RANDOMIZED POLYNOMIAL-TIME ROOT COUNTING IN PRIME POWER RINGS

LEANN KOPP, NATALIE RANDALL, J. MAURICE ROJAS, AND YUYU ZHU

ABSTRACT. Suppose $k, p \in \mathbb{N}$ with p prime and $f \in \mathbb{Z}[x]$ is a univariate polynomial with degree d and all coefficients having absolute value less than p^k. We give a Las Vegas randomized algorithm that computes the number of roots of f in $\mathbb{Z}/\left(p^k\right)$ within time $d^3(k \log p)^{2+o(1)}$. (We in fact prove a more intricate complexity bound that is slightly better.) The best previous general algorithm had (deterministic) complexity exponential in k. We also present some experimental data evincing the potential practicality of our algorithm.

1. INTRODUCTION

Suppose $k, p \in \mathbb{N}$ with p prime and $f \in \mathbb{Z}[x]$ is a univariate polynomial with degree $d \geq 1$ and all coefficients having absolute value less than p^k. Let $N_{p,k}(f)$ denote the number of roots of f in $\mathbb{Z}/\left(p^k\right)$ (see, e.g., [25, 23, 2, 19, 15, 29] for further background on prime power rings). Computing $N_{p,k}(f)$ is a fundamental problem occuring in polynomial factoring [22, 10, 5, 26, 16], coding theory [3], and cryptography [20]. The function $N_{p,k}(f)$ is also a basic ingredient in the study of Igusa zeta functions and algorithms over $\mathbb{Q}_p$ [17, 12, 11, 5, 31, 6, 21, 30, 7, 1].

In spite of the fundamental nature of computing $N_{p,k}(f)$, the fastest earlier general algorithms had complexity exponential in k: [9] gave a deterministic algorithm taking time $\left(d \log(p) + 2^k\right)^{O(1)}$. While the O-constant was not stated in [9], the proof of the main theorem there indicates that the dependence on k in their algorithm is linear in e^k. Note that counting the roots via brute-force takes time $dp^k(k \log p)^{1+o(1)}$, so the algorithm from [9] is preferable, at least theoretically, for $p \geq 3$. Here, we present a simpler, dramatically faster randomized algorithm (Algorithm 2.3 of the next section) that appears practical for all p.

Theorem 1.1. *Following the notation above, there is a Las Vegas randomized algorithm that computes $N_{p,k}(f)$ in time $kd^3(k \log p)^{1+o(1)} + \left(dk \log^2 p\right)^{1+o(1)}$. In particular, the number of random bits needed is $O(dk \log(dk) \log p)$, and the space needed is $O(dk^2 \log p)$ bits.*

We prove Theorem 1.1 in Section 3 below. In our context, *Las Vegas randomized* means that, with a fixed error probability (which we can take to be, say, $\frac{1}{3}$), our algorithm under-estimates the number of roots. Our algorithm otherwise gives a correct root count, *and* always correctly announces whether the output count is correct or not. This type of randomization is standard in many number-theoretic algorithms, such as the fastest current algorithms for factoring polynomials over finite fields or primality checking (see, e.g., [2, 18, 8]).

At a high level, our algorithm here and the algorithm from [9] are similar in that they reduce the main problem to a collection of computations, mostly in the finite field $\mathbb{Z}/(p)$, indexed by the nodes of a tree of size depending on f and k. Also, both algorithms count by partitioning the roots in $\mathbb{Z}/\left(p^k\right)$ into clusters having the same mod p reduction. One subtlety to be aware of is that we compute the *number* of roots in $\mathbb{Z}/\left(p^k\right)$, *without* listing all of them. Indeed, the number of roots in $\mathbb{Z}/\left(p^k\right)$ can be as high as, say, $p^{k-\lceil k/2 \rceil}$ (when $k \geq d = 2$) or $p^{d/p}$ (when $d = kp$): simply consider the polynomials x^2 and $(x^p - x)^{d/p}$. So we can't attain a time or space bound sub-exponential in k unless we do something more clever than naively store every root (see Remark 1.2 below).

Partially supported by NSF grant CCF-1409020, and NSF REU grants DMS-1757872 and DMS-1460766.

In finer detail, the algorithm from [9] solves a "small" polynomial system at each node of a recursion tree (using a specially tailored Gröbner basis computation [13]), while our algorithm performs a univariate factorization in $(\mathbb{Z}/(p))[x]$ at each node of a smaller recursion tree. Our use of fast factorization (as in [18]) is why we avail to randomness, but this pays off: Gaining access to individual roots in $\mathbb{Z}/(p)$ (as suggested in [9]) enables us to give a more streamlined algorithm.

Remark 1.2. *von zur Gathen and Hartlieb presented in* [15] *a randomized polynomial-time algorithm to compute all factorizations of certain $f \in (\mathbb{Z}/(p^k))[x]$. (Examples like $x^2 = (x-p)(x+p) \in (\mathbb{Z}/(p^2))[x]$ show that unique factorization fails badly for $k \geq 2$, and the number of possible factorizations can be exponential in k.) Their algorithm is particularly interesting since it uses a compact data structure to encode all the (possibly exponentially many) factorizations of f. Unfortunately, their algorithm has the restriction that p^k not divide the discriminant of f. Their complexity bound, in our notation, is the sum of $d^7 k \log(p)(k\log(p)+\log d)^2$ and a term involving the complexity of finding the mod p^k reduction of a factorization over $\mathbb{Z}_p[x]$ (see Remarks 4.10–4.12 from* [15]*). The complexity of just counting the number of possible factorizations (or just the number of possible linear factors) of f from their data structure does not appear to be stated.* $\diamond$

Creating an efficient classification of the roots of f in $\mathbb{Z}/\left(p^k\right)$ (and improving the data structure from [15] by removing all restrictions on f), within time polynomial in $d + k\log p$, is a problem we hope to address in future work.

For the reader interested in implementations, we have a preliminary `Maple` implementation of Algorithm 2.3 freely downloadable from `www.math.tamu.edu/~rojas/count.map` . A few timings (all done on a Dell XPS13 Laptop with 8Gb RAM and a 256Gb ssd, running `Maple 2015` within Ubuntu Linux 14.04) are listed below:

$f(x)$	p^k	Brute-force[1]	Algorithm 2.3
Random degree 15	2^{250}	$\approx 2\times 10^{62}$ years	0.077sec.
Random degree 75	10009^{15}	$\approx 5\times 10^{47}$ years	0.116sec.
$(x-1234)^3(x-7193)^4(x-2030)^{12}$	123456791^1	9min. 18sec.	20.075sec
$(x-1234)^3(x-7193)^4(x-2030)^{12}$	123456791^{23}	$\approx 10^{173}$ years	1min. 50.323sec.

Our `Maple` implementations of brute-force and Algorithm 2.3 here are 5 lines long and 16 lines long, respectively. In particular, our random f above were generated by taking uniformly random integer coefficients in $\{0,\ldots,p^k-1\}$ and then multiplying 5 (or 25) random cubic examples together: This results in longer timings for our code than directly picking a single random polynomial of high degree. The actual numbers of roots in the last two examples are respectively 3 and
83524650739763670783591272793501499347381420700990366689774050080031654011699848668752654
47353154003992420920966387632512203162958040452324632454082330808872546949259397 3.

1.1. **A Recurrence from Partial Factorizations.** Throughout this paper, we will use the integers $\{0,\ldots,p^k-1\}$ to represent elements of $\mathbb{Z}/\left(p^k\right)$, unless otherwise specified. With this understanding, we will use the following notation:

Definition 1.3. *For any $f \in \mathbb{Z}[x]$ we let $\tilde{f}$ denote the mod p reduction of f and, for any root $\zeta_0 \in \{0,\ldots,p-1\}$ of $\tilde{f}$, we call ζ_0* degenerate *if and only if $f'(\zeta_0) = 0 \bmod p$. Letting* ord_p :

[1]The timings in years were based on extrapolating (without counting the necessary expansion of laptop memory beyond 8Gb) from examples with much smaller k already taking over an hour.

$\mathbb{Z} \longrightarrow \mathbb{N} \cup \{0\}$ *denote the usual* p*-adic valuation with* $\operatorname{ord}_p(p) = 1$*, we then define* $s(f, \varepsilon) := \min_{j \geq 0} \left\{ j + \operatorname{ord}_p \frac{f^{(j)}(\varepsilon)}{j!} \right\}$ *for any* $\varepsilon \in \{0, \ldots, p-1\}$*. Finally, fixing* $k \in \mathbb{N}$*, let us inductively define a set* $T_{p,k}(f)$ *of pairs* $(f_{i,\zeta}, k_{i,\zeta}) \in \mathbb{Z}[x] \times \mathbb{N}$ *as follows: We set* $(f_{0,0}, k_{0,0}) := (f, k)$*. Then, for any* $i \geq 1$ *with* $(f_{i-1,\mu}, k_{i-1,\mu}) \in T_{p,k}(f)$ *and any* degenerate *root* $\zeta_{i-1} \in \{0, \ldots, p-1\}$ *of* $\tilde{f}_{i-1,\mu}$ *with* $s_{i-1} := s(f_{i-1,\mu}, \zeta_{i-1}) \in \{2, \ldots, k_{i-1,\mu} - 1\}$*, we define* $\zeta := \mu + p^{i-1}\zeta_{i-1}$*,* $k_{i,\zeta} := k_{i-1,\mu} - s_{i-1}$ *and* $f_{i,\zeta}(x) := \left[\frac{1}{p^{s_{i-1}}} f_{i-1,\mu}(\zeta_{i-1} + px) \right] \bmod p^{k_{i,\zeta}}$*.* $\diamond$

The "perturbations" $f_{i,\zeta}$ of f will help us keep track of how the roots of f in $\mathbb{Z}/\left(p^k\right)$ cluster (in a p-adic metric sense) about the roots of $\tilde{f}$. Since $\frac{f^{(j)}(\varepsilon)}{j!}$ is merely the coefficient of x^j in the Taylor expansion of $f(x + \varepsilon)$ about $x = 0$, it is clear that $\frac{f^{(j)}(\varepsilon)}{j!}$ is always an integer (under the assumptions above) provided $\varepsilon \in \mathbb{Z}$.

We will see in the next section how $T_{p,k}(f)$ can be identified with a finite rooted directed tree. In particular, it is easy to see that the set $T_{p,k}(f)$ is always finite since, by construction, only $f_{i,\zeta}$ with $i \leq \lfloor (k-1)/2 \rfloor$ and $\zeta \in \mathbb{Z}/(p)$ are possible (see also Lemma 3.6 of Section 3 below).

Example 1.4. *Let us take* $p = 3$*,* $k = 7$*, and* $f(x) := x^{10} - 10x + 738$*. A simple calculation then shows that* $\tilde{f}_{0,0}(x) = x(x-1)^9$*, which has roots* $\{0, 1\}$ *in* $\mathbb{Z}/(3)$*. The root* 0 *is non-degenerate so the only possible* $f_{1,\zeta}$ *would be an* $f_{1,1} = f_{1,0+1}$*.*

In particular, $s(f_{0,0}, 1) = 4$ *and thus* $k_{1,1} = 3$ *and* $f_{1,1}(x) = 21x^4 + 13x^3 + 5x^2 + 9 \bmod 3^3$*. Since* $\tilde{f}_{1,1}(x) = x^2(x-1)$ *and* 1 *is a non-degenerate root of* $\tilde{f}_{1,1}$*, we see that the only possible* $f_{2,\zeta}$ *would be an* $f_{2,1} = f_{2,1+0}$*.*

Since $s(f_{1,1}, 0) = 2$ *we then obtain* $k_{2,1} = 1$*, and* $f_{2,1}(x) = 2(x-1)(x-2) \bmod 3$*, which has only non-degenerate roots. So by Definition 1.3 there can be no* $f_{3,\zeta}$ *and thus our collection of pairs* $T_{p,k}(f)$ *consists of just* 3 *pairs.* $\diamond$

Using base-p expansion, there is an obvious bijection between the ring $\mathbb{Z}_p$ of p-adic integers and the set of root-based paths in an infinite p-ary tree $\mathbb{T}_p$. It is then natural to use the leafs of a finite subtree of $\mathbb{T}_p$ to store the roots of f in $\mathbb{Z}/\left(p^k\right)$. This type of tree structure was studied earlier by Schmidt and Stewart in [27, 28], from the point of view of classification and (in our notation) upper bounds on $N_{p,k}(f)$. However, it will be more algorithmically efficient to instead endow our set $T_{p,k}(f)$ with a tree structure. The following fundamental lemma relates $N_{p,k}(f)$ to a recursion tree structure on $T_{p,k}(f)$.

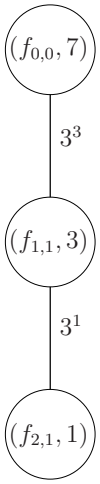
Lemma 1.5. *Following the notation above, let* $n_p(f_{0,0})$ *denote the number of non-degenerate roots of* $\tilde{f}_{0,0}$ *in* $\mathbb{Z}/(p)$*. Then, provided* $k \geq 2$ *and* $f_{0,0}$ *is not identically* 0 *in* $(\mathbb{Z}/(p))[x]$*, we have*

$$N_{p,k}(f_{0,0}) \;=\; n_p(f_{0,0}) + \left(\sum_{\substack{\zeta_0 \in \mathbb{Z}/(p) \\ s(f_{0,0},\zeta_0) \geq k}} p^{k-1} \right) + \sum_{\substack{\zeta_0 \in \mathbb{Z}/(p) \\ s(f_{0,0},\zeta_0) \in \{2,\ldots,k-1\}}} p^{s(f_{0,0},\zeta_0)-1} N_{p,k-s(f_{0,0},\zeta_0)}(f_{1,\zeta_0}).$$

We prove Lemma 1.5 in the next section, where it will immediately follow that Lemma 1.5 applies recursively, i.e., our root counting formula still holds if one replaces $(f_{0,0}, k, f_{1,\zeta_0}, \zeta_0)$ with $(f_{i-1,\mu}, k_{i-1,\mu}, f_{i,\mu+p^{i-1}\zeta_{i-1}}, \zeta_{i-1})$. There we also show how Lemma 1.5 leads to our recursive algorithm (Algorithm 2.3) for computing $N_{p,k}(f)$. In essence, the third sum term above is what creates children for a node corresponding to $f_{i-1,\mu}$.

Note that by construction, $s(f,\zeta_0)\geq 2$ implies that ζ_0 is a degenerate root of $\tilde{f}$. So the last two sum terms in the formula (from Lemma 1.5 above) range over certain degenerate roots of $\tilde{f}$. Note also that $N_{p,k}(f)$ depends only on the residue class of f mod p^k, so we will often abuse the notations $N_{p,k}(f)$ and $s(f,\zeta_0)$ by allowing $f\in\left(\mathbb{Z}/\left(p^k\right)\right)[x]$ as well. The following example illustrates how $N_{p,k}(f)$ can be computed recursively.

Example 1.6.
Revisiting Example 1.4, let us count the roots in $\mathbb{Z}/(3^7)$ of $f(x) := x^{10} - 10x + 738$. Lemma 1.5 and our earlier computation of $T_{p,k}(f)$ then tell us that $N_{3,7}(f) = 1 + 3^3 N_{3,3}(f_{1,1})$ and $N_{3,3}(f_{1,1}) = 1 + 3^1 N_{3,1}(f_{2,1})$ where $f_{2,1}(x) = 2(x-1)(x-2)$. So we obtain $N_{3,7}(f) = 1 + 3^3(1 + 3^1 \cdot 2) = 190$. (Our `Maple` *implementation confirmed this count in under 4 milliseconds.) We illustrate the corresponding tree structure (defined in Section 3 below) on the right. Note that the powers of 3 in the expression $1 + 3^3(1 + 3^1 \cdot 2)$ appear as edge labels in our tree, but the contribution of* non*-degenerate roots to our count is not notated on our tree.* $\diamond$



While the tree from our example above has just 3 nodes, the earlier tree structure from [27, 28] would have resulted in over 190 nodes. We will now fully detail how to efficiently reduce root counting over $\mathbb{Z}/\left(p^k\right)$ to computing p-adic valuations and factoring in $(\mathbb{Z}/(p))[x]$.

2. Algebraic Preliminaries and Our Algorithm

Let us first recall the following version of Hensel's Lemma:

Lemma 2.1. *(See, e.g.,* [24, Thm. 2.3, Pg. 87, Sec. 2.6]*.) Suppose $k\in\mathbb{N}$, $f\in\mathbb{Z}[x]$ is not identically 0 in $(\mathbb{Z}/(p))[x]$, and $\zeta_0\in\mathbb{Z}/(p)$ is a non-degenerate root of $\tilde{f}$. Then there is a unique $\zeta\in\mathbb{Z}/\left(p^k\right)$ with $\zeta=\zeta_0 \bmod p$ and $f(\zeta)=0 \bmod p^k$.* ■

The following lemma enables us to understand the lifts of degenerate roots of $\tilde{f}$.

Lemma 2.2. *Following the notation of Lemma 2.1, suppose instead that $\zeta_0\in\mathbb{Z}/(p)$ is a root of $\tilde{f}$ of (finite) multiplicity $m\geq 2$. Suppose also that $k\geq 2$ and that there is a $\zeta\in\mathbb{Z}/\left(p^k\right)$ with $\zeta=\zeta_0 \bmod p$ and $f(\zeta)=0 \bmod p^k$. Then $s(f,\zeta_0)\in\{2,\ldots,m\}$.*

Proof of Lemma 2.2: We may assume, by base-p expansion that $\zeta=\zeta_0+p\zeta_1+\cdots+p^{k-1}\zeta_{k-1}$ for some $\zeta_1,\ldots,\zeta_{k-1}\in\{0,\ldots,p-1\}$. Note that $f'(\zeta_0)=0 \bmod p$ since ζ_0 is a degenerate root. Note also that $j+\operatorname{ord}_p\frac{f^{(j)}(\zeta_0)}{j!}\geq 2$ for all $j\geq 2$. Letting $\sigma:=\zeta_1+p\zeta_2+\cdots+p^{k-2}\zeta_{k-1}$ we then see by Taylor expansion that $f(\zeta)=f(\zeta_0)+f'(\zeta_0)p\sigma+\cdots+\frac{f^{(k-1)}}{(k-1)!}p^{k-1}\sigma^{k-1} \bmod p^k$. So $f(\zeta)=0 \bmod p^k$ implies that $f(\zeta_0)=0 \bmod p^2$ and thus $s(f,\zeta_0)\geq 2$.

To conclude, by Taylor expansion about ζ_0, our multiplicity assumption implies that $\frac{f^{(m)}(\zeta_0)}{m!}$ is an integer not divisible by p. So $m+\operatorname{ord}_p\frac{f^{(m)}(\zeta_0)}{m!}=m$ and thus $s(f,\zeta_0)\leq m$. ■

We are now ready to state our main algorithm.

Algorithm 2.3 (`RandomizedPrimePowerRootCounting`(f,p,k)).
Input. $(f,p,k)\in\mathbb{Z}[x]\times\mathbb{N}\times\mathbb{N}$ *with* p *prime and* $f(x)=c_0+\cdots+c_dx^d$.
Output. *An integer* $M\leq N_{p,k}(f)$ *that, with probability at least* $\frac{2}{3}$, *is exactly* $N_{p,k}(f)$.
Description.
1: **Let** $v:=\min\limits_{i\in\{0,\ldots,d\}}\operatorname{ord}_pc_i$ *and* $f_{0,0}:=f$.
2: **If** $v\geq k$
3: **Let** $M:=p^k$. **Return**.
4: **Elseif** $v\in\{1,\ldots,k-1\}$
5: **Let** $M:=p^v$`RandomizedPrimePowerRootCounting`$\left(\frac{f_{0,0}(x)}{p^v},p,k-v\right)$. **Return**.
6: **End(If)**.
7: **Let** $M:=\deg\gcd(\tilde{f}_{0,0}/\gcd(\tilde{f}_{0,0},\tilde{f}'_{0,0}),x^p-x)$.
8: **For** $\zeta_0\in\mathbb{Z}/(p)$ *a degenerate root of* $\tilde{f}_{0,0}$ **do**[2]
9: **Let** $s:=s(f_{0,0},\zeta_0)$.
10: **If** $s\geq k$
11: **Let** $M:=M+p^{k-1}$.
12: **Elseif** $s\in\{2,\ldots,k-1\}$
13: **Let** $M:=M+p^{s-1}$`RandomizedPrimePowerRootCounting`$(f_{1,\zeta_0},p,k-s)$.
14: **End(If)**.
15: **End(For)**.
16: **If** *the preceding* **For** *loop did* not *access all the degenerate roots of* $\tilde{f}_{0,0}$
17: *Print* ``Sorry, your Las Vegas factoring method failed. You have an under-count so you should try re-running.''
18: **End(If)**.
19: Print ``If you've seen no under-count messages then your count is correct!''
20: **Return**.

Before proving the correctness of Algorithm 2.3, it will be important to prove our earlier key lemma.

Proof of Lemma 1.5: Proving our formula clearly reduces to determining how many lifts each possible root $\zeta_0\in\mathbb{Z}/(p)$ of $\tilde{f}_{0,0}$ has to a root of $f_{0,0}$ in $\mathbb{Z}/\left(p^k\right)$. Toward this end, note that Lemma 2.1 implies that each non-degenerate ζ_0 lifts to a unique root of $f_{0,0}$ in $\mathbb{Z}/\left(p^k\right)$. In particular, this accounts for the summand $n_p(f_{0,0})$ in our formula. So now we merely need to count the lifts of the degenerate roots.

Assume $\zeta_0\in\mathbb{Z}/(p)$ is a degenerate root of $\tilde{f}_{0,0}$, write $\zeta=\zeta_0+p\zeta_1+\cdots+p^{k-1}\zeta_{k-1}\in\mathbb{Z}/\left(p^k\right)$ via base-p expansion as before, set $\sigma:=\zeta_1+p\zeta_2+\cdots+p^{k-2}\zeta_{k-1}$, and let $s:=s(f_{0,0},\zeta_0)$. Clearly then, $f_{0,0}(\zeta)=p^sf_{1,\zeta_0}(\sigma) \bmod p^k$ and, by construction, $f_{1,\zeta_0}\in\mathbb{Z}[x]$ and is not identically 0 in $(\mathbb{Z}/(p))[x]$.

If $s\geq k$ then $f_{0,0}(\zeta)=0 \bmod p^k$ independent of σ. So there are exactly p^{k-1} values of $\zeta\in\mathbb{Z}/\left(p^k\right)$ with $\zeta=\zeta_0 \bmod p$. This accounts for the second summand in our formula.

If $s\leq k-1$ then ζ is a root of $f_{0,0}$ with $\zeta=\zeta_0 \bmod p$ if and only if $f_{1,\zeta_0}(\sigma)=0 \bmod p^{k-s}$. Also, $s\geq 2$ (thanks to Lemma 2.2) because ζ_0 is a degenerate root. Since the base-p digits $\zeta_{k-s+1},\ldots,\zeta_{k-1}$ do not appear in the last equality, the number of possible lifts ζ of ζ_0 is thus

[2]Here we use the fastest available Las Vegas factoring algorithm over $(\mathbb{Z}/(p))[x]$ (currently [18]) to isolate the degenerate roots of $\tilde{f}$. Such factoring algorithms enable us to correctly announce failure to find all the degenerate roots, should this occur. We describe in the next section how to efficiently control the error probability.

exactly p^{s-1} times the number of roots $\zeta_1 + p\zeta_2 + \cdots + p^{k-s-1}\zeta_{k-s} \in \mathbb{Z}/\left(p^{k-s}\right)$ of f_{1,ζ_0}. So this accounts for the third summand in our formula and we are done. ■

We are at last ready to prove the correctness of Algorithm 2.3.

Proof of Correctness of Algorithm 2.3: Assume temporarily that Algorithm 2.3 is correct when $f_{0,0}$ is *not* identically 0 in $(\mathbb{Z}/(p))[x]$. Since (for any integers a, x, y with $a \leq k$) $p^a x = p^a y \bmod p^k \iff x = y \bmod p^{k-a}$, Steps 1–6 of our algorithm then clearly correctly dispose of the case where f is identically 0 in $(\mathbb{Z}/(p))[x]$. So let us now prove correctness when f is *not* identically 0 in $(\mathbb{Z}/(p))[x]$. Applying Lemma 1.5, we then see that it is enough to prove that the value of M is the value of our formula for $N_{p,k}(f)$ when the **For** loops of Algorithm 2.3 runs correctly.

Step 7 ensures that the value of M is initialized as $n_p(f)$. Steps 8–15 (once the **For** loop is completed) then simply add the second and third summands of our formula to M thus ensuring that $M = N_{p,k}(f)$, *provided* the **For** loop has run correctly, along with all the **For** loops in the recursive calls to `RandomizedPrimePowerRootCounting`. Should any of these **For** loops run incorrectly, Steps 16–20 ensure that our algorithm correctly announces an under-count.[3] So we are done. ■

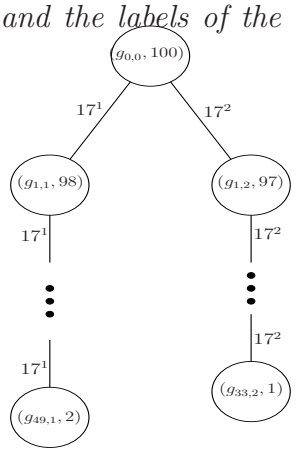
3. Our Complexity Bound: Proving Theorem 1.1

Let us now introduce a tree structure on $T_{p,k}(f)$ that will enable our complexity analysis.

Definition 3.1. *Let us identify the elements of $T_{p,k}(f)$ with nodes of a labelled rooted directed tree $\mathcal{T}_{p,k}(f)$ defined inductively as follows:*

1. *We set $f_{0,0} := f$, $k_{0,0} := k$, and let $(f_{0,0}, k_{0,0})$ be the label of the root node of $\mathcal{T}_{p,k}(f)$.*
2. *The non-root nodes of $\mathcal{T}_{p,k}(f)$ are uniquely labelled by each $(f_{i,\zeta}, k_{i,\zeta}) \in T_{p,k}(f)$ with $i \geq 1$.*
3. *There is an edge from node $(f_{i',\zeta'}, k_{i',\zeta'})$ to node $(f_{i,\zeta}, k_{i,\zeta})$ if and only if $i' = i-1$ and there is a degenerate root $\zeta_{i-1} \in \mathbb{Z}/(p)$ of $\tilde{f}_{i',\zeta'}$ with $s(f_{i',\zeta'}, \zeta_{i-1}) \in \{2, \ldots, k_{i',\zeta'} - 1\}$ and $\zeta = \zeta' + p^{i-1}\zeta_{i-1} \in \mathbb{Z}/(p^i)$.*
4. *The label of a directed edge from node $(f_{i',\zeta'}, k_{i',\zeta'})$ to node $(f_{i,\zeta}, k_{i,\zeta})$ is $p^{s\left(f_{i',\zeta'}, (\zeta-\zeta')/p^{i'}\right)-1}$.*

In particular, the edges are labelled by powers of p in $\{p^1, \ldots, p^{k-2}\}$, and the labels of the nodes lie in $\mathbb{Z}[x] \times \mathbb{N}$. ◇

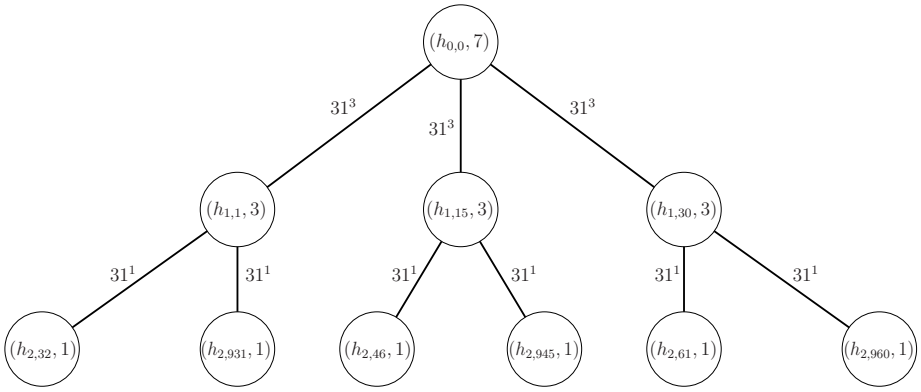


Example 3.2. *Letting $g(x) := x^5 - 8x^4 + 25x^3 - 38x^2 + 28x - 8$, the tree $\mathcal{T}_{17,100}(g)$ is drawn to the right: Note that $\mathcal{T}_{17,100}(g)$ has depth $\lfloor (100-1)/2 \rfloor = 49$ and exactly $1 + \lfloor (100-1)/2 \rfloor + \lfloor (100-1)/3 \rfloor = 83$ nodes. To count the roots of g in $\mathbb{Z}/(17^{100})$ one can then easily calculate that $g_{0,0}(x) = (x-1)^2(x-2)^3$, $g_{1,1}(x) = x^2(4913x^3 - 867x^2 + 51x - 1)$ and $g_{1,2}(x) = x^3(289x^2 + 34x + 1)$. The last two polynomials have* no *nonzero roots mod* 17. *A bit more computation then yields $\tilde{g}_{i,1}(x) = -x^2$ for all $i \in \{1, \ldots, 49\}$ and $\tilde{g}_{j,2}(x) = x^3$ for all $j \in \{1, \ldots, 33\}$. Also, $N_{17,2}(g_{49,1}) = 17$ by Lemma 1.5 and $N_{17,1}(g_{33,2}) = 1$ trivially. So by Lemma 1.5 once more, $g_{0,0}$ has exactly $17 \cdot 17^{49} \cdot 1 + 17^2 \cdot (17^2)^{32} \cdot 1 = 17^{50} + 17^{66}$ roots in $\mathbb{Z}/(17^{100})$. Expanded in base-*10*, this count is* 162042453765370612419692325878157575935987567591343647038024548627637893995166018. ◇

[3] Note that checking for an under-count can be reduced to an irreducibility check in $\mathbb{F}_p[x]$, which can be done in deterministic polynomial-time: See, e.g., [15, Cor. 14.35, Algor. 14.36, & Thm. 14.37, pp. 406–408].

Remark 3.3. *Our trees $\mathcal{T}_{p,k}(\cdot)$ thus encode algebraic expressions for our desired root counts $N_{p,k}(\cdot)$. In particular, the children of a node labelled (f_i, k_i) yield terms (corresponding to the child nodes) that one sums to get the root count $N_{p,k_i}(f_i)$, and the edge labels yield weights multiplying the corresponding terms. (The contribution from non-degenerate roots is not visible from the tree but does influence each $N_{p,k_i}(f_i)$, as detailed by Lemma 1.5.)* $\diamond$

Example 3.4. *Suppose we set $p=31$, $k=7$, and we define $h(x)$ to be*
$x^{12} - 60x^{11} - 4420x^{10} + 275040x^9 + 8287728x^8 - 502626240x^7 - 8802489280x^6 - 10069291727x^5$
$-6168330858x^4 - 10982634616x^3 + 6650045702x^2 - 4862117081x - 6450915579$.
Then the tree $\mathcal{T}_{31,7}(h)$ has the following structure:



In particular, the polynomials corresponding to the depth 1 nodes are exactly
$h_{1,1} = 9610x^6 + 13640x^5 + 25563x^4 + 2511x^3 + 9417x^2 + 13640x + 14992$,
$h_{1,15} = 22103x^6 + 1674x^5 + 11825x^4 + 26443x^3 + 29205x^2 + 1674x + 26240$, *and*
$h_{1,30} = 24986x^6 + 28520x^5 + 22228x^4 + 2542x^3 + 29541x^2 + 28520x + 12618$.
Also, the polynomials corresponding to the depth 2 nodes are exactly $h_{2,1+1\cdot 31} = h_{2,1+30\cdot 31} = 14x^2$ and $h_{2,15+1\cdot 31} = h_{2,15+30\cdot 31} = h_{2,30+1\cdot 31} = h_{2,30+30\cdot 31} = 4x^2$. So Lemma 1.5 tells us that h has exactly $6 \cdot 31^4 \cdot 1 = 5541126$ roots in $\mathbb{Z}/(31^7)$.

The following lemma will be central in our complexity analysis.

Lemma 3.5. *Suppose $k, p \in \mathbb{N}$ with p prime, $f \in \mathbb{Z}[x]$ has degree d, and $(f_{i-1,\mu}, k_{i-1,\mu})$ is any node of $\mathcal{T}_{p,k}(f)$. Then $\sum \deg \tilde{f}_{i,\zeta} \leq \deg \tilde{f}_{i-1,\mu}$, where the sum ranges over all child nodes $(f_{i,\zeta}, k_{i,\zeta})$ of $(f_{i-1,\mu}, k_{i-1,\mu})$.* ■

Lemma 3.5 follows immediately from the last sentence of Assertion (3) of the more refined lemma below:

Lemma 3.6. *Following the notation of Lemma 3.5, we have that:*

(1) The depth of $\mathcal{T}_{p,k}(f)$ is at most $\lfloor (k-1)/2 \rfloor$.
(2) The degree of the root node of $\mathcal{T}_{p,k}(f)$ is at most $\lfloor d/2 \rfloor$.
(3) The degree of any non-*root node of $\mathcal{T}_{p,k}(f)$ labelled $(f_{i,\zeta}, k_{i,\zeta})$, with parent $(f_{i-1,\mu}, k_{i-1,\mu})$ and $\zeta_{i-1} := (\zeta - \mu)/p^{i-1}$, is at most $\lfloor s(f_{i-1,\mu}, \zeta_{i-1})/2 \rfloor$. In particular, $\deg \tilde{f}_{i,\zeta} \leq s(f_{i-1,\mu}, \zeta_{i-1}) \leq k_{i-1,\mu} - 1 \leq k-1$ and $\sum_{\substack{(f_{i,\zeta},k_{i,\zeta}) \text{ a child} \\ \text{of } (f_{i-1,\mu},k_{i-1,\mu})}} s(f_{i-1,\mu}, \zeta_{i-1}) \leq \deg \tilde{f}_{i-1,\mu}$.*
(4) $\mathcal{T}_{p,k}(f)$ has at most $\left\lfloor \frac{d}{2} \right\rfloor$ nodes at depth $i \geq 1$, and thus a total of no more than $1 + \left\lfloor \frac{d}{2} \right\rfloor \left\lfloor \frac{k-1}{2} \right\rfloor$ nodes.

Proof of Lemma 3.6:

Assertion (1): *By Definitions 1.3 and 3.1, the labels* $(f_{i,\zeta}, k_{i,\zeta})$ *satisfy* $2 \leq k_{i-1,\mu} - k_{i,\zeta} \leq k_{i-1,\mu} - 1$ *for any child* $(f_{i,\zeta}, k_{i,\zeta})$ *of* $(f_{i-1,\mu}, k_{i-1,\mu})$*, and* $1 \leq k_{i,\zeta} \leq k-2$ *for all* $i \geq 1$*. So considering any root to leaf path in* $\mathcal{T}_{p,k}(f)$*, it is clear that the depth of* $\mathcal{T}_{p,k}(f)$ *can be no greater than* $1 + \lfloor (k-2-1)/2 \rfloor = \lfloor (k-1)/2 \rfloor$*.* ■

Assertion (2): *Since* $\tilde{f}_{0,0} = \tilde{f}$ *has degree* $\leq d$*, and the multiplicity of any degenerate root of* $\tilde{f}_{0,0}$ *is at least* 2*, we see that* $\tilde{f}_{0,0}$ *has no more than* $\lfloor d/2 \rfloor$ *degenerate roots in* $\mathbb{Z}/(p)$*. Every edge emanating from the root node of* $\mathcal{T}_{p,k}(f)$ *corresponds to a unique degenerate root of* $\tilde{f}_{0,0}$ *(and not every degenerate root of* $\tilde{f}$ *need yield a valid edge emanating from the root of* $\mathcal{T}_{p,k}(f)$*), so we are done.* ■

Assertion (3): *The degree bound for non-root nodes follows similarly to the degree bound for the root node: Letting* $s := s(f_{i-1,\mu}, \zeta_{i-1})$*, it suffices to prove that* $\deg \tilde{f}_{i,\zeta} \leq s$ *for all* $i \geq 1$*. Note that we must have*

$$s = \min_{j \in \{0,\ldots,k_{i,\zeta}-1\}} \left\{ j + \operatorname{ord}_p \frac{f_{i-1,\mu}^{(j)}(\zeta_{i-1})}{j!} \right\},$$

since $f_{i,\zeta} \in \left(\mathbb{Z}/\left(p^{k_{i,\zeta}}\right)\right)[x]$ *for* $i \geq 1$*. So then, the coefficient of* x^ℓ *in* $f_{i-1,\mu}(\zeta_{i-1} + px)$ *must be divisible by* p^{s+1} *for all* $\ell \geq s+1$*. In other words, the coefficient of* x^ℓ *in* $f_{i,\zeta}(x)$ *must be divisible by* p *for all* $\ell \geq s+1$*, and thus* $\deg \tilde{f}_{i,\zeta} \leq s$*. That* $s \leq k_{i-1,\mu} - 1$ *follows from the definition of* $s(f,\zeta)$*, and* $k_{i-1,\mu} \leq k$ *since* $k_{0,0} := k$ *and (thanks to Definition 1.3)* $k_{i-1,\mu} > k_{i,\zeta}$*.*

To prove the final bound, note that Lemma 2.2 implies that each term $s(f_{i-1,\mu}, \zeta_{i-1})$ *in the sum is at most the multiplicity of the root* ζ_{i-1} *of* $\tilde{f}_{i-1,\mu}$*. Since the sum of the multiplicities of the degenerate roots of* $\tilde{f}_{i-1,\mu}$ *is no greater than* $\deg \tilde{f}_{i-1,\mu}$*, we are done.* ■

Assertion (4): *By Assertion (3), the sum of the degrees of the* $\tilde{f}_{1,\zeta_0}$ *(as* $(f_{1,\zeta_0}, k_{1,\zeta_0})$ *ranges over all depth* 1 *node labels of* $\mathcal{T}_{p,k}(f)$*) is no greater than* $\deg \tilde{f}_{0,0}$*, which is at most* d*.*

By applying Assertion (3) to all nodes of depth $i \geq 2$*, the sum of the degrees of the* $\tilde{f}_{i,\zeta}$ *(as* $(f_{i,\zeta}, k_{i,\zeta})$ *ranges over all depth* i *node labels of* $\mathcal{T}_{p,k}(f)$*) is no greater than the sum of the degrees of the* $\tilde{f}_{i-1,\mu}$ *(as* $(f_{i-1,\mu}, k_{i-1,\mu})$ *ranges over all depth* $i-1$ *node labels of* $\mathcal{T}_{p,k}(f)$*).*

Since $\deg \tilde{f}_{0,0} \leq d$ *we thus obtain that, for* every *depth* i*, the sum of the degrees of the* $\tilde{f}_{i,\zeta}$ *(as* $(f_{i,\zeta}, k_{i,\zeta})$ *ranges over all depth* i *node labels of* $\mathcal{T}_{p,k}(f)$*) is no greater than* d*. So by the final part of Assertion (3), our tree* $\mathcal{T}_{p,k}(f)$ *has no more than* $\lfloor d/2 \rfloor$ *nodes at any fixed depth* ≥ 1*. So by Assertion (1) we are done.* ■

We are at last ready to prove our main theorem.

Proof of Theorem 1.1: *Since we already proved at the end of the last section that Algorithm 2.3 is correct, it suffices to prove the stated complexity bound for Algorithm 2.3. Proving that Algorithm 2.3 runs as fast as stated will follow easily from (a) the fast randomized Kedlaya-Umans factoring algorithm from* [18] *and (b) applying Lemmata 3.5 and 3.6 to show that the number of necessary factorizations and p-adic valuation calculations is well-bounded.*

More precisely, the **For** *loops and recursive calls of Algorithm 2.3 can be interpreted as a depth-first search of* $\mathcal{T}_{p,k}(f)$*, with* $\mathcal{T}_{p,k}(f)$ *being built along the way. In particular, we begin at the root node by factoring* $\tilde{f}_{0,0} = \tilde{f}$ *in* $(\mathbb{Z}/(p))[x]$ *via* [18]*, in order to find the degenerate roots of* $\tilde{f}$*. (Factoring in fact dominates the complexity of the gcd computation that gives us* $n_p(f_{0,0})$*, if we use a deterministic near linear-time gcd algorithm such as that of Knuth and Schönhage*

(see, e.g., [4, Ch. 3]*).) This factorization takes time* $(d^{1.5}\log p)^{1+o(1)} + \left(d\log^2 p\right)^{1+o(1)}$ *and requires* $O(d\log p)$ *random bits.*

Now, in order to continue the recursion, we need to compute p-adic valuations of polynomial coefficients in order to find the $s(f_{0,0},\zeta_0)$ *and determine the edges emanating from our root. Expanding each* $f_{0,0}(\zeta_0+px)$ *can clearly be done mod* p^k*, so each such expansion takes time no worse than* $d^2(k\log p)^{1+o(1)}$ *via Horner's method and fast finite ring arithmetic (see, e.g.,* [2, 14]*). Computing* $s(f_{0,0},\zeta_0)$ *then takes time no worse than* $d(k\log p)^{1+o(1)}$ *using, say, the standard binary method for evaluating powers of p. There are no more than* $\lfloor d/2\rfloor$ *possible* ζ_0 *(thanks to Lemma 3.6), so the total work so far is*

$$d^3(k\log p)^{1+o(1)} + \left(d\log^2 p\right)^{1+o(1)}.$$

(To simplify our bound, we are rolling multiplicative constants into the exponent, at the price of a negligible increase in the little-$o(\cdot)$ *terms in the exponent.) Note that now, computing the expansion* $f_{0,0}(\zeta_0+px)$ *dominates the factorization of* $\tilde{f}_{0,0}$*.*

The remaining work can then be bounded similarly, but with one small twist: By Assertion (4) of Lemma 3.6, the number of nodes at depth i of our tree is never more than $\lfloor d/2\rfloor$ *and, by Lemma 3.5, the sum of the degrees of the* $\tilde{f}_{i,\zeta}$ *at level i is no greater than d.*

Now observe that (for $i\geq 2$*) the amount of work needed to compute the* $s(f_{i-1,\mu},\zeta_{i-1})$ *at level* $i-1$ *(which are used to define the polynomials at level i) is no greater than* $d\cdot d(k\log p)^{1+o(1)}$*, and this will be dominated by the subsequent computations of the expansions of the* $f_{i,\zeta}$*. In particular, by the basic calculus inequality* $r_1^t+\cdots+r_\ell^t \leq (r_1+\cdots+r_\ell)^t$ *(valid for any* $r,t\geq 1$*), the total amount of work for the factorizations for each subsequent level of* $\mathcal{T}_{p,k}(f)$ *will be*

$$d^{1.5}(\log p)^{1+o(1)} + \left(d\log^2 p\right)^{1+o(1)},$$

with $O(d\log p)$ *random bits needed. The expansions of the* $f_{i,\zeta}$ *at level i will take time no greater than* $d^3(k\log p)^{1+o(1)}$ *to compute. So our total work at each subsequent level is then*

$$d^3(k\log p)^{1+o(1)} + \left(d\log^2 p\right)^{1+o(1)}.$$

So then, the total amount of work for our entire tree will be

$$kd^3(k\log p)^{1+o(1)} + k\left(d\log^2 p\right)^{1+o(1)}.$$

and the number of random bits needed is $O(dk\log p)$*.*

We are nearly done, but we must still ensure that our algorithm has the correct Las Vegas properties. In particular, while finite field factoring can be assumed to succeed with probability $\geq 2/3$*, we use multiple calls to finite field factoring,* each of which could fail. *The simplest solution is to simply run our finite field factoring algorithm sufficiently many times to reduce the* over-all *error probability. In particular, thanks to Lemma 3.6, and the basic union bound for probabilities, it is enough to enforce a success probability of* $O\left(\frac{1}{dk}\right)$ *for each application of finite field factoring. This implies that we should run the algorithm from* [18] $O(\log(dk))$ *many times each time we need a factorization over* $(\mathbb{Z}/(p))[x]$*. So, multiplying our last total by* $\log(dk)$*, this yields a final complexity bound of*

$$kd^3(k\log p)^{1+o(1)} + \left(dk\log^2 p\right)^{1+o(1)}$$

(since computing the expansions of the $f_{i,\mu}(\zeta_{i-1}+x)$ *dominates our complexity) and a total number of* $O(dk\log(dk)\log p)$ *random bits needed.*

To conclude, note that as our algorithm proceeds with depth first search, we need only keep track of collections of $f_{i,\zeta}$ occuring along a root-to-leaf path in $\mathcal{T}_{p,k}(f)$. A polynomial of degree d with integer coefficients all of absolute value less than p^k requires $O(dk \log p)$ bits to store, and Lemma 3.6 tells us that the depth of $\mathcal{T}_{p,k}(f)$ is $O(k)$. So we never need more than $O(dk^2 \log p)$ bits of memory. ■

Acknowledgements

We are grateful to Qi Cheng, Shuhong Gao, and Daqing Wan for many useful conversations. We also humbly thank Joachim von zur Gathen for his kind encouragement, and the referee for suggestions that clarified our exposition.

References

[1] Martín Avendaño; Ashraf Ibrahim; J. Maurice Rojas; and Korben Rusek, *"Faster p-adic Feasibility for Certain Multivariate Sparse Polynomials,"* Journal of Symbolic Computation, special issue in honor of 60th birthday of Joachim von zur Gathen, vol. 47, no. 4, pp. 454–479 (April 2012).

[2] Eric Bach and Jeff Shallit, *Algorithmic Number Theory, Vol. I: Efficient Algorithms,* MIT Press, Cambridge, MA, 1996.

[3] Jérémy Berthomieu; Grégoire Lecerf; and Guillaume Quintin, *"Polynomial root finding over local rings and application to error correcting codes,"* Applicable Algebra in Engineering, Communication, and Computing, December 2013, Volume 24, Issue 6, pp. 413–443.

[4] P. Bürgisser, M. Clausen, and M.A. Shokrollahi, *Algebraic complexity theory*, with the collaboration of Thomas Lickteig, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], 315, Springer-Verlag, Berlin, 1997.

[5] David G. Cantor and Daniel M. Gordon, *"Factoring polynomials over p-adic fields,"* Algorithmic number theory (Leiden, 2000), pp. 185–208, Lecture Notes in Comput. Sci., 1838, Springer, Berlin, 2000.

[6] Wouter Castryck; Jan Denef; and Frederik Vercauteren, *"Computing Zeta Functions of Nondegenerate Curves,"* International Mathematics Research Papers, vol. 2006, article ID 72017, 2006.

[7] Antoine Chambert-Loir, *"Compter (rapidement) le nombre de solutions d'équations dans les corps finis,"* Séminaire Bourbaki, Vol. 2006/2007, Astérisque No. 317 (2008), Exp. No. 968, vii, pp. 39-–90.

[8] Qi Cheng, *"Primality Proving via One Round in ECPP and One Iteration in AKS,"* Journal of Cryptology, July 2007, Volume 20, Issue 3, pp. 375–387.

[9] Qi Cheng; Shuhong Gao; J. Maurice Rojas; and Daqing Wan, *"Counting Roots for Polynomials Modulo Prime Powers,"* Proceedings of ANTS XIII (Algorithmic Number Theory Symposium, July 16–20, 2018, University of Wisconsin, Madison), Mathematical Sciences Publishers (Berkeley, California), to appear.

[10] Alexander L. Chistov, *"Efficient Factoring [of] Polynomials over Local Fields and its Applications,"* in I. Satake, editor, Proc. 1990 International Congress of Mathematicians, pp. 1509–1519, Springer-Verlag, 1991.

[11] Henri Cohen, *A course in computational algebraic number theory,* Graduate Texts in Mathematics, 138, Springer-Verlag, Berlin, 1993.

[12] Jan Denef, *"Report on Igusa's local zeta function,"* Séminaire Bourbaki 1990/1991 (730-744) in Astérisque 201–203 (1991), pp. 359–386.

[13] Shuhong Gao, Frank Volny IV, and Mingsheng Wang, *"A new framework for computing Gröbner bases"*, *Mathematics of Computation*, 85 (2016), no. 297, 449–465.

[14] Joachim von zur Gathen and Jürgen Gerhard, *"Modern Computer Algebra,"* 3rd ed., Cambridge University Press, 2013.

[15] Joachim von zur Gathen and Silke Hartlieb, *"Factoring Modular Polynomials,"* J. Symbolic Computation (1998) **26**, pp. 583–606.

[16] Jordi Guàrdia; Enric Nart; Sebastian Pauli, *"Single-factor lifting and factorization of polynomials over local fields,"* Journal of Symbolic Computation 47 (2012), pp. 1318–1346.

[17] Jun-Ichi Igusa, *"Complex powers and asymptotic expansions I: Functions of certain types,"* Journal für die reine und angewandte Mathematik, 1974 (268–269): 110–130.

[18] Kiran Kedlaya and Christopher Umans, *"Fast polynomial factorization and modular composition,"* SIAM J. Comput., 40 (2011), no. 6, pp. 1767–1802.
[19] Adam R. Klivans, *Factoring Polynomials Modulo Composites,* Master's Thesis, Carnegie Mellon University Computer Science Technical Report CMU-CS-97-136, 1997.
[20] Alan G. B. Lauder, *"Counting solutions to equations in many variables over finite fields,"* Found. Comput. Math. 4 (2004), no. 3, pp. 221–267.
[21] Alan G. B. Lauder and Daqing Wan, *"Counting points on varieties over finite fields of small characteristic,"* Algorithmic number theory: lattices, number fields, curves and cryptography, pp. 579-–612, Math. Sci. Res. Inst. Publ., 44, Cambridge Univ. Press, Cambridge, 2008.
[22] Arjen K. Lenstra; Hendrik W. Lenstra (Jr.); Laszlo Lovász, *"Factoring polynomials with rational coefficients,"* Math. Ann. 261 (1982), no. 4, pp. 515–534.
[23] Bernard R. McDonald, *Finite Rings with Identity,* Marcel Dekker, Inc., New York, 1974.
[24] Ivan Niven; Herbert S. Zuckerman; and Hugh L. Montgomery, *An Introduction to the Theory of Numbers,* fifth edition, John Wiley & Sons, Inc., 1991.
[25] R. Raghavendran, *"Finite associative rings,"* Compositio Mathematica, tomme 21, no. 2 (1969), pp. 195–229.
[26] Ana Sălăgean, *"Factoring polynomials over $\mathbb{Z}_4$ and over certain Galois rings,"* Finite Fields and Their Applications 11 (2005), pp. 56–70.
[27] Wolfgang M. Schmidt, *"Solutions trees of polynomial congruences modulo prime powers,"* Number theory in progress (K. Győry, H. Iwaniec, J. Urbanowicz, eds.), Vol. 1, proceedings of a conference in honor of 60th birthday of Andrzej Schnizel (Zakopane, Poland, June 30–July 9, 1997), Walter de Gruyter GhmH & Co. KG, D-10785 Berlin, 1999.
[28] Wolfgang M. Schmidt and C. L. Stewart, *"Congruences, Trees, and p-adic Integers,"* Transactions of the American Mathematical Society, vol. 349, No. 2, Feb. 1997, pp. 605–639.
[29] Carlo Sircana, *Factoring polynomials over $\mathbb{Z}/(n)$,* Ph.D. thesis, University of Pisa, Italy, 2016.
[30] Daqing Wan, *"Algorithmic theory of zeta functions over finite fields,"* Algorithmic number theory: lattices, number fields, curves and cryptography, pp. 551–578, Math. Sci. Res. Inst. Publ., 44, Cambridge Univ. Press, Cambridge, 2008.
[31] W. A. Zuniga-Galindo, *"Computing Igusa's Local Zeta Functions of Univariate Polynomials, and Linear Feedback Shift Registers,"* Journal of Integer Sequences, Vol. 6 (2003), Article 03.3.6.

Email address: LHK0002@auburn.edu

Mathematics Department, Auburn University, 221 Parker Hall, Auburn, Alabama 36849

Email address: natgrandall@gmail.com

Department of Mathematics, Austin College, 900 N. Grand Ave. Sherman, TX 75090

Email address: rojas@math.tamu.edu

TAMU 3368, College Station, TX 77843-3368

Email address: zhuyuyu@math.tamu.edu

TAMU 3368, College Station, TX 77843-3368