Lecture 8: Univariate Polynomial Factoring over Finite Fields

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science

rafael.oliveira.teaching@gmail.com

February 3, 2021

Overview

- Why facotring?
- Warm-up: computing square roots over finite fields
- Extending the Algorithm: Cantor-Zassenhaus Algorithm
- Acknowledgements

- Why facotring?
- Warm-up: computing square roots over finite fields
- Extending the Algorithm: Cantor-Zassenhaus Algorithm
- Acknowledgements

Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)

Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)

Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)

Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)

Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)
  3. any Euclidean domain is a PID (and hence UFD)

Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)
  3. any Euclidean domain is a PID (and hence UFD)
  4. $\mathbb{Q}[x, y]$ is a UFD but *not* a PID

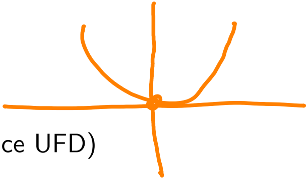
Why Factoring?

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)
  3. any Euclidean domain is a PID (and hence UFD)
  4. $\mathbb{Q}[x, y]$ is a UFD but *not* a PID
  5. $\mathbb{Z}[x]$ is a UFD but *not* a PID

Why Factoring?

$(y-x^2)\,x \cdot y$

- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)
  3. any Euclidean domain is a PID (and hence UFD)
  4. $\mathbb{Q}[x, y]$ is a UFD but *not* a PID
  5. $\mathbb{Z}[x]$ is a UFD but *not* a PID



- Over UFDs, it makes sense to talk about *greatest common divisor* and they are very useful in symbolic computation and algebraic geometry.
  1. Factoring polynomials
  2. Irreducible components of hypersurfaces
  3. Multiplicity of roots, factors and components

$f(x)$ α root of $x \iff x-\alpha \mid f(x)$

Square Roots over $\mathbb{F}_p$

$x^p - x = x(x^{p-1} - 1)$
$= x(x^{\frac{p-1}{2}} - 1)(x^{\frac{p-1}{2}} + 1)$

α square root of a in $\mathbb{F}_p$
$\Updownarrow$
$x - \alpha$ factor $x^2 - a$

- **Input:** Let $p \in \mathbb{N}$ be an odd prime and $a \in \mathbb{F}_p$
- **Output:** factors of $x^2 - a$ over $\mathbb{F}_p[x]$

1. If $x^2 - a$ factors, it will factor as $(x - \alpha)(x + \alpha)$ for some $\alpha \in \mathbb{F}_p$
2. By Fermat's little theorem, $b^p - b \equiv 0 \mod p$ for any $b \in \mathbb{F}_p$, so

$x^2 - x = x(x-1)$
$= (x-0)(x-1)$

$$x^p - x = \prod_{b \in \mathbb{F}_p} (x - b)$$

$x^3 - x = x(x-1)(x-2)$ over $\mathbb{F}_3 = \mathbb{Z}_3$

$x - 2 \equiv x + 1 \mod 3$
$x(x-1)(x+1) = x^3 - x$

3. So both $x - \alpha$, $x + \alpha$ divide $x^p - x$
4. $x^p - x = x \cdot f_1(x) \cdot f_2(x)$, where

$$f_1(x) = x^{(p-1)/2} - 1 \quad \text{and} \quad f_2(x) = x^{(p-1)/2} + 1$$

5. If α is root of f_1 and $-\alpha$ is root of f_2, then $\gcd(f_1, x^2 - a) = x - \alpha$ and we can factor!

$\deg(f_1) \cdot \deg(x^2 - a) = p$

6. *Two issues*: will this split always happen? And can we avoid computing that GCD?

over $\mathbb{F}_p$ $\log p$
running time poly($\log p$)

Square roots over $\mathbb{F}_p$

- α is a root of f_1 and $-\alpha$ is a root of f_2 iff

$$\alpha^{(p-1)/2} \equiv 1 \quad \text{and} \quad (-\alpha)^{(p-1)/2} \equiv -1 \quad \text{mod } p$$

$f_1(\alpha) \equiv 0$ $\qquad$ $f_2(-\alpha) \equiv 0$

Square roots over $\mathbb{F}_p$

- α is a root of f_1 and $-\alpha$ is a root of f_2 iff

$$\alpha^{(p-1)/2} \equiv 1 \quad \text{and} \quad (-\alpha)^{(p-1)/2} \equiv -1$$

- If $p \equiv 3 \mod 4$ we know that f_1, f_2 split the roots of $x^2 - a$ and thus we are good!
- How do we make this work in general?

$$(-\alpha)^{\frac{p-1}{2}} = (-1)^{\frac{p-1}{2}} \cdot \alpha^{\frac{p-1}{2}} = -\alpha^{\frac{p-1}{2}}$$

$$p \equiv 3 \mod 4 \implies p = 4k+3 \quad k \in \mathbb{Z}$$

$$\frac{p-1}{2} = 2k+1 \text{ odd}$$

Square roots over $\mathbb{F}_p$

- α is a root of f_1 and $-\alpha$ is a root of f_2 iff

$$\alpha^{(p-1)/2} \equiv 1 \quad \text{and} \quad (-\alpha)^{(p-1)/2} \equiv -1$$

- If $p \equiv 3 \mod 4$ we know that f_1, f_2 split the roots of $x^2 - a$ and thus we are good!
- How do we make this work in general?
- Factoring $g(x) = x^2 - a$ equivalent to factoring

$$h(x) = (x - d)^2 - c^2 a$$

Square roots over $\mathbb{F}_p$

- α is a root of f_1 and $-\alpha$ is a root of f_2 iff

$$\alpha^{(p-1)/2} \equiv 1 \quad \text{and} \quad (-\alpha)^{(p-1)/2} \equiv -1$$

- If $p \equiv 3 \mod 4$ we know that f_1, f_2 split the roots of $x^2 - a$ and thus we are good!
- How do we make this work in general?
- Factoring $g(x) = x^2 - a$ equivalent to factoring

$$h(x) = (x - d)^2 - c^2 a$$

- $g(x) = (x - \alpha)(x + \alpha)$ if, and only if,

$$h(x) = (x - d - c\alpha)(x - d + c\alpha)$$

$g(\alpha) = 0$

$h(d + c\alpha) = (d + c\alpha - d)^2 - c^2\alpha^2$

$= (c\alpha)^2 - c^2\alpha^2 = 0$

Square roots over $\mathbb{F}_p$

- α is a root of f_1 and $-\alpha$ is a root of f_2 iff

$$\alpha^{(p-1)/2} \equiv 1 \quad \text{and} \quad (-\alpha)^{(p-1)/2} \equiv -1$$

- If $p \equiv 3 \mod 4$ we know that f_1, f_2 split the roots of $x^2 - a$ and thus we are good!
- How do we make this work in general?
- Factoring $g(x) = x^2 - a$ equivalent to factoring

$$h(x) = (x - d)^2 - c^2 a$$

- $g(x) = (x - \alpha)(x + \alpha)$ if, and only if,

$$h(x) = (x \overbrace{- d - c\alpha}^{\beta_1})\overset{\neq}{\,}(x \overbrace{- d + c\alpha}^{\beta_2})$$

- So, if g factors, we can try to find "good" (c, d) so that $f_1(x), f_2(x)$ "split" the factors of h

$f_1(\beta_1) = 0$ and $f_2(\beta_2) = 0$

Square roots over $\mathbb{F}_p$

- What if we pick $c, d \in \mathbb{F}_p$ at random? What is the probability that $f_1(x)$ has only one of the roots of h as a factor?

Square roots over $\mathbb{F}_p$

- What if we pick $c, d \in \mathbb{F}_p$ at random? What is the probability that $f_1(x)$ has only one of the roots of h as a factor?
- If $a_1 \neq a_2$ and $b_1 \neq b_2$ over $\mathbb{F}_p$:

$$\Pr_{c,d}[c \cdot a_1 + d = b_1 \text{ and } c \cdot a_2 + d = b_2] = \frac{1}{p^2}$$

$\boxed{a_1 \neq a_2}$

$c \cdot a_1 + d$, $c \cdot a_2 + d$ linearly independent

$\mathbb{F}_p^2$

$$\underbrace{\begin{pmatrix} a_1 & 1 \\ a_2 & 1 \end{pmatrix}}_{\text{invertible}} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Square roots over $\mathbb{F}_p$

- What if we pick $c, d \in \mathbb{F}_p$ at random? What is the probability that $f_1(x)$ has only one of the roots of h as a factor?
- If $a_1 \neq a_2$ and $b_1 \neq b_2$ over $\mathbb{F}_p$:

$$\Pr_{c,d}[c \cdot a_1 + d = b_1 \text{ and } c \cdot a_2 + d = b_2] = \frac{1}{p^2}$$

- On the other hand:

$$\Pr_{b_1}[b_1 \text{ is root of } \overbrace{x^{(p-1)/2}}^{\text{has } \frac{p-1}{2} \text{ roots in } \mathbb{F}_p}] = \frac{1}{2}$$

$$\Pr_{b_2}[b_2 \text{ is not root of } \underbrace{x^{(p-1)/2}}_{\text{has } \frac{p-1}{2} \text{ roots}}] = \frac{1}{2}$$

Square roots over $\mathbb{F}_p$

$a_1 = \alpha$ $a_2 = -\alpha$

- What if we pick $c, d \in \mathbb{F}_p$ at random? What is the probability that $f_1(x)$ has only one of the roots of h as a factor?
- If $a_1 \neq a_2$ and $b_1 \neq b_2$ over $\mathbb{F}_p$:

$h(x) = (x - b_1)(x - b_2)$

$$\Pr_{c,d}[c \cdot a_1 + d = b_1 \text{ and } c \cdot a_2 + d = b_2] = \frac{1}{p^2}$$

get split

- On the other hand:

$\frac{1}{4} \times 2$

f_1

$$\Pr_{b_1}[b_1 \text{ is root of } x^{(p-1)/2}] = \frac{1}{2}$$

f_2

$$\Pr_{b_2}[b_2 \text{ is not root of } x^{(p-1)/2}] = \frac{1}{2}$$

- Thus, with probability $\approx 1/2$, uniform random choice of c, d gives us that $f_1(x)$ splits $h(x)$

Square Root Algorithm

1. Pick random $c, d \in \mathbb{F}_p$ and compute $h(x)$

$$h(x) = (x - d - c\alpha)(x - d + c\alpha)$$

Square Root Algorithm

1. Pick random $c, d \in \mathbb{F}_p$ and compute $h(x)$
2. Compute $\ell(x) \equiv f_1(x) \mod h(x)$

$$\gcd(f_1, h) = \gcd(f_1 \bmod h, h)$$

$$x^{\frac{p-1}{2}} \bmod h(x) \qquad O(\log p)$$

Square Root Algorithm

1. Pick random $c, d \in \mathbb{F}_p$ and compute $h(x)$
2. Compute $\ell(x) \equiv f_1(x) \mod h(x)$
3. Compute $r(x) = \gcd(h(x), \ell(x))$

Square Root Algorithm

1. Pick random $c, d \in \mathbb{F}_p$ and compute $h(x)$
2. Compute $\ell(x) \equiv f_1(x) \mod h(x)$
3. Compute $r(x) = \gcd(h(x), \ell(x))$
4. If $r(x) = 1$ or $r(x) = h(x)$, go back to step 1

Square Root Algorithm

1. Pick random $c, d \in \mathbb{F}_p$ and compute $h(x)$
2. Compute $\ell(x) \equiv f_1(x) \mod h(x)$
3. Compute $r(x) = \gcd(h(x), \ell(x))$
4. If $r(x) = 1$ or $r(x) = h(x)$, go back to step 1
5. Otherwise we found a root of $h(x)$

$\deg(r) = 1$

$\Rightarrow r$ proper factor of h.

~~Example~~ Fermat's little theorem

- ~~$f(x) = 6x^3 - 42x^2 + 72x - 60$ and $g(x) = 2x^2 - 6x - 20$~~

$p \mid \to \dfrac{p!}{a!\,(p-a)!}$, $p \nmid a!\,(p-a)!$

$$a^p \equiv [(a-1)+1]^p \equiv$$

$$\equiv (a-1)^p + \cancel{\binom{p}{1}(a-1)^1}_{0} + \cancel{\binom{p}{2}(a-1)^2}_{0} + \cdots + \cancel{\binom{p}{p-1}(a-1)^{p-1}}_{0}$$

$$+ \underline{1}$$

$\boxed{p=5}$

$$\equiv (a-1)^p + 1 = (a-2)^p + 2 \equiv \cdots$$

$0 \equiv p \boxed{\dfrac{(p-1)}{2}}$

$$\equiv a \mod p$$

$\to \dfrac{p(p-1)(p-2)}{3}$

$$\Longrightarrow \quad a^p - a \equiv 0 \mod p$$

- Why facotring?
- Warm-up: computing square roots over finite fields
- **Extending the Algorithm: Cantor-Zassenhaus Algorithm**
- Acknowledgements

Challenges to generalize previous algorithm

Want : factor $f(x) \in \mathbb{F}_q[x]$

- We can extend the previous algorithm to factor any polynomial over $\mathbb{F}_{p^k}$, but we need to deal with the following issues

$q = p^k$

Challenges to generalize previous algorithm

- We can extend the previous algorithm to factor any polynomial over $\mathbb{F}_{p^k}$, but we need to deal with the following issues
- Our algorithm can only factor degree 1 factors

Need to find polynomials which are divisible by irreducible polynomials of higher degree.

Challenges to generalize previous algorithm

- We can extend the previous algorithm to factor any polynomial over $\mathbb{F}_{p^k}$, but we need to deal with the following issues
- Our algorithm can only factor degree 1 factors

  Need to find polynomials which are divisible by irreducible polynomials of higher degree.
- Cannot factor polynomials which have square factors

$$f(x) = (x-a)^2$$

previous algorithm: could <u>not</u> split $f(x)$

Challenges to generalize previous algorithm

- We can extend the previous algorithm to factor any polynomial over $\mathbb{F}_{p^k}$, but we need to deal with the following issues
- Our algorithm can only factor degree 1 factors

  Need to find polynomials which are divisible by irreducible polynomials of higher degree.
- Cannot factor polynomials which have square factors
- The trick to "split" a polynomial through a high degree polynomial needs to be generalized to work for higher-degree irreducible factors.

Challenges to generalize previous algorithm

$p^k \; (= q)$ $\quad$ p odd

- We can extend the previous algorithm to factor any polynomial over $\mathbb{F}_{p^k}$, but we need to deal with the following issues
- Our algorithm can only factor degree 1 factors

  Need to find polynomials which are divisible by irreducible polynomials of higher degree.
- Cannot factor polynomials which have square factors
- The trick to "split" a polynomial through a high degree polynomial needs to be generalized to work for higher-degree irreducible factors.
- Algorithm only works over odd prime fields.

$\mathbb{F}_{2^k}[x]$

Challenges to generalize previous algorithm

- We can extend the previous algorithm to factor any polynomial over $\mathbb{F}_{p^k}$, but we need to deal with the following issues
- Our algorithm can only factor degree 1 factors

  Need to find polynomials which are divisible by irreducible polynomials of higher degree.
- Cannot factor polynomials which have square factors
- The trick to "split" a polynomial through a high degree polynomial needs to be generalized to work for higher-degree irreducible factors.
- Algorithm only works over odd prime fields.
- From now on, let $q = p^k$ be a power of a prime.

Reducing to the Square-Free case

- Over finite fields, we can define the derivative of a polynomial in a formal way (and has similar properties to the usual derivative). If $f(x) = f_0 + f_1 x + \cdots + f_d x^d$ then

$$f'(x) = f_1 + 2 \cdot f_2 x + \cdots + d \cdot f_d \cdot x^{d-1}$$

$$x^{\ell} \longrightarrow \ell \cdot x^{\ell-1}$$

Reducing to the Square-Free case

- Over finite fields, we can define the derivative of a polynomial in a formal way (and has similar properties to the usual derivative). If $f(x) = f_0 + f_1 x + \cdots f_d x^d$ then
$$f'(x) = f_1 + 2 \cdot f_2 x + \cdots d \cdot f_d \cdot x^{d-1}$$
- The property that we will need is the one on square factors:

If $f = g^2 \cdot h$ for some polynomials $g, h \in \mathbb{F}_{p^k}[x]$, then
$$g \mid \gcd(f, f')$$

$$f' = 2g \cdot g' \cdot h + h' \cdot g^2$$

Reducing to the Square-Free case

- Over finite fields, we can define the derivative of a polynomial in a formal way (and has similar properties to the usual derivative). If $f(x) = f_0 + f_1 x + \cdots f_d x^d$ then
$$f'(x) = f_1 + 2 \cdot f_2 x + \cdots d \cdot f_d \cdot x^{d-1}$$
- The property that we will need is the one on square factors:

  If $f = g^2 \cdot h$ for some polynomials $g, h \in \mathbb{F}_{p^k}[x]$, then
$$g \mid \gcd(f, f')$$
- So we can get a square-free polynomial simply by dividing by the GCD:
$$\boxed{\frac{f}{\gcd(f, f')}}$$ is square-free

contains all original factors of f

this already "factored f a little bit"

Reducing to the Square-Free case

- Over finite fields, we can define the derivative of a polynomial in a formal way (and has similar properties to the usual derivative). If $f(x) = f_0 + f_1 x + \cdots f_d x^d$ then

$$f'(x) = f_1 + 2 \cdot f_2 x + \cdots d \cdot f_d \cdot x^{d-1}$$

- The property that we will need is the one on square factors:

If $f = g^2 \cdot h$ for some polynomials $g, h \in \mathbb{F}_{p^k}[x]$, then

$$g \mid \gcd(f, f')$$

- So we can get a square-free polynomial simply by dividing by the GCD:

$$\frac{f}{\gcd(f, f')}$$

- Need to be careful: what if $\gcd(f, f') = f$?
- This can happen iff $f' = 0$

Reducing to the Square-Free case

- Over finite fields, we can define the derivative of a polynomial in a formal way (and has similar properties to the usual derivative). If $f(x) = f_0 + f_1 x + \cdots f_d x^d$ then

$$f'(x) = f_1 + 2 \cdot f_2 x + \cdots d \cdot f_d \cdot x^{d-1}$$

- The property that we will need is the one on square factors:

If $f = g^2 \cdot h$ for some polynomials $g, h \in \mathbb{F}_{p^k}[x]$, then

$$g \mid \gcd(f, f')$$

- So we can get a square-free polynomial simply by dividing by the GCD:

$$\frac{f}{\gcd(f, f')}$$

- Need to be careful: what if $\gcd(f, f') = f$?
- This can happen iff $f' = 0$
- $f' = 0$ iff the only non-zero monomials of f are powers of p

$\mathbb{Z}_5[x]$ $\quad x^5 + x^{10} \longrightarrow \cancel{5} \cdot x^4 + \cancel{10} \cdot x^9 = 0$

Reducing to the Square-Free case

- Over finite fields, we can define the derivative of a polynomial in a formal way (and has similar properties to the usual derivative). If $f(x) = f_0 + f_1 x + \cdots f_d x^d$ then

$$f'(x) = f_1 + 2 \cdot f_2 x + \cdots d \cdot f_d \cdot x^{d-1}$$

- The property that we will need is the one on square factors:

If $f = g^2 \cdot h$ for some polynomials $g, h \in \mathbb{F}_{p^k}[x]$, then

$$g \mid \gcd(f, f')$$

- So we can get a square-free polynomial simply by dividing by the GCD:

$$\frac{f}{\gcd(f, f')}$$

$\mathbb{F}_q$

- Need to be careful: what if $\gcd(f, f') = f$?
- This can happen iff $f' = 0$
- $f' = 0$ iff the only non-zero monomials of f are powers of p
- Example: $x^3 + 2x^6 = (x + 2x)^3$ over $\mathbb{Z}_3[x]$

$\mathbb{F}_q$

Getting irreducible factors of higher degree

- To be able to find irreducible factors of high degree, need to find analogue of $\underline{x^q - x}$ for higher degree irreducibles

$$\underbrace{x^q - x = \prod_{a \in \mathbb{F}_p} (x - a)}$$

has all irreducible Polynomials of deg 1 and only irreducible factors of degree 1

[1] Will prove the moreover part later.

Getting irreducible factors of higher degree

- To be able to find irreducible factors of high degree, need to find analogue of $x^q - x$ for higher degree irreducibles

$$x^q - x = \prod_{a \in \mathbb{F}_p} (x - a)$$

- **Lemma:** $x^{q^d} - x$ is a multiple of any degree d irreducible polynomial over $\mathbb{F}_q[x]$. Moreover, if $g(x)$ is irreducible and divides $x^{q^d} - x$, then $\deg(g) = d$.[1]
  1. Let $g(x)$ be an irreducible polynomial of degree d over $\mathbb{F}_q[x]$

[1] Will prove the moreover part later.

Getting irreducible factors of higher degree

- To be able to find irreducible factors of high degree, need to find analogue of $x^q - x$ for higher degree irreducibles

$$x^q - x = \prod_{a \in \mathbb{F}_p} (x - a)$$

- **Lemma:** $x^{q^d} - x$ is a multiple of any degree d irreducible polynomial over $\mathbb{F}_q[x]$. Moreover, if $g(x)$ is irreducible and divides $x^{q^d} - x$, then $\deg(g) = d$.[1]
    1. Let $g(x)$ be an irreducible polynomial of degree d over $\mathbb{F}_q[x]$
    2. Let $\mathbb{K} = \mathbb{F}_q[x]/(g(x))$. $\mathbb{K}$ is a field which contains all polynomials of degree $\leq d - 1$
    3. Practice problem: prove this.

[1] Will prove the moreover part later.

Getting irreducible factors of higher degree

- To be able to find irreducible factors of high degree, need to find analogue of $x^q - x$ for higher degree irreducibles

$$x^q - x = \prod_{a \in \mathbb{F}_p} (x - a)$$

- **Lemma:** $x^{q^d} - x$ is a multiple of any degree d irreducible polynomial over $\mathbb{F}_q[x]$. Moreover, if $g(x)$ is irreducible and divides $x^{q^d} - x$, then $\deg(g) = d$.[1]
  1. Let $g(x)$ be an irreducible polynomial of degree d over $\mathbb{F}_q[x]$
  2. Let $\mathbb{K} = \mathbb{F}_q[x]/(g(x))$. $\mathbb{K}$ is a field which contains all polynomials of degree $\leq d - 1$
  3. Practice problem: prove this.
  4. Thus, for all $\alpha \in \mathbb{K}$, we have $\alpha^{|\mathbb{K}|} - \alpha = 0$

[1] Will prove the moreover part later.

Getting irreducible factors of higher degree

- To be able to find irreducible factors of high degree, need to find analogue of $x^q - x$ for higher degree irreducibles

$$x^q - x = \prod_{a \in \mathbb{F}_p} (x - a)$$

- **Lemma:** $x^{q^d} - x$ is a multiple of any degree d irreducible polynomial over $\mathbb{F}_q[x]$. Moreover, if $g(x)$ is irreducible and divides $x^{q^d} - x$, then $\deg(g) = d$.[1]
  1. Let $g(x)$ be an irreducible polynomial of degree d over $\mathbb{F}_q[x]$
  2. Let $\mathbb{K} = \mathbb{F}_q[x]/(g(x))$. $\mathbb{K}$ is a field which contains all polynomials of degree $\leq d - 1$
  3. Practice problem: prove this.
  4. Thus, for all $\alpha \in \mathbb{K}$, we have $\alpha^{|\mathbb{K}|} - \alpha = 0$
  5. Since $x \in \mathbb{K}$, we have that $x^{|\mathbb{K}|} - x \equiv 0 \mod g(x)$ $\quad g \mid x^{|\mathbb{K}|} - x$

[1] Will prove the moreover part later.

Getting irreducible factors of higher degree

- To be able to find irreducible factors of high degree, need to find analogue of $x^q - x$ for higher degree irreducibles

$$x^q - x = \prod_{a \in \mathbb{F}_p} (x - a)$$

$f_0 + f_1 x + \cdots + f_{d-1} x^{d-1}$

- **Lemma:** $x^{q^d} - x$ is a multiple of any degree d irreducible polynomial over $\mathbb{F}_q[x]$. Moreover, if $g(x)$ is irreducible and divides $x^{q^d} - x$, then $\deg(g) = d$.[1]
  1. Let $g(x)$ be an irreducible polynomial of degree d over $\mathbb{F}_q[x]$
  2. Let $\mathbb{K} = \mathbb{F}_q[x]/(g(x))$. $\mathbb{K}$ is a field which contains all polynomials of degree $\leq d - 1$

     $1, x, x^2, \ldots, x^{d-1}$
  3. Practice problem: prove this.
  4. Thus, for all $\alpha \in \mathbb{K}$, we have $\alpha^{|\mathbb{K}|} - \alpha = 0$
  5. Since $x \in \mathbb{K}$, we have that $x^{|\mathbb{K}|} - x \equiv 0 \mod g(x)$
  6. $|\mathbb{K}| = q^d$, since each polynomial of degree $\leq d - 1$ is a distinct element

[1] Will prove the moreover part later.

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

$\ell \leq g$

1. Iterate the following for $i = 1, 2, \ldots, \ell$

$$\underbrace{(x-1)(x-2)}_{g_1(x)} \underbrace{(x^2+x+1)(x^2+3x+1)}_{g_2(x)}$$

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

1. Iterate the following for $i = 1, 2, \ldots, \ell$
2. While $g(x)$ not a unit
   - Compute $\underline{g_i(x)} = \gcd(\underline{g(x)}, \underline{x^{q^i} - x})$

only has deg i factors

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

1. Iterate the following for $i = 1, 2, \ldots, \ell$
2. While $g(x)$ not a unit
   - Compute $g_i(x) = \gcd(g_i(x), x^{q^i} - x)$

To complete our full factorization algorithm, we need to generalize the factor splitting trick.

we can assume that
g is square-free and
g only has factors irreducible
of degree d.

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

1. Iterate the following for $i = 1, 2, \ldots, \ell$
2. While $g(x)$ not a unit
   - Compute $g_i(x) = \gcd(g_i(x), x^{q^i} - x)$

To complete our full factorization algorithm, we need to generalize the factor splitting trick.

- Now can assume that $g(x)$ is a product of irreducible factors of *same degree* d

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

1. Iterate the following for $i = 1, 2, \ldots, \ell$
2. While $g(x)$ not a unit
    - Compute $g_i(x) = \gcd(g_i(x), x^{q^i} - x)$

To complete our full factorization algorithm, we need to generalize the factor splitting trick.

- Now can assume that $g(x)$ is a product of irreducible factors of *same degree* d
- So we need to find a polynomial which is multiple of some of the factors of g, but not all.

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

1. Iterate the following for $i = 1, 2, \ldots, \ell$
2. While $g(x)$ not a unit
   - Compute $g_i(x) = \gcd(g_i(x), x^{q^i} - x)$

To complete our full factorization algorithm, we need to generalize the factor splitting trick.

- Now can assume that $g(x)$ is a product of irreducible factors of *same degree* d
- So we need to find a polynomial which is multiple of some of the factors of g, but not all.
- Here we simply use the polynomials

$$f_1(x) = x^{(q^d-1)/2} - 1 \quad \text{and} \quad f_2(x) = x^{(q^d-1)/2} + 1$$

$x^{\frac{q^d-1}{2}} - 1$ $\qquad$ $x^{\frac{q^d-1}{2}} + 1$

Algorithm to get degree d irreducible factors

Now we can factor $g(x) = g_1(x)g_2(x)\cdots g_\ell(x)$ where each $g_t(x)$ is a product of factors of degree exactly t

1. Iterate the following for $i = 1, 2, \ldots, \ell$
2. While $g(x)$ not a unit
   - Compute $g_i(x) = \gcd(g_i(x), x^{q^i} - x)$

To complete our full factorization algorithm, we need to generalize the factor splitting trick.

- Now can assume that $g(x)$ is a product of irreducible factors of *same degree* d
- So we need to find a polynomial which is multiple of some of the factors of g, but not all.
- Here we simply use the polynomials

$$f_1(x) = x^{(q^d-1)/2} - 1 \quad \text{and} \quad f_2(x) = x^{(q^d-1)/2} + 1$$

- But how do we perform the random step?

Splitting Trick

- In the warm-up part, we needed to get a random transformation of the roots, by making $g(x) = x^2 - a$ into $h(x) = (x - d)^2 - c^2 a$. How do we generalize this for higher degree irreducible polynomials?

Splitting Trick

- In the warm-up part, we needed to get a random transformation of the roots, by making $g(x) = x^2 - a$ into $h(x) = (x - d)^2 - c^2 a$. How do we generalize this for higher degree irreducible polynomials?
- We want a map that with high probability contains one of our irreducible factors of degree d

Splitting Trick

- In the warm-up part, we needed to get a random transformation of the roots, by making $g(x) = x^2 - a$ into $h(x) = (x-d)^2 - c^2 a$. How do we generalize this for higher degree irreducible polynomials?
- We want a map that with high probability contains one of our irreducible factors of degree d
- We also saw that for any $\underline{T(x)}$, the polynomial $\underline{T(x)}^{q^d} - T(x)$ is a $\underline{\text{multiple}}$ of any irreducible factor of degree d and

$$T(x)^{q^d} - T(x) = \underline{T(x)} \cdot \underline{f_1(T(x))} \cdot \underline{f_2(T(x))}$$

$g(x) \mid$

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

Splitting Trick

- In the warm-up part, we needed to get a random transformation of the roots, by making $g(x) = x^2 - a$ into $h(x) = (x-d)^2 - c^2 a$. How do we generalize this for higher degree irreducible polynomials?
- We want a map that with high probability contains one of our irreducible factors of degree d
- We also saw that for any $T(x)$, the polynomial $T(x)^{q^d} - T(x)$ is a multiple of any irreducible factor of degree d and

$$T(x)^{q^d} - T(x) = T(x) \cdot f_1(T(x)) \cdot f_2(T(x))$$

- **Lemma:** let $h(x) \in \mathbb{F}_q[x]$ be irreducible and of degree d, and let $D > d$. Then:

splitting step

$$\longrightarrow \Pr_{T(x)} [h(x) \mid f_1(T(x))] \approx \frac{1}{2}$$

$\deg(T) = D$

$\hat{T} \equiv T \mod h(x)$ $\quad$ $\hat{T} \in \mathbb{K} = \mathbb{F}_q[x]/(h(x))$

$\alpha^{q^d} - \alpha \equiv 0$ over $\mathbb{K}$ $\quad$ $\alpha \boxed{\widetilde{f_1}(\alpha)} f_2(\alpha) \equiv 0 \mod h$

Splitting Trick

- In the warm-up part, we needed to get a random transformation of the roots, by making $g(x) = x^2 - a$ into $h(x) = (x-d)^2 - c^2 a$. How do we generalize this for higher degree irreducible polynomials?
- We want a map that with high probability contains one of our irreducible factors of degree d
- We also saw that for any $T(x)$, the polynomial $T(x)^{q^d} - T(x)$ is a multiple of any irreducible factor of degree d and

$$T(x)^{q^d} - T(x) = T(x) \cdot f_1(T(x)) \cdot f_2(T(x))$$

2d

- **Lemma:** let $h(x) \in \mathbb{F}_q[x]$ be irreducible and of degree d, and let $D > d$. Then:

$\underline{T}$ mod $f_1 \cdot f_2$ 2d

$$\Pr_{T(x)} [h(x) \mid f_1(T(x))] \approx \frac{1}{2}$$

- **Lemma:** For any $\underline{T_1, T_2} \in \mathbb{F}_q[x]$ of degree $< d$, and irreducible polynomials $\underline{f_1, f_2} \in \mathbb{F}_q[x]$ of degree d

Chinese Remainder Thm

factors of g

$$\Pr_{T(x)} [T(x) \equiv \underline{T_1} \bmod f_1(x) \text{ and } T(x) \equiv \underline{T_2} \bmod f_2(x)] \approx \frac{1}{q^2}$$

where $T(x) \in \mathbb{F}_q[x]$ is of degree $\le 2d - 1$

Algorithm

- **Input**: ~~primitive polynomials $f, g \in \mathbb{Z}[x]$~~ $g(x) \in \mathbb{F}_q[x]$
- **Output**: ~~$h = \gcd(f, g)$~~
  irreducible factors of g

– get square-free part of g
by $g / \gcd(g, g')$

– get $g_i(x) = \prod$ (irreducible factors of g of deg i)
by $\gcd(g, x^{q^i} - x) =: g_i$

– random $T(x) \in \mathbb{F}_q[x]$ $d < \deg(T) < 2d$
get factor (w.h.p.) by $\gcd\left(g_d, T(x)^{\frac{q^d-1}{2}} - 1\right)$

Algorithm

- **Input**: primitive polynomials $f, g \in \mathbb{Z}[x]$
- **Output:** $h = \gcd(f, g)$
- Algorithm:
  1. Compute $b = \gcd(LT(f), LT(g))$, and set $B \in \mathbb{N}$

Algorithm

- **Input**: primitive polynomials $f, g \in \mathbb{Z}[x]$
- **Output:** $h = \gcd(f, g)$
- Algorithm:
  1. Compute $b = \gcd(LT(f), LT(g))$, and set $B \in \mathbb{N}$
  2. Pick random prime $p \in [2B, 4B]$
  3. Compute $p(x) = \gcd_{\mathbb{Z}_p[x]}(f, g)$

Algorithm

- **Input**: primitive polynomials $f, g \in \mathbb{Z}[x]$
- **Output:** $h = \gcd(f, g)$
- Algorithm:
  1. Compute $b = \gcd(LT(f), LT(g))$, and set $B \in \mathbb{N}$
  2. Pick random prime $p \in [2B, 4B]$
  3. Compute $p(x) = \gcd_{\mathbb{Z}_p[x]}(f, g)$
  4. Compute $q, f^*, g^* \in \mathbb{Z}[x]$ with height $< p/2$ satisfying:

$$q \equiv bp \mod p, \quad f^* \cdot q \equiv b \cdot f \mod p, \quad g^* \cdot q \equiv b \cdot q \mod p$$

Algorithm

- **Input**: primitive polynomials $f, g \in \mathbb{Z}[x]$
- **Output:** $h = \gcd(f, g)$
- Algorithm:
  1. Compute $b = \gcd(LT(f), LT(g))$, and set $B \in \mathbb{N}$
  2. Pick random prime $p \in [2B, 4B]$
  3. Compute $p(x) = \gcd_{\mathbb{Z}_p[x]}(f, g)$
  4. Compute $q, f^*, g^* \in \mathbb{Z}[x]$ with height $< p/2$ satisfying:

     $$q \equiv bp \mod p, \quad f^* \cdot q \equiv b \cdot f \mod p, \quad g^* \cdot q \equiv b \cdot q \mod p$$

  5. If

     $$\|f^*\|_1 \cdot \|q\|_1 \leq B \quad \text{and} \quad \|g^*\|_1 \cdot \|q\|_1 \leq B$$

     Return q.
     Otherwise go back to step 2.

Algorithm

- **Input**: primitive polynomials $f, g \in \mathbb{Z}[x]$
- **Output:** $h = \gcd(f, g)$
- Algorithm:
  1. Compute $b = \gcd(LT(f), LT(g))$, and set $B \in \mathbb{N}$
  2. Pick random prime $p \in [2B, 4B]$
  3. Compute $p(x) = \gcd_{\mathbb{Z}_p[x]}(f, g)$
  4. Compute $q, f^*, g^* \in \mathbb{Z}[x]$ with height $< p/2$ satisfying:
  $$q \equiv bp \mod p, \quad f^* \cdot q \equiv b \cdot f \mod p, \quad g^* \cdot q \equiv b \cdot q \mod p$$
  5. If
  $$\|f^*\|_1 \cdot \|q\|_1 \leq B \quad \text{and} \quad \|g^*\|_1 \cdot \|q\|_1 \leq B$$
  Return q.
  Otherwise go back to step 2.
- Correctness follows by previous slides, and probability the our random prime does not work is $\leq 1/2$.

Acknowledgement

Based entirely on

- Lecture 5 from Madhu's notes
  http://people.csail.mit.edu/madhu/FT98/

$p = 2 \quad q = 2^k$