Discrete Math, Fourteenth Problem Set (July 18)

REU 2003

Instructor: László Babai
Scribe: Ivona Bezakova

0.1 Repeated Squaring

For the primality test we need to compute $a^{X-1} \pmod{X}$. There are two problems with the straightforward method (multiplying $X-1$ copies of a and reducing modulo X). First, the resulting number a^{X-1} has exponentially many digits (nearly 2^n binary digits if X has n binary digits and $a=2$). The space requirement can be reduced to linear in the bit-length of the input by doing all operations modulo X. Second, an exponential number of modular multiplications is needed, namely $X-2 \approx 2^n$. We can reduce this number to $\approx n$ by "repeated squaring" – see the "Pseudocodes for basic algorithms in Number Theory" handout.

0.2 Factoring of integers

Given a number X we want to find its prime factors. The "trial division" method tests all integers up to $\sqrt{X}$, requiring $\approx 2^{n/2}$ trials where n is the bit-length of X. Several factoring algorithms are known which run in time $2^{c\sqrt{n\log n}}$. While this is still very far from polynomial time, it is significantly better than trial division. We shall see the idea.

Remark 0.1. The best factoring algorithm known, called the *number field sieve*, runs in time $2^{c\sqrt[3]{n\log^2 n}}$ [Lenstra, Lenstra, Manasse and Pollard (1990)]. This statement is based on some plausible but unproven number theoretic assumptions ("heuristic analysis"). Given the significance of factoring integers, efficient factoring algorithms are of great interest even if we cannot rigorously prove their efficiency. Dixon's remains the only factoring algorithm with a fully proven $2^{c\sqrt{n\log n}}$ upper bound on its running time.

0.3 Types of randomized algorithms

A *Las Vegas* algorithm is an algorithm that runs in polynomial time and produces an answer with probability $\geq 1/2$; alternatively, it may say "don't know." Whenever an answer is produced, it is guaranteed to be correct. By repeating the algorithm k times using independent coin flips, the probability of not getting an answer is reduced to $1/2^k$.

A *Monte Carlo* algorithm with one-sided error runs in polynomial time and it produces a correct answer with probability $\geq 1/2$. The primality tests discussed in the previous class are examples of Monte Carlo algorithms with one-sided error, i. e., the answer "X is composite" is always correct but "X is prime" may be an incorrect answer; the probability that a composite X is misclassified as prime is less than $1/2^k$ (after k repetitions).

0.4 Representing a prime as a sum of two squares

We proved in this class that if p is aq prime and $p \equiv 1 \pmod 4$ then there exist integers a, b such that $p = a^2 + b^2$. The goal is to find such a and b in polynomial time (i. e., in time $(\log p)^{O(1)}$).

Rabin and Shallit showed that we can find a, b in randomized polynomial time (a Las Vegas algorithm). As a subroutine, they need to factor polynomials in $\mathbb{F}_p$. This can be done in randomized polynomial time using a Las Vegas algorithm by Berlekamp. (Polynomial-time factoring of polynomials over $\mathbb{Q}$ is a consequence of the lattice basis reduction algorithm. This algorithm does not work for factoring over $\mathbb{F}_p$ and no deterministic polynomial-time algorithm is known for this case.)

How can we guarantee that a factoring is correct? By multiplying the factors we can verify that the product equals the input polynomial. But we also need to test that the factors are irreducible. Indeed, we have a deterministic test for irreducibility of polynomials over $\mathbb{F}_p$.

0.5 Deterministic test of irreducibility of polynomials over $\mathbb{F}_p$

Let $f(x) \in \mathbb{F}_p[x]$ be a polynomial over $\mathbb{F}_p$. We say that f is *irreducible* if there do not exist polynomials g, h of lower degree than f such that $f \equiv gh \pmod p$.

Let $\mathcal{F} \supseteq \mathbb{F}_p$ be the algebraic closure of $\mathbb{F}_p$. Let $\alpha \in \mathcal{F}$ be a root of f, i.e. $f(\alpha) = 0$. If f is an irreducible polynomial of degree k then $\mathbb{F}_p[\alpha] = \mathbb{F}_{p^k}$.

"Fermat's Little Theorem for $\mathbb{F}_{p^k}$" states that for all $\alpha \in \mathbb{F}_{p^k}$, if $\alpha \neq 0$ then $\alpha^{p^k-1} = 1$ (equality in the field $\mathbb{F}_{p^k}$). Therefore $x - \alpha$ divides g.c.d. $(f, x^{p^k-1} - 1)$ and since f is irreducible, this implies that f divides $x^{p^k-1} - 1$. Moreover, f does not divide $x^{p^\ell-1} - 1$ for any $\ell < k$.

Exercise 0.2. Let $f \in F_p[x]$ be a polynomial of degree k. Prove that f is irreducible if and only if both of the following conditions hold:

1. $f \mid x^{p^k-1} - 1$;
2. $(\forall \ell < k)(f \nmid x^{p^\ell-1} - 1)$.

Remark 0.3. Above we sketched the "only if" part.

Exercise 0.4. Given a polynomial $f \in F_p[x]$ and $t \leq 2^n$, compute g.c.d. $(f, x^t - 1)$ in time polynomial in n, $\log p$, and the degree of f. *Hint.* Compute repeated squares $x, x^2, x^4, \ldots \pmod{f}$.

Combining the preceding two exercises we obtain a polynomial time algorithm for testing irreducibility of polynomials over $\mathbb{F}_p$.

0.6 Las Vegas factoring of quadratic polynomials over $\mathbb{F}_p$

Let $f \in \mathbb{F}_p[x]$ be a quadratic polynomial. We want to factor f over $\mathbb{F}_p$. First we test whether f is irreducible; if so, we are done. Otherwise we know that f can be factored as $f(x) = (x-a)(x-b)$ where $a, b \in \mathbb{F}_p$. We want to find a and b.

Definition 0.5. We say that a polynomial g *splits* polynomial f if $1 \neq$ g.c.d. $(f, g) \neq f$.

It suffices to find a polynomial that splits f.

If $a = b$ then f' splits f. So if f' does not split f then we know that $a \neq b$. We may also assume that $ab \neq 0$ (why?)

Exercise 0.6. Prove: the roots of the polynomial $x^{(p-1)/2} - 1$ are exactly the quadratic residues mod p. (Note: we used this fact in proving that $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$; and this fact in turn was one of the chief ingredients of our proof that primes $\equiv 1 \pmod 4$ can be written as the sum of two squares.)

We start with an easy case. Suppose $\left(\frac{a}{p}\right) \neq \left(\frac{b}{p}\right)$. In this case, according to the preceding exercise, $x^{(p-1)/2} - 1$ splits f.

If now $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$, our next trick is to consider the polynomials $g_r(x) = f(x+r)$. Ideally, we want to find r such that $\left(\frac{a-r}{p}\right) \neq \left(\frac{b-r}{p}\right)$. Then $x^{(p-1)/2} - 1$ splits g_r and therefore we can factor f. The next exercise shows that we have an excellent chance of finding r by just picking it at random.

Exercise 0.7. Prove: For a random $r \in \mathbb{F}_p$ the probability $P\left(\left(\frac{a-r}{p}\right) \neq \left(\frac{b-r}{p}\right)\right) \approx 1/2$.

Exercise$^+$ 0.8. Generalize the algorithm to factoring polynomials of arbitrary degree over $\mathbb{F}_p$ into their irreducible factors.

0.7 Back to the $p = a^2 + b^2$ problem

In this section we describe the Rabin–Shallit polynomial-time Las Vegas algorithm to represent a given prime of the form $4k + 1$ as the sum of two squares.

We know that -1 is a quadratic residue in $\mathbb{F}_p$ for $p \equiv 1 \pmod 4$, i.e. there exists t such that $t^2 \equiv -1 \pmod p$. Our first task is to find such t? This is equivalent to factoring the polynomial $x^2 + 1 = (x + t)(x - t) = x^2 - t^2 \in F_p[x]$.

Definition 0.9. The domain of **Gaussian integers** is defined as $\mathbb{Z}[i] = \{a + bi \ : \ a, b \in \mathbb{Z}\}$.

Divisibility, primes, g.c.d. are defined among Gaussian integers the same way as among integers. (Recall: by definition, the greatest common divisor is not "largest" among the common divisors but one that is a common multiple of all common divisors. So its existence is not evident.)

There are four units (numbers dividing 1) among Gaussian integers, $1, -1, i$, and $-i$. Let us take a look at Gaussian primes. 2 and 5 are not primes in $\mathbb{Z}[i]$ (because $5 = (2 + i)(2 - i)$) but 3 remains a prime.

Exercise 0.10. For $z = a + bi$, let $N(z) = a^2 + b^2$ (the "norm" of z). Note that $N(z) = |z|^2$. Prove: $N(zw) = N(z)N(w)$.

Exercise 0.11. Prove: if $z, w \in \mathbb{Z}[i]$ and $w \mid z$ then $N(w) \mid N(z)$.

Exercise 0.12. Prove that if p is a prime number (in $\mathbb{Z}$) and $p \equiv -1 \pmod 4$ then p is a prime in $\mathbb{Z}[x]$.

Exercise 0.13. If p is a prime number (in $\mathbb{Z}$) and $p \equiv 1 \pmod 4$ then $p = a^2 + b^2 = (a + bi)(a - bi)$. Prove: $a + bi$ is prime in $\mathbb{Z}[i]$ (as is $a - bi$). *Hint.* Suppose $c + di \mid a + bi$. Compare their norms.

Exercise 0.14. Prove that Euclid's algorithm works in $\mathbb{Z}[i]$. (The remainder must have smaller norm than the divisor.) It follows that for $z, w \in \mathbb{Z}[i]$, g.c.d.(z, w) exists and is a linear combination of z and w with coefficients from $\mathbb{Z}[i]$.

Corollary 0.15. *Every Gaussian integer has unique prime factorization in $\mathbb{Z}[i]$ (unique apart from order and units). (Prove!)*

The algorithm

Let $p \equiv 1 \pmod 4$. By factoring the polynomial $x^2 + 1$ over $\mathbb{F}_p$, we found, in Las Vegas polynomial time, and integer t such that $p \mid t^2 + 1$. Now $t^2 + 1 = (t + i)(t - i)$. Since $p = a^2 + b^2$ (a, b unknown) and $a + bi$ is a prime in $\mathbb{Z}[i]$, it follows that $a + bi$ divides either $t + i$ or $t - i$ (but not both – why?). Therefore $t + i$ splits p: if $a + bi \mid t + i$ then $a + bi = \text{g.c.d.}\,(t + i, p)$. Otherwise, $a - bi = \text{g.c.d.}\,(t + i, p)$. In either case, we find a, b using Euclid's algorithm in $\mathbb{Z}[i]$.

Exercise$^+$ 0.16. Give an alternative proof of the theorem that primes $\equiv 1 \pmod 4$ can be written as the sum of two squares. Use Gaussian integers.

0.8 Factoring integers in time, exponential in $\sqrt{n \log n}$

J. Dixon was the first to show that n-digit integers can be factored in $e^{c\sqrt{n \log n}}$ steps. Let X be an n-digit integer ($X \approx 2^n$) which we want to factor.

Definition 0.17. We say that z is a v-*smooth* integer if all primes dividing z are $\leq v$.

Suppose we can find a, b such that $a^2 \equiv b^2 \pmod{X}$ and $a \not\equiv \pm b \pmod{X}$. Then X divides $(a+b)(a-b)$ and $a+b$ splits X.

Exercise 0.18. If X is odd and not a prime power then there exist a, b such that $a^2 \equiv b^2 \pmod{X}$ and $a \not\equiv \pm b \pmod{X}$.

How do we find such a and b? Let v be a threshold number $v = 2^{\sqrt{n \log n}}$ and let $P = \{p \text{ is a prime} : \ p \leq v\}$. Let $r_i \in \{1, \ldots, X\}$ be a random number. Let $s_i = (r_i^2 \bmod X)$. If s_i is not v-smooth, discard it and try again. Let $N = |P|$. Keep generating the r_i until $N+1$ smooth numbers s_i are found.

Suppose $s_i = p_1^{\alpha_{i_1}} \ldots p_N^{\alpha_{i_N}} \pmod{X}$ where $P = \{p_1, \ldots, p_N\}$.

Consider the matrix

$$\begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,N} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N+1,1} & \alpha_{N+1,2} & \cdots & \alpha_{N+1,N} \end{pmatrix}$$

Since this matrix is $(N+1) \times N$, its rows are linearly dependent $\pmod 2$. Therefore there exists a nonempty set $S \subseteq \{1, \ldots, N\}$ such that $(\forall j)(\sum_{i \in S} \alpha_{ij} = 2\beta_j)$. Thus

$$\prod_{i \in S} r_i^2 \equiv \prod_{j=1}^{N} p_j^{2\beta_j} \pmod{X}$$

Let $a := \prod_{i \in S} r_i \bmod X$ and $b := \prod_{j=1}^{N} p_j^{\beta_j} \bmod X$. Now $a^2 \equiv b^2 \pmod{X}$.

There are two catches in this sketch. First, the smooth numbers have to happen sufficiently frequently. Second, the probability of $a \not\equiv \pm b \pmod{X}$ must be significant.

The second claim seems more plausible and can be proven by using a clever counting argument. Here we present intuition behind the first claim.

Let $\psi(X, v)$ be the number of v-smooth numbers less than X. Let $k = \sqrt{n / \log n}$, i.e. $v^k \approx X$. Clearly, $\psi(v^k, v) \geq \binom{\pi(v)}{k} > (\pi(v)/k)^k$ (the product of any k primes $\leq v$ is smooth). What is the probability that a random number is smooth?

$$\frac{\psi(v^k, v)}{v^k} \geq \left(\frac{\pi(v)}{k}\right)^k \frac{1}{v^k} \sim \left(\frac{c}{\sqrt{n \log n}} \frac{1}{\sqrt{n / \log n}}\right)^{\sqrt{n / \log n}} = (c/n)^{\sqrt{n / \log n}} \geq c_1^{\sqrt{n \log n}}.$$

Thus, the expected number of trials to find a random smooth number is less than $(1/c_1)^{\sqrt{n \log n}}$. Note, however, that the s_i are not generated uniformly at random, so the actual proof is necessarily more delicate. But it uses the same simple lower bound on $\psi(v^k, v)$.