Arithmetic mod m

Have selected a modulus m, integer ≥ 2.

Define ① $\mathbb{Z}_m = \{0, 1, 2, 3, \ldots, m-1\}$

correspond to possible remainders from division alg., divisor is m.

② $+_m$: $a +_m b = (a+b) \bmod m$

$\cdot_m$: $a \cdot_m b = (ab) \bmod m$

$37 +_6 (4) \cdot_6 (7)$

$= 37 +_6 (28 \bmod 6)$

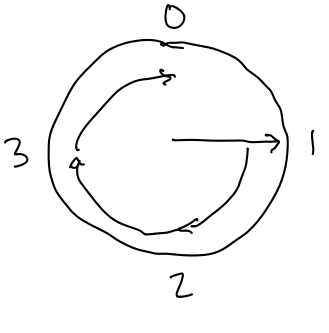
$= 37 +_6 4 = 41 \bmod 6$

$= 5$

Ex.] Work mod 4 $\qquad \mathbb{Z}_4 = \{0, 1, 2, 3\}$

$+_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\cdot_4$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1



$27 \cdot_4 (-5) \equiv (27 \bmod 4) \cdot_4 (-5 \bmod 4)$

$= 3 \cdot_4 3 = 1$

Solving congruence equations

Solve

① $3x + 1 \equiv 2 \pmod{4}$

Goal: Find all integers x such that

$4 \mid (3x+1-2)$

Add 3 to both sides

$$3x + 4 \equiv 5 \pmod{4}$$

② $3\underline{x} \equiv 1 \pmod{4}$

Undo mult. by 3: Multiply by 3's multiplicative inverse (3)

$$3 \cdot 3x \equiv 3 \cdot 1 \pmod{4}$$

$$\underline{9}x \equiv 3 \pmod{4}$$

$$\underline{x} \equiv 3 \pmod{4}$$

So any x equiv. to 3 mod 4 solves:

$$\boxed{\ldots, -5, -1, 3, 7, 11, 15, \ldots}$$

Euclidean Algorithm

Our main purpose in learning it: To find gcd(a, b) and ... (see below)

How it works:

1. Start with two (usually positive) integers a, b. Call the larger one $\overset{\text{dividend}}{r_0}$, the smaller $\overset{\text{divisor}}{r_1}$.
2. Iterate the division algorithm:
   - Divide r_0 by r_1—that is, find integers q_1 and r_2 (Note: $0 \leqslant r_2 < r_1$) so that

     $$r_0 = q_1 r_1 + \underset{\text{might be } 0}{r_2} \qquad r_0 = \underline{\quad}\, r_1 + \underset{\uparrow \atop r_2}{\underline{\quad}}$$

   - Shift roles r_1 into the former role of r_0, r_2 into the former role of r_1, and repeat, using the division algorithm to find q_2 and r_3. (Note: $0 \leqslant r_3 < r_2$.)

     $$r_1 = q_2 r_2 + r_3$$

   We continue to shift roles of the r_j and repeat. This process produces a strictly decreasing sequence

   $$r_0,\ r_1,\ r_2,\ \ldots,\ r_n$$

   until a remainder, call it r_{n+1}, finally is zero, which is our **stopping criterion**.

Example 1:

Perform the Euclidean algorithm with $a = \overset{r_1}{276}$, $b = \overset{r_0}{324}$. ■

$$\overset{r_0}{324} = \underline{1}\;\overset{r_1}{276} + \underline{\overset{r_2}{48}}$$

Note: Any integer divisor of two of $\{r_0, r_1, r_2\}$ divides all 3.

$$\overset{r_1}{276} = \underline{5}\,(\overset{r_2}{48}) + \underline{\overset{r_3}{36}}$$

Note: Any divisor of two of these nos. $\{r_1, r_2, r_3\}$ divides all three.

Coupled w/ previous: Any divisor of two of the nos. $\{r_0, r_1, r_2, r_3\}$ divides all.

$$\overset{r_2}{48} = \underline{1}\,(\overset{r_3}{36}) + \underline{\overset{r_4}{12}}$$

$$36 = \underline{3}\,(12) + \underset{\uparrow \atop \text{stop!}}{\underline{0}}$$

Observe: ① Any divisor of two in the list $\{r_0, r_1, \ldots, r_4\}$ divides all of them

② r_4 divides r_3 making it the $\gcd(r_3, r_4) = \gcd(r_0, r_1)$.

See website https://www.extendedeuclideanalgorithm.com/calculator.php.

Example 2:

Perform the Euclidean algorithm with $a = 4312$, $b = 585$.

■

$r_0 = 4312$, $r_1 = 585$

$$4312 = \underline{7}\,(585) + \underline{217}$$

$$585 = \underline{2}\,(217) + \underline{151}$$

$$217 = \underline{1}\,(151) + \underline{66}$$

$$151 = \underline{2}\,(66) + \underline{19}$$

$$66 = \underline{3}\,(19) + \underline{9}$$

$$19 = \underline{2}\,(9) + \underline{1}$$

$$9 = \underline{9}\,(1) + \underline{0}$$

Call the <u>last nonzero remainder</u> $r_n = \underline{1}$

for this pair of nos.

$$\gcd(4312, 585) = 1$$

We say 4312 and 585 are <u>relatively prime</u>.

Extended Euclidean Algorithm

Our main purpose in learning it:

Extend Euclidean alg. so as to write

$\gcd(a, b)$ as $\underline{\text{linear}}$ $\underline{\text{comb}}$ $ta + sb$

$$324 = 1(276) + 48 \qquad \boxed{48} = \boxed{324 - 276}$$

$$276 = 5(48) + 36 \implies 36 = 276 - 5(48)$$

$$\underline{48 = 36 + 12} \implies \boxed{12 = 48 - 36}$$

rewrite r_n in terms of r_{n-1}, r_{n-2}

insert my prescription for $r_{n-1} = 36$

$$\begin{aligned} 12 &= 48 - [276 - 5(48)] \\ &= 48 - 276 + 5(48) \\ \underline{12} &= 6(48) - 276 \\ &= 6(324 - 276) - 276 \\ &= 6(324) - 6(276) - 276 \\ &= 6(324) - 7(276) \end{aligned}$$

$$t \cdot a + s \cdot b$$

Definition 1: For any $n \geqslant 2$ define $\mathbb{Z}_n$ to be the set of integers $\{0, 1, 2, 3, \ldots, n-1\}$.

Solving congruences

Consider the function

$$f(x) \;=\; 7x + 4 \mod 12.$$

The implied domain of this function is the set of integers, and the codomain is the list of remainders $\{0, 1, 2, \ldots, 11\}$ which are possible when dividing an integer by 12. That is, the codomain is $\mathbb{Z}_{12}$. When we look to *solve* the (congruence) equation

$$7x + 4 \equiv 9 \pmod{12}, \tag{1}$$

we seek to describe those inputs x to f which produce the particular output 9.

From these facts

1. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$.
2. If $a \equiv b \pmod{m}$ and $c \in \mathbb{Z}$, then $ac \equiv bc \pmod{m}$.

we know that we can perform some of the basic steps of algebra. With regards to the example equation (1), we solve by

- adding 8 to both sides, which gets rid of the $+4$ since $(8 + 4) \mod 12 = 0$

$$\left.\begin{array}{ll} \text{new LHS :} & 7x + 4 + 8 \equiv 7x + 12 \equiv 7x \pmod{12} \\ \text{new RHS :} & 9 + 8 \equiv 17 \equiv 5 \pmod{12} \end{array}\right\} \quad \Rightarrow \quad 7x \equiv 1 \pmod{12}.$$

- multiplying both sides by 7, since $(7 \cdot 7) \mod 12 = 1$.

$$\left.\begin{array}{ll} \text{new LHS :} & 7 \cdot 7x \equiv 49x \equiv x \pmod{12} \\ \text{new RHS :} & 7 \cdot 5 \equiv 11 \pmod{12} \end{array}\right\} \quad \Rightarrow \quad x \equiv 11 \pmod{12}.$$

These two steps have led to the solution: the integers x which satisfy Equation (1) are those which are equivalent to 11 $\pmod{12}$.

The general linear congruence equation, with modulus $m \geqslant 2$, looks like

$$ax + b \equiv n \pmod{m}. \tag{2}$$

It can be solved in much the same way as above—adding $(-b)$, the **additive inverse** of b, to both sides, then multiplying by the **multiplicative inverse** of a—provided that that $\gcd(a, m) = 1$ (a sufficient condition for a to *have* a multiplicative inverse $\pmod{m}$).

For small integers a, m it is generally possible to figure out

- what the $\gcd(a, m)$ is, and
- when $\gcd(a, m) = 1$, which number $\bar{a} \in \mathbb{Z}_m$ is the multiplicative inverse—i.e., satisfies $a\bar{a} \equiv 1 \pmod{m}$.

When these cannot be determined so easily, we resort to the Euclidean and Extended Euclidean Algorithms.[1]

Example 3:

Show that the integers 311 and 6215 are relatively prime, and then find the multiplicative inverse of $311 \pmod{6215}$.

Answer: We perform steps of the Euclidean algorithm (left side) and, rewrite (right side) the equations to express the newest remainder r_j in terms of two prior ones r_{j-1} and r_{j-2} (helpful steps for the extended Euclidean algorithm):

$$\begin{array}{lcl} 6215 = (19)(311) + 306 & \Rightarrow & 306 = (1)(6215) - (19)(311) \\ 311 = (1)(306) + 5 & & 5 = (1)(311) - (1)(306) \\ 306 = (61)(5) + 1 & & 1 = (1)(306) - (61)(5) \\ 5 = (5)(1) + 0 & & \end{array}$$

The last nonzero remainder is $\gcd(6215, 311)$, and since it is 1, the two numbers are relatively prime.

To find the multiplicative inverse, we use the equations, starting at the bottom, on the right-hand side, continually substituting the next-higher equation:

$$\begin{aligned} 1 &= (1)(306) - (61)(5) \qquad \text{(bottom equation on the right)} \\ &= (1)(306) - (61)[(1)(311) - (1)(306)] \;=\; (62)(306) - (61)(311) \\ &= (62)[(1)(6215) - (19)(311)] - (61)(311) \;=\; (62)(6215) - (1239)(311) \\ &= (62)(6215) + (-1239)(311) \end{aligned}$$

If we consider the operations above as happening $\pmod{6215}$, we have

$$(62)(6215) + (-1239)(311) \equiv (4976)(311) \equiv 1 \pmod{6215}.$$

Thus, 4976 is the multiplicative inverse $\pmod{6215}$ of 311.

■

Example 4: Affine ciphers

[1] An app that implements both the Euclidean and Extended Euclidean Algorithms is linked to the class webpage. The direct url is https://www.extendedeuclideanalgorithm.com/calculator.php.

Affine ciphers are based on functions of the form

$$f(x) := ax + b \mod 26.$$

When $\gcd(a, 26) = 1$, such a function $f\colon \mathbb{Z}_{26} \to \mathbb{Z}_{26}$ is *bijective* and is, thus, *invertible*.

One equates the 26 letters of the English alphabet with the numbers 0–25: $a \leftrightarrow 0$, $b \leftrightarrow 1, \ldots,$ $z \leftrightarrow 25$. This makes a natural map between simple strings of letters and finite sequences of integers, such as

$$\text{the word } \textit{pencil} \quad \leftrightarrow \quad 15, 4, 13, 2, 8, 11,$$

which, in its numerical equivalent, could hardly be said to be "encrypted." However, if we let $y = f(x)$, then an encrypted version of $x_1, x_2, x_3, x_4, x_5, x_6$ would be $y_1, y_2, y_3, y_4, y_5, y_6$, with

$$\begin{aligned}
x_1 &= 15: & y_1 &= f(15) = 15a + b \mod 26,\\
x_2 &= 4: & y_2 &= f(4) = 4a + b \mod 26,\\
x_3 &= 13: & y_3 &= f(13) = 13a + b \mod 26,\\
x_4 &= 2: & y_4 &= f(2) = 2a + b \mod 26,\\
x_5 &= 8: & y_5 &= f(8) = 8a + b \mod 26,\\
x_6 &= 11: & y_6 &= f(11) = 11a + b \mod 26.
\end{aligned}$$

In the case where $a = 19$ and $b = 4$, these encrypted values would be $3, 2, 17, 16, 0, 5$, though we would generally transmit this as its alphabetic equivalent *dcrqaf*.

The person on the receiving end needs the inverse function to f in order to perform decryption of the message. We can obtain it in the same manner described above—by solving the congruence equation $ax + b \equiv y \pmod{26}$. If $\bar{a}$ is the multiplicative inverse of $a \pmod{26}$, then

$$ax + b \equiv y \pmod{26} \qquad \Rightarrow \qquad x \equiv \bar{a}(y - b) \pmod{26}.$$

One needs $\gcd(a, 26) = 1$, of course, so that $\bar{a}$ exists, in which case $g(y) = \bar{a}(y - b) \mod 26$ is the inverse function to $f(x) = ax + b \mod 26$.

You can explore affine ciphers without the tedium of all the letter-to-number conversions at http://www.calvin.edu/~scofield/courses/m100/materials/scriptForms/affineTranslator.shtml a link which appears on the class webpage. While affine ciphers are a neat application of congruences, they are quite easily broken.

■

Systems of linear congruences

A **system of congruences** is nothing more than multiple individual congruences, with the requirement that any solution x must be an integer which simultaneously satisfies them all. First, an example.

Example 5:

Find integer solutions x which simultaneously solve the linear congruences

$$3x + 2 \equiv 4 \pmod{7} \qquad \text{and} \qquad 2x + 3 \equiv 2 \pmod{9}.$$

Answer: Using the techniques described in the previous section, we can solve the first of these congruences,[2] obtaining

$$x \equiv 3 \pmod{7}.$$

The integers which satisfy this requirement come from the list

$$\ldots, -39, -32, -25, -18, -11, -4, 3, 10, 17, 24, 31, 38, 45, \ldots,$$

and take the form $x = 7s + 3$, $s \in \mathbb{Z}$. That $x \equiv 3 \pmod{7}$, or $7 \mid (x - 3)$, stands as one requirement, the one imposed by the first congruence equation, on our solutions x. So, we insert the form $x = 7s + 3$, $s \in \mathbb{Z}$. into the second congruence equation:

$$2x + 3 \equiv 2(7s + 3) + 3 \equiv 2 \pmod{9} \qquad \text{and solve to obtain} \qquad s \equiv 4 \pmod{9},$$

which means $9 \mid (s - 4)$, or $s = 9t + 4$, $t \in \mathbb{Z}$. Putting these statements, together,

$$x \;=\; 7s + 3 \;=\; 7(9t + 4) + 3 \;=\; 63t + 31, \qquad t \in \mathbb{Z},$$

which is the same as saying the solutions to this system of two linear congruences are precisely those x which satisfy

$$x \equiv 31 \pmod{63}.$$

One can check that these x, ones in the list

$$\ldots, -158, -95, -32, 31, 94, \ldots$$

are all in the list of solutions to $3x + 2 \equiv 4 \pmod{7}$, but also satisfy the other congruence relation $2x + 3 \equiv 2 \pmod{9}$.

■

We solved this system of linear congruence equations by a method Rosen calls **back substitution**. Specifically, we

- solved the first congruence, putting it in the form $x \equiv b \pmod{m}$.
- employed the fact that the integers which satisfy $x \equiv b \pmod{m}$ all take the form $x = ms + b$, where s is an integer.

[2] The key is that there be multiplicative inverse to 3 in mod 7 arithmetic, which is so, since $\gcd(3, 7) = 1$.

- (back-)substituted the expression $ms + b$ in for x in the next congruence equation. One can repeatedly solve one congruence to get an expression that is inserted into the next one.

Is it possible to solve any system consisting of an arbitrary list of linear congruence relations in this manner? Indeed, not, when the system doesn't even have solutions, like in these scenarios:

(i) Suppose one of your congruences were

$$2x \equiv 1 \pmod{4}.$$

Even with no knowledge of the other congruences in the system, this one fouls things up by itself, since it has no (integer) solutions.

(ii) Suppose two of your congruences were

$$x \equiv 5 \pmod{7} \qquad \text{and} \qquad x \equiv 11 \pmod{14}.$$

The two lists of numbers these generate,

$$\ldots, -16, -9, -2, 5, 12, 19, \ldots$$

and

$$\ldots, -17, -3, 11, 25, 39, \ldots$$

have no overlaps—no common entries in both lists.

In these two scenarios, there are no solutions, so back-substitution will fail.

Is it possible to categorize situations in which solutions exist? That is one role the next theorem plays.

Theorem 1 (Chinese Remainder Theorem): Suppose the moduli m_1, m_2, ..., m_n in the system of congruences

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_n \pmod{m_n} \end{aligned} \tag{3}$$

are pairwise relatively prime. Then the system has solutions which can be characterized by $x \equiv A \pmod{m}$, with $m := m_1 m_2 \cdots m_n$, for some $A \in \{0, 1, 2, \ldots, m-1\}$.

Had we known this theorem prior to doing Example 3, we would have known the system was solvable because

- both congruences could be put into the form $x \equiv b_i \pmod{m_i}$. This required the existence of multiplicative inverses to 3 in $\bmod 7$ arithmetic and to 2 in $\bmod 9$ arithmetic. And
- the moduli, 7 and 9, were relatively prime.

We still had to do some work to figure out that $A = 31$ (for that example). If, however, two of our congruences have moduli $m_1 = 7$ and $m_2 = 14$ (as in Scenario (ii) above), then we have no assurance of a solution, since $\gcd(7, 14) \neq 1$.

An alternative to back-substitution. Often the proof of a mathematical fact is constructive, providing an algorithm. In particular, the proof of the Chinese Remainder Theorem offers an alternative to back-substitution for solving a system of linear congruences already in the form (3). Under the assumptions of the theorem, if we use the moduli $m_1, \ldots, m_n$ to form the related list of n numbers

$$M_i = \frac{m_1 m_2 \cdots m_n}{m_i} = \frac{m}{m_i}, \qquad k = 1, \ldots, n,$$

(i.e, M_1 is the product of the moduli leaving m_1 out, M_2 is the product of moduli leaving m_2 out, etc.), then for each $1 \leqslant i \leqslant n$, the numbers m_i and M_i are relatively prime. This means that M_i has a multiplicative inverse, call it y_i, in $\pmod{m_i}$ arithmetic. Having found these multiplicative inverses so that $M_i y_i \equiv 1 \pmod{m_i}$, $1 \leqslant i \leqslant n$, we may take

$$A = \sum_{i=1}^{n} a_i M_i y_i \mod m_1 m_2 \cdots m_n = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots a_n M_n y_n \mod m.$$

The solutions, then, of the system (3) are those $x \equiv A \pmod{m}$.

See Example 5 in the Rosen text, pp. 278–279 to see this method used to solve the system of congruences posed in Example 4, p. 277.

Representations of (large) integers and the Chinese Remainder Theorem

The integers 0–255 are all representable as 8-bit binary strings:

n	8-bit binary representation	n	8-bit binary representation
0	00000000	127	01111110
1	00000001	⋮	⋮
2	00000010	252	11111100
3	00000011	253	11111101
4	00000100	254	11111110
⋮	⋮	255	11111111

The Chinese Remainder Theorem indicates that the system of congruences (3) offers an *alternative way* of representing uniquely all integers $k \in \mathbb{Z}_m$ (still taking $m = m_1 m_2 \cdots m_n$)—namely, we represent k as the n-tuple $(k \mod m_1, k \mod m_2, \ldots, k \mod m_n)$. With the understanding that the tuple $(32, 18, 91)$ is the representation of some number k with $0 \leqslant k < (97)(98)(99) = 941,094$ using the three moduli $m_1 = 97$, $m_2 = 98$, $m_3 = 99$ (pairwise relatively prime), we could recover k by solving the system

$$x \equiv 32 \pmod{97}, \qquad x \equiv 18 \pmod{98}, \qquad x \equiv 91 \pmod{99}.$$

See Example 7, p. 279 for the representations of numbers in $\mathbb{Z}_{12}$ (i.e., 0–11) when the two relatively prime moduli $m_1 = 3$ and $m_2 = 4$ are used.