PAWS 2025: ANALYSIS AND IMPLEMENTATION OF ALGORITHMS IN NUMBER THEORY
PROBLEM SET 2

THOMAS BOUCHET, KATE FINNERTY, ASIMINA S. HAMAKIOTES, YONGYUAN HUANG

The goal for Problem Set 2 is to become comfortable using the key definitions in Lecture 2 and connect them to the complexity concepts from the first problem set and lecture. The questions are loosely in ascending order of difficulty. Feel free to skip around and try whatever exercises would be the most helpful for you. Try as many as you can but don't feel like you need to complete them all!

1. Beginner Problems

Question 1: Recall that an algebraic extension can be written as a quotient $\mathbb{Q}[x]/(f)$, where $f(x)$ is irreducible. Also, recall that the ring of integers of an extension is the ring of all algebraic integers (roots of monic polynomials whose coefficients are in $\mathbb{Z}$) contained in said extension.

(a) Show that the ring of integers for $\mathbb{Q}(\sqrt{5})$ is $\mathbb{Z}[\frac{1+\sqrt{5}}{2}]$.
(b) For d a squarefree integer, describe a generator of the ring of integers for $\mathbb{Q}(\sqrt{d})$.

Question 2: Show that $\mathrm{Res}_y(y^m m_\alpha(x/y), m_\beta(y))$ has $\alpha\beta$ as a root, so factoring this polynomial will result in finding the minimal polynomial of $\alpha\beta$. Similarly, show that you can recover the minimal polynomial of α/β from $\mathrm{Res}_y(m_\alpha(xy), m_\beta(y))$.

Question 3: Consider the quadratic number field $K = \mathbb{Q}(\sqrt{-7})$. Note that $\sqrt{-7}+1$ is an element of K. Can you find its minimal polynomial? How is it related to the minimal polynomial of $\sqrt{-7}$? Can you now find an algorithm to compute the minimal polynomial of any element $a + b\sqrt{-7} \in K$? Can you generalize this to any quadratic number field?

Question 4: Consider a large monic polynomial over $\mathbb{Z}[x]$ that one wants to factor. One way you could create it in Magma is with:

```
R<t,y> := PolynomialRing(Integers(),2);
S<x> := PolynomialRing(Integers());
degree := 4;
f := x^degree + Evaluate(Random(degree-1,R,100),[x,1]);
```

This polynomial is almost surely irreducible. We can approximate one of the roots by:

```
r := Roots(PolynomialRing(ComplexField())!f)[1][1];
```

Find the minimal polynomial of the root by checking what algebraic (integer) relations $1, r, r^2, r^3, r^4, r^5$ hold. If you find an algebraic relation of degree smaller than 5, the polynomial is reducible. Otherwise, does it show that it is irreducible? We will see that this can be done with LLL in Lecture 4.

Question 5: Let $f(x) = x^3 - x - 2$.

(a) Find all roots of $f(x)$ mod 2 in $\mathbb{Z}/2\mathbb{Z}$.
(b) Recall Hensel's Lemma for integers (Lecture 2 Lemma 2.6). Which of the roots of $f(x)$ in $\mathbb{Z}/2\mathbb{Z}$ lift to a root in $\mathbb{Z}/2^k\mathbb{Z}$ for every $k \geq 1$?
(c) For each root of $f(x)$ in $\mathbb{Z}/2\mathbb{Z}$ that lifts to a root in in $\mathbb{Z}/2^k\mathbb{Z}$ for every $k \geq 1$, find its approximation modulo 2^5.

2. Intermediate Problems

Question 6: Prove that a number field is an algebraic extension of $\mathbb{Q}$.

Question 7: Let $f(x) \in \mathbb{Q}[x]$ be an irreducible polynomial and α be a root of f. Recall $\mathbb{Q}(\alpha)$ denotes the smallest field that contains both $\mathbb{Q}$ and α. Explain why $\mathbb{Q}[\alpha] = \{p(\alpha) : p \in \mathbb{Q}[x]\} = \mathbb{Q}(\alpha)$. Note that it is enough to write the inverse of α as a polynomial (over $\mathbb{Q}$) in α. What is the degree of $\mathbb{Q}(\alpha)$ in terms of the degree of $f(x)$?

Question 8: Consider the biquadratic number field $\mathbb{Q}(\sqrt{a}, \sqrt{b})$. Follow the proof of the Primitive Element Theorem to find a primitive element θ such that $\mathbb{Q}(\sqrt{a}, \sqrt{b}) = \mathbb{Q}(\theta)$. Can you find a way to compute the minimal polynomial of θ? Can you write $\sqrt{a}$ and $\sqrt{b}$ as polynomials in θ? If you want, you can pick specific values for a and b.

Question 9: Recall the following discussion of multiplication in number fields: To make multiplication more efficient, we can precompute and store those reductions. Suppose $\theta^n = -t_{n-1}\theta^{n-1} - \cdots - t_0$. Then for $k \geq n$, we have

$$\theta^{n+k} = \sum_{i=0}^{n-1} r_{i,k}\theta^i, \tag{1}$$

where $r_{i,0} = -t_i$ and

$$r_{k+1,i} = \begin{cases} r_{k,i-1} - t_i r_{k,n-1} & \text{if } i \geq 1, \\ -t_0 r_{k,n-1} & \text{if } i = 0. \end{cases}$$

Show that pre-computing the constants $r_{i,k}$ as in (1) takes $O(kn)$ field operations.

3. Advanced Problems

Proposition 1 (Prop. 2.2 from the notes)**.** *Let $\overline{f}(x) \in \mathbb{F}_p[x]$ be squarefree and assume that its decomposition into irreducibles is $\overline{f}(x) = \prod_{1 \leq i \leq r} f_i(x)$. The polynomials $T(x) \in \mathbb{F}_p[x]$ with $\deg(T(x)) < \deg(\overline{f}(x))$ for which for each i with $q \leq i \leq r$ there exists $s_i \in \mathbb{F}_p$ with $T(x) \equiv s_i \pmod{f_i(x)}$, are exactly the p^r polynomials $T(x)$ such that $\deg(T(x)) < \deg(\overline{f}(x))$ and $T(x)^p \equiv T(x) \pmod{\overline{f}(x)}$.*

Question 10: Show that the Berlekamp algorithm for small p terminates and correctly computes the factorization of $\overline{f}$ into irreducibles. You can follow the following steps.

(1) As a warm-up, let $\overline{f}(x) \in \mathbb{F}_p(x)$ be a polynomial of degree n. Show that $\overline{f}(x)$ is irreducible if and only if
   (i) $x^{p^n} \equiv x \pmod{\overline{f}(x)}$; and

(ii) for each prime $q|n$, $\gcd(x^{p^{n/q}} - x, \overline{f}(x)) = 1$.

(2) Prove Proposition 1.
(3) Using the notation of Step 2 of the algorithm, show that any polynomial $T(x)$ in the kernel of $Q - I$ holds that $T(x)^p \equiv T(x) \pmod{\overline{f}(x)}$.
(4) Explain why the dimension of $\ker(Q - I)$ is exactly r and why the column vector $(1, 0, \ldots, 0)^t$ belongs to the kernel.
(5) Let $T(x)$ be a polynomial corresponding to a V_j. Explain why the polynomials F from Step 4 of the algorithm correspond to irreducible factors once we have $k = r$.

Question 11: You can generate random integer polynomials of degree d with coefficients in $[-b, b]$ in Magma by running the script

```
R<t,y> := PolynomialRing(Integers(),2);
S<x> := PolynomialRing(Integers());
f := Evaluate(Random(d,R,b),[x,1]);
```

Create a polynomial that has 10 random divisors of degree 3. Run the steps of the Berlekamp algorithm with at least two primes and compare results.