

Mathematics 55– Spring 2005
Lecture 9 (Monday 2/7/2005)
Euclidean Algorithm and Congruence

Announcements: Solutions to problem sets 1,2 are posted on course web site. Quiz Tuesday will focus on the reading assignments, §§1.8-2.6.

Definition of congruence: $a \equiv b \pmod{m}$ (here $m \in \mathbb{N}$) means that $m|(a-b)$. We say that a is congruent to b modulo m . Another way (useful) to put this: $a = b + km$ for some integer k .

Theorem: Suppose that $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. Then $(a+c) \equiv (b+d) \pmod{m}$ and likewise $ac \equiv bd \pmod{m}$. (Both of these are very easy to prove; just write $a = b + km$ and $c = d + jm$, where $k, j \in \mathbb{Z}$, and follow your nose; I presented a proof of the first conclusion.)

Example: : What was called “clock arithmetic” when I was in grade school. Take $m = 12$. If we add five hours to 10 o’clock we get 3 o’clock, not 15 o’clock; we tell time modulo $m = 12$.

Arithmetic modulo 2

a	b	$a + b \pmod{2}$	$ab \pmod{2}$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Statement of the Euclidean Algorithm: Purpose: Find $\gcd(a, b)$ where $a, b \in \mathbb{N}$.

(Recall an inefficient algorithm that we learned last time: Factor $a = p_1^{c_1} \cdots p_N^{c_N}$ and $b = p_1^{d_1} \cdots p_N^{d_N}$. Then the gcd is $p_1^{f_1} \cdots p_N^{f_N}$ where each $f_j = \min(c_j, d_j)$. The trouble with this algorithm is that factoring large integers (with say 128 digits in their binary expansion) takes a fairly long time. The algorithm we’ll learn is quite a bit faster for finding the gcd when a, b are fairly large.)

Proposition: If $a = qb + r$ where a, b, q, r are nonnegative integers and $a, b \geq 1$ then $\gcd(a, b) = \gcd(b, r)$.

Proof: We will prove that any common divisor of a, b is a common divisor of b, r , and vice versa. Since these two pairs have all the same common divisors, in particular they must then have the same largest common divisor.

Suppose that c divides both a, b . Then c divides qb . Since it divides a , it also divides $r = a - qb$ (by a theorem from earlier in the text). Thus c divides r as well as b .

The converse is proved in the same way, using the formula $a = qb + r$ to conclude that $c|a$. □

Description of the Euclidean algorithm: Start with a pair (a, b) . If $a = b$ then $\gcd(a, b) = a = b$ and we're done. If not, reorder the numbers if necessary so that $a > b$. To prepare, define $r_0 = a$ and $r_1 = b$.

Now apply the *division algorithm* to express $r_0 = q_1 r_1 + r_2$ for some integer remainder r_2 satisfying $0 \leq r_2 < r_1$. We now change the question, by considering $\gcd(r_1, r_2)$ instead of $\gcd(r_0, r_1)$.

Note two things: • According to the preceding Proposition, this question has the same answer as the question we started with. • The new pair (r_1, r_2) is smaller than the original pair (r_0, r_1) , in the sense that $r_1 < r_0$ and $r_2 < r_1$; we've replaced the given problem by a simpler problem, which has exactly the same answer!

If $r_2 = 0$ the procedure terminates (and see below for what happens next). Otherwise apply division algorithm to express $r_1 = q_2 r_2 + r_3$ for some integers q_2, r_3 with $0 \leq r_3 < r_2$. If $r_3 = 0$ then stop; otherwise replace the pair (r_1, r_2) by the "smaller" pair (r_2, r_3) and continue the process.

This produces a sequence of remainders $r_0 > r_1 > r_2 > \dots$. All are nonnegative, so eventually we must reach 0. Define the index n so that r_{n+1} is the first remainder to equal zero, and r_n is the last to be strictly > 0 .

Once the process stops we have $r_{n-1} = q_n r_n + 0$, so $r_n | r_{n-1}$, so $\gcd(r_{n-1}, r_n) = r_n$. By repeated application of the Proposition we thus conclude that

$$\boxed{r_n = \gcd(r_{n-1}, r_n) = \dots = \gcd(r_0, r_1) = \gcd(a, b)}$$

□

We worked a couple of quick examples; it's quite easy to apply this algorithm in practice.

The proof of the Euclidean algorithm gives a stronger conclusion that is quite important:

Key theorem: For any positive integers a, b , there exist integers s, t such that

$$\boxed{\gcd(a, b) = sa + tb.}$$

Example: $\gcd(5, 3) = 1$ can be expressed as $1 = 2 \cdot 3 + (-1) \cdot 5$. Note that we had to use a negative coefficient in front of 5.

(Those of you who've taken Math 54 may catch a whiff of linear algebra here; $sa + tb$ is a *linear combination* of a, b with *integers* coefficients. Those who haven't taken M54, don't worry about this.)

Comment (without proof) on efficiency/complexity: If $a > b$ this algorithm takes $O(\log(a))$ steps. The alternative method using prime factorization, in contrast, is not $O([\log(a)]^p)$ steps for any finite power p .

Two **corollaries**: (i) If $c|ab$ and c, a are relatively prime, then $c|b$. (*Example*: : $6|4 \cdot 9$, yet 6 does not divide either 4 or 9; just because $c|ab$ is by itself not sufficient to force c to divide one factor or the other. (ii) If p is prime and $p|ab$ then $p|a$ or $p|b$. (6 is not prime, so again this fails to apply to the preceding example.)

Proof of (i): Write $1 = sa + tc$ and multiply through by c to get $b = s(ab) + (tb)c$. Now c divides tb since it divides itself, and c divides sab since it is assumed to divide ab . Therefore c divides $sab + tb = b$. \square

(ii) is a consequence of (i), for if p does not divide a then p, a must be relatively prime. (The only positive integer divisors of p are 1 and p itself; if p doesn't divide a then the only common divisor is 1, hence $\gcd(p, a) = 1$.) \square