Notes 4

I blew some proofs last time! No base case for Nim.

7 Induction in Number Theory

Number theory is about natural numbers: divisibility, primality, fractions...

Number theory used to be for "entertainment" but has recently become *very* important (more later).

Definition 7.1 *a divides b (written $a \mid b$) when $(\exists c) b = ac$. b is* divisible *by a.*

Theorem 7.2 *If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$ and $a \mid (b - c)$.*

Proof. By definition of divisibility, $(\exists t) b = at$ and $(\exists s) c = as$.
Therefore, $b + c = at + as = a(t + s)$.
Therefore, by definition, $a \mid b + c$. ∎

Theorem 7.3 *If $a \mid b$ then $a \mid bc$ for any c.*

7.1 GCDs

Remember die hard? Why did 5 and 3 buckets work but 6 and 3 not? What if buckets are 21 and 133, and goal is 13 gallons?

Definition 7.4 *The* greatest common divisor *of x and y, written $GCD(x, y)$, is the largest natural number q such that $q \mid x$ and $q \mid y$.*

We can prove GCD is well defined. A divisor always exists, and the set of divisors is bounded above and this has a largest element (proof by well ordering—any finite set has a largest element).

GCD of 24 and 36? 21 and 133?

You will prove in homework: In die hard, buckets of size x, y always produced numbers divisible by GCD of x and y.

Theorem 7.5 *GCD of x and y has form $ax + by$ for some integers a and b.*

Proof. Let $S = \{l \mid l > 0, l = ax + by, a, b \in \mathbb{N}\}$. Let $l = \min(S) = ax + by$ (exists since S is nonempty—contains $x^2 + y^2$). Let $g = \mathrm{GCD}(x, y)$.

- $g \leq l$. Proof: $g \mid x$ and $g \mid y$ means $g \mid ax + by = l$.
- $g \geq l$. It suffices to show l is a common divisor, since g is greatest such. Proof: First show $l \mid x$. Write $x = ql + r$ by division algorithm.

$$\begin{aligned} r &= x - ql \\ &= x - q(ax + by) \\ &= (1 - qa)x + (-qb)y \end{aligned}$$

  But, by division algorithm, $0 \leq r < l$. Thus $r \notin S$, so $r = 0$. Thus $l \mid x$. Similarly, $l \mid y$. Thus $l \mid ax + by$.

■

7.2 Euclid's Algorithm

The above proof tells us gcd has a particular form, but it doesn't tell us how to find it: it is a non-constructive existential proof (we can't examine infinitely many $ax + by$). An algorithm tells us how to actually find the answer. Euclid's algorithm for GCD is one of the first algorithms.

Definition 7.6 *$x \bmod y$ is the remainder on division of x by y.*

Note $x \bmod y$ is well defined by the division algorithm. $x \bmod y = x$, but for $y > 0$, we know that $0 \leq x \bmod y < y$.

Procedure Euclid(x, y):

- if $y > x$, swap x and y.
- if $y = 0$, then return x.
- otherwise, return Euclid$(x \bmod y, y)$.

For any algorithm, there are two things to prove:

- it gives an answer (termination)
- it gives the right answer (correctness)

7.2.1 Termination

Theorem 7.7 *Euclid's algorithm terminates after at most $x + y$ steps.*

Proof. Strong Induction on y. Let $P(y)$ be: $(\forall x)$Euclid(x, y) terminates in at most $y + 1$ steps. Base case: $y = 0$. We stop after 1 step since we just return x. Inductive step: Since $x \bmod y < y$, the value of the second argument to Euclid$(y, x \bmod y)$ is less than the value of the second argument to Euclid(x, y). Therefore, the time taken for Euclid(x, y) is one step to reduce to Euclid$(y, x \bmod y)$ plus (by strong induction) $x \bmod y + 1 \leq y$ steps to solve the recursive subproblem. Thus the total time is at most $y + 1$, proving the induction step. ∎

Note that the above proof does induction one one variable while holding the other variable fixed. This is perfectly legitimate. I said "by induction on y" in the proof because there was potential confusion regarding x.

The proof relies on a reduction in the value of the second argument to work. We will use the same approach in the remaining theorems in this section, but won't mention it again.

In class, I did a bad induction: I used a base case of $x = y = 0$, which was invalid because we never actually arrive at that case. We stop as soon as $y = 0$, so I can't make an inductive step at that point. Instead, I have to directly prove all possible base cases—that is, any case where $y = 0$ and x has any value.

7.2.2 Correctness

How do we prove Euclid's algorithm works? We show that it works "at each step." It follows by induction that it works all the way.

Lemma 7.8 *GCD(x, y) = GCD$(y, x \bmod y,)$.*

Proof. If $g \mid x$ and $g \mid y$, then $g \mid (x - qy) = r$ by Theorem 7.2. If $g \mid y$ and $g \mid r$, then $g \mid (qy + r) = x$ by Theorem 7.2. Therefore, (x, y) and (y, r) have same divisors; thus same GCD. ∎

Theorem 7.9 *Euclid(x, y) = GCD(x, y).*

Proof. By induction on y. Inductive hypothesis: for all x, Euclid$(x, y) = \text{GCD}(x, y)$. Base case: $y = 0$ immediate from algorithm. Inductive step: use previous lemma. ∎

Example 1 We find GCD$(949, 273)$.

	949	273
3	273	130
2	130	13
10	13	0

7.3 Finding GCD coefficients

The previous algorithms finds the GCD. It can be modified to give the desired coefficients.

Procedure Euclid-Coefficients(x, y) where $x \geq y$, $\mathrm{GCD}(x, y) = g$.

- if $y = 0$, return $(1, 0)$.
- Write $x = qy + r$
- Use Euclid-Coefficients(y, r) to find a, b such that $ay + br = g$.
- Return $(b, a - qb)$ (claiming $bx + (a - qb)y = g$).

Theorem 7.10 *Euclid-Coefficients(x, y) returns a, b such that $ax + by = GCD(x, y)$.*

Proof. Induction on y. Say $\mathrm{GCD}(x, y) = g$. Base case $y = 0$ trivial. Otherwise, say $x = qy + r$. By induction, recursive call returns (a, b) with $ay + br = g$. Then $g = ay + br = ay + b(x - qy) = bx + (a - bq)y$. ∎

Example 2 Table method for getting GCD and Coefficients: run Euclid's algorithm, then fill in the coefficients from the bottom up.

	949	273	-2	7
3	273	130	1	-2
2	130	13	0	1
10	13	0	1	0

7.4 RSA cryptosystem

RSA is a *public key* cryptosystem: anyone can encrypt a message to me, but no one can decrypt it:

- I find two large primes p and q, set $n = pq$.

- I find e,d such that $ed \bmod (p-1)(q-1) = 1$
- I publish e, n.
- Given message M, coded message $C = M^e \bmod n$ can be computed by anyone, since e, n are public.
- Given encoded message C, message $M = C^d \bmod n$.

The last step follows from the following theorem:

Theorem 7.11 *$M^{ed} \bmod n = M$.*

Proof. Textbook. ∎

No one knows how to "break" RSA:

- No one knows how to compute p, q efficiently from n.
- No one knows how to compute d efficiently without p, q (finding d actually lets you find p, q).
- No one knows how to find M efficiently without d (ie, solve $C = M^e$ for M).

Where do we get n, e, and d? There are fast algorithms for generating large prime numbers p and q, which of course gives n. But how do we find e and d? Recall our objective: if $m = (p-1)(q-1)$, we want

$$de \bmod m = 1.$$

In other words, $de = qm + 1$. Or, $de - qm = 1$. From our discussion of Euclid's algorithm, we see this equation has a solution (in $\mathbb{N}$) if and only if $\mathrm{GCD}(e, m) = 1$. So we can pick any old e, and so long as it is relatively prime to m (which is almost always the case) the above equation will have a solution that tells us d.

Used by netscape.

7.5 Fast Exponentiation

In recitation, we will show how to compute the big exponentiations needed to implement RSA quickly. It relies on the fact that show $x^{ab} \bmod n = (x^a \bmod n)^b \bmod n$.

7.6 Factoring

If we knew p and q, it would be easy to find d given e and break RSA. So RSA relies on the "hardness" of factoring. But maybe we could find different primes p', q' such that $n = p'q'$? If such solution are common enough, then just trying lots of possibilities might give us one. Fortunately, the *Fundamental Theorem of Arithmetic* shows that this isn't the case:

Theorem 7.12 *Every number has a* unique *factorization into primes (except for reordering the factors)*

So the only way to factor n is to find p, q.

Lemma 7.13 *Let p be a prime. If $p \mid ab$, the $p \mid a$ or $p \mid b$.*

Proof. If $p \mid a$, we are done, so assume not. It follows that $\mathrm{GCD}(p, a) = 1$ (since the only divisors of p are p and 1, and p doesn't divide a). It follows that there is an equation $sa + tp = 1$. It follows that $sab + tpb = b$. But $p \mid sab$ (since $p \mid ab$) and $p \mid tpb$, so $p \mid b$ (by Theorem 7.2). ∎

Corollary 7.14 *Let p be a prime. If $p \mid a_i a_2 \cdots a_k$, then $(\exists i) p \mid a_i$.*

Proof. (of the fundamental theorem). We proved last time that every number has a factorization. We need to show it is unique. We argue by (strong) induction. The base case $n = 1$ id trivial. So suppose the claim true for all numbers less than n. Suppose one factorization is $n = p_1 p_2 \cdots p_k$, with the p_i written in nondecreasing order. Suppose there is a *different* factorization into primes $n = q_1 q_2 \cdots q_l$. It follows that $p_1 \mid q_1 \cdots q_l$. This, by the previous corollary, $(\exists i) p_1 \mid q_i$. But then $p_1 = q_i$, implying two factorizations of n/p_1. This contradicts our inductive hypothesis, so the theorem if proved. ∎

Unique factorization gives another way to compute GCD: factor both numbers, and multiply out all their common factors.