CS 487 / · · ·
Introduction to Symbolic Computation

University of Waterloo
Éric Schost
eschost@uwaterloo.ca
Finite fields algorithms
Overview

Error correcting algorithms over finite fields need:

• **Root finding:** given $P$ in $\mathbb{F}[x]$ ($\mathbb{F}$ a finite field), find all the roots of $P$ in $\mathbb{F}$.

**Example:** With $\mathbb{F} = \mathbb{Z}/31\mathbb{Z} = \{0, 1, 2, \ldots, 30\}$,

$$P = x^5 + 22x^4 + 5x^3 + 4x^2 + 6x + 13,$$

find $P(2) = P(19) = 0$.

Here, $P = (x + 12)^2(x + 29)(x^2 + 2)$, but we don’t know it.

• **Discrete logarithm:** given $\alpha$ and $\beta$ in $\mathbb{F}$, find $k$ such that $\alpha^k = \beta$.

**Example:** With $\mathbb{F} = \mathbb{Z}/31\mathbb{Z}$, $\alpha = 3$, $\beta = 2$, find

$$3^{24} = 2.$$
Let \( q := |\mathbb{F}| \).

**Naive approaches:** try everything.
- to find the roots of \( P \), try all elements in \( \mathbb{F} \).
  
  **Runtime:** \( q \times O(d) = O(dq) \) operations +, \( \times \) in \( \mathbb{F} \), with \( d = \deg(P) \).

- to find the discrete logarithm, compute all powers of \( \alpha \)
  
  **Runtime:** \( O(q) \) multiplications in \( \mathbb{F} \).

**Want:** polynomial in \( \log(q) \).
- unknown in both cases
- random polynomial time for root-finding
- “heuristic” subexponential time for discrete logs.
Preamble to root-finding: squarefree part
Suppose first that $P$ is in $\mathbb{Q}[x]$, and that it factors as

$$P = P_1^{e_1} \cdots P_s^{e_s},$$

with $P_1, \ldots, P_s$ the **irreducible factors** of $P$ and $e_1, \ldots, e_s$ their **multiplicities**.

We want

$$\overline{P} = P_1 \cdots P_s.$$

(same roots, no multiplicities) Of course, we don’t know the $P_i$’s.

**Prop.**

$$\overline{P} = \frac{P}{\gcd(P', P)}.$$
Proof on an example

Take \( P = (x - r_1)^{31}(x - r_2)^{62} \); then its derivative is

\[
P' = 31(x - r_1)^{30}(x - r_2)^{62} + 62(x - r_1)^{31}(x - r_2)^{61}
\]

\[
= 31(x - r_1)^{30}(x - r_2)^{61}(x - r_2 + 2(x - r_1)).
\]

What is their GCD?

- it has to be \((x - r_1)^{s_1}(x - r_2)^{s_2}\)
- \(s_1\) must be equal to 30
- \(s_2\) must be equal to 61

\[
\gcd(P, P') = (x - r_1)^{30}(x - r_2)^{61}
\]

\[
P/ \gcd(P, P') = (x - r_1)^{1}(x - r_2)^{1}
\]
**Summary**

**Complexity:**
- the cost of computing GCD’s and divisions
- \( O(M(d) \log(d)) \) if \( \deg(P) = d \)

**Over finite fields:**
- the algorithm becomes a bit more complicated
- the previous example fails over \( \mathbb{Z}/31\mathbb{Z} \):
  \[
P' = 0 \text{ because } 31 \mod 31 = 62 \mod 31 = 0
  \]
- Our first example becomes \( x^4 + 10x^3 + 9x^2 + 20x + 14 \).
  This is \( (x + 12)(x + 29)(x^2 + 2) \), but we don’t know it.
Squares
Squares a finite field

Suppose that $\mathbb{F}$ is a finite field where $2 \neq 0$ (iff $1 \neq -1$).

• excludes $\mathbb{F}_2 = \{0, 1\}$
• and also binary fields $\mathbb{F}_2[x]/P(x)$
• the algorithm becomes more complicated.

Prop. If $\mathbb{F}$ has $q$ elements,

• there are $(q - 1)/2$ squares (excluding zero)
• $\alpha$ is a square iff $\alpha^{(q-1)/2} = 1$ (or $\alpha = 0$)
In $\mathbb{F} = \mathbb{Z}/31\mathbb{Z} = \{0, 1, 2, \ldots, 30\}$, we have $0^2 = 0$ and

\[
\begin{align*}
1^2 &= 30^2 = 1 & 6^2 &= 25^2 = 5 & 11^2 &= 20^2 = 28 \\
2^2 &= 29^2 = 4 & 7^2 &= 24^2 = 18 & 12^2 &= 19^2 = 20 \\
3^2 &= 28^2 = 9 & 8^2 &= 23^2 = 2 & 13^2 &= 18^2 = 14 \\
4^2 &= 27^2 = 16 & 9^2 &= 22^2 = 19 & 14^2 &= 17^2 = 10 \\
5^2 &= 26^2 = 25 & 10^2 &= 21^2 = 7 & 15^2 &= 16^2 = 8
\end{align*}
\]

So the (nonzero) squares are

\[
\{1, 2, 4, 5, 7, 8, 9, 10, 14, 16, 18, 19, 20, 25, 28\}.
\]
0. If $b$ is nonzero, $b^2$ is nonzero.

1. Every square has at least two square roots:
   - if $a = b^2$, then $a = (-b)^2$, and $b \neq -b$

2. It cannot have more than two:
   - $b^2 = c^2 \implies (b - c)(b + c) = 0$
   - so if $b - c \neq 0$, $b + c = 0$ \hspace{1cm} (\mathbb{F} is a field)

3. So the $(q - 1)$ nonzero elements in $\mathbb{F}$ map to $(q - 1)/2$ squares.
1. For any $a$ nonzero in $\mathbb{F}$, $a^{q-1} = 1$.

   I don’t have a one-line proof, but it’s true.
1. For any \( a \) nonzero in \( \mathbb{F} \), \( a^{q-1} = 1 \).
   
   I don’t have a one-line proof, but it’s true.

2. So \( (a^{(q-1)/2})^2 = 1 \).
Proof, 2.

1. For any $a$ nonzero in $\mathbb{F}$, $a^{q-1} = 1$.
   
   I don’t have a one-line proof, but it’s true.

2. So $(a^{(q-1)/2})^2 = 1$.

3. So $a^{(q-1)/2} = \pm 1$. 
Proof, 2.

1. For any $a$ nonzero in $\mathbb{F}$, $a^{q-1} = 1$.
   I don’t have a one-line proof, but it’s true.

2. So $(a^{(q-1)/2})^2 = 1$.

3. So $a^{(q-1)/2} = \pm 1$.

4. If $a = b^2$, then

   $a^{(q-1)/2} = b^{q-1} = 1$. 

5. There are $(q-1)/2$ squares, so that gives us $(q-1)/2$ solutions to $a^{(q-1)/2} = 1$. There are no other solutions.

6. So for the non-squares, $a^{(q-1)/2} = -1$. 

13 / 31
1. For any \(a\) nonzero in \(\mathbb{F}\), \(a^{q-1} = 1\).
   
   I don’t have a one-line proof, but it’s true.

2. So \((a^{(q-1)/2})^2 = 1\).

3. So \(a^{(q-1)/2} = \pm 1\).

4. If \(a = b^2\), then
   
   \[a^{(q-1)/2} = b^{q-1} = 1.\]

5. There are \((q-1)/2\) squares, so that gives us \((q-1)/2\) solutions to \(a^{(q-1)/2} = 1\). There are no other solutions.
Proof, 2.

1. For any $a$ nonzero in $\mathbb{F}$, $a^{q-1} = 1$.
   
   I don’t have a one-line proof, but it’s true.

2. So $(a^{(q-1)/2})^2 = 1$.

3. So $a^{(q-1)/2} = \pm 1$.

4. If $a = b^2$, then
   
   \[ a^{(q-1)/2} = b^{q-1} = 1. \]

5. There are $(q - 1)/2$ squares, so that gives us $(q - 1)/2$ solutions to $a^{(q-1)/2} = 1$. There are no other solutions.

6. So for the non-squares, $a^{(q-1)/2} = -1$. 
Part 1: keep only linear factors
Something we saw in the proof.

\[ x^q - x = \prod_{a \in F} (x - a). \]

Example: over \( \mathbb{Z}/31\mathbb{Z} \),

\[ x(x - 1)(x - 2) \cdots (x - 30) = x^{31} - x. \]
1. (Optional) Make $P$ squarefree.

We saw that before.

2. Replace $P$ by $Q = \gcd(P, x^q - x)$

So $Q = \prod_{a \in \mathbb{F}, P(a) = 0} (x - a)$

Our first example: $P = x^4 + 10x^3 + 9x^2 + 20x + 14$ gives

$$Q = \gcd(P, x^{31} - x) = x^2 + 10x + 7.$$ 

Then, $Q = (x + 12)(x + 29)$ (but we don’t know it!)
We have to be careful: \( \deg(x^q - x) = q \).

- Lecture 3: we saw that
  \[
gcd(P, x^q - x) = \deg(P, (x^q - x) \mod P)
  \]
- To compute \((x^q - x) \mod P\), we compute \(x^q \mod P\)
- We saw this question a few times:
  - lecture 2, Mersenne numbers, compute \(2^n \mod 19\)
  - midterm, compute \(F^m \mod x^n\)
  - asn 3, compute \((a + b\psi)^n\)
  - final?
- \(O(\log(q))\) multiplications modulo \(P\), so \(O(M(d) \log(q))\)
- After that, GCD is \(O(M(d) \log(d))\)
Part 2: using squares
At this stage, suppose \( Q = (x - r_1) \cdots (x - r_s) \), for some unknown \( r_i \)'s, all distinct.

Take \( A \) in \( \mathbb{F}[x] \), of degree less than \( s \), and compute

\[
B = \gcd(A^{(q-1)/2} - 1, Q).
\]

What are the roots of \( B \)?

- It must be some of the \( r_i \)'s (because they must be roots of \( Q \))
- It is those for which \( A(r_i)^{(q-1)/2} - 1 = 0 \), that is, \( A(r_i) \) is a nonzero square.

So \( B \) is a factor of \( Q \), and hopefully not a trivial one.
Example

Take $Q = x^2 + 10x + 7 = (x - 19)(x - 2)$, and let’s try some $A$’s.

Remember that $q = 31$, so $(q - 1)/2 = 15$.

- with $A = x + 2$, we find $B = x + 29$
  $A(19) = 21$, $A(2) = 4$

- with $A = x$, we find $B = x^2 + 10x + 7$
  $A(19) = 19$, $A(2) = 2$

- with $A = 3x$, we find $B = 1$.
  $A(19) = 26$, $A(2) = 6$

If $B \neq 1$ and $B \neq Q$, we are done, because here $\deg(Q) = 2$.

In general: recursive call.
$21 / 31 \mathbb{Z} = \frac{21}{31}$

- red: $A = 2x + 2$ ✓
- green: $A = x$ ×
- blue: $A = 3x$ ×

Squares, satisfy:
- $a^{15} = 1$
- not squares (except 0):
  - $a^{15} \neq 1$
Sample space: choosing $s$ coeffs of $A \equiv$ choosing $A(r_1), \ldots, A(r_s)$. $q^s$ choices.

We lose if

- either all $A(r_i)$’s are nonzero squares
  \[
  \left(\frac{q-1}{2}\right)^s \simeq \frac{q^s}{2^s} \text{ choices.}
  \]
  \[\text{\hspace{1in} } B = Q\]

- or all are non-squares
  \[
  \left(\frac{q+1}{2}\right)^s \simeq \frac{q^s}{2^s} \text{ choices.}
  \]
  \[\text{\hspace{1in} } B = 1\]
Probability of losing is

\[
\frac{q^s}{2^s} + \frac{q^s}{2^s} = \frac{1}{2^{s-1}} \leq \frac{1}{2}
\]

Probability \( p \) of winning is \textbf{at least} 1/2, so we expect to win in \( O(1) \) trials.

\[
E(\text{trials}) = 1 \cdot p + 2 \cdot p(1 - p) + 3 \cdot p(1 - p)^2 + \cdots \]
\[
= \frac{1}{p} \leq 2.
\]
Complexity of splitting

Let \( s := \deg(Q) \leq d \).

**We have to be careful:** \( A^{(q-1)/2} \) has degree \( \theta(sq) \).

- again,

  \[
  \gcd(A^{(q-1)/2} - 1, Q) = \gcd(A^{(q-1)/2} - 1 \mod Q, Q)
  \]

- takes \( O(\log(q)) \) multiplications mod \( Q \), so \( O(M(s) \log(q)) \)

- After that, GCD is \( O(M(s) \log(s)) \)

**Summary:** the cost to split \( Q \) into 2 factors is

\[
O_E(M(s) \log(qs))
\]

\((O_E = \text{expected runtime})\)
The overall recursion

The recursive calls are organized in a binary tree, with factors at their nodes

- the sum of the degrees $s_i$ at a given level is $s$
- the cost at each node is

$$O_E(M(s_i) \log (qs_i))$$

- so the overall cost at a given level is

$$O_E(M(s) \log (qs))$$

**Question:** expected depth of the tree?
Recall $Q = (x - r_1) \cdots (x - r_s)$.

Fix $i$ and $j$. The probability that $r_i, r_j$ are in the same node after $k$ steps is at most $\frac{1}{2^k}$. 
Recall $Q = (x - r_1) \cdots (x - r_s)$.

Fix $i$ and $j$. The probability that $r_i, r_j$ are in the same node after $k$ steps is at most $\frac{1}{2^k}$.

The probability $p_k$ that some $r_i, r_j$ are in the same node after $k$ steps is at most $\frac{s^2}{2^k}$.

This is the probability that the depth is greater than $k$. 
The overall recursion

The probability that the depth is exactly $k$ is $p_{k-1} - p_k$, so

$$E(\text{depth}) = \sum_{k \geq 1} k(p_{k-1} - p_k) = \sum_{k \geq 0} p_k.$$ 

- if $k \leq \log(s^2)$, we use the bound $p_k \leq 1$
- if $k > \log(s^2)$, we use the bound $p_k \leq \frac{s^2}{2^k}$

(this is the same trick as for the expected height of a skiplist)

First sum is at most $\log(s^2) = 2 \log(s)$, second sum is at most 2. So

$$E(\text{depth}) = O(\log(s))$$

and the total cost is $O_E(M(s) \log(qs) \log(s))$. 
Bonus: discrete logarithm
After all this work, we found a root $r$ of $P$, and we would like to find $e$ such that

$$\alpha^e = r.$$ 

**Remark:** $e$ is less than $q = |\mathbb{F}|$, because

$$\alpha^{q-1} = 1.$$ 

**Naive algorithm:** compute

$$1, \alpha, \alpha^2, \ldots, \alpha^{q-1}$$

(stop as soon as you find $r$.) At most $q$ multiplications.
Let $t = \sqrt{q}$, and write

$$e = e_0 + e_1 t$$

with

$$e_0, e_1 \in \{0, \ldots, t - 1\}$$
Let \( t = \sqrt{q} \), and write
\[
e = e_0 + e_1 t
\]
with
\[
e_0, e_1 \in \{0, \ldots, t - 1\}
\]
Then
\[
q^e = q^{e_0 + e_1 t} = q^{e_0} q^{e_1} \quad \text{with} \quad q' = q^t.
\]
So \( q^e = r \) means
\[
q^{e_0} q^{e_1} = r \iff q^{e_0} = \frac{r}{q^{e_1}}.
\]
Let $t = \sqrt{q}$, and write

$$e = e_0 + e_1 t$$

with

$$e_0, e_1 \in \{0, \ldots, t - 1\}$$

Then

$$q^e = q^{e_0 + e_1 t} = q^{e_0} q^{e_1} \quad \text{with} \quad q' = q^t.$$ 

So $q^e = r$ means

$$q^{e_0} q^{e_1} = r \iff q^{e_0} = \frac{r}{q^{e_1}}.$$ 

**Runtime:**

- $O(\sqrt{q})$ multiplications / inverses.
- needs hashing / balanced BST for lookup.
Suppose we don’t know about fast dictionaries. We have elements

- $v_1, \ldots, v_t$
- $w_1, \ldots, w_t$

and we wonder if $v_i = w_j$ for some $i, j$. 
Suppose we don’t know about fast dictionaries. We have elements

- $v_1, \ldots, v_t$
- $w_1, \ldots, w_t$

and we wonder if $v_i = w_j$ for some $i, j$.

**Naive:** compute all differences; this is $O(t^2)$. 

Bonus of the bonus
Suppose we don’t know about fast dictionaries. We have elements

- $v_1, \ldots, v_t$
- $w_1, \ldots, w_t$

and we wonder if $v_i = w_j$ for some $i, j$.

**Naive:** compute all differences; this is $O(t^2)$.

**Not so naive:**

- compute $(x - v_1) \cdots (x - v_t)$
- evaluate it at $w_1, \ldots, w_t$
  - quasi-linear
  - if we find a zero, this gives $w_j$
- decide whether $w_j$ cancels $(x - v_1) \cdots (x - v_{t/2})$ or $(x - v_{t/2+1}) \cdots (x - v_t)$
  - quasi-linear
- recurse to get $v_i$