11

Division Mod n,

Linear Integer Equations,

Random Numbers,

The Fundamental Theorem of Arithmetic

Bezout's Lemma

Let's look at the values of 4x + 6y when x and y are integers. If x is -6 and y is 4 we get zero. If x is −1 and y is 1 we get 2. In fact a little experimentation will convince you that you can get all the even integers but only even integers. That is 4x + 6y generates the collection of even integers. We will denote this set by 2Z.[1] Similarly 3Z would represent the collection of all multiples of 3: $3Z = \{...-12,-9,-6,-3,0,3,6,9,12,...\}$. It turns out that if we look at expressions of the from ax + by, or ax + by + cz where a, b and c are fixed integers (for example, 2x + 7y or 5x + 10y + 5z) we always get all multiples of some integer (which were defined earlier as *modules*). For example if we look at all values of 5x + 10y + 5z, we get 5Z which is $\{...-10,-5,0,5,10,...\}$. It would be very useful if you would take 15 minutes to see for yourself what numbers you can generate with 4x + 6y and with 6x + 9y and with 4x + 7y. This all can summarized by one of the most useful theorems in discrete mathematics:

[1] Z in this context is used as it is generally used in abstract algebra to mean *integer*. This is from the German word for number, *zahl*.

The equation ax + by = c where all letters are integers and a, b, and c are known, is solvable if and only if (a, b)|c.

Bezout's Lemma

This theorem is sometimes known as *Bezout's lemma*. The theorem also holds true if we extend it to more than 2 variables. For example, $5x + 7y + 4z$ takes on the values $GCD(5,7,4)\cdot Z = Z$. That is, $5x + 7y + 4z$ takes on the values of all integers. Similarly $4x + 12y + 8z$ takes on the values of $GCD(4,12,8)Z = 4Z$.

Example $4x + 12y = 8$ is solvable because $(4,12) = 4$ and $4|8$, but $4x + 12y = 6$ is not solvable because $4 \nmid 6$ (4 does not divide 6). $3x + 6y = 15$ has a solution because $(3,6) = 3$ and $3|15$. Similarly, $8x + 10y = 29$ does not have a solution because $(8,10) = 2$ and $2 \nmid 29$.

If the proof is too difficult to follow, jump ahead to the next heading. But first make sure you understand what the theorem says.[1] There is a second proof in the second appendix to this section.

To prove Bezout's lemma, we need to prove our assertion above that expressions of the form ax + by where a and b are fixed integers are all multiples of a single constant d. Let d be the smallest positive value taken on by expressions of the form ax + by. Suppose that there is a number c of the form ax + by that is not a multiple of d. Suppose also that c is positive (if c is negative then $-c$ is a positive value taken on by ax + by). By the division algorithm we can divide d into c so that we get a positive remainder less than d. It can't be zero because we assume that d does not divide c. That is, there is a q and an r such that $d > r > 0$ and $c = d\cdot q + r$. Let x_0, y_0, x_1, and y_1 be the integers that satisfy $ax_0 + by_0 = d$ and $ax_1 + by_1 = c$. Then

[1]For those who have had abstract algebra, there is an easier route to Bezout's lemma. First, note that with a and b fixed, the expression ax + by generates a subgroup of the integers under addition. Hence, this group must be cyclic. Denote the positive generator of this set as d. It is easy to show that d divides both a and b. It is also easy to show that any common divisor of a and b divides d. Hence d = (a,b).

$a(x_1 - qx_0) + b(y_1 - qy_0) = ax_1 - qax_0 + by_1 - qby_0 = ax_1 + by_1 - q(ax_0 + by_0) = c - q \cdot d = r$. That is, r is a value taken on by ax + by. But r is less than d and d is by definition the smallest positive value of ax + by and we have a contradiction. Hence ax + by takes on all of the multiples of some number d and takes on only those values.

To finish the proof of Bezout's lemma, let us suppose that a, b, and c are fixed integer values (they can be negative as well as positive). We still denote the smallest positive value achieved by ax + by as d. Let (a,b) = g; then we can write a = a'g, and b = b'g. We now have that ax + by = a'gx + b'gy = g(a'x + b'y) = c. Hence g **must** divide c. Also it must divide d since d is a value that c can have. (If it doesn't, then ax + by = c has no solution.) Now, two values achieved by ax + by are a and b. The first occurs when x = 1 and y = 0 and the second occurs when x = 0 and y = 1. Since d divides all values achieved by ax + by, it must divide a and b. Hence it is a common divisor of a and b. But since the largest common divisor of a and b, g, divides d, then d and g must be equal.

Relatively Prime Integers: $\perp$

We say two integers, a and b, are relatively prime if they have no common positive factor other than 1: GCD(a, b) = 1. This is also denoted: $a \perp b$.[1] For example: $3 \perp 5$, $4 \perp 9$. If n is any positive integer, $1 \perp n$ and $n \perp (n+1)$.

[1]This notation was suggested in *Concrete Mathematics: A Foundation for Computer Science* by Graham, Knuth and Patashnik, Addison-Wesley, 1989.

Euclid's Lemma

Bezout's lemma gives us one of the most useful facts of number theory, which is known as *Euclid's lemma*.

If $a \mid bc$ and $a \perp b$ then $a \mid c$.

Proof: By Bezout's lemma there is a solution to $ax + by = 1$. We multiply both sides by c to get $acx + bcy = c$. a divides both terms on the left hand side because it is a factor of the first term and we were given that $a \mid bc$. Hence $a \mid c$.

Division in Z_n

Remember Z_n is arithmetic modulo n in a slightly different guise. In Z_n we restrict ourselves to the integers $\{0,1,2,...,n-1\}$. Once we are in Z_n, these are the only integers that exists. In Z_n addition, multiplication, and subtraction are very much like in the integers. The tricky operation is division. We will define two kinds of elements in Z_n. The first are *units*. x is a unit if it has a multiplicative inverse. That is, x is a unit if there exists an element x' such that $x \cdot x' = 1$. An element, $x \neq 0$, is a *divisor of zero* if there exists an element $x'' \neq 0$ such $x \cdot x'' = 0$. Suppose that x has an inverse and is a divisor of zero. Then there exists x' so that $x \cdot x' = 1$ and there exists an element x" such that $x \cdot x'' = 0$. Multiply both sides of $x \cdot x' = 1$ by x" so that $x'' \cdot x \cdot x' = x''$; $(x'' \cdot x) \cdot x' = x''$; $0 \cdot x' = x''$; $0 = x''$. But by definition $x'' \neq 0$. This is a contradiction, and it means that **an element cannot be both a unit and a divisor of zero**.

We will now show that every element in Z_n, other than 0, is a unit or a divisor of zero. Suppose that element a is relatively prime to n. Then, from the preceding section, we know that $a \cdot x + b \cdot n = 1$ has a solution. Hence, $ax \equiv 1 \pmod{n}$ has a solution and a is a unit. Suppose that element a is not relatively prime to n. Then $(a, n) = d$ with $d > 1$. We can write $a = a' \cdot d$ and $n = n' \cdot d$ with $1 < n' < n$. (Remember that in Z_n, $n = 0$.) Then $a \cdot n' = a' \cdot d \cdot n' = a' \cdot 0 = 0$. Hence, **in Z_n every non-zero element is a divisor of zero or is a unit**.

If n is a prime number, then every non-zero number in Z_n is a unit. In such a case, we have division. We divide a by b, by multiplying a by the inverse of b which we denote here as b^{-1}. **We define $a/b = a \cdot b^{-1}$.** For example in Z_7, division is well defined. In Z_7, $3/5 = 3 \cdot 3 = 2$; $2/4 = 2 \cdot 2 = 4$; $5/1 = 5 \cdot 1 = 5$.

If n is a composite number, on the other hand, then we have some elements that are divisors of zero. For example, in Z_6, we can divide by 1 and 5 only and 2, 3, and 4 are divisors of zero.

☐ **Exercise 1** Compare the multiplication tables of Z_6 and Z_7 to see why one system has division and the other does not.

☐ **Exercise 2** In Z_{12}, divide 4 by 2, 3, 5, 7,8, and 11.

Cancellation in Modulo n Arithmetic

It is now easy to see that if (c, n) = 1 that cancellation can be performed on $ac \equiv bc \pmod{n}$. Since c is relatively prime to n there exists c^{-1} such that $cc^{-1} \equiv 1 \pmod{n}$. Multiplying both sides of $ac \equiv bc \pmod{n}$ by c^{-1} we get $ac\,c^{-1} \equiv bcc^{-1} \pmod{n}$ and then $a \equiv c \pmod{n}$. If (c, n) ≠ 1 then all bets are off. For example, $2 \cdot 4 \equiv 2 \cdot 1 \pmod{6}$ but $4 \not\equiv 1 \pmod{6}$. We can however state the following rule:

$$ac \equiv bc \pmod{n} \quad \Rightarrow \quad a \equiv b \left(\mod \frac{n}{(c,n)} \right)$$

☐ **Exercise 3** Prove the general law of cancellation just given. (Hint, this problem can be solved by using Bezout's lemma and then Euclid's Lemma.)

How to Solve ax ≡ b (mod n)

Bezout's lemma will answer whether ax ≡ b (mod n) has a solution for x (a, b, and n are given). By definition, ax ≡ b (mod n) if n|ax – b. But n divides ax – b is equivalent to saying that there exists some integer c such that ax – b = cn. In other words ax ≡ b (mod n) has a solution if and only if there is a solution to ax – cn = b. By Bezout's lemma, ax - cn = b has a solution (for x and c) if and only if (a, n)|b. One trick to solve ax ≡ b (mod n) for x is to solve ax – cn = b for x and c. How to solve this is shown in the next section. However, there is another method which always works as well.

Example We want to find solutions to 2x ≡ 5 (mod 7). It has a solution if (2, 7)|5, which it does. To find the solution, we add the modulus, which is congruent to 0, to the right hand side. This gives us 2x ≡ 12 (mod 7), and thus the solution is x = 6.

Example With the congruence 2x ≡ 5 (mod 6), (2, 6) = 2 which doesn't divide 5. Hence there is no solution.

Example We want to solve 12x ≡ 18 (mod 42) which has a solution since (12, 42)|18. The first step to solving this problem is to perform cancellation. Since 6 divides both sides, we can reduce the problem to $2x \equiv 3 \left(\bmod \frac{42}{(6,42)} \right) \equiv 3 \pmod 7$.

Now the general solution technique will be explained shortly. But in this case it goes as follows: By adding 7 (which is congruent to 0) to the right side, we get the solution x = 5. By repeatedly adding 7 to 5 we get 6 solutions, within modulus 42, which are: 5, 12, 19, 26, 33, 39.

Consider now the problem ax ≡ b (mod p) where a, b, and p are given and p is prime. If p does not divide a, then (a, p) = 1 and the problem has a solution. If p does divide a, then a is congruent to 0 and the problem then is equivalent to 0x ≡ b (mod p) which leads to

$0 \equiv b \pmod{p}$. That is if b is a multiple of p then any x will work otherwise there is no solution.

Example Consider $ax \equiv b \pmod{7}$. Since 7 is a prime, if a is not a multiple of 7 there is a solution. For example $2x \equiv 1 \pmod{7}$ has the solution x = 4. $3x \equiv 1 \pmod{7}$ has the solution x = 5. $4x \equiv 1 \pmod{7}$ has the solution x = 2. $5x \equiv 1 \pmod{7}$ has the solution x = 3. $6x = 1 \pmod{7}$ has the solution x = 5.

All Solutions of $ax \equiv b \pmod{n}$

Now let us get a little more ambitious. Instead of just solving $ax \equiv b \pmod{n}$ we want all of its solutions. If k is a solution of $ax \equiv b \pmod{n}$, then $k + r \cdot n$ (r any integer) is also a solution. This is because $n \equiv 0 \pmod{n}$. To make things interesting let us then concentrate on the solutions with the range from 1 to n − 1. That is we want all solutions within the modulus. Another way of looking at it is that we want all of the solutions of ax = b in the number system Z_n.

Again: $ax \equiv b \pmod{n}$ has a solution if and only $(a, n)|b$. In this case, that $ax \equiv b \pmod{n}$ has a solution, we can cancel (a, n) out of a and b to get the new equation: $\frac{a}{(a,n)}x \equiv \frac{b}{(a,n)} \left(\bmod \frac{n}{(a,n)} \right)$. We rewrite this as $a'x \equiv b' \pmod{n'}$. We now have that

n' is relatively prime to a' (because we have already divided out their common factor). Using Bezout's Lemma we can show that by repeatedly adding n' to the right hand side of $a'x \equiv b' \pmod{n'}$ we can eventually solve for x. Given a solution, x, of $a'x \equiv b' \pmod{n'}$, it is a solution of $ax \equiv b \pmod{n}$ but so is $x + t \cdot \frac{n}{(a,n)}$ (where t is any integer). This follows from that fact that $a \cdot \frac{n}{(a,n)} = n \cdot \frac{a}{(a,n)}$ and again that $n \equiv 0 \pmod{n}$. Our question then becomes, how many

multiples are there of $\frac{n}{(a,n)}$ within the modulus n? The answer is (a, n). Putting all of this together we get:

ax ≡ b (mod n) has a solution if and only if (a, n)|b. In that case, in the system z_n there are (a, n) solutions. If y is one such solution, then the others are $y + \frac{n}{(a,n)}$, $y + 2\cdot\frac{n}{(a,n)}$, and so on (where n can be any integer).

How Not to Solve ax ≡ b (mod n): Multiplication Can be Evil

Consider the problem 9x ≡ 3 mod 12. If we multiply both sides by 2, we get 6x ≡ 6 mod 12. This yields x ≡ 1 mod 12, which is **not** a solution. In fact using the technique above the solutions within modulus 12 are 3, 7, and 12. Note these are also solutions to 6x ≡ 6 mod 12. But when we multiplied 9x ≡ 3 mod 12 by 2, we picked up anomalous solutions. The reason, put simply, is that in mod 12 arithmetic there are divisors of zero, such as 3 and 4. These complicate matters. Solution by multiplication is safe then when n is a prime. However, if we are trying to solve ax ≡ b (mod p) either a is a multiple of p or it is relatively prime to p. In the first case any number x solves the equation. In the second case, we know there is exactly one solution within the modulus p. Also, we know that a has an inverse mod p, a^{-1}, and $x \equiv a^{-1}\cdot b \pmod{p}$.

How to Solve ax + by = c

Bezout's lemma tells us whether ax + by = c has a solution, but it does not tell us how to find it. The following examples show two general technique that always work. The first is preferable for working by hand; the second is preferable for programming. However, once we have one solution, there are an infinite number of solutions. It is easy to check that if ax + by = c has a solution, x_0 and y_0, and if t is any integer, then another solution is given by $x = x_0 + bt$ and $y = y_0 - bt$. However, this may not be the most general solution. Remember that ax + by = c has a solution if and only if (a, b)|c. If (a, b) is greater than 1, we can divide it out of the equation and yield a simpler equation.

Example We want to solve 13x + 21y = 3. This is solvable since (13, 21)|3. The best way to solve it (by hand) is as follows: We solve for x and y separately, and we can do it in either order. However, this time we will do y first for the reason that we will then work with the smaller modulus, that is mod 13 instead of mod 21. In terms of y, we have that $21y \equiv 3 \pmod{13}$. This can be written $8y \equiv 3 \pmod{13}$. We solve this as above, by adding the modulus, 13, to the right hand side. We get $8y \equiv 16$. This time we only have added the modulus once. We have $y \equiv 2 \pmod{13}$. The second part of this method is to rewrite the last relation as $y = 2 + 13t$. We then substitute this into the original equation to get $13x + 21(2 + 13t) = 3$. This reduces to $13x = -273t - 39$. Hence x = -3 - 21t.

We can summarize this technique as follows:

Given ax + by = c.

1. If (a, b)∤c Then no solution [stop].
2. Divide ax + by = c by (a, b) to get Ax + By = C
3. Solve $Ax \equiv C \pmod{|B|}$ to get x = h + |B|t {We use |B| to ensure a positive modulus.
4. Substitute x = h + |B|t in Ax + By = C to solve for y {y will be of the form y = k - At.

It is not hard to prove that

Given the problem ax + by = c. If x = u and y = v are any pair of solutions, then the general solution is of the form: $x = u + \frac{b}{(a,b)}t$ and $y = v - \frac{a}{(a,b)}t$

A second technique for solving equations of the form ax + by = c is to apply the Euclidean algorithm to a and b, and then to work the algorithm backwards from the solution to solve for x and y. This is most easily shown by an example. The general technique is the ***Extended Euclidean Algorithm*** which is given in the first appendix to this section.

Example We would like to solve 7x + 9y = 5. Since (7, 9) = 1 and since 1 divides 5 we know that a solution exists. Our problem is to find it. We know in particular that 7x + 9y = 1 has a solution, so we first find that. The secret is to use the Euclidean algorithm on 7 and 9 and to write down the steps. By retracing those steps backwards we can solve the problem. Using the Euclidean algorithm to find the GCD of 7 and 9, we have: 9 = 7·1 + 2; 7 = 2·3 + 1; 2 = 2·1. The method requires that we take the second to the last equation, which in this case is 7 = 2·3 + 1. We rewrite it to solve for its remainder which will always be the GCD; in this case we get 1 = 7 – 2·3. We then work back to the first equation; for each equation we solve for the remainder and substitute that in the earlier equation, until we have solve for the as a function of a and b. In this case, there is only one previous equation: 9 = 7·1 + 2. Solving for the remainder we get 2 = 9 – 7·1. Substituting for 2 in 1 = 7 – 2·3, we get 1 = 7 –(9 – 7·1)·3. We want to rearrange this to give our GCD which is 1 in terms of 7 and 9. We get: 1 = 4·7 – 3·9. To solve our original problem, we multiply both sides of this by 5 to get: 5 = 20·7 – 15·9. Hence the solution to the problem is x = 20 and y = –15. The general solution is x = 20 + 9t and y = –15 – 7t.

The same method shown in the last two examples can be extended to solve problems like $6x + 10y + 15z = 7$. Bezout's lemma can be extended in the obvious manner to show whether a solution exists, and then the above technique for finding the solution can be extended as well.

- ☐ **Exercise 4** Solve $9x + 12y = 5$.
- ☐ **Exercise 5** Solve $9x + 12y = 6$.
- ☐ **Exercise 6** Solve $45x + 50y = 20$.

Generation of Random Numbers

Random number generation on computers means generating a sequence of numbers that seem random but are not. There is one technique that has been dominant since the early days of computers and is due to the eminent number theorist D. H. Lehmer. Although many texts give us this algorithm, I think that I can make it clearer by using a simple example.

First let me tell you a simple fact that is not easy to prove, and which we will not prove, but is easy to understand. If n is a prime, then we know Z_n has division. The non-zero elements are all units. It is to prove that the product of units is always a unit. That is if x and y are units with inverses x' and y', then y'x' is the inverse of xy since xyy'x' = x(yy')x' = x1x' = xx' = 1. The remarkable fact, which I ask you to accept on faith, is that when n is prime the units of Z_n can all be written as powers of a single unit, which is known as a *primitive root modulo n*.

The Lehmer algorithm generates a repeating cycle of integers. These integers are often rewritten as *uniform variates*. That means real numbers between 0 and 1 such that each number is equally likely.[1] The algorithm requires a prime p and then generates a cycle of $p - 1$ integers. The most common prime used is $2^{31} - 1$. This means that a cycle of $2^{31} - 2$ integers is generated, with $2^{31} - 2 = 2{,}147{,}483{,}646$.

Example We will illustrate the algorithm by using p = 11. This of course is totally useless for generating random numbers but illustrates the algorithm perfectly well. We

[1]Clearly we generate a finite subset of the interval from 0 to 1 but the object is to appear like a uniform sample from that interval.

need to find a primitive root in Z_{11}. That is we need a unit that when multiplies by itself gives all ten units of Z_{11}. It is not simple to find such a unit. In the case of Z_{11} though there are only 10 candidates and we can find one by trial and error. Such a primitive root is 8. If we multiply 8 by itself we get 9. If we multiply 9 by 8 we get 6, and if we multiply 6 by 8 we get 4. The whole sequence is 8, 9, 6, 4, 10, 3, 2, 5, 7, 1. This is a pseudo-random sequence of length 10 and it contains each of the integers 1 through 10 just once. To turn it into a uniform sequence we divide each integer by 11. This gives us the sequence .727, .818, .545, .365, .909, .273, .182, .455, .636, .091.

The preceding algorithm of Lehmer's is known as the *multiplicative linear congruential generator*. It can be summarized as follows:

1 Start with a large prime p and a primitive root mod p denoted by r.

2 Choose an integer, x, between 0 and p – 1.

3 Compute $x \leftarrow x \cdot r \pmod{p}$.

4 Output x

5 Go to 3.

Notice that the algorithm as stated does not terminate. In reality it terminates when you have computed the last number that you want. If it is your aim to produce a sequence of uniform variates, replace line 4 by: **Output x/p**. Frequently, the algorithm is made slightly more complex by adding an additive constant. The algorithm would then be stated:

1 Start with a large prime p, a primitive root mod p denoted by r, and an additive constant c, such that $0 < c < p$.

2 Choose an integer, x, between 0 and p.

3 Compute $x \leftarrow x \cdot r + c \pmod{p}$.

4 Output x

5 Go to 3.

This algorithm is called a *mixed linear congruential generator*.

The Fundamental Theorem of Arithmetic

Earlier we defined *relatively prime*, but we have not defined primes yet (they did show up in the section on induction but that was an example that you could defer if necessary). A positive integer is a *prime* if it is greater than 1, and if its only positive divisors are itself and one. For example, the first ten primes are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Students often wonder why 1 is excluded as a prime. There are many reasons. A simple reason is this. If you know that a number n is divided by some prime p, that tells you something about n. However, knowing that 1 divides n tells you only that n is an integer, which is not information since we are restricting ourselves to integers. The Fundamental Theorem of Arithmetic says that each positive integer can be written in one and only one way as a product of primes except for the order of the numbers. If we were to write the primes in order of increasing size then the order would be unique. We will have to build up to the theorem with several preliminary results. First we need more definitions. A number greater than 1 that is not a prime is a *composite* number. (1 is a unit.)

The Well Ordering Theorem: A nonempty set of positive integers has a least element.

Proof: Let S be a set of positive integers without a least element. We will construct a set T of positive integers not in S. 1 must belong to T, because if it were in S, it would be the least element. Now suppose 1, 2, 3 through n were in T (this is the induction hypothesis) then n + 1 cannot be in S because it would be the least element. Hence n + 1 must belong to T. Therefore T contains all the positive integers and S must be empty.

Theorem: If n is a composite number then n has a some divisor, d, with $1 < d < n$.

Proof: Since n is composite, it is greater than 1. If it has no divisors other than 1 and n it would be a prime. Any other divisor would be between 1 and n since no positive integer can be divided (without remainder) by a larger integer.

Theorem: If n is a composite number then n has a some prime divisor, p, with $1 < p < n$.

Proof: From the previous theorem, we know that there is a factor d, with $1 < d < n$. If d is a prime, we are finished. Otherwise d is a composite and we can employ the previous theorem to find a new divisor f, such that f divides d and $1 < f < d$. Hence f divides n and $1 < f < n$. Again, if f is a prime we are done. Otherwise we can repeat the process as until we reach a prime divisor of n. If we did not find such a prime, we would have an infinite decreasing sequence of positive integers. Amongst other things, this violates The Well Ordering Theorem.

Theorem: A composite number n, can be written as a product of primes.

Proof: Given a positive integer n, we know from the previous theorem that it can be written as $n = p_1c_1$ for some prime p_1. If c_1 is a prime we are finished. Otherwise, we apply the same theorem to c_1 to get the decomposition $n = p_1p_2c_2$ where p_2 is a prime. If c_2 is a prime, we are finished. Again, just as in the proof of the last theorem we can continue this process until we have a product of primes. Suppose otherwise, that after n steps we have $n = p_1p_2..p_nc_n$ where c_n is a composite number greater than 1. Since each prime is at least equal to 2, we then have $n \leq 2^{n+1}$ which is absurd (note $p_1p_2..p_nc_n$ has n + 1 terms each at least 2).

Theorem: If p is a prime and $p|a \cdot b$ (with a and b positive integers) then either p|a or p|b or both.

Proof: If $p \nparallel a$ then $p \perp a$ (they are relatively prime). Then Euclid's lemma applies and p|b.

Theorem: If $p|a_1a_2...a_k$ then for some i, $1 \leq i \leq k$, $p|a_i$.

Proof: Proof is by induction on k, the number of terms. If k = 1, the theorem is trivially true. If k = 2 then this was proven true in the last theorem. Suppose that the theorem is true for k > 1 (or 2 since we have verified that). This is the induction hypothesis. Now consider the case $p|a_1a_2...a_{k+1}$. We write this as $p|(a_1a_2...a_k)a_{k+1}$ where we look at the product as a product of 2 terms instead of k + 1 terms. By the previous theorem either $p|a_{k+1}$ or $p|a_1a_2...a_k$. If $p|a_{k+1}$ is true we are finished. Otherwise $p|a_1a_2...a_k$ and the induction hypothesis applies.

The Fundamental Theorem of Arithmetic: A composite number n, can be written as a product of primes in one way and only one way other than the order of the terms.

Proof: Suppose $n = p_1p_2...p_h = q_1q_2...q_k$ where the terms in both expressions are primes. Assume $h \le k$ (otherwise rename the p's as q's and vice versa). Since $p_1|n$ then $p_1|q_1q_2...q_k$. By our previous theorem p_1 must divide q_i for some i. Since q_i is also a prime $p = q_i$. We factor p out of both products. We then do the same thing for p_2 through p_h. This might leave 1 = 1 implying that both products contained exactly the same terms. Note that it also implies that both expressions have the same number of occurrences of each prime. The only alternative would be $1 = q_aq_b...q_r$ where $q_aq_b...q_r$ is the product of the remaining primes from the right side expression. Clearly this is impossible, hence both $p_1p_2...p_h$ and $q_1q_2...q_k$ contain exactly the same primes.

Example Given two positive integers a and b, each can be written as a product of primes. $a = p_1^{c_1} p_2^{c_2} \dots p_h^{c_h}$ $b = p_1^{d_1} p_2^{d_2} \dots p_h^{d_h}$. In this representation the primes p_1 through p_h are distinct. We can give both numbers the same prime factors because if, say, p_1 does not divide b, we simply make its exponent equal to 0. Now let us define m_i as the minimum of c_i and d_i, and similarly we'll define M_i as the maximum of c_i and d_i. You should be able to prove that the greatest common divisor of a and b, (a,b), is $p_1^{m_1} p_2^{m_2} \dots p_h^{m_h}$. Similarly you should be able to prove that the least common multiple of a and b, [a,b], is $p_1^{M_1} p_2^{M_2} \dots p_h^{M_h}$. As a numerical example consider a = 18 and b = 60. The prime factorizations are $18 = 2 \cdot 3^2$ and $b = 2^2 \cdot 3 \cdot 5$. Then $(18,60) = 2 \cdot 3$ and $[18,60] = 2^2 \cdot 3^2 \cdot 5$.

Appendix A to Section 11

The Extended Euclidean Algorithm

(This is adapted primarily from Knuth's *The Art of Computer Programming*, Vol II.)[1]

Given a, b, return u_1, u_2, u_3 such that $u_1a + u_2b = u_3 = (a, b)$

$(u_1, u_2, u_3) \leftarrow (1, 0, a)$
$(v_1, v_2, v_3) \leftarrow (0, 1, b)$
While $v_3 \neq 0$
    $q \leftarrow \lfloor u_3/v_3 \rfloor$
    $(w_1, w_2, w_3) \leftarrow (u_1, u_2, u_3) - q(v_1, v_2, v_3)$
    $(u_1, u_2, u_3) \leftarrow (v_1, v_2, v_3)$
    $(v_1, v_2, v_3) \leftarrow (w_1, w_2, w_3)$
Return (u_1, u_2, u_3)

Recursive Version of Extended-Euclidean Algorithm

Given a, b, return u_1, u_2, u_3 such that $u_1a + u_2b = u_3 = (a, b)$

Extended-Euclidean(a, b)
    If b = 0 then return (1, 0, a) [exit]
    $(v_1, v_2, v_3) \leftarrow$ Extended-Euclidean(b, a mod b)
    $(u_1, u_2, u_3) \leftarrow (v_2, v_1 - \lfloor a/b \rfloor v_2, v_3)$
    Return (u_1, u_2, u_3)

---

[1]This algorithm uses matrix notation, specifically we have the convention that d(a, b, c) = (da, db, dc). Matrices will be covered in section 18. I used this convention (from Knuth) because it is, I think, by far the clearest way to state the algorithm and to study the algorithm and to prove that the algorithm works.

Appendix B to Section 11

An Inductive Proof of Bezout's Lemma

Theorem: Given positive integers a and b, there exists integers x and y such that $ax + by = d = (a, b)$.

Proof: The theorem is true if a and b each equal 1. The proof is by induction on $s = a + b$. Hence the theorem is true for $s = 2$. Let the theorem by true for s equal to 2 up to n.[1] We want to show that the theorem is true for $s = n + 1$. Suppose that $a + b = n + 1$. We can assume that $a \geq b$ (otherwise exchange a and b). If $a = k \cdot b$ for some integer k then $d = b$ and a solution is given by $x = 0$ and $y = 1$. Hence we can assume that a is not a multiple of b. Then by the division algorithm there exists integers q and r such that $a = b \cdot q + r$ where $b > r > 0$. We have that $b + r < a + b = n + 1$, hence by the induction hypothesis there exists integers u and v such that $ub + vr = (b, r)$. But $(b, r) = (a, b) = d$ (we proved this before: it is <u>the</u> key to the Euclidean algorithm). We now have that $ub + vr = ub + v(a - bq) = d$. In other words, $av + b(u - qv) = d$; this gives us $x = v$ and $y = (u - qv)$ and we are done.

[1]Normally induction is done on the positive integers starting at 1. However, it was shown in Section 6 that the same technique can be applied to the set of all integers greater than some integer k. In the proof here, $k = 2$.

Appendix C to Section 11

A Program For Inverses Mod n

It is simple to write a program into a modern programmable calculator to find inverses for multiplication mod n. Such calculators often have a built in GCD function. If yours does not have this use the Euclidean algorithm.

Program Inverse

Input modulus n
Input value c
    If GCD(n, c) ≠ 1 **Then**
        Output "No Inverse";
        Exit;
    x ← 1
    Repeat While 1 = 1
        x ← x·c
        If x·c mod n = 1 **Then**
            Output x
            Exit

Here is what the program looks like in my TI-86:

```
PROGRAM: Inverse
:Prompt nn
:Prompt cc
:If gcd(nn,cc)≠1
:Then
:Display "Nope"
:Stop
:Else
:1→x
:While 1==1
:mod(x*cc,nn)→1
:If mod(x*cc,nn)==1
:Then
:Disp x
:Stop
:end
:end
```

1. Multiplication tables:

Z_6	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Z_7	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Notice that in Z_7 each non-zero element has an inverse.

2. The units in Z_{12} are those numbers relatively prime to 12: 1, 5, 7, 11. Hence we cannot divide by 2, 3, or 8. Be careful: in Z_{12} each unit is its own inverse, but in general this is **not** the case. If it were, then things would be much easier. $4/5 = 4\cdot 5 = 8$. $4/7 = 4\cdot 7 = 4$. $4/11 = 4\cdot 11 = 8$.

3. $ac \equiv bc \pmod n$ implies $n \mid (ac - bc)$; $n \mid c(a-b)$. Let $(c,n) = d.$; $n = dn'$; $c = dc'$. We have $dn' \mid dc'(a-b)$; $n' \mid c'(a-b)$. But $(n',c') = 1$; hence by Euclid's lemma $n' \mid (a-b)$.

4. $(9,12) = 3$. Since 3 does not divide 5, this problem has no solution.

5. $(9,12) = 3$. Since 3 does divide 6, this problem has a solution. We divide the equation by 3, to get $3x + 4y = 2$. Solving for y first (arbitrarily; we could do x first) we have $4y \equiv 2 \pmod 3$. Adding 3 to the right hand side twice, we get $4y \equiv 8 \pmod 3$. This

gives us $y \equiv 2 \pmod 3$. Hence, $y = 2 + 3t$ where t is any integer. We substitute this in $3x + 4y = 2$ to get $3x + 4(2 + 3t) = 2$. This yields $x = -2 -4t$. Again t is any integer, however, once t is chosen for one variable it must be the same value for the other variable.

6. $(45,50) = 5$. Since 5 does divide 20, this problem has a solution. Dividing the equation by 5, we get $9x + 10y = 4$. Solving for y, we have $10y \equiv 4 \pmod 9$. Adding 9 to the right hand side four times we get $10y \equiv 40 \pmod 9$. This gives us $y \equiv 4 \pmod 9$ or $y = 4 + 9t$. Substituting this in $9x + 10y = 4$, we get $9x + 10(4 + 9t) = 4$. This yields $x = -4 -10t$.