Lecture 11 - Basic Number Theory.

Boaz Barak

October 20, 2005

Divisibility and primes Unless mentioned otherwise throughout this lecture all numbers are non-negative integers. We say that a divides b, denoted $a|b$ if there's a k such that $ka = b$. We say that p is *prime* if for $a > 0$, $a|p$ only for $a = 1$ and $a = p$

Unique factorization.

Theorem 1 (Unique factorization)**.** *For every $n > 0$, there are unique primes $p_1, \ldots, p_k$ such as n is the multiplication of these primes.*

We typically order the primes from small to big, and group together multiplications of the same prime, and so the unique factorization of n is its representation of the form $p_1^{i_1} \cdot p_2^{i_2} \cdots p_\ell^{i_\ell}$.

Basic property of prime and co-prime numbers. Two easy consequences of the unique factorization theorem:

- If p and q are co-prime and both $p|n$ and $q|n$, then $pq|n$.
- If $p|ab$ then either $p|a$ or $p|b$.

How many primes exist. Another nice fact to know about primes is that there are infinitely many of them. (It is not immediately obvious from the unique factorization theorem — initially you might think that perhaps the only primes are $\{2, 3, 5\}$ and all other numbers are of the form $2^i 3^j 5^k$)

In fact, we have the following theorem:

Theorem 2 (Chebychev's theorem)**.** *Let $p(n)$ denote the number of primes between 1 and n. Then, $p(n) = \Omega(\frac{n}{\log n})$.*

This means in particular that if you choose a random ℓ-bit integer, with probability $\Omega(\frac{1}{\ell})$ it will be prime. Chebychev's theorem actually has a very short and simple proof (see Shoup's book). It is known actually that $p(n) = \frac{n}{\ln n}(1 + o(1))$. This is called the prime number theorem (there's also an OK proof for this).

g.c.d For two integers a and b, their g.c.d is the largest d such that $d|a$ and $d|b$. The g.c.d can be shown to be the largest common part of their factorization. That is, if p, q, r are primes and $a = pq^2$ and $b = q^3r$ then $gcd(a, b) = q^2$. If a and b factor into disjoint sets of primes then $gcd(a, b) = 1$. In particular for every two different primes p, q $gcd(p, q) = 1$. If $gcd(a, b) = 1$ we say that a and b are *co-prime* to one another.

Modulu For every two numbers a and b there is unique k and r such that $0 \leq r \leq b - 1$ and $a = kb + r$. In this case we say that $r = a \pmod b$. Clearly $b|a$ iff $a \pmod b = 0$. Also note that for all a, b, c

$$a + b \pmod c = (a \pmod c + b \pmod c) \pmod c$$

and

$$a \cdot b \pmod c = (a \pmod c \cdot b \pmod c) \pmod c$$

If $a \pmod b = a' \pmod b$ we say that a and a' are *equivalent* modulu b, sometimes denoting this by $a \equiv_b a'$.

We denote by $\mathbb{Z}_b$ the set $\{0, \ldots, b - 1\}$. When we add or multiply two elements from $\mathbb{Z}_b$ we use addition/multiplication modulu b.

Chinese reminder theorem. Let p and q be two prime numbers (actually can be also just co-prime) and let $n = pq$. Consider the following function from $\mathbb{Z}_n$ to $\mathbb{Z}_p \times \mathbb{Z}_q$: $f(x) = \langle x \pmod p, x \pmod q \rangle$. We claim the following properties of this function:

1. $f(\cdot)$ preserves addition: $f(x + x') = f(x) + f(x')$. (In the right hand side $f(x) + f(x')$ means that we add the first element of both pairs mod p and the second element mod q. This follows from the fact that the modulu operation has this property.
2. $f(\cdot)$ preserves multiplication: $f(x \cdot x') = f(x) \cdot f(x')$. Again, this follows from the fact that the modulu operation has this property.
3. $f(\cdot)$ is one-to-one. Indeed, if there exist $x > x'$ with $f(x) = f(x')$ then $f(x - x') = \langle 0, 0 \rangle$. Which means that $p|x - x'$ and $q|x - x'$ which implies $pq = n|x - x'$ which can't happen for a number between 1 and $n - 1$.
4. $f(\cdot)$ is onto. This follows from the fact that $|Z_n| = |Z_p| \cdot |Z_q|$.

Operations we can do efficiently We can do the following operations efficiently (polynomial in the number of bits it takes to describe the inputs)

1. Addition and multiplication modulu some n
2. Exponentiation modulu n. We can not compute $x^y \pmod{n}$ by repeated multiplications since that can take y operation which is too many. Rather we separate y to a sum of powers of two (binary notation): $y = 2^i + 2^j + 2^k$ thus we need to compute $x^{2^i} cdot x^{2^j} \cdot x^{2^k}$. We can compute x^{2^i} in i multiplications by repeated squaring.
3. Taking inverse modulu n. If $gcd(x, n) = 1$ then the extended gcd algorithm gives a y such that $xy \pmod{n} = 1$. We sometimes denote $y = x^{-1}$.

Non-trivial efficient operations. We'll show we can do the following two things efficiently:

1. Take a square root modulu a prime. That is, for a prime p and $a \in Z_p$, find b such that $a = b^2 \pmod{p}$ if such a b exists.
2. Primality testing: given a number n decide whether it is a prime or a composite number.

Fermat's little theorem We'll use the following theorem of Fermat: for every prime p and number $1 \leq a \leq p - 1$. $a^{p-1} = 1 \pmod{p}$. We note that this is actually a consequence of a more general theorem on groups.

Facts about square roots. When we work in $\mathbb{Z}_p$, we denote by $-x$ the number such that $x - x = 0 \pmod{p}$. In other words, $-x = p - x$. Note that it's always the case that $x \neq -x$ since otherwise we'd have $2x = p$ which means that p is even. We know that over the reals any number a has either zero square roots (if its negative) or two square roots $+\sqrt{a}$ and $-\sqrt{a}$ if its positive. It turns out a similar thing holds for $\mathbb{Z}_p$: every $a \in \mathbb{Z}_p$ has either no square roots, or two square roots of the form x and $-x$.

To prove this first note that if $x^2 = a \pmod{p}$ then $(-x)^2 = a \pmod{p}$. Thus, if a has any square roots it has at least two of them. Now we'll prove that if x and y are square roots of the same value then $x = \pm y$. Indeed, if $x^2 = y^2 \pmod{p}$ this means that $x^2 - y^2 = 0 \pmod{p}$ or that $p|(x+y)(x-y)$. Since p is prime this means that either $p|x+y$ (meaning $x = -y \pmod{p}$) or $p|x - y$ (meaning $x = y \pmod{p}$).

Taking square root modulu prime: We're given a prime p and a number a which has a square root x, and we want to find x (or $-x$). We can assume p is odd (if p is the only even prime, namely two, then we can easily solve this problem mod p). $p \pmod{4}$ can be either 1 or 3. We start with the case that $p \pmod 4 = 3$. That is, $p = 4t+3$. In this case we claim that a^{t+1} is a square root of a.

Indeed, write $a = x^2$. Then $(a^{t+1})^2 = x^{4(t+1)} = x^{4t+4} = x^{p-1+2} = x^{p-1}x^2 = 1 \cdot a$.

See `http://www.wisdom.weizmann.ac.il/~oded/PS/RND/l11.ps` for the algorithm in the case $p = 1 \pmod 4$. (We note that in that case we use a probabilistic algorithm).

Square roots modulu composites We note the following property about square roots modulu composites: if an odd number n is a product of (powers of) at least 2 distinct primes, then every number a that has square root mod n, has at least 4 square roots. Indeed, if n is of this form then $n = pq$ for some co-prime p and q (i.e., p is the power of the first prime, and q is the rest).

If $x^2 = a \pmod n$ then consider the Chinese-remainder function $f(\cdot)$ and denote $f(x) = (x', x'')$ and $f(a) = \langle a', a'' \rangle$. Then, we get that $\langle x'^2, x''^2 \rangle = \langle a'^2, a''^2 \rangle$ but this holds also for all four possible combinations $\langle \pm x', \pm x'' \rangle$.

Primality testing: Let $SQRT(a,p)$ denote our algorithm that on input a, p outputs either "fail" or a number x such that $x^2 = a \pmod p$. We'll use that to test whether n is prime. To test whether n is prime, we first check that n is odd and is not a power of some number. If not, we choose a random number $1 \leq x \leq n-1$, compute $a = x^2 \pmod n$ and run $SQRT(a,p)$. If it returns "fail" decide that n is a composite. If it returns some number x' such that $x'^2 = a \pmod p$ then if $x' = \pm x$ then decide that n is a prime. Otherwise decide that n is a composite.

Theorem 3. *If n is prime then our algorithm finds this with probability at least 0.99. If n is composite then algorithm finds this with probability 0.1.*

(Note that we can amplify the success probability of this algorithm using generic techniques.)

Proof. First for our analysis We first make $SQRT$ into a deterministic algorithm by simply choosing coins for $SQRT$ and hardwiring it into to the algorithm. The case of n prime is pretty easy. Suppose n is a composite which is odd and is not a prime power. For every x, say that x is "good" if $SQRT(x^2)$ is either "fail" or is equal to $x' \neq \pm x$. Since there are at least 4 roots for every a, we get that at least two of them are good (there are at most two bad roots for each a). If we hit a good x then we output the right answer. $\square$