Homework 1

Algorithmic Number Theory (Fall 2014)
Rutgers University
Swastik Kopparty

Due Date: November 19, 2014

Answer any 2 questions.

Questions

1. Recall the Gauss_η algorithm for finding short vectors in a 2 dimensional lattice L (here $\eta \in (0, 1]$). We assume $L \subseteq \mathbb{Z}^n$, and that L is specified by giving as input a basis $\{a, b\}$, where all coordinates of a, b are at most n bits long.

   (a) Assume $\|a\| \leq \|b\|$

   (b) Repeat the following:

      i. Define $m = \lceil \frac{\langle a,b \rangle}{\|a\|^2} \rfloor$.

      ii. Set $b = b - m \cdot a$.

      iii. Swap(a, b)

      Until $\|a\| \geq \eta \cdot \|b\|$.

   (c) Output b.

   We already saw that Gauss_1 finds THE shortest vector in L, and that $\mathsf{Gauss}_{0.9}$ halts in $\mathsf{poly}(n)$ steps (but only finds an approximately shortest vector of L).

   Show that for every L, Gauss_1 halts at most one step after $\mathsf{Gauss}_{0.9}$ halts. Thus Gauss_1 is also an efficient algorithm.

2. We will now see an algorithm that approximately solves the closest vector problem (CVP). Let $L \subseteq \mathbb{R}^n$ be a given lattice (w.l.o.g. it is full-rank) , and let $y \in \mathbb{R}^n$. Our goal is to find $x \in L$ such that $\|x - y\|$ is almost as small as possible.

   The algorithm is as follows:

   - Let $b_1, \ldots, b_n$ be an LLL-reduced basis for L. (In particular, in the Gram-Schmidt orthonormal coordinate system $(u_1, \ldots, u_n) \in (R^n)^n$, the column vectors $b_1, \ldots, b_n$ form an upper triangular matrix. Let $d_1, \ldots, d_n$ be the diagonal entries of this upper triangular matrix.)
   - Find $x \in L$ such that for each $i \in [n]$:

   $$|\langle x, u_i \rangle - \langle y, u_i \rangle| \leq \frac{d_i}{2}.$$

   - Output this x.

Show how to implement the second step of the algorithm efficiently.

Show that the x output by the algorithm satisfies:

$$\|x - y\| \leq 2^{O(n)} \cdot \min_{z \in L} \|z - y\|.$$

3. Let L be a given lattice in $\mathbb{Z}^d$, specified by a basis, where each basis vector has each coordinate at most n bits long. For a given $p \in [1, \infty) \cup \{\infty\}$, we want to find the $x \in L \setminus \{0\}$ that minimizes $\|x\|_p$. Give a $\mathsf{poly}(2^{\mathsf{poly}(d)}, n)$ time algorithm to do this.

   Assuming the result of the previous problem, also show how to solve the CVP problem exactly under these norms.

   Use your method to give a polynomial time algorithm for the following problem. Given $N \in \mathbb{Z}$ and $x \in \mathbb{Q}$ as input, find $a, b, c \in \mathbb{Z}$ with $|a|, |b|, |c| \leq N$ such that

   $$|x - (a + b\sqrt[3]{2} + c(\sqrt[3]{2})^2)|$$

   is as small as possible.

   (The running time should be polynomial in $\log N$ and the number of bits in the numerator and denominator of x).

4. **Problem Updated on Nov. 8. Unfortunately the previous problem was too simple to be correct.** We will now see the p-adic analogue of our algorithm to factor a given polynomial $F(X) \in \mathbb{Q}[X]$. The overall strategy is the same: find an $\tilde{\alpha}$ which is sufficiently "close" to a root α of $F(X)$, and then find a small-coefficient polynomial of low degree which almost-vanishes at $\tilde{\alpha}$. What changes is the notion of closeness; we will use the p-adic metric instead of the standard absolute difference metric.

   (a) Let $F(X) \in \mathbb{Z}[X]$ be a polynomial of degree n, where each coefficient is an n-bit integer. Show that there is a prime $p \leq 2^{\mathsf{poly}(n)}$, and an integer $a < p$, such that:

   - $F(a) \equiv 0 \mod p$,
   - $F'(a) \not\equiv 0 \mod p$,
   - The leading coefficient of $F(X)$ is not divisible by p.

   (I don't know how one can find such p and a efficiently. Under the GRH, using the "effective Chebotarev density theorem", one can show that with probability $\geq \frac{1}{\mathsf{poly}(n)}$, a large random integer $p \approx 2^{\mathsf{poly}(n)}$ will be a prime be such that $F(X)$ has at least one root a in $\mathbb{F}_p$. This a can then be found by Berlekamp's algorithm.)

   Assume for the next few parts of this problem that such p and a have been given to you. In the last (optional) part of this problem, you will see how one can do away with this assumption.

   (b) Let a, p be as above. Give a $\mathsf{poly}(n, k)$ time algorithm to find an integer $a_k \in [0, p^k)$ such that $a_k \equiv a \mod p$, with $F(a_k) \equiv 0 \mod p^k$.

   (c) Let $k = \mathsf{poly}(n)$.

   If F is reducible, show that there exists nonzero $G(X) \in \mathbb{Z}[X]$ such that:

   - $G(a_k) \equiv 0 \mod p^k$.
   - each coefficient of $G(X)$ is at most 2^{n^2} in absolute value.

Conversely, if such a nonzero $G(X) \in \mathbb{Z}[X]$ exists, show that $G(X)$ and $F(X)$ have a nontrivial GCD in $\mathbb{Q}[X]$.

(d) Show that we can efficiently determine if such a $G(X)$ as above exists, and if it does exist, we can find it efficiently. Use this to complete the description of the efficient factoring algorithm for polynomials in $\mathbb{Q}[X]$ (assuming that we are given a, p as help).

(e) **Optional:** One way to use the above ideas to get a self-contained efficient algorithm for factoring polynomials over $\mathbb{Q}$ is as follows. You should think about what it takes to implement this algorithm efficiently.

We first find a prime $p \leq n^{10}$, an integer $k \leq n$ and an element $\alpha \in \mathbb{F}_{p^k}$ such that, if $\bar{F}(X)$ is the reduction of $F(X)$ mod p, then we have:

- $\bar{F}(\alpha) = 0$,
- $\bar{F}'(\alpha) \neq 0$,
- $\deg(F(X)) = \deg(\bar{F}(X))$.

Show that this can be done efficiently.

We then try to use α to find a root of $F(X)$ in some field $L \supset \mathbb{Q}$. This field L will play the role of the complex numbers in the factoring algorithm from class. L will also be similar to $\mathbb{C}$ in the sense that L will be a finite algebraic extension of the completion of $\mathbb{Q}$ according to some metric (just like $\mathbb{C}$ is an extension of $\mathbb{R}$, which in turn is the completion of $\mathbb{Q}$ according to the usual metric).

Let $\bar{h}(T) \in \mathbb{F}_p[T]$ be a monic irreducible polynomial of degree k (so that $\mathbb{F}_{p^k} = \mathbb{F}_p[T]/\bar{h}(T)$). Let $h(T) \in \mathbb{Z}[T]$ be a monic irreducible polynomial of degree k such that $h(T) \mod p$ equals $\bar{h}(T)$.

Let $\mathbb{Q}_p$ be the field of p-adic numbers, $\mathbb{Z}_p$ be the ring of p-adic integers. Let L be the extension of $\mathbb{Q}_p$ given by $\mathbb{Q}_p[T]/h(T)$. Let R be the integral closure of $\mathbb{Z}_p$ in L. Let $\mathfrak{p}$ be the unique prime ideal of R. Let $\pi \in \mathfrak{p} \setminus \mathfrak{p}^2$. We have $\mathfrak{p} = \pi \cdot R$.

Note that $R/\mathfrak{p} = \mathbb{F}_{p^k}$. Let $a \in R$ be such that $a \mod \mathfrak{p} = \alpha$. Then we have:

- $F(a) \equiv 0 \mod \langle \pi \rangle$.
- $F'(a) \not\equiv 0 \mod \langle \pi \rangle$.
- The leading coefficient of $F(X) \in R[X]$ is not divisible by π.

We can then use Hensel lifting to find, for each k, $a_k \in R$ s.t. $F(a_k) \equiv 0 \mod \langle \pi^k \rangle$.

Having found a_k for large enough $k = \mathsf{poly}(n)$, we then search for a polynomial $G(X) \in \mathbb{Z}[X]$ s.t. $G(a_k) \equiv 0 \mod \langle \pi^k \rangle$, each coefficient of G is at most 2^{n^2}, and $\deg(G) < \deg(F)$. If such a G exists, it is a factor of $F(X)$.

5. Suppose p is a given prime, g is a given generator of $\mathbb{F}_p^*$, and you have access to an algorithm $A_{p,g}(x)$. $A_{p,g}$ has the property that for at least 0.01 fraction of the $x \in \mathbb{F}_p^*$, we have:

$$g^{A_{p,g}(x)} = x \mod p,$$

(i.e., for 0.01 fraction of the $x \in \mathbb{F}_p^*$, $A_{p,g}(x)$ is the discrete log of x to the base g).

Give a $\mathsf{poly}(\log p)$ time randomized algorithm (which can invoke $A_{p,g}$ as a subroutine) which computes the discrete log of a given $x \in \mathbb{F}_p^*$ for *every* $x \in \mathbb{F}_p^*$.

6. Let $\mu(n)$ be the Mobius function (i.e., $\mu(n) = (-1)^{\#\text{ primes dividing } n}$ if n is squarefree, $\mu(n) = 0$ otherwise).

A conjecture of Sarnak says that for every polynomial time computable function $f : \mathbb{N} \to [-1, 1]$,

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(n)\mu(n) \right| = o(1).$$

(In words: no polynomial time computable function can correlate with μ; this would express a strong form of pseudorandomness of μ). Note that if factoring can be done in $\mathsf{polylog}(n)$ time, then this conjecture is false.

Put in your best effort and find a $\mathsf{polylog}(n)$ time computable $f(n)$ so that $\left| \frac{1}{N} \sum_{n=1}^{N} f(n)\mu(n) \right|$ is as large as possible (as a function of N). You can use any fact you want about primes. You may also want to look up "smooth numbers".

In the other direction, if we insist that f satisfies: $\left| \frac{1}{N} \sum_{n=1}^{N} f(n)\mu(n) \right| = \Omega(1)$, we can try to come up with such an f which can be computed as fast as possible. How low can you make the running time of f. You should able to make it n^{ϵ} for every $\epsilon > 0$, and even faster as the course progresses.