A Faster Randomized Algorithm for Counting Roots in $\mathbb{Z}/(p^k)$

Leann Kopp

Department of Mathematics,
Texas A&M University
NSF DMS – 1757872

July 16, 2018

Outline

Today we will...

- Review the randomized algorithm for counting mod p^k
- See some specific examples regarding computational time
- Go over the complexity bound of the algorithm
- Discuss a bound on the number of roots mod p^k

Factorization

- Counting roots in $\mathbb{Z}/(p)[x]$ is easy if you allow randomization, thanks to the work of Zieler, Berlekamp, and many others, dating back to the 1960's.
- These methods take advantage of $\mathbb{Z}/(p)[x]$ being a unique factorization domain.

Factorization

- Counting roots in $\mathbb{Z}/(p)[x]$ is easy if you allow randomization, thanks to the work of Zieler, Berlekamp, and many others, dating back to the 1960's.
- These methods take advantage of $\mathbb{Z}/(p)[x]$ being a unique factorization domain.
  - One simple method: compute the $\gcd(x^p - x, f)$ in $\mathbb{Z}/(p)$
  - By Fermat's Little Theorem, we know that for a prime p, $x^p - x = x(x-1)\cdots(x-(p-1)) \bmod p$

Factorization

- Counting roots in $\mathbb{Z}/(p)[x]$ is easy if you allow randomization, thanks to the work of Zieler, Berlekamp, and many others, dating back to the 1960's.
- These methods take advantage of $\mathbb{Z}/(p)[x]$ being a unique factorization domain.
  - One simple method: compute the $\gcd(x^p - x, f)$ in $\mathbb{Z}/(p)$
  - By Fermat's Little Theorem, we know that for a prime p, $x^p - x = x(x-1)\cdots(x-(p-1)) \bmod p$
- $\mathbb{Z}/(p^k)[x]$ is not a unique factorization domain, so we have to be more careful when counting in $\mathbb{Z}/(p^k)[x]$.

Factorization

- Counting roots in $\mathbb{Z}/(p)[x]$ is easy if you allow randomization, thanks to the work of Zieler, Berlekamp, and many others, dating back to the 1960's.
- These methods take advantage of $\mathbb{Z}/(p)[x]$ being a unique factorization domain.
  - One simple method: compute the $\gcd(x^p - x, f)$ in $\mathbb{Z}/(p)$
  - By Fermat's Little Theorem, we know that for a prime p, $x^p - x = x(x-1)\cdots(x-(p-1)) \bmod p$
- $\mathbb{Z}/(p^k)[x]$ is not a unique factorization domain, so we have to be more careful when counting in $\mathbb{Z}/(p^k)[x]$.
  - Example: $(x+3)^2 = x(x+6) \mod 9$.

Background to Algorithm

Consider the Taylor expansion of a polynomial $f \in \mathbb{Z}[x]$ of degree d, where $\zeta \in \mathbb{Z}$ is a root of the mod p reduction of f and $\varepsilon \in \{0, \ldots, p^k - 1\}$:

$$f(\zeta + p\varepsilon) = f(\zeta) + f'(\zeta)p\varepsilon + \cdots + \frac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1} \bmod p^k.$$

Background to Algorithm

Consider the Taylor expansion of a polynomial $f \in \mathbb{Z}[x]$ of degree d, where $\zeta \in \mathbb{Z}$ is a root of the mod p reduction of f and $\varepsilon \in \{0, \ldots, p^k - 1\}$:

$$f(\zeta + p\varepsilon) = f(\zeta) + f'(\zeta)p\varepsilon + \cdots + \tfrac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1} \bmod p^k.$$

Definition: Let $s \in \{1, \ldots, k\}$ be the maximal integer such that p^s divides each of $f(\zeta), \ldots, \frac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1}$.

Background to Algorithm

Consider the Taylor expansion of a polynomial $f \in \mathbb{Z}[x]$ of degree d, where $\zeta \in \mathbb{Z}$ is a root of the mod p reduction of f and $\varepsilon \in \{0, \ldots, p^k - 1\}$:

$$f(\zeta + p\varepsilon) = f(\zeta) + f'(\zeta)p\varepsilon + \cdots + \tfrac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1} \bmod p^k.$$

Definition: Let $s \in \{1, \ldots, k\}$ be the maximal integer such that p^s divides each of $f(\zeta), \ldots, \frac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1}$.

Lemma (Hensel's Lemma)

If $f \in \mathbb{Z}[x]$ is a polynomial with integer coefficients, p is prime, and $\zeta_J \in \{0, \ldots, p^{J-1} - 1\}$ is a root of $f \pmod{p^J}$ and $f'(\zeta_J) \neq 0 \pmod{p}$, then there is a unique $\zeta \in \{0, \ldots, p^{J+1} - 1\}$ with $f(\zeta) = 0 \pmod{p^{J+1}}$ and $\zeta = \zeta_J \pmod{p^J}$.

Overview of Algorithm

For each root ζ of the mod p reduction of f, we have the following:

- By Hensel's Lemma, we have a single lift (a unique ε) when $s = 1$.

Overview of Algorithm

For each root ζ of the mod p reduction of f, we have the following:

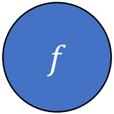
- By Hensel's Lemma, we have a single lift (a unique ε) when $s = 1$.
- When $s = k$, the expression $f(\zeta) + \cdots + \frac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1}$ vanishes identically mod p^k, giving p^{k-1} lifts.

Overview of Algorithm

For each root ζ of the mod p reduction of f, we have the following:

- By Hensel's Lemma, we have a single lift (a unique ε) when $s = 1$.
- When $s = k$, the expression $f(\zeta) + \cdots + \frac{1}{(k-1)!} f^{(k-1)}(\zeta)(p\varepsilon)^{k-1}$ vanishes identically mod p^k, giving p^{k-1} lifts.
- When $s \in \{2, \ldots, k-1\}$, we can reapply the algorithm to an instance of counting roots for the polynomial $f_\zeta(\varepsilon) = \frac{f(\zeta)}{p^s} + \frac{f'(\zeta)}{p^{s-1}}\varepsilon + \cdots + \frac{f^(k-1)(\zeta)}{(k-1)!p^{s-(k-1)}}\varepsilon^{k-1}$ in $\mathbb{Z}/(p^{k-s})$.

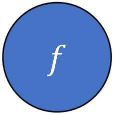
Illustration of Complexity Bound



With $p = 3, k = 7$:

$$f(x) = x^{10} - 10x + 738$$
$$f(x) = x(x+2)^9 \mod 3$$

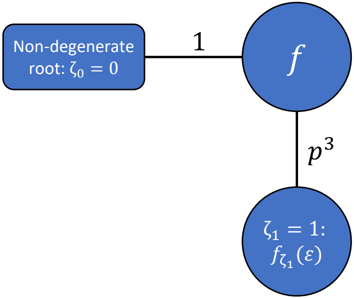
Illustration of Complexity Bound



With $p = 3, k = 7$:

$f(x) = x^{10} - 10x + 738$
$f(x) = x(x+2)^9 \mod 3$
$\zeta_0 = 0$ with $s(0,0) = 1$
$\zeta_1 = 1$ with $s(1,1) = 4$

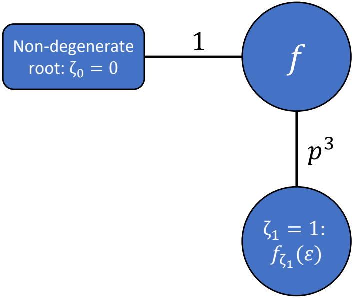
Illustration of Complexity Bound



With $p = 3, k = 7$:

$f(x) = x^{10} - 10x + 738$
$f(x) = x(x+2)^9 \mod 3$
$\zeta_0 = 0$ with $s(0,0) = 1$
$\zeta_1 = 1$ with $s(1,1) = 4$

Illustration of Complexity Bound

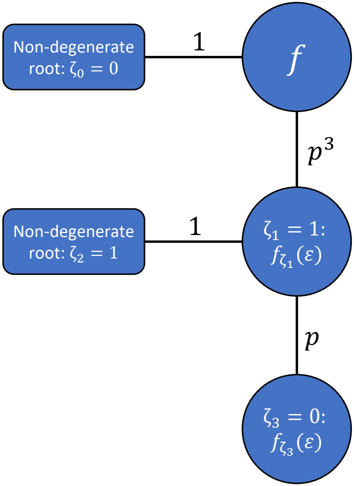


With $p = 3, k = 7$:

$f(x) = x^{10} - 10x + 738$
$f(x) = x(x+2)^9 \mod 3$
$\zeta_0 = 0$ with $s(0,0) = 1$
$\zeta_1 = 1$ with $s(1,1) = 4$

$f_{\zeta_1}(\varepsilon) = x^3 + 2x^2 \mod 3, k = 3$
$\zeta_2 = 1$ with $s(2,1) = 1$
$\zeta_3 = 0$ with $s(3,1) = 2$

Illustration of Complexity Bound

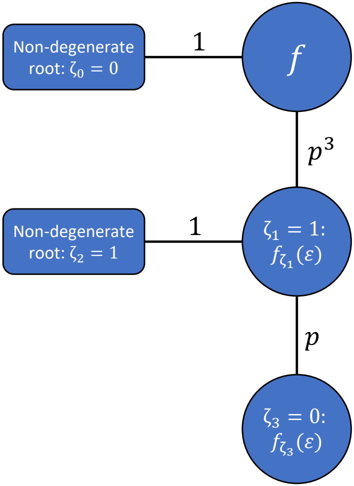


With $p = 3, k = 7$:

$f(x) = x^{10} - 10x + 738$
$f(x) = x(x+2)^9 \mod 3$
$\zeta_0 = 0$ with $s(0,0) = 1$
$\zeta_1 = 1$ with $s(1,1) = 4$

$f_{\zeta_1}(\varepsilon) = x^3 + 2x^2 \mod 3, k = 3$
$\zeta_2 = 1$ with $s(2,1) = 1$
$\zeta_3 = 0$ with $s(3,0) = 2$

Illustration of Complexity Bound



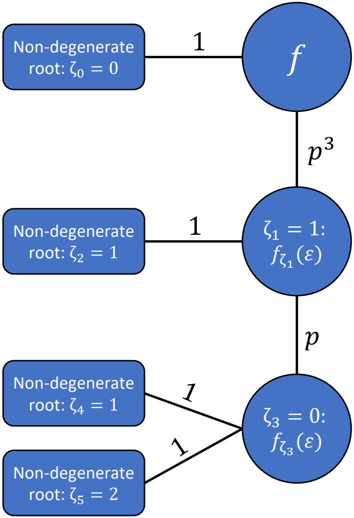
With $p = 3, k = 7$:

$f(x) = x^{10} - 10x + 738$
$f(x) = x(x+2)^9 \mod 3$
$\zeta_0 = 0$ with $s(0,0) = 1$
$\zeta_1 = 1$ with $s(1,1) = 4$

$f_{\zeta_1}(\varepsilon) = x^3 + 2x^2 \mod 3, k = 3$
$\zeta_2 = 1$ with $s(2,1) = 1$
$\zeta_3 = 0$ with $s(3,0) = 2$

$f_{\zeta_3}(\varepsilon) = 2x^2 + 1 \mod 3, k = 1$
$\zeta_4 = 1$ with $s(4,1) = 1$
$\zeta_5 = 2$ with $s(5,2) = 1$.

Illustration of Complexity Bound



With $p = 3, k = 7$:

$f(x) = x^{10} - 10x + 738$
$f(x) = x(x+2)^9 \mod 3$
$\zeta_0 = 0$ with $s(0,0) = 1$
$\zeta_1 = 1$ with $s(1,1) = 4$

$f_{\zeta_1}(\varepsilon) = x^3 + 2x^2 \mod 3, k = 3$
$\zeta_2 = 1$ with $s(2,1) = 1$
$\zeta_3 = 0$ with $s(3,0) = 2$

$f_{\zeta_3}(\varepsilon) = 2x^2 + 1 \mod 3, k = 1$
$\zeta_4 = 1$ with $s(4,1) = 1$
$\zeta_5 = 2$ with $s(5,2) = 1.$

Ideas Behind Proof of Complexity Bound

- We use Kedlaya-Umans fast $\mathbb{Z}/(p)[x]$ factoring algorithm, which takes time $d^{1.5+o(1)}(\log p)^{1+o(1)} + d^{1+o(1)}(\log p)^{2+o(1)}$ for a degree d polynomial.

Ideas Behind Proof of Complexity Bound

- We use Kedlaya-Umans fast $\mathbb{Z}/(p)[x]$ factoring algorithm, which takes time $d^{1.5+o(1)}(\log p)^{1+o(1)} + d^{1+o(1)}(\log p)^{2+o(1)}$ for a degree d polynomial.
- To simplify, the complexity is less than or equal to (the number of nodes in the recursion tree)(the complexity of factoring over $(\mathbb{Z}/(p))[x]$).

Ideas Behind Proof of Complexity Bound

- We use Kedlaya-Umans fast $\mathbb{Z}/(p)[x]$ factoring algorithm, which takes time $d^{1.5+o(1)}(\log p)^{1+o(1)} + d^{1+o(1)}(\log p)^{2+o(1)}$ for a degree d polynomial.
- To simplify, the complexity is less than or equal to (the number of nodes in the recursion tree)(the complexity of factoring over $(\mathbb{Z}/(p))[x]$).
- The depth and branching of our recurrence tree is strongly limited by the value of k.
- Optimizing parameters, the worst case is when $d \approx e \approx 2.71828$ and the depth is $\frac{k}{e}$.

Complexity Bound

- The randomized algorithm counts the number of roots in time $d^{1.5+o(1)}(\log p)^{2+o(1)}(1.12)^k$.

Complexity Bound

- The randomized algorithm counts the number of roots in time $d^{1.5+o(1)}(\log p)^{2+o(1)}(1.12)^k$.
- This complexity bound refers to counting roots, not to finding/storing them; we can have over $\lfloor \frac{d}{k} \rfloor p^{k-1}$ roots, so we cannot achieve this same time bound if we store every single root.

Complexity Bound

- The randomized algorithm counts the number of roots in time $d^{1.5+o(1)}(\log p)^{2+o(1)}(1.12)^k$.
- This complexity bound refers to counting roots, not to finding/storing them; we can have over $\lfloor \frac{d}{k} \rfloor p^{k-1}$ roots, so we cannot achieve this same time bound if we store every single root.

Example: The polynomial $f(x) = (x-2)^7(x-1)^3$ with $p = 17, k = 7$ has $24,221,090$ roots (this is greater than $\lfloor \frac{d}{k} \rfloor p^{k-1} = 24,137,569$). Computing this number of roots took .004 seconds, but storing and listing them all would take much longer.

Complexity Bound

- The randomized algorithm counts the number of roots in time $d^{1.5+o(1)}(\log p)^{2+o(1)}(1.12)^k$.
- This complexity bound refers to counting roots, not to finding/storing them; we can have over $\lfloor \frac{d}{k} \rfloor p^{k-1}$ roots, so we cannot achieve this same time bound if we store every single root.

Example: The polynomial $f(x) = (x-2)^7(x-1)^3$ with $p = 17, k = 7$ has $24,221,090$ roots (this is greater than $\lfloor \frac{d}{k} \rfloor p^{k-1} = 24,137,569$). Computing this number of roots took .004 seconds, but storing and listing them all would take much longer.
(Counting the number of roots using brute force took 39 minutes).

Comparing Methods

- The randomized algorithm counts the number of roots in time $d^{1.5+o(1)}(\log p)^{2+o(1)}(1.12)^k$.
- In comparison, brute force counting takes time $\approx p^k$.
- We expect the randomized algorithm to be faster even for p as small as 2 (1.12^k vs. 2^k).

Data for $p = 2$

The table below shows the average difference in computation time for the number of roots of 100 random polynomials of degree less than or equal to 100 in $\mathbb{Z}/(2^k)$ for the given k, between brute force and the randomized algorithm (negative implies brute force was faster):

k	8	9	10	11	15
Avg Diff (in seconds)	-0.0011	-0.00029	0.0028	0.01701	0.32499

Timing Data

- We expect the randomized algorithm to take the longest when a polynomial has many degenerate roots because a polynomial of this type will require many recursive calls.
- Polynomials with many degenerate roots do take longer than a random polynomial, but overall the randomized algorithm still outperforms other methods.

Timing Data

- We expect the randomized algorithm to take the longest when a polynomial has many degenerate roots because a polynomial of this type will require many recursive calls.
- Polynomials with many degenerate roots do take longer than a random polynomial, but overall the randomized algorithm still outperforms other methods.
- Example: Counting roots of $f(x) = (x-1)(x-2)^2 \cdots (x-10)^{10}$ in $\mathbb{Z}/(31^{10})$ took 6.4 seconds using the randomized algorithm.
- For comparison, a random polynomial of the same degree (55) took 1 millisecond with the randomized algorithm, and counting roots using brute force for just 31^6 took 2.7 hours.

Example with Large Number of Roots

The number of roots of a polynomial in $\mathbb{Z}/(p^k)$ can be very large, especially for polynomials with many degenerate roots mod p.

Example: $(x-2)^{50}$ has 5132842958629010337866366828195 (31 digit number) roots in $\mathbb{Z}/(25^{29})$.

Example with Large Number of Roots

The number of roots of a polynomial in $\mathbb{Z}/(p^k)$ can be very large, especially for polynomials with many degenerate roots mod p.

Example: $(x-2)^{50}$ has 5132842958629010337866366828195 (31 digit number) roots in $\mathbb{Z}/(25^{29})$.

But we do have an upper bound on the number of roots based on the sizes of k, p and d.

Steps Toward a Bound on the Number of Roots

Lemma

If a root ζ of the mod p reduction of f has multiplicity j, then $s_\zeta \leq j$, where s_ζ is the greatest integer such that p^{s_ζ} divides each of $f(\zeta), \ldots, \frac{f^{(k-1)}(\zeta)}{(k-1)!}p^{k-1}\varepsilon^{k-1}$.

Proof: If ζ has multiplicity j, then $f(\zeta) = \cdots = f^{j-1}(\zeta) = 0 \pmod{p}$, but $f^{(j)}(\zeta) \neq 0 \pmod{p}$. So $\frac{f^j(\zeta)}{j!}p^j$ is divisible by p^j but not p^{j+1} and therefore $s_\zeta \leq j$.

Steps Toward a Bound on the Number of Roots

Lemma

If a root ζ of the mod p reduction of f has multiplicity j, then $s_\zeta \leq j$, where s_ζ is the greatest integer such that p^{s_ζ} divides each of $f(\zeta), \ldots, \frac{f^{(k-1)}(\zeta)}{(k-1)!}p^{k-1}\varepsilon^{k-1}$.

Proof: If ζ has multiplicity j, then $f(\zeta) = \cdots = f^{j-1}(\zeta) = 0 \pmod{p}$, but $f^{(j)}(\zeta) \neq 0 \pmod{p}$. So $\frac{f^j(\zeta)}{j!}p^j$ is divisible by p^j but not p^{j+1} and therefore $s_\zeta \leq j$.

This is important because it tells us that we can have at most $\lfloor \frac{d}{s} \rfloor$ roots of f mod p for each s.

Bound on the Number of Roots

Theorem

Let p be a prime, $f \in \mathbb{Z}[x]$ a polynomial of degree d, and $k \in \mathbb{N}$ such that $d \geq k \geq 2$. Then the number of roots of f in $\mathbb{Z}/(p^k)$ is less than or equal to $\min\{d, p\}p^{k-1}$.

Bound on the Number of Roots

Theorem

Let p be a prime, $f \in \mathbb{Z}[x]$ a polynomial of degree d, and $k \in \mathbb{N}$ such that $d \geq k \geq 2$. Then the number of roots of f in $\mathbb{Z}/(p^k)$ is less than or equal to $\min\{d, p\}p^{k-1}$.

We know there are polynomials with more than $\min\{\lfloor \frac{d}{k} \rfloor, p\}p^{k-1}$ roots in $\mathbb{Z}/(p^k)$, so our bound is within a factor of k of optimality.

Root Bound Examples

Polynomials with $d \geq p$ having p^k roots in $\mathbb{Z}/(p^k)$:

1. $f(x) = (x^p - x)^k$ is a polynomial of degree pk with p^k roots in $\mathbb{Z}/(p^k)$.

Root Bound Examples

Polynomials with $d \geq p$ having p^k roots in $\mathbb{Z}/(p^k)$:

1. $f(x) = (x^p - x)^k$ is a polynomial of degree pk with p^k roots in $\mathbb{Z}/(p^k)$.
2. $g(x) = (x^{p^k - p^{k-1}} - 1)x^k$ has degree $p^k - p^{k-1} + k$ and also vanishes on all of $\mathbb{Z}/(p^k)$.

Sharp Bound for $k = 2$

When $k = 2$, there are a maximum of $\min\{\lfloor \frac{d}{2} \rfloor, p\}p + (d \bmod k)$ roots in $\mathbb{Z}/(p^2)$.

This upper bound is sharp. For example, the with $p = 5$ the degree 3 polynomial $(x-1)^2x$ has $\lfloor \frac{3}{2} \rfloor \cdot 5 + (3 \bmod 2) = 6$ roots in $\mathbb{Z}/(p^2)$.

Conclusions

- We have a Las Vegas randomized algorithm for counting the number of roots of a degree d polynomial $f \in \mathbb{Z}[x]$ in $\mathbb{Z}/(p^k)$.
- The complexity of the randomized algorithm is given by $\mathcal{O}(1.12^k)$.
- We see time improvements for computations using the randomized algorithm over brute-force counting even for $p = 2$.
- An upper bound on the number of roots is $\min\{d, p\}p^{k-1}$, and a sharp upper bound for $k = 2$ is given by $\min\{\lfloor \frac{d}{2} \rfloor, p\}p + (d \bmod k)$.