(A^{\frac{q^r-1}{2}} - 1\right)\right].$$

Claim! $\mathbb{P}_A\left[g_1 \mid A^{\frac{q^r-1}{2}} - 1\right] = \frac{1}{2} - \frac{1}{2q^r}$.

Proof. Note that

$$g_1 \mid \left(A^{\frac{q^r-1}{2}} - 1\right) \iff g_1 \mid (A \bmod g)^{\frac{q^r-1}{2}} - 1.$$

Consider the field

$$\mathcal{L} = \frac{\mathbb{F}_q[x]}{\langle g_1(x)\rangle} \cong \mathbb{F}_{q^r}.$$

Consider $B(y) = y^{\frac{q^r-1}{2}} - 1$.

Obs: # zeroes of $B(y)$ in $\mathcal{L} = \frac{q^r-1}{2}$.

↳ precisely the nonzero squares

Thus, the claim is reduced to:

$$\mathbb{P}\left[\, (A \bmod g_1) \text{ is a zero of } B(y)\right]$$

If $A \bmod g_1$ is uniformly distributed in $\mathcal{L}$, then the probability above is $\frac{q^r-1}{2q^r} = \frac{1}{2} - \frac{1}{2q^r}$.

And it IS uniformly distributed in $\mathcal{L}$. ∎

(Every such A is of the form $c + hg$ for unique c, h with deg $\le r-1$.)

Similarly $\mathbb{P}\left[g_2 \mid A^{\frac{q^r-1}{2}} - 1\right] = \frac{1}{2} - \frac{1}{2q^r}$.

If divisibility by q_1 and q_2 are indep events, then we are done.
And the independence is due to Chinese Remainder Theorem.

Thus, the desired probability is:

$$2\left[\left(\frac{1}{2}-\frac{1}{2q^r}\right)-\left(\frac{1}{2}-\frac{1}{2q^r}\right)^2\right]$$

$$= 2\left\{\frac{1}{2}-\frac{1}{2q^r}-\frac{1}{4}+\frac{1}{2q^r}-\frac{1}{4q^{2r}}\right\}$$

$$= 2\left\{\frac{1}{4}-\frac{1}{4q^{2r}}\right\}$$

$$= \frac{1}{2}-\frac{1}{2q^{2r}}.$$

□

Lecture 14 (24-09-2021)

24 September 2021 17:30

Today: Deterministic algorithm for factoring univariate poly over $\mathbb{F}_q$. (Berlekamp)

- Running time: $\text{poly}(\deg, \log q, \text{char})$.
  (So if we have prime field, not so good.)

Input: $f \in \mathbb{F}_q[x]$, given as a list of coefficients $f_0, \dots, f_d$.
$p = \text{char}(\mathbb{F}_q)$

Output: A non-trivial factor of f.

Assume that preprocessing has been done (as in Lec 13).
→ f is square free
→ product of distinct irreducible of same deg.

Two steps:

Algo 1: $\text{poly}(d, q)$

Algo 2: Change above slightly to get $\text{poly}(d, \log q, p)$

Algo 1:

1. Find a poly $g \in \mathbb{F}_q[x]$ s.t.
   ① $1 \le \deg(g) < \deg(f)$,
   ② $f \mid g^q - g$.

   (This step won't work if f is irred.)

2. For every $\alpha \in \mathbb{F}_q$, compute $\text{GCD}(f, g(x) - \alpha)$.

Theorem 1. If $f \in \mathbb{F}_q[x]$ is NOT irreducible, then Algo 1

finds a non-trivial factor.
Moreover, runtime of algo $\leq$ poly(d, q).

Proof. 1) Assume that step 1 works, i.e., we have found $g \in \mathbb{F}_q[x]$ satisfying ① and ②.
Want to show that step 2 finds a nontrivial factor of f in time poly(d, q).

Runtime is easy: q iterations, GCD computation is poly(d).
Correctness:
Note $$z^q - z = \prod_{\alpha \in \mathbb{F}_q} (z - \alpha).$$

Thus, $$g(x)^q - g(x) = \prod_{\alpha \in \mathbb{F}_q} (g(x) - \alpha).$$

$$f \mid g(x)^q - g(x) \Rightarrow f \mid \prod_{\alpha \in \mathbb{F}_q} (g(x) - \alpha).$$

NOTE: Did not assume f is irred. for this part. Thus, Step 1 will not work if f is irred.

$\Rightarrow \exists \alpha \in \mathbb{F}_q$ s.t. GCD$(f, g(x) - \alpha)$ is nonconstant. Since $\deg(g(x)) < d$, we get that GCD is a NON-TRIVIAL factor.

2) Step 1 works: That is, if f is reducible, then
ⓐ a desired g exists, and
ⓑ can find one in time poly(d, q).

Note that condition ② is an $\mathbb{F}_q$-linear constraint: If
$f \mid g_1^q - g_1$ and $f \mid g_2^q - g_2$, then
$$f \mid (g_1 + \alpha g_2)^q - (g_1 + \alpha g_2).$$

Will try to set up linear system in coefficients of g.

Let $g \in \mathbb{F}_q[x]$ be arbitrary with deg $\leq d-1$.

$$g(x) = g_0 + g_1 x + \cdots + g_{d-1} x^{d-1}.$$

Then, $g(x)^q = g_0 + g_1 x^q + \cdots + g_{d-1} x^{q(d-1)}$.

Want: $g_0, \ldots, g_d \in \mathbb{F}_q$ s.t.

$$(g^q - g)(x) = \sum_{i=0}^{d-1} g_i (x^{iq} - x^i)$$

is divisible by $f(x)$.

- Compute

$R_i(x) := (x^{iq} - x^i) \bmod f(x)$ for $i = 0, \ldots, d-1$.

↳ deg $\leq d-1$.

Want $g_0, \ldots, g_{d-1} \in \mathbb{F}_q$ s.t.

$$\underbrace{g_0 R_0(x) + \cdots + g_{d-1} R_{d-1}(x)}_{\text{collect all coefficients of } x^i \text{ and equate to } 0} = 0.$$

- We get a linear system in g_i over $\mathbb{F}_q$.
- Solve the system to get a solution.

BUT we need a solution which gives a sol^n with $g_i \neq 0$ for some $i > 0$.

Runtime of this step = poly(d, q).

Note: each $\alpha \in \mathbb{F}_q$ satisfies $\alpha^q - \alpha = 0$.

Thus, we have q "trivial" solutions.

Suffices to show that the linear system has $\geq q+1$ distinct solutions.

Now we use that f is reducible.

Write $f = A \cdot B$ s.t. $\gcd(A, B) = 1$ with A & B nonconst.

Observe: $\forall \alpha, \beta \in \mathbb{F}_q$, if $g(x) = \alpha \mod A$, and
$g(x) = \beta \mod B$,
then $f \mid g(x)^q - g(x)$.
(That is, if g is constant mod A & B, then $f \mid g^q - g$.)

Proof: $g^q(x) - g(x) = \alpha^q - \alpha = 0 \mod A$.
$\|$ (by FLT) $g^q - g = 0 \mod B$.
By CRT, $g^q - g = 0 \mod AB$.
$\|$
f ∎

Now, for each $(\alpha, \beta) \in \mathbb{F}_q^2$, consider the polynomial

$g_{\alpha,\beta}(x)$ obtained by CRT s.t. $g_{\alpha,\beta} \equiv \alpha \mod A$,
$\beta \mod B$.

Since $q^2 > q$, we have a non-deg $=1$ sol^n.

NOTE: The algo to find g DOES NOT use A and B.
The algo just solves the linear system.
The analysis above shows (using A and B) that the linear system WILL have a solution of desired form. ∎

Algo 2: (Improving runtime to $\text{poly}(d, \log q, p)$.

Step 1: Find $g \in \mathbb{F}_q[x]$ s.t.

Step 1: Find $g \in \mathbb{F}_q[x]$ s.t.

① $1 \le \deg(g) \le d-1$,

② $f \mid g^p - g$.

Step 2. For all $\alpha \in \mathbb{F}_p$, compute $\gcd(f, g(x) - \alpha)$.

↳ prime subfield of $\mathbb{F}_q$

Using the earlier proof, it suffices to show Step 1 works in desired time.

$$g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_{d-1} x^{d-1}$$

Condition ② is not $\mathbb{F}_q$ linear anymore, only $\mathbb{F}_p$.

$g_i \in \mathbb{F}_q$. Consider $\mathbb{F}_q = \frac{\mathbb{F}_p[Y]}{\langle h(Y) \rangle}$ for h irred of deg k.

$$g_i = g_{i,0} + g_{i,1}[Y] + \cdots + g_{i,k-1}[Y]^{k-1}$$

$g_{i,j} \in \mathbb{F}_p$

$$g_i^p = g_{i,0} + g_{i,1}^p [Y]^p + \cdots + g_{i,k-1}^p [Y]^{p(k-1)}.$$

$[Y] = Y \bmod h$.

$f \mid g^p - g$ leads to a linear system (over $\mathbb{F}_p$) in the $g_{i,j}$. Solve this system to get a solution.

Now, need to argue that $\exists$ non trivial solution.

The same as before! Note that the deg 0 sol'n's are only when $g(x) = \alpha \in \mathbb{F}_p$.

Thus, p deg $=0$ sol'n's.

But p^t many solutions exist via CRT as before. ∎

But p' many solutions exist via CRT as before. ☺

Q. Do the following:

Input: $f(x) \in \mathbb{F}_q[x]$, $\deg(f) = d$.

Output: $\alpha \in \mathbb{F}_{q'}$, for some extension $\mathbb{F}_{q'} \supseteq \mathbb{F}_q$ s.t. $f(\alpha) = 0$.

Lecture 15 (28-09-2021)

28 September 2021 17:27

Today and next class: Bivariate factorisation

Input: $f(x, y) \in \mathbb{F}[x,y]$, deg d, given as a list of monomials.

Goal: Output a factorisation of f into a product of irreducibles. (Over $\mathbb{F}$ or maybe an extension.)
As in the case of univariate, the following suffices:

Output: A nontrivial factor of f.

Will use the following univariate problem as a subroutine:
Input: $h(x) \in \mathbb{F}[x]$
Output: $\alpha \in \mathbb{K}$ s.t. $h(\alpha) = 0$, where $\mathbb{K}$ is an ext^n of $\mathbb{F}$.

Note: If $\mathbb{F}$ is finite, we have seen how to get an irreducible factor of a polynomial. Then we can get an extension where there is a root.
For $\mathbb{F}$ infinite, we haven't seen any such routine but will assume that we have one such.

High level view:

- Preprocess f to put it in a "necessary" form:

① f is not a p^{th} power, where $p = \text{char}(\mathbb{F})$.
(If $p = 0$, nothing to do.)

② Make sure that $f(x, y)$ is monic when viewed as a univariate in $(\mathbb{F}[x])[y]$ (or maybe in

$(\mathbb{F}[y])[x]$.

③ $\frac{\partial f}{\partial y} \neq 0$. [Non-singularity w.r.t. y.]

④ $f(x,y)$ is squarefree.

⑤ $f(0, y) \in \mathbb{F}[y]$ is squarefree

(Idea is to change f to f^* at each stage s.t. f^* satisfies the conditions and a factorisation of f^* gives one of f.)

At the end, we will have a polynomial $\tilde{f}$ with the desired properties and

factors of $\tilde{f}$ $\longleftrightarrow$ factors of f.

— Iterative derivative based algorithm to factor f, assuming that it is preprocessed.

(Will resemble Newton iteration.)

Will use the following form of Gauss' Lemma:

Lemma. $f(x, y) \in \mathbb{F}[x][y]$ is monic in y and $g(x,y) \in \mathbb{F}(x)[y]$ is monic in y and divides f over $\mathbb{F}(x)$.

Then, $g(x, y) \in \mathbb{F}[x][y]$

Will also use Resultants, GCD algorithms, etc. from earlier classes.

Today: preprocessing

PREPROCESSING.

① Is not a p^{th} power.

— 1.1. Compute $e \in \mathbb{N}$ s.t. f is a p^e-th power but

Not a p^{e+1}-th power.

For every monomial $m_{i,j} = x^i y^j$ with a nonzero coefficient in f, define

$$e_{i,j} = \max_e \left(p^e \text{ divides both } i \text{ and } j\right)$$

Then, put $e := \min_{m_{i,j}} e_{i,j}$.

f is a p^e-th power but not p^{e+1} th power

$\Updownarrow$ Assume $\mathbb{F}$ is perfect, i.e., $\mathbb{F} = \mathbb{F}^p$.

$\forall m_{i,j}$ in f, we have $p^e \mid i$ & $p^e \mid j$

AND

$\exists m_{i,j}$ in f s.t. $p^{e+1} \nmid i$ or $p^{e+1} \nmid j$.

For concreteness, we shall assume

- 1.2. $f \longrightarrow f^*$ s.t. ① f^* is not a p^{th} power
② factors of f and f^* are "closely related"

Write $f(x, y) = \sum_{j=0}^{d} f_j(x)\, y^j$.

$p^e \mid j$ s.t. $f_j(x)$ is nonzero.

$$f^*(x, y) := \sum_{j=0}^{d} f_j(x)\, y^{j/p^e}$$ → NOT a p^{th} power.

Let us now go from a nontrivial factor of f^* to one of f in an efficient manner:

Algo. Note that $f^*(x, y^{p^e}) = f(x, y)$.

Let $h(x,y)$ be a nontrivial factor of $f(x,y)$.
$\Rightarrow h^*(x, y^{p^e}) \mid f^*(x, y^{p^e}) = f(x,y)$.

Moreover, $h(x,y) := h^*(x, y^{p^e})$ is a nontrivial factor of $f(x,y)$. ✓

Now, if f is reducible, we need to ensure f^* is reducible. Let h be an irred factor (non trivial). Then, $h^{p^e} \mid f$.
Write $h(x,y)^{p^e} = h^*(x, y^{p^e})$.
Then, $h^* \mid f^*$ and h^* is nontrivial.

①&③ Assume f is not a p^{th} power. (Also WLOG $\exists j$ s.t. coeff. of y^j in $f^*(x,y)$ is nonzero and $p \nmid j$.)
Now, we process f to make it monic in y and $\frac{\partial f}{\partial y} \not\equiv 0$.

– Want to find a matrix

$$Z = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \in \mathbb{F}^{2\times 2}$$

s.t.
① Z is invertible,
② $f\left(Z \cdot \begin{bmatrix} x \\ y \end{bmatrix}\right)$ is monic in y and $d = \deg(f)$,
③ derivative condition holds for.

$$F = f(z_{11}x + z_{12}y,\ z_{21}x + z_{22}y) \in \mathbb{F}[x,y,\bar{z}]$$
$$= \sum_{i=0}^{d} y^i \cdot f_i(\bar{z}, x)$$
↙ deg in y cannot increase

Claim: ① $f_d(\bar{z}, x) \neq 0$,
② $f_1(\bar{z}, x) \in \mathbb{F}[x]$,

③ $f_j(\bar{z}, x) \neq 0$, where j is s.t. p_{t_j} from earlier.

Note: $m_{ij}(x,y) = x^i y^j$

$$\downarrow$$

$$(z_{11}x + z_{12}y)^i (z_{21}x + z_{22}y)^j \in \mathbb{F}[\bar{z}, x][y]$$

deg in y $= i+j$.

coefficient of y^{i+j} $= z_{12}^i z_{22}^j$

$= m_{ij}(z_{12}, z_{22})$.

Write $f(x,y) = h_0(x,y) + \cdots + h_d(x,y)$.

↳ homogeneous components of f of corresp. total degree

Then, coefficient of y^ℓ in F is equal to $h_\ell(z_{12}, z_{22})$.

$\not\equiv 0$

This proves ① and ②.

To prove ③, put $\bar{z} = \bar{z}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ to get

$f_j(\bar{z}_0, x) =$ coefficient of y^j in $f \not\equiv 0$. ∎

Claim: Can efficiently find $A \in \mathbb{F}^{2\times 2}$ s.t.

① $\det(A) \neq 0$,

② $f_d(A) \neq 0$,

③ $f_j(A, x) \neq 0$.

Proof: Take $Q(\bar{z}) := \det(\bar{z})\, f_d(\bar{z})\, f_j(\bar{z}, x)$.

We have shown Q is not the zero poly.

Thus, $\exists A$ s.t. $Q(A) \neq 0$. (Use a polynomial

identity theorem.)

Assume f large enough. $\square$

(Or take extension.)

This fulfills our requirement.

Lecture 16 (01-10-2021)

01 October 2021 17:23

Preprocessing phase:

DID THIS

1. Ensure $f(x, y)$ is not a p^{th} power.
2. $f(x, y)$ is monic in y
3. $\frac{\partial f}{\partial y}(x, y) \neq 0$

} did this by a linear change of variable

4. $f(x, y)$ is squarefree
5. $f(0, y) \in \mathbb{F}[y]$ is squarefree

---

④ **Squarefreeness**

$g^2(x,y) \mid f(x,y) \Rightarrow \gcd\left(f(x,y), \frac{\partial f}{\partial y}(x,y)\right)$ is nontrivial

↓ we have ensured that this is nonzero!

Thus, if f is not squarefree, then we already have a nontrivial factor.

Just need to see if we can compute GCD. We can do it in the usual way since f and $\frac{\partial f}{\partial y}$ is monic in y.

(Possibly get a polynomial in $\mathbb{F}(x)[y]$ but we Gauss to conclude that it is in $\mathbb{F}[x][y]$.)

⑤ To ensure: $f(0, y) \in \mathbb{F}[y]$ is square free.

Know so far: $f(x,y)$ — monic in y, $\frac{\partial f}{\partial y} \neq 0$, square free

Recall: $f(x, y) \in (\mathbb{F}[x])[y]$ is squarefree iff

$$\mathrm{Disc}_y(f) = \mathrm{Res}_y\left(f, \frac{\partial f}{\partial y}\right) \neq 0.$$

$$f = f_0(x) + y \cdot f_1(x) + \cdots + y^d \cdot f_d(x)$$

f_d: nonzero field constant, i.e., $\in \mathbb{F}^\times$

$$\frac{\partial f}{\partial y} = \tilde{f}_0(x) + y \cdot \tilde{f}_1(x) + \cdots + y^e \cdot \tilde{f}_e(x)$$

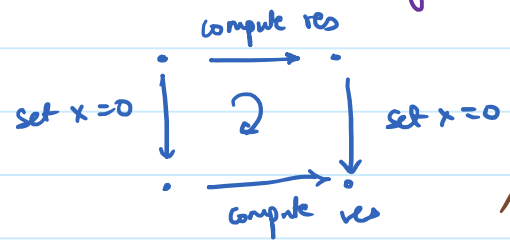
$(\tilde{f}_e \neq 0, \ e \leq d-1)$

$$\det \begin{bmatrix} f_d & & & \tilde{f}_e & \cdots \\ f_{d-1} & f_d & & \vdots & \\ \vdots & \vdots & \ddots & \tilde{f}_0 & \\ f_0 & \vdots & & & \\ & f_0 & & & \end{bmatrix} \in \mathbb{F}[x]$$

$$r(x) := \mathrm{Res}\left(f, \frac{\partial f}{\partial y}\right) \in \mathbb{F}[x], \quad \deg \leq 2d^2$$

If $r(0) \in \mathbb{F}$ is nonzero, then we are done already.

(Why?!
The following diagram commutes.



Otherwise, $\exists \alpha \in \mathbb{F}$ s.t. $r(\alpha) \neq 0$. $(\because r(x) \neq 0)$
(Assume $\mathbb{F}$ large enough.)
Now, put $f^*(x,y) = f(x+\alpha, y)$.
Everything now works nicely.)

X END OF PRE PROCESSING X

Going forward: Assume f has ① - ⑤.

Special case: f has a factor of the form $Y - g(x)$ for some $g(x) \in \mathbb{F}[x]$.

Things we know:

① $f(x,y)$ is squarefree

$$\Rightarrow \quad (y - g(x)) \mid f(x, y)$$

AND

$$(y - g(x))^2 \nmid f(x, y)$$

② $y - g(x) \mid f(x, y) \iff f(x, g(x)) = 0.$

Assume: We have an efficient algorithm for computing roots of univariates over $\mathbb{F}$?

Can we use this to get anything about g?

Note: $f(x, g(x)) \in \mathbb{F}[x]$ of deg $\leq O(d^2)$

Want a $g(x)$ of deg $< d$ s.t. $f(x, g(x)) = 0$. ← 0 polynomial

$\Updownarrow$

$\forall k \in \mathbb{N}, k \leq d^2$: coeff of x^k in $f(x, g(x))$ is 0.

$\Updownarrow$

② $\forall k \in \mathbb{N}, k \leq d^2$: $f(x, g(x)) = 0 \mod x^k$

Put $k = 1$ in ②:

$f(x, g(x)) = 0 \mod x$

$g = g_0 + g_1 x + \cdots + g_r x^r$

g_0 = g_1 = ... = g_r ↓ want to find these

$f(x, g_0 + g_1 x + \cdots + g_r x^r) = 0 \mod x$

$\Updownarrow$

going mod x really means setting $x = 0$.

$f(0, g_0) = 0$ in $\mathbb{F}$

$\Updownarrow$

g_0 is a root of $f(0, Y) \in \mathbb{F}[Y]$

'Finding' g_0: $f(X, Y) \xrightarrow[X=0]{} f(0, Y) \in \mathbb{F}[Y]$

$\downarrow$ use efficient root finder

get a list: $r_0, \ldots, r_d$ of roots of $f(0, Y)$

One of them is g_0. (Dunno which one.)

Rough Algo: ① For each g_0 in $\{r_0, \ldots, r_d\}$, find a g_1.

② Keep building inductively.

Will show: Going from g_0 gives a unique g_1, so no more branching.

What we have:

- $\alpha \in \mathbb{F}$ s.t. $f(0, \alpha) = 0$.
- $f(0, Y)$ is square free.

Want: $\beta \in \mathbb{F}$ s.t. $f(X, \alpha + \beta X) = 0 \mod \langle X^2 \rangle$.

Lemma. $\forall \alpha \in \mathbb{F}$: ($f(X, \alpha) = 0 \mod \langle X \rangle$ AND $\frac{\partial f}{\partial Y}(0, \alpha) \neq 0$

$\Downarrow$

$\exists!\ \beta \in \mathbb{F}$ s.t. $f(X, \alpha + \beta X) = 0 \mod \langle X^2 \rangle$.)

Moreover, β can be efficiently computed.

Lemma*. $\forall k \in \mathbb{N}$, $\forall \alpha_0, \ldots, \alpha_k \in \mathbb{F}$:

$f(X, \alpha_0 + \cdots + \alpha_k X^k) = 0 \mod \langle X \rangle^{k+1}$

AND $\frac{\partial f}{\partial Y}(0, \alpha_0) \neq 0$

AND $\frac{\partial f}{\partial y}(0, \alpha_0) \neq 0$

$\Downarrow$

$\exists!\ \beta \in \mathbb{F}$ s.t. $f(x, \alpha_0 + \cdots + \alpha_k x^k + \beta x^{k+1}) = 0 \mod \langle x \rangle^{k+2}$.

↓ can be efficiently computed.

Proof of Lemma: Want $\beta \in \mathbb{F}$ s.t. $f(x, \alpha + \beta x) = 0 \mod x^2$.

Know: $f(x, \alpha) = 0 \mod x$, $\frac{\partial f}{\partial y}(0, \alpha) \neq 0$.

Taylor:

$$f(x, \alpha + \beta x) = f(x, \alpha) + (\beta x)\frac{\partial f}{\partial y}(x, \alpha) + \frac{(\beta x)^2}{2!}\frac{\partial^2 f}{\partial y^2}(x, \alpha) + \cdots$$

Going mod x^2, our desired condition becomes

$$f(x, \alpha) + \beta x \cdot \frac{\partial f}{\partial y}(x, \alpha) = 0 \mod \langle x^2 \rangle$$

↓ our hypothesis ensures that $\exists!\ \beta \in \mathbb{F}$ satisfying above

Exercise: Prove Lemma* and verify that we have found $y_-(x)$.

Next: Arbitrary factors

Lecture 17 (05-10-2021)

05 October 2021 17:25

Don't assume $Y - g(X)$ is a factor.

Algo:

① Find α_0 s.t. $f(x, \alpha_0) = 0 \mod \langle x \rangle$, i.e., $f(0, \alpha_0) = 0$.
Since $f(0,y)$ is squarefree, this ensures $\frac{\partial}{\partial y} f(0, \alpha_0) \neq 0$.

② For $k = 2d^2 + 1$, use Lemma* to get
$A(x) = a_0 + \cdots + a_k x^k$ s.t.
$f(x, A(x)) \equiv 0 \mod \langle x \rangle^{k+1}$.

③ Find a polynomial $H(x,y) \neq 0$ s.t.
① $\deg_y(H) \leq d-1$,
② $\deg_x(H) \leq d$,
③ $H(x, A(x)) \equiv 0 \mod \langle x \rangle^{k+1}$.
(we have preprocessed to make sure $\deg_y(f) = d$.)
If no such H, output "f irred".

④ Output $\gcd_y(f, H)$.
If $\gcd = 1$, output "f irred".

How do we find H? Note that ③ is a linear condition. Thus, we get a linear system in coefficients of H.

$$H(x,y) = \sum_{i=0}^{d} \sum_{j=0}^{d-1} h_{ij} x^i y^j \quad ; \quad h_{ij} \in \mathbb{F}.$$

Want h_{ij} s.t. $H(x, A(x)) = \sum_i \sum_j h_{ij} x^i (A(x))^j = 0 \mod \langle x \rangle^{k+1}$.

System of homog. linear solution in h_{ij}.
Size $O(d^2)$. Solve for a nonzero solution.

If no nonzero sol^n: output f is irred.

Running time: poly(d) + Root finding call.

Correctness:

Lemma. Assume f is reducible.
① There is a nonzero sol^n to the linear system.

② Any nonzero sol^n will give a nontrivial GCD.

Proof. ① Write $f(x,y) = g_1(x,y) \cdots g_t(x,y)$
for $t \geq 2$, and
(i) g_i — irred,
(ii) $g_i \neq g_j$ $\forall i \neq j$.

Obs: $1 \leq \deg_y(g_i) \leq d-1$, since f is monic in y with $\deg_y(f) = d$.

Claim: $\exists\, i \in \{1, \dots, t\}$ s.t. $g_i(x, A(x)) = 0 \bmod \langle x \rangle^{k}$.
Then, can take $H = g_i$.

Proof.
$$g_1(x, A(x)) \cdot \cdots \cdot g_t(x, A(x)) = f(x, A(x)) = 0 \bmod \langle x \rangle^{k+1}. \quad (*)$$

In particular, the product is $0 \bmod x$, i.e.,

$$f(0, A(0)) = g_1(0, A(0)) \cdot g_2(0, A(0)) \cdots g_t(0, A(0)) = 0$$

Since $f(0,y)$ is square free, $\exists!\, i \in \{1, \dots, t\}$ s.t.
$g_i(0, A(0)) = 0$.

Thus, for $j \neq i$, $g_j(x, A(x))$ is NOT divisible by x. Thus, $(*)$ tells us that $g_i(x, A(x)) \equiv 0 \bmod x^{k+1}$!

② Pick a nontrivial sol^n H.

Claim 1: $\deg_y(H) \geq 1$.
Proof. Else it is just $H(x,y) = h(x) \in F[x]$.
Then, $H(x, A(x)) = h(x) \equiv 0 \bmod \langle x \rangle^{k+1}$.
But $\deg h = \deg_x H \leq d \leq k+1$.
$\Rightarrow h = 0$ itself. ∎

View: $f(x,y)$, $H(x,y) \in F[x][y]$

$$f(x,y) = y^d + f_{d-1}(x) y^{d-1} + \cdots + f_1 y + f_0.$$
$$H(x,y) = H_\ell(x) y^\ell + \cdots + H_1(x) \cdot y + H_0(x).$$

$$C(x,y) = \begin{bmatrix} 1 & & & H_0 \end{bmatrix}$$

$$S_y(f, H) = \begin{bmatrix} 1 & & & H_d & \\ f_{d-1} & 1 & & H_{d-1} & H_d \\ f_{d-2} & f_{d-1} & \cdots & \vdots & H_{d-1} & \ddots \\ \vdots & \vdots & & & \vdots & \end{bmatrix}_{(d+1)\times(d+1)}$$

TS: $\det S_y(f, H) = Res_y(f, H) \equiv 0.$

Note $Res_y(f, H) \in \mathbb{F}[x]$ of deg $\leq d(d+1) < 2d^2$.

Thus, suffices to show it is $0 \bmod x^{k+1}$!

"Recall": $Res_y(f, H) = f(x,y)\,B(x,y) + H(x,y)\cdot C(x,y)$ for $B, C \in \mathbb{F}[x, y]$.

This is in $\mathbb{F}[x]$!

Thus, putting $y = A(x)$ gives

$$Res_y(f, H) = f(x, A(x))\,B(x, A(x)) + H(x, A(x))\cdot C(x, A(x)) \equiv 0 \bmod x^{2d^2+1}.$$

This finishes the proof. ∎

Exercise. Can you adapt this to multivariate. (Assume it is preprocessed.)

Algebraic / Arithmetic circuits.

Typical computational question:

I/P: $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$

Goal: Do something with f.

Examples:
1. $a \in \mathbb{F}^n$, evaluate $f(a)$
2. Check if $f(x) \equiv 0$. (Polynomial identity testing (PIT))
3. Is f irreducible?

④ Output irreducible factors of f

⑤ Given g, output $f+g$, fg.

⑥ Does $f \mid g$?

More interested in: How is f given?

① List of coefficients. $\binom{n+d}{d} \sim 2^{2n}$ $(n=d)$ Very large.

Ex: $f(x_1, \ldots, x_n) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$.

I/p:

$\partial = (a_1, \ldots, a_n) \in \mathbb{F}^n$.

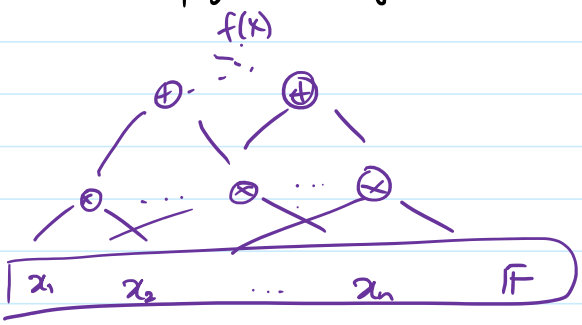
Want: $f(\partial)$.

Naive way: 2^n operations.

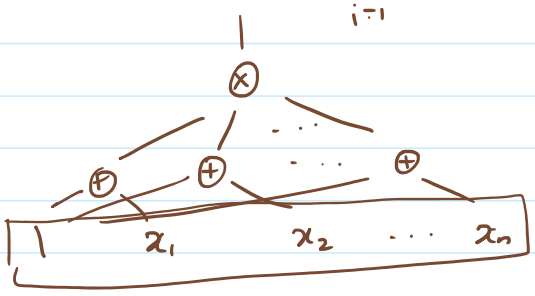
But note: $f(x_1, \ldots, x_n) = \prod_{i=1}^{n} (1 + x_i)$.

Now, $f(\partial)$ is computable in $O(n)$ time.

In most application, polynomials given in arithmetic circuits.



For example, $f(\ldots) = \prod_{i=1}^{n} (1 + x_i)$



Formal def^n: Directed acyclic graph. (Directed upwards)

Leaves → variables & field constants.

Edges → + and ×

Size = # edges / # vertices

Depth = length of the longest path from input

to output.

"Recall": P = Decision problem with det. poly time algo.

VP = poly. families with d = poly(n), circuits of size poly(n)

Valiant, after Leslie Valiant

NP = Easy to verify

Analog ↑

VNP

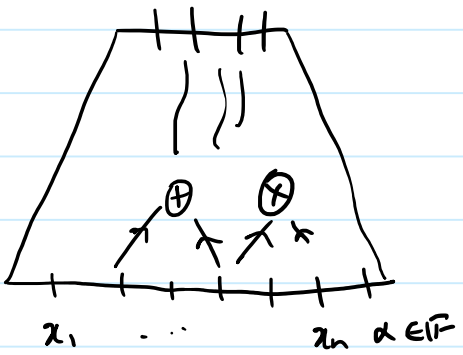
(non uniform) P v/s NP ~closely related~ VP v/s VNP

assuming stuff like GRH.

Lecture 18 (08-10-2021)

08 October 2021 17:24

Arithmetic / Algebraic circuits



Will assume fan-in 2, i.e., $\oplus$ and $\otimes$ take only two inputs

- Size → either # wires → edges or # gates → vertices
- Depth

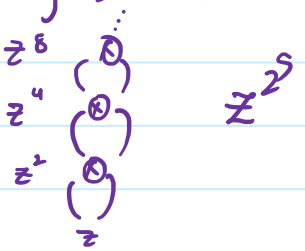
n = # variables

d = degree

s = circuit size

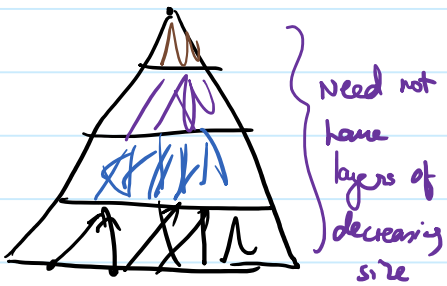
d = poly(n)

Note, we can compute high degree poly using low s



Can assume circuit is layered:

Can partition gates into layers s.t. only edges from lower to upper (nothing with same layer)



Layered circuit ≡ Parallel computation

gates in a layer ≡ # processors

layers ≡ # Rounds of computation

→ poly(n)

→ poly(log n)

} considered efficient in circuit theory

NC

↳ Nick's class

VP / VNP

VP / VNP

$$\mathrm{Det}_n(x) := \det\begin{bmatrix} & | & \\ & x_{ij} & \\ & | & \end{bmatrix}$$

n^2 variables

$$= \sum_{\sigma \in S_n} \mathrm{sign}(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)}$$

$n!$ monomials

However, this has a circuit of size $O(n^4)$
(Exponent might be diff.)

Thm Det_n has a circuit of size $O(n^4)$ and depth $O(\log^2 n)$.

Berkowitz / Ciaky → Char $>> 0$ or $= 0$.
'94 Schonage → all fields
'98 Mahajan-Vinay → Combinatorial circuit

Fact. Det. is 'almost' complete for VP.

$P \in \mathbb{F}[x_1, \cdots, x_n]$, deg (d), size s ckt
Then, $\exists$ matrix M_p of size $(sd)^{\mathrm{poly}\log(sd)}$ with entries being $x_1, \ldots, x_n$ or $\alpha \in \mathbb{F}$ s.t.

$$P = \det(M_p).$$

Open problem: Can we remove 'almost'?

Theorem. (Valiant - Skyum - Beskowitz - Rackoff '82)
P — n variate, deg d, size s ckt, $s = d = \mathrm{poly}(n)$.
Then, P has a ckt of size $\mathrm{poly}(sd)$ and depth $O(\log s \cdot \log d)$.

Permanent

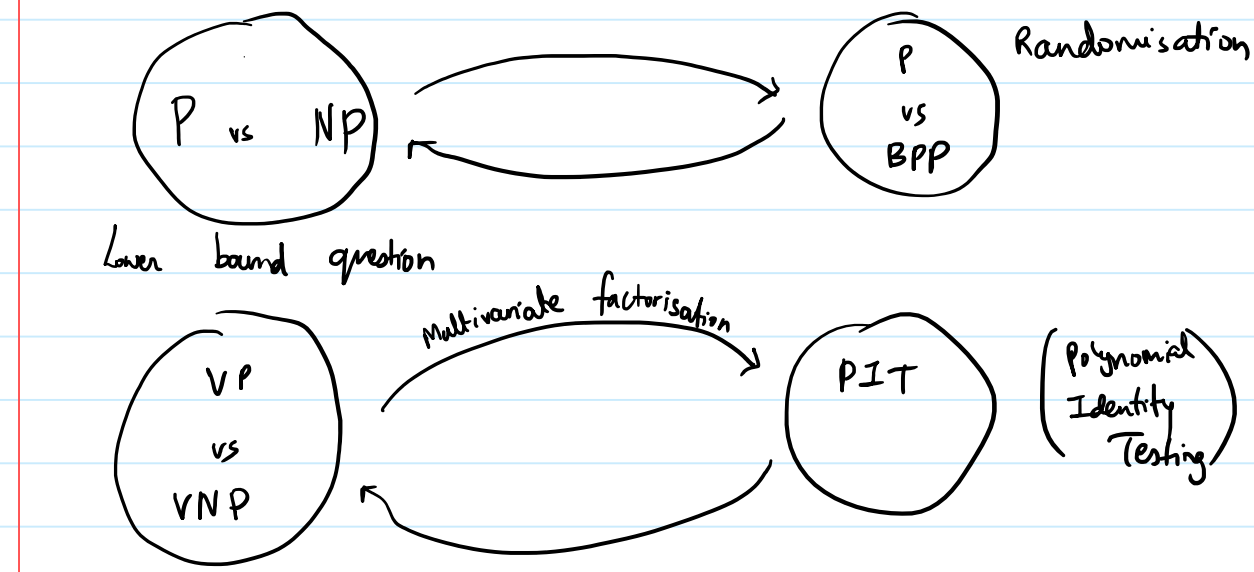
$$\text{Perm}_n = \sum_{\sigma \in S_n} \prod_{i \in [n]} x_{i,\sigma(i)}.$$

$= \text{perm}\left[\downarrow \; x_{ij} \right]$

Valiant (conjecture): Perm_n does not have a ckt of size poly (n).

$\hookrightarrow$ VP $\neq$ VNP (Conjecture)

Fact. Perm_n is complete for VNP.



PIT: I/P : Ckt C, size s, deg d, n-variables

Goal : Is $C \equiv 0$?

White box setup : Look inside ckt

Black box : recs s, d, n and queries to C.

– PIT has a simple randomised algorithm.

$S \subseteq \mathbb{F}$, $|S| = 100d$, S2-lemma: $\Pr_{a \in S^n}[P(a) = 0] \leq \frac{1}{100}$.

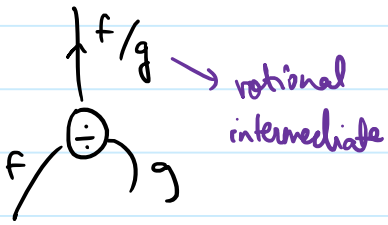
Fact. $\exists c \in \mathbb{N}$ s.t. $\forall s, d, n \in \mathbb{N}$ $\exists H_{s,d,n} \subseteq \mathbb{F}^n$, $|H| \leq (sdn)^c$ s.t. (like 10 or 20)

$\forall P \in \mathbb{F}[x_1, \ldots, x_n]$, deg d, ckt size s,

$\forall P \in \mathbb{F}[x_1, \ldots, x_n]$, deg d, ckt size s,
if $P \neq 0$, then $\exists a \in H_{s,d,n}$ s.t. $P(a) \neq 0$.

Getting back to circuits:

What if we got a division gate?



Theorem [Strassen '70]

$\mathbb{F}$ - large enough.
If $P \in \mathbb{F}[x_1, \ldots, x_n]$, deg d, size s ckt with $\times, \div, +$ gates,
then P has a ckt of size $(sd)^{10}$ with $\times, +$ gates.

Q. P, deg d, size s ckt.

$$P = P_0 + P_1 + \cdots + P_d,$$

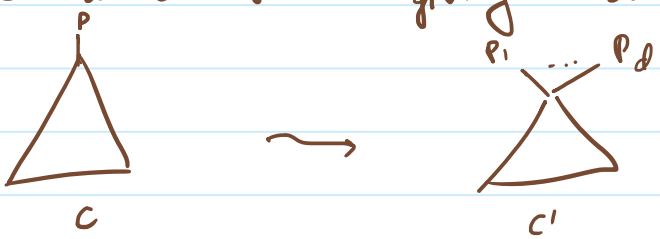
P_i = homogeneous component of P of deg i.

What is the complexity of the homogeneous components?

Thm. (Strassen)

$\forall i$, P_i has a ckt of size $\in O(sd^2)$.
Moreover, the ckt can be assumed homogeneous, i.e.,
the polynomial computed at every gate is homog.

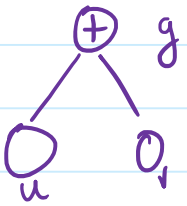
Proof We construct a circuit C' giving d outputs: each P_i.

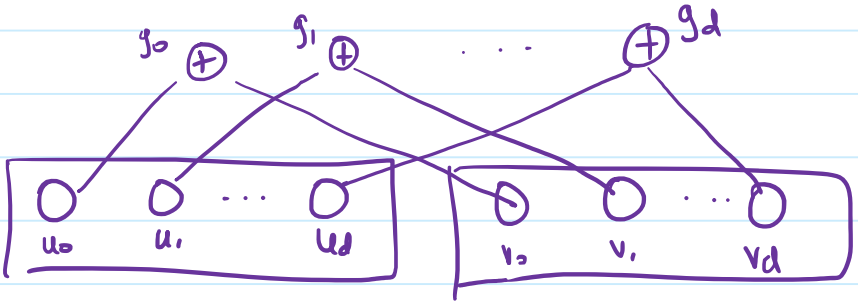


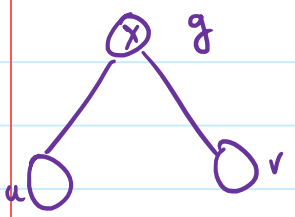
Gates: For each gate in C, we create $d+1$ gates.

$g_i \rightarrow$ computes degree i component of g

Wires:

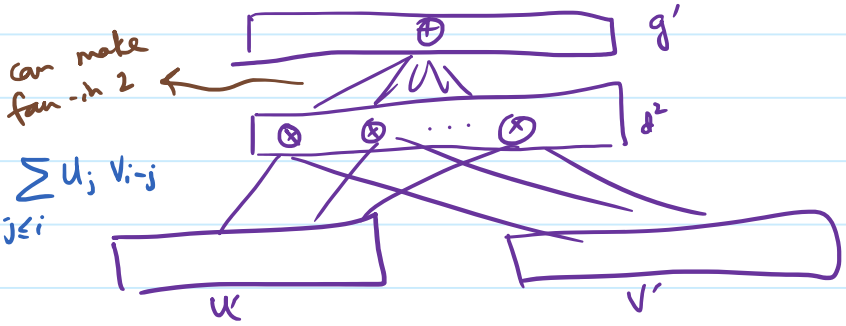






$g_i = \sum_{j \le i} u_j v_{i-j}$

can make fan-in 2



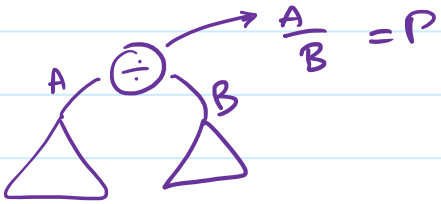
The above construction should make everything clear. ☐

The division reduction is similar. Keep track of numerator and denominator (not even necessarily in lowest form). But in the final stage, we need to do one division.

Lecture 19 (12-10-2021)

12 October 2021 17:30

From the last time, we can eliminate division by introducing one division gate at last point. At that point, we know that one input divides the other.



① Ensure that the denominator has a non-zero constant term.

Since $B \neq 0$, $\exists \alpha \in \mathbb{F}^*$ s.t. $B(\alpha) \neq 0$.

Then, can translate and do it.

↳ $P(X+a) \longrightarrow P(X)$

② Use ① for division elimination.

$$P = \frac{A}{1+B'}, \quad B(0) = 0.$$

(Can scale to make $B(0)=1$.)

Note: $(1-B')(1+B') = 1-B'^2 \equiv 1 \mod \langle x_1, \ldots, x_n \rangle^2$

$(1-B'+B'^2)(1+B') = 1+B'^3$,

$(1-B'+\cdots+(-1)^k B'^k)(1+B') = 1+(-1)^k B'^{k+1}$

$\equiv 1 \mod \langle x_1, \ldots, x_n \rangle^{k+1}$

Note $P(X) \longrightarrow$ polynomial of deg d.

$P(x) \mod \langle x_1, \ldots, x_n \rangle^{d+1} = P(x)$.

Thus, computing P modulo $\langle x_1, \ldots, x_n \rangle^{d+1}$ is sufficient.

Corollary. Define

$$Q = (1 - B' + B'^2 + \cdots + (-1)^r B^{d+1}) \cdot A.$$

Homogeneous components of deg $\le d$ of Q is P.

↓

From this we are done.

Theorem: Det$_n$ can be computed by a circuit of size poly(n) and depth $O(\log^2 n)$.

(Our proofs will break down over some fields. Does work over fields like $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, etc.)

Proof: $X = \begin{pmatrix} & j & \\ i & & \\ & x_{ij} & \end{pmatrix}_{n \times n}$ ← formal variables x_{ij}

Eigenvalues of X are roots of $\mathrm{Det}(X - \lambda I)$.

$\uparrow$

$\mathbb{F}(\bar{X})[\lambda]$

↳ n^2 variables)

Let $\mathbb{K}$ = Algebraic closure of $\mathbb{F}(\bar{x})$.

$\mathrm{Det}(X - \lambda I) \in \mathbb{K}[\lambda]$.

$\lambda_1, \ldots, \lambda_n \in \mathbb{K}$ eigenvalues of X

① $\lambda_1 + \cdots + \lambda_n = \mathrm{Tr}(X) = \sum_i x_{ii}$.

② $\lambda_1 \cdots \lambda_n = \pm \mathrm{Det}(X)$.

③ $\lambda_1^2 + \cdots + \lambda_n^2 = \mathrm{Tr}(X^2)$.

(λ_j^i are e-vals of X^i, with some mult.

④ [illegible]

④ In fact, $\lambda_1^i + \cdots + \lambda_n^i = \mathrm{Tr}(X^i)$. (of X^i, with some mult.)

Digression:

Symmetric Polynomials

① $P_i(y_1, \ldots, y_n) = y_1^i + \cdots + y_n^i$.

Power symmetric polynomials.

② $E_i(y_1, \ldots, y_n) = \sum_{\substack{S \subseteq [n] \\ |S| = i}} \prod_{j \in S} x_j$.

Elementary symmetric polynomials.

In particular, $E_n(y_1, \ldots, y_n) = y_1 \cdots y_n$.

Fact: Both $\{P_i\}_{i=1}^{n}$ and $\{E_i\}_{i=1}^{n}$ generate the ring of symmetric polynomials.

Newton - Girard Identities:

$\forall k$: $\quad k E_k = P_1 E_{k-1} - P_2 E_{k-2} + \cdots + (-1)^{k-1} P_k$.

(Eg. $n=2$, $k=2$: $\quad 2ab = (a+b)(a+b) - (a^2+b^2)$)

in matrix form

$$\begin{bmatrix} 1 & & & \\ -P_1/2 & 1 & & \\ -P_1/3 & P_2/3 & 1 & \\ \vdots & \vdots & & \ddots \\ & & & 1 \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} P_1 \\ -P_2/2 \\ P_3/3 \\ \\ \end{bmatrix}$$

Char $> n$

Char $> n$

$$\begin{bmatrix} \vdots & \vdots & \ddots & \\ -p_1/k & p_2/k & \cdots & 1 \end{bmatrix} \begin{bmatrix} \vdots \\ E_n \end{bmatrix} = \begin{bmatrix} \\ (-1)^{n-1} p_n/n \end{bmatrix}$$

$M = I + M'$; E ; P

$$E = (I + M')^{-1} \cdot P$$

$$= (I - M' + \cdots + (-1)^{n-1} M'^{n-1}) P$$

Note $M'^n = 0$. $\therefore I + M' = I - M' + M'^2 + \cdots$ (finite sum)

Circuit for $E_n(\lambda_1, \ldots, \lambda_n) = \text{Det}_n$:

① Compute $x^2, x^3, \ldots, x^n$.

↳ Compute $x^2, x^4, x^8, \ldots$ and then compute each.

$O(\log n)$ depth and poly(n) size.

② Compute $p_1, \ldots, p_n$ of $\lambda_1, \ldots, \lambda_n$.

$$p_i = \text{Tr}(x^i)$$

③ Have access to entries of M'.

Compute $M'^2, M'^3, \ldots, M'^{n-1}$ as before.

Depth increase by $O(\log n)$.

Fan-in $= O(n)$.

④ Compute $(I - M' + M'^2 + \cdots) P$.

⑤ Convert to fan-in 2 → depth increases by factor of $O(\log n)$

Parallel Algo for Bipartite Matchings:

Given a bipartite graph, does it have a perfect matching?

$G \rightsquigarrow$ graph on $\{1, \ldots, n\} \sqcup \{1, \ldots, n\}$.

Define

$$\tilde{G}(i,j) = \begin{cases} x_{ij} & \text{if } (i,j) \text{ is an edge} \\ 0 & \text{else} \end{cases}$$

Lemma: $\det(\tilde{G}) \neq 0 \iff G$ has a perfect matching.

$\det(\tilde{G}) \neq 0$ as a polynomial

Pf. ($\Leftarrow$) Suppose $(1, i_1), \ldots, (n, i_n)$ is a perfect matching.

Then, the monomial $x_{1,i_1} \cdots x_{n,i_n}$ has a nonzero coeff in $\det(\tilde{G})$.

($\Rightarrow$) A nonzero monomial gives a perf. matching. ∎

Combine Schwartz-Zippel with the circuit to finish the problem.

Lecture 20 (19-10-2021)

19 October 2021 17:30

Today: Multivariate factorisation.

Notation: $n \to$ no. of variables

$d \to$ degree.

$f \in \mathbb{F}[x_1, \ldots, x_n]$

Q. How is f given?

① coeff vec. of $f \to f \in \mathbb{F}^{\binom{n+d}{d}}$ ↳ "dense representation"

② Sparse representation → only specify nonzero coefficients

Assume that input is "sparse" → poly(n) nonzero

↳ ISSUE ↳ factor can be dense

$$x^n - 1 = (x-1)(x^{n-1} + x^{n-2} + \cdots + x + 1)$$

$$\prod_{i=1}^{n} (x_i^n - 1) = \prod_{i=1}^{n} (x_i - 1)(1 + x_i + \cdots + x_i^{n-1})$$

$\prod_{i=1}^{n}(x_i^n - 1)$: 2^n monomials

$\prod_{i=1}^{n}(1 + x_i + \cdots + x_i^{n-1})$: n^n monomials

Open Q: $f \in \mathbb{F}[x_1, \ldots, x_n]$, deg d, s-sparse

$g \mid f$. How dense can g be?

Another example: Let $\mathrm{char}(\mathbb{F}) = p$.

Then, $(x_1 + \cdots + x_n)^d \mid (x_1 + \cdots + x_n)^p \quad \forall\, d < p$

$(x_1 + \cdots + x_n)^d$: dense

$x_1^p + \cdots + x_n^p$: sparse

③ f is given as a circuit.

Thm. (Kaltofen, 86)
$\mathbb{F}$ – field of char 0, $n, d \in \mathbb{N}$
$f \in \mathbb{F}[x_1, \ldots, x_n]$, deg d, has a circuit of size s.
If $g \mid f$, then g has a circuit of size at most $(s \cdot d \cdot n)^{10}$.

Moreover, given the ckt for f, can efficiently (with randomness) construct ckts for all its irreducible factors.

In fact: Can be done with blackbox access to the circuit of f.
↳ Given query access to f, we can construct query access for any irreducible factor $g \mid f$.

Over finite fields: $\mathbb{F}_{p^k}$.
Output: $g_i^{p^{r_i}}$ for $r_i \in \mathbb{N}$.

Open Q. $\mathbb{F} \to$ char $= p$
$g \in \mathbb{F}[x_1, \ldots, x_n]$, deg d s.t. g^p has a ckt of size s.
What is an upper bound on the ckt size of g?

(Last year: Andrews – (2020) g has a ckt of size $(sdn)p^n$.)
Only result we know.

Warm-up: Circuits.

- f has a circuit of size s
  $\Rightarrow$ all the homogeneous components of f have a (homogenous) circuit of size $O(d^2 s)$.

Porism. $f \in \mathbb{F}[x_1, \ldots, x_n][y, z]$ has a ckt of size s.

$$f(y, z) = \sum_{i,j} y^i z^j f_{ij}(\bar{x}).$$

Then, for each i, j, $f_{ij}(\bar{x})$ has a circuit of size $O(d^2 s)$.

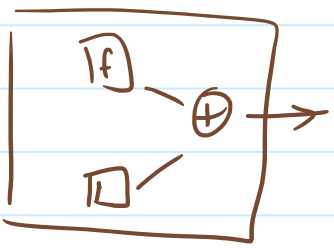
Sketch. Given ckt of f, split each gate g into d^2 gates, keeping track of coeffs of $y^i z^j$. ☐

Note: the above required access to circuit explicitly.

Q Given black board access to f, can we efficiently simulate black box access to its homogeneous components?

Example.
- If given black box to f, can we get $f+1$?

  Yes.



- $f = f_0 + f_1 + \cdots + f_d$; $f_i \to$ deg i homog. component

$f(x_1, \ldots, x_n)$

Introduce new variable t.

$x_i \longmapsto x_i t$

$$f(x_1 t, \ldots, x_n t) = f_0 + t f_1 + t^2 f_2 + \cdots + t^d f_d.$$

$\in \mathbb{F}[\bar{x}][t]$.

We wish to now find the coefficients of t above.

Let $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n$.

Th.

Let $a = (a_1, \ldots, a_n) \in \mathbb{F}^n$.

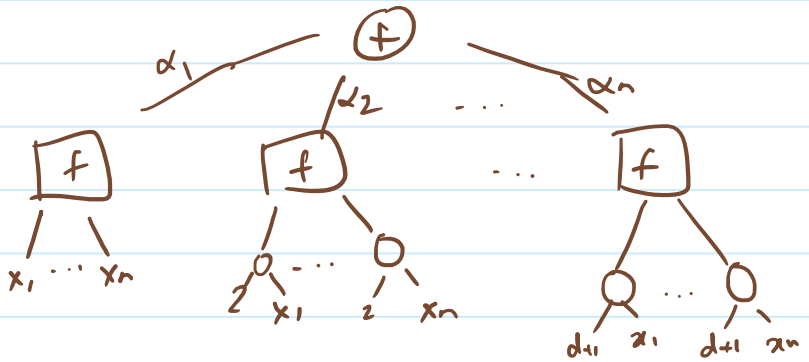
Then,

$$g_a(t) := f(a_1 t, \ldots, a_n t) = f_0(a) + t \cdot f_1(a) + \cdots + t^d f_d(a).$$

Can get $f_i(a)$ by interpolation.

$$f_i(a) = \alpha_{1,i}\, g_a(1) + \cdots + \alpha_{d+1,i}\, g_a(d+1).$$

$\alpha_{j,i}$ depends only on i



- Similarly, one can do the coefficient of $y^i z^j$ problem by bivariate interpolation.

Algo. Essentially the same as that in the bivariate case.

$$f \in \mathbb{F}[x_1, \ldots, x_n][y] = \mathbb{F}[\bar{x}][y]$$

① Preprocessing : - Not a p^{th} power → homework will assume char = 0 onwards anyway
- Monic in y ⇝ $f \in \mathbb{F}[\bar{x}][y]$
- Square free

$f(0, \ldots, 0, y)$ ⇝ square free

→ Can simulate circuit of $x_i \mapsto x_i + \alpha_i y$.

Choose α_i appropriately.

In fact, the transformation can be done with a blackbox.

Check if

Check if $\gcd_y\left(f, \frac{\partial f}{\partial y}\right) = 1$ using resultant. Everything as in bivariate goes through.

↓

we can form the matrix as before by our earlier discussion on homogeneous components

If $\neq 1$, compute GCD using Euclid as before.

Similarly, can make sure $f(\bar{0}, y)$ is sq. free as before.

② Find an approximate root of $f \in \mathbb{F}[\bar{x}][y]$.

Want $g_k(\bar{x})$ s.t. $f_k(\bar{x}, g_k(\bar{x})) = 0 \mod \langle \bar{x} \rangle^k$.

$\langle x_1, \ldots, x_n \rangle^k$

- $f(\bar{0}, y) \in \mathbb{F}[y]$.
  Want $\alpha \in \mathbb{F}$ s.t. $f(\bar{0}, \alpha) = 0$. (Univariate root finder.)
  Then, $f(\bar{x}, \alpha) \equiv 0 \mod \langle x \rangle$.

- Find $\alpha_0, \alpha_{11}, \ldots, \alpha_{1n} \in \mathbb{F}$ s.t.
  $f(\bar{x}, \alpha_0 + \alpha_{11} x_1 + \cdots + \alpha_{1n} x_n) \equiv 0 \mod \langle \bar{x} \rangle^2$.
  ⋮
  $f(\bar{x}, \alpha_0 + \alpha_1(\bar{x}) + \cdots + \alpha_k(\bar{x})) \equiv 0 \mod \langle x \rangle^k$.
  $\alpha_i \to$ homogeneous deg i

Let us mimic the bivariate case to get α_1.

- Have: $\alpha_0 \in \mathbb{F}$ s.t. ① $f(\bar{x}, \alpha_0) = 0 \mod \langle \bar{x} \rangle$.
  ② $\frac{\partial f}{\partial y}(\bar{0}, \alpha_0) \neq 0$. (Sq free)

Want: Homog. linear poly $\alpha_1(\bar{x}) = \alpha_{11} x_1 + \alpha_{12} x_2 + \cdots + \alpha_{1n} x_n$ s.t.

$f(\bar{x}, \alpha_0 + \alpha_1(\bar{x})) \equiv 0 \mod \langle x \rangle^2$.

Soln: Taylor + Truncation as before.

We have

$$f(\bar{x}, \alpha_0 + \alpha_1(\bar{x})) = f(\bar{x}, \alpha_0) + \alpha_1(\bar{x}) \cdot \frac{\partial f}{\partial y}(\bar{x}, \alpha_0) + \frac{\alpha_1(\bar{x})^2}{2!} \frac{\partial^2 f}{\partial y}(\bar{x}, \alpha_0) + \cdots$$

Since α_1 is deg 1. homog., we wish to solve

$$f(\bar{x}, \alpha_0) + \alpha_1(\bar{x}) \cdot \frac{\partial f}{\partial y}(\bar{x}, \alpha_0) = 0 \mod \langle \bar{x} \rangle^2.$$

The above is equivalent to

$$f(\bar{x}, \alpha_0) + \alpha_1(\bar{x}) \frac{\partial f}{\partial y}(\bar{0}, \alpha_0) = 0 \mod \langle \bar{x} \rangle^2.$$

↳ nonzero field constant
Can divide and solve.

Can similarly get the higher $g_k(\bar{x})$.

$$g_k(\bar{x}) = g_{k-1}(\bar{x}) - \frac{f(\bar{x}, g_{k-1}(\bar{x}))}{\frac{\partial f}{\partial y}(\bar{0}, \alpha_0)} \mod \langle \bar{x} \rangle^k$$

↳ easy to get ckt -
Only nontrivial part is $f(\bar{x}, g_{k-1}(\bar{x}))$.
Put circuit of $g_{k-1}(\bar{x})$ at bottom of f where the input of y goes.

Note this is a bootstrap construction!

How do we get mod? We don't! Just take

the boxed formula (without the mod). It still has the property we want!

$$\text{ckt size}(g_k) = \Theta\left(\text{ckt size}(g_{k-1}) + \text{ckt size}(f)\right).$$
$$\leq O(k \cdot s).$$

Stop at $k = 2d^2+1$.

⑤ Linear system.

Recall. Bivariate: Want $H(x,y) \neq 0$ s.t.

① $d_y(H) < d$,

② $d_x(H) \leq d$,

③ $H(x, g_k(x)) = 0 \mod \langle x \rangle^k$.

We showed: (I) Reducible $f \Rightarrow$ nonzero solution

(II) Any nonzero solution H gives $\gcd(f, H) \neq 1$ if f reducible.

Now, we want $H(\bar{x}, y)$ s.t.

① $d_y(H) < d$,

② $d_{\bar{x}}(H) \leq d$,

③ $H(\bar{x}, g_k(\bar{x})) = 0 \mod \langle x \rangle^k$, $\left(k = 2d^2+1\right)$

④ Non-zero.

The same proof and everything works.
However, last time we had a smaller linear system.
This time, it will be something like $\binom{n+k}{k}$.

↳ too big.

Final Algorithm:

① Preprocess as before

② View $f \in \mathbb{F}(x_1, \ldots, x_n)[y]$.

Define $F \in \mathbb{F}(\bar{x})[t, y]$ as

$$F(t, y) := f(tx_1, \ldots, tx_n, y).$$

How do we relate the factorisations?

① f factors $\Rightarrow$ F factors

$f = g_1 g_2 \rightsquigarrow F = G_1 G_2$

② $F = G_1 G_2 \hookrightarrow f = g_1 g_2$ ($t = 1$)

We can ensure G_1 & G_2 are monic in y (Gauss) of deg ≥ 1. Thus, putting $t = 1$ keeps the factorisation non trivial.

Now factor F using $F(y, t) = \sum_{i,j} F_{ij}\, y^i t^j$.

↳ we have these, from the beginning.

Lecture 21 (22-10-2021)

22 October 2021 17:37

Hardness v/s Randomness

Hardness: Lower bounds for arithmetic circuits:

$\{f_n\}_{n \in \mathbb{N}}$ — Explicit

(0) f_n is n-variate

(1) $\exists c \in \mathbb{N}$ s.t. $\forall n^{\text{th}}$ $\deg(f_n) \leq n^c$

(2) $\forall n \gg 0$, f_n does NOT have a small ckt, i.e., $\exists c_2 \in \mathbb{N}$ s.t. f_n cannot be computed by a ckt of size $< n^{c_2}$.

(or can think of c_2 as poly $\log n$)

Example. $\{\text{Perm}_n\}_{n \in \mathbb{N}}$ ① $\deg(\text{Perm}_n) = n$,

② # var $= n^2$, (not exactly what we said earlier)

③ Perm does not have poly(n) ckts. (Believed!)

Randomness: Deterministic algorithms for polynomial identity testing (PIT)

I/p : ckt C, n-variables, deg d, size s

Goal : $C \overset{?}{\equiv} 0$

Randomised algo: ① $S \subseteq \mathbb{F}$, $|S| \geq 100d$,

② Pick $a \in_{\text{randomly}} S^n$, query $C(a)$.

For a nonzero C,

For a nonzero C,

$$\Pr_{a \in S^n}[C(a) = 0] \leq \frac{1}{100}.$$

Naive deterministic algo:

① $S \subseteq \mathbb{F}$, $|S| = d+1$.

② Query C on all of S^n.

If $C|_{S^n} \equiv 0$, then $C \equiv 0$.

Running time $\leq (d+1)^n \cdot$ Query time

Thm. [Kabanets - Impagliazzo '04]

$s, d : \mathbb{N} \to \mathbb{N}$ poly bounded.

If $\{\text{Perm}_n\}_n$ does not have poly(n) size ckts, then there is a det. algo for PIT for ckts of size $s(n)$, deg $d(n)$ that runs in time poly$(n)^{n^\varepsilon}$ for some $\varepsilon < 1$. (They also proved the other direction.)

Proof. The PIT algorithm.

Find polynomials $\phi_1, \ldots, \phi_n$ on m-variables $y_1, \ldots, y_m$ s.t.

① $m < n$ (think $m = n^\varepsilon$),

② deg (ϕ_i) is "small" (poly(n)),

③ $\forall$ ckts C, n-variate, size $s(n)$, deg $d(n)$

$C(x_1, \ldots, x_n) \neq 0$ iff $C(\phi_1, \ldots, \phi_n) \neq 0$,

④ ϕ_i's are explicit, i.e., can evaluate somewhat efficiently.

Hitting set generators

How does the above help?

I/p : C, n-vars, deg d, size s.

Algorithm : ① $\tilde{C}(y_1, \ldots, y_m) = C(\phi_1(\bar{y}), \ldots, \phi_n(\bar{y}))$.

② $\deg(\tilde{C}) \leq \deg(C) \cdot \left(\max_i \deg(\phi_i(\bar{y}))\right)$

$\leq (\text{poly } n)^2 = \text{poly } n$

$\#\text{vars}(\tilde{C}) \leq m \quad (\sim n^\varepsilon)$

③ Invoke the trivial algo. on $\tilde{C}$ and we're done.

Easy to note : Algo is correct. $(C \equiv 0 \iff \tilde{C} \not\equiv 0)$

Running time : ② $\text{PIT}(\tilde{C}) \rightsquigarrow (\text{poly}(n))^m = (\text{poly}(n))^{n^\varepsilon}$

$(d \cdot \deg(\phi) + 1)^{n^\varepsilon}$

① $C \longmapsto \tilde{C}$

$\check{C}$

Running time : $(d \cdot \deg(\phi) + 1)^{n^\varepsilon} \cdot \left(\text{Time to Compute } \phi \text{ on all of } S^n\right)$

(Assuming const to compute C)

① Constructing a good enough hitting set generator (HSG) suffices.

② Use hardness of permanent to construct HSG.

Let us first try to construct HSG for $m = n-1$.

$$y_1, \ldots, y_m \longrightarrow \phi_1(\bar{y}) = y_1, \ldots, \phi_{n-1}(\bar{y}) = y_{n-1} = y_m, \quad \phi_n(\bar{y}) = \mathrm{Perm}_n(\bar{y})$$

(deal with issues when m not perf. square)

Prop: ① $m < n$,
② $\deg(\phi) \sim \sqrt{n}$,
③ $C \neq 0 \Rightarrow C(\phi_1, \ldots, \phi_n) \neq 0$.
④ $n^{\sqrt{n}}$ time computable.

only need to check this!

Proof $C(x_1, \ldots, x_n) \longrightarrow C(\phi_1, \ldots, \phi_n)$
$C(y_1, \ldots, y_{n-1}, \mathrm{Perm}(\bar{y}))$.

Let us do it in multiple steps.

$$C(x_1, \ldots, x_n) \to C(y_1, x_2, \ldots x_n) \to C(y_1, y_2, x_3, \ldots, x_n) \to \vdots \to C(y_1, \ldots, y_{n-1}, x_n) \to C(y_1, \ldots, y_{n-1}, \mathrm{Perm}(\bar{y}))$$

all of these are nonzero if $C \neq 0$.

$C(y_1, \ldots, y_{n-1}, x_n) \not\equiv 0$ AND $C(y_1, \ldots, y_{n-1}, \mathrm{Perm}(\bar{y})) \equiv 0$

$\Updownarrow$

$(x_n - \mathrm{Perm}(\bar{y})) \mid C(\bar{y}, x_n)$

last class

$\Downarrow$

X_n - Perm($\overline{Y}$) has a "small" = poly(n) ckt

$\Downarrow$

Contradicts that Perm($\overline{Y}$) does not have small ckts.

Thus, $\phi_1, \ldots, \phi_n$ is indeed an HSG.

But now we need to do it for $m = n^{\epsilon}$.

(Nisan-Wigderson Construction)

Combinatorial designs

$T_1, \ldots, T_n \subseteq \{y_1, \ldots, y_m\}$

① Explicit construction (poly(n)),

② $|T_i| = \sqrt{m}$,

③ $|T_i \cap T_j| \leq \delta = \delta(\epsilon)$.

Thm. Deterministic efficient construction of combinatorial designs.

Using designs for HSG:

$\phi_i = \text{Perm}(T_i)$

Need to show:

C, size s, deg d, n-var nonzero $\Rightarrow$ $C(\phi_1, \ldots, \phi_n) \neq 0$.

(other properties easy to check)

Assume $C \not\equiv 0$ but $\tilde{C} \equiv 0$.

$C(x_1, \ldots, x_n) \neq 0 \longrightarrow C(\phi_1, x_2, \ldots) \neq 0$

$\longrightarrow C(\phi_1, \ldots, \phi_{i-1}, x_i, \ldots) \neq 0$

$\longrightarrow C(\ldots, \phi_i, x_{i+1}, \ldots) = 0$

$\rightarrow C(\phi_1, \ldots, \phi_n) = 0.$

Thus, we have $C(\phi_1, \ldots, \phi_{i-1}, X_i, \ldots, X_n) \neq 0$

AND

$C(\phi_1, \ldots, \phi_i, X_{i+1}, \ldots, X_n) = 0.$

$\Rightarrow (X_i - \phi_i) \mid C(\phi_1, \ldots, \phi_{i-1}, X_i, \ldots, X_n).$

variables are X_i, T_i

vars are $X_i, \ldots, X_n, T_1, \ldots, T_{i-1}$

{ "relevant" vars

$X_i, T_1 \cap T_i, \ldots, T_{i-1} \cap T_i$

Set all irrelevant variables so that

$C(\phi_1, \ldots, \phi_{i-1}, X_i, \ldots, X_n)(\bar{a}) \neq 0.$

$\uparrow \mathbb{F}^{\sim}$

$C_2 = C(\phi_1(\bar{a}, T_1 \cap T_i), \ldots, \phi_{i-1}(\bar{a}, T_{i-1} \cap T_i), X_i, \ldots).$

Claim: ① C_2 has a not-too-large ckt. ← very few vars.

② $X - \text{Perm}(T_i) \mid C_2.$ ← easy to see

As before, we are done. ∎

Lecture 22 (26-10-2021)

26 October 2021 17:22

I/P: n, in binary. (Size $\approx \log n$)

Goal: Find if n prime.

Theorem (Agrawal - Biswas '96-ish)

There is a randomised algo. A s.t. on input $n \in \mathbb{N}$:
① A runs in time poly $(\log n)$ always.
② If n is prime, A accepts n.
③ If n is composite, A rejects with pr $= 0.9$.

Attempt 1. Fermat's Little Theorem based primality testing.

① Pick $a \in \{1, \dots, n-1\}$.
② Check if $a^{n-1} \equiv 1 \bmod n$.
③ If YES, accept n.
Else, reject n.

→ Runs in poly $(\log n)$
→ Accepts primes.
→ What about composites?

Carmichael Numbers: Composite n s.t.
$\forall a \in \mathbb{Z}_n^*$, $a^{n-1} \equiv 1 \pmod n$.

Fact: Infinitely many such numbers exist.

Agrawal - Biswas Test:

Lemma $(x+1)^n = x^n + 1$ in $\mathbb{Z}_n[x]$
$\Updownarrow$
n is a prime.

Clear. ⇒ if $q^k \| n$, then $q^k \nmid \binom{n}{q}$.

A-B Algorithm

① Check if $n = a^b$ for $a, b > 1$.
① Pick $d = \Theta(\log n)$.
② Pick $u(x) \in \mathbb{Z}_n[x]$, uniformly at random, monic, deg d.
③ Check if
$$u(x) \mid (x+1)^n - x^n - 1 \quad \text{in } \mathbb{Z}_n[x].$$
If yes, accept. Else, reject.

① Time complexity: poly $(\log n)$.

② If n prime, it is correct.

③ For composite n, it rejects with some "decent" prob.
↳ T. show.

Claim. Let n be composite and $q \mid n$ be prime.
Then,
$$(x+1)^n - x^n - 1 \neq 0 \quad \text{in } \mathbb{Z}_q[x].$$

Proof. Write $n = q^k m$ with $q \nmid m$.

$(X+1)^n - X^n - 1 \neq 0$ in $\mathbb{F}_q[X]$.

Proof. Write $n = q^k m$ with $q \nmid m$.

$$\binom{n}{q^k} = \binom{n}{q^k} \cdot \frac{n-1}{q^k-1} \cdot \cdots \cdot \frac{n-q^k+1}{1} \not\equiv 0 \bmod p. \quad \square$$

Take away: If n - composite and $q \mid n$ a prime divisor, then
$(X+1)^n - X^n - 1 \neq 0$ in $\mathbb{Z}_p[X]$.

Claim. If n is composite, then

$$\Pr_u \left[u \text{ divides } (X+1)^n - X^n - 1 \right] < 1 - \frac{1}{2d}$$

Claim $\Rightarrow$ Rejection prob. $\geq \frac{1}{2d} = \frac{1}{\Theta(\log n)}$.
Repeat algo $10 \log n$ times + reject with probability ≥ 1.

Thm, the claim finishes the algorithm.

Proof of claim: By contradiction.

Suppose $\Pr_u \left[u \text{ divides } \cdots \right] \geq 1 - 1/2d$

for some composite n.

u uniformly, deg d, monic

$$\Pr_{\tilde{u} \in_{\text{unif}} \mathbb{F}_q[X] \text{ deg } d, \text{ monic}} \left[\tilde{u} \mid (X+1)^n - X^n - 1 \text{ in } \mathbb{F}_q[X] \right] \geq 1 - 1/2d.$$

$\therefore (X+1)^n - X^n - 1$ is a nonzero poly having

$$\geq q^d \left(1 - \frac{1}{2d}\right) \text{ monic, deg } d \text{ factors.}$$

OTOH, there are $\geq \frac{q^d}{d}$ monic, deg d irred polys in $\mathbb{F}_q[X]$.

$\therefore \geq \frac{q^d}{2d}$ distinct irreducible deg d polyns. divide $(X+1)^n - X^n - 1$ in $\mathbb{F}_q[X]$.

But $\frac{q^d}{2d} \cdot d > n$ since $d = \Theta(\log n)$. $\longrightarrow\!\!\leftarrow$