

Modular Arithmetic

Victor Adamchik

Fall of 2005

■ Plan

1. Review
2. Applications of Modular Arithmetic
3. Solving Linear Congruences
4. Chinese Remainder Theorem
5. Arithmetic with Large Integers

■ Review

Definition. $a \bmod n$ means the remainder when a is divided by n

$$a = qn + r$$

Definition. (modulo equivalence)

$$a \equiv b \pmod{n} \text{ if and only if } n \mid (a - b)$$

We will say that a and b are equivalent modulo n . We will also write modulo equivalence as

$$a \equiv_n b$$

Theorem. \equiv_n is an [equivalence relation](#) on the integers.

An [equivalence class](#) consists of those integers which have the same remainder on division by n . The equivalence classes are also known as [congruence classes](#) modulo n . Rather than say the integers a and b are equivalent we say that they are congruent modulo n .

Definition. The set of all integers congruent to a modulo n is called the [residue class](#) $[a]$.

Example. Residue classes mod 3:

$$[0] = \{\dots, -6, -3, 0, 3, 6, \dots\}$$

$$[1] = \{\dots, -5, -2, 1, 4, 7, \dots\}$$

$$[2] = \{\dots, -4, -1, 2, 5, 8, \dots\}$$

Definition. The set of residue classes is denoted by Z_n

$$Z_n = \{0, 1, 2, \dots, n-1\}$$

Theorem (arithmetic on Z_n) When we are doing +, - or * modulo n , we can replace a number by another number in the same residue class.

Example. Compute

$$414 * 463 \text{ mod } 413$$

$$1 * 50 = 50$$

Note (cancelation property). Though it seems that arithmetic on Z_n is the same as on Z , do not be deceived. The product of two non-zero elements of Z_n can sometimes be 0.

Question. Is the following implication correct?

$$x a \equiv_n x b \implies a \equiv_n b$$

Only if $\text{GCD}(x, n) = 1$.

Proof.

$$x a \equiv_n x b \implies n \mid (x a - x b) \implies n \mid x(a - b) \implies n \mid (a - b) \implies a \equiv_n b$$

Definition.

$$Z_n^* = \{x \in Z_n \mid \text{GCD}(x, n) = 1\}$$

Example.

$$Z_0 = \{0, 1, 2, 3, 4, 5\}$$

$$Z_6^* = \{1, 5\}$$

Advantage of Z_n^* is that it has a cancelation property.

Definition. Euler's phi function (or totient function) is the size of Z_n^*

$$\phi(n) = |Z_n^*|$$

$\phi(n)$ is the number of integers $1 \leq k < n$ coprime to n .

Example.

$$\phi(12) = |\{1, 5, 7, 11\}| = 4$$

It's easy to see that if p is prime then $\phi(p) = p - 1$. More on this function later.

Note. A number's **multiplicative inverse** x^{-1} defined by

$$x * x^{-1} = 1$$

is another difference between Z_n and Z . In Z only two elements have inverses: 1 and -1. We have proved that if p is prime then each element in Z_p has an inverse. We say that Z_p is a finite **field**, a set in which we can $+$, $-$, $*$, $/$. The prove is based on the following theorem

Theorem (Ferma's little theorem) If p is prime and $p \nmid a$, then

$$a^{p-1} \equiv_p 1$$

According to this theorem, the inverse is defined by (assuming p is prime)

$$a^{-1} = a^{p-2}$$

■ Applications of Modular Arithmetic

Problem 1. How do we efficiently store people's records?

If we use Social Security number as the key, we will have to deal with an array of size 10^{10} .

The solution - **hashing**. Store records in the table at index $h(k)$ defined by

$$h(k) = k \pmod{N}$$

Here, h is a hash function, and N is an array size. Regardless of chosen N , collisions might happen

$$h(k_1) = h(k_2)$$

You will deal with collisions in 15-211.

Problem 2. How do we generate a random number?

There is nothing random about `random()` or `rand()` function. The standard technique is to create a [pseudorandom sequence](#). The most commonly used procedure for generating pseudorandom numbers is the *linear congruential method* (Lehmer, 1949). We generate a sequence of pseudorandom numbers x_k , by successively using the congruence

$$x_{n+1} = a * x_n + b \pmod{m}, \quad n > 0$$

$$x_0 < m \quad \text{is the seed}$$

with appropriate values of $0 < a < m$, $0 \leq b < m$, $m > 0$ and $\text{GCD}(b, m) = 1$

For example,

$$x_{n+1} = 7 * x_n + 4 \pmod{9}, \quad n > 0$$

$$x_0 = 3$$

This generates the following sequence

$$x_1 = (7 * x_0 + 4) = (7 * 3 + 4) = 25 = 7 \pmod{9}$$

$$x_2 = (7 * x_1 + 4) = (7 * 7 + 4) = 53 = 8 \pmod{9}$$

$$x_3 = (7 * x_2 + 4) = (7 * 8 + 4) = 60 = 6 \pmod{9}$$

and so on

This recurrence is eventually periodic. So, it is desirable to use a recurrence with the long period.

Problem 3. ISBN numbers for published books.

The ISBN number for our textbook is 0-13-184-868-2.

The information is decoded in the first 9 digits. The last digit is for parity check

$$1 * a_1 + 2 * a_2 + \dots + 9 * a_9 = a_{10} \pmod{11}$$

Applying this to our textbook

$$1 * 0 + 2 * 1 + 3 * 3 + 4 * 1 + 5 * 8 + 6 * 4 + 7 * 8 + 8 * 6 + 9 * 8 = 255 = 2 \pmod{11}$$

The sum of 9 digits is 255 and it equals to the last digit mod 11.

Problem 4. The Gauss' Easter formula.

The church set the Easter on first Sunday after the first full moon in spring. The Easter sunday always moves in the period of 22nd March until 25th April. F. Gauss developed the exact formula to calculate the date of Easter. Here is its short version

$$\begin{aligned} a &= \text{year} \pmod{11} \\ b &= \text{year} \pmod{4} \\ c &= \text{year} \pmod{7} \\ d &= 19a + M \pmod{30} \\ e &= 2b + 4c + 6d + N \pmod{7} \end{aligned}$$

where constants M and N are

$$\begin{aligned} M &= 24 \\ N &= 5 \end{aligned}$$

valid throughout the period between 1900 and 2099 years. Then Easter falls on $(22 + d + e)$ March or $(d + e - 9)$ April.

```
year = 2006;
a = Mod[year, 19]; b = Mod[year, 4]; c = Mod[year, 7];
M = 24; n = 5; (* 1900 - 2099 *)
d = Mod[19 a + M, 30];
e = Mod[2 b + 4 c + 6 d + n, 7];
{22 + d + e (*march*), d + e - 9 (*april*)}

{47, 16}
```

In 2006 the catholic church will celebrate Easter on April 16. Dates are according the Gregorian calendar.

Here is the full version of the algorithm (see <http://www.smart.net/~mmontes/nature1876.html>), published in *Nature*, 1876 April 20, vol. 13, p. 487.

```

year = 2006;
a = Mod[year, 19];
b = IntegerPart[year / 100];
c = Mod[year, 100];
d = IntegerPart[b / 4];
e = Mod[b, 4];
f = IntegerPart[(b + 8) / 25];
g = IntegerPart[(b - f + 1) / 3];
h = Mod[19 * a + b - d - g + 15, 30];
i = IntegerPart[c / 4];
k = Mod[c, 4];
l = Mod[32 + 2 * e + 2 * i - h - k, 7];
m = IntegerPart[(a + 11 * h + 22 * l) / 451];
EasterMonth = IntegerPart[(h + l - 7 * m + 114) / 31]; (*3=March,4=April*)
p = Mod[h + l - 7 * m + 114, 31];
EasterDate = p + 1;
{EasterDate, EasterMonth}

{16, 4}

```

■ Solving Linear Congruences.

Definition. A relation of the form

$$a * x \equiv b \pmod{n}$$

is called a [linear congruence](#). We will also write this as

$$a * x \equiv_n b$$

How many solutions does it have? It's clear that if x_0 is a solution then every element from a congruent class is also a solution.

Example.

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

Here are solutions

$$\dots, -1, 3, 7, 10, \dots$$

Therefore, uniqueness can only be mod n .

$$x = 3 + 4k, \text{ where } k \in \mathbb{Z}$$

Example.

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

The congruence is satisfied for two representatives

$$x = 1 \text{ and } x = 3$$

Therefore, the equations has two solutions

$$x = 1 + 4k \text{ and } x = 3 + 4k, \text{ where } k \in \mathbb{Z}$$

Theorem. *The linear congruence*

$$a * x \equiv b \pmod{n}$$

has a unique solution iff $\text{GCD}(a, n) = 1$.

Proof. \Leftarrow)

First we prove that the equation has a solution. To solve the equation means to find the inverse of a

$$x \equiv b * a^{-1} \pmod{n}$$

But as we know the inverse is not necessarily exists in \mathbb{Z}_n . Therefore, some restrictions required. In the next paragraph we show how to find an inverse using the Extended Euclidean Algorithm.

Given $\text{GCD}(a, n) = 1$, it follows by the EEA that $\exists s, r \in \mathbb{Z}$

$$a * r + n * s = 1$$

Consider this equation mod n

$$a * r \equiv 1 \pmod{n}$$

This means that r is the multiplicative inverse of a

$$r \equiv a^{-1} \pmod{n}$$

Therefore, the solution is

$$x \equiv b * r \pmod{n}$$

Next, we prove uniqueness. We assume that there are two solutions X and Y .

$$\left. \begin{array}{l} a * X \equiv b \pmod{n} \\ a * Y \equiv b \pmod{n} \end{array} \right\} \implies a * X \equiv a * Y \pmod{n} \implies$$

$$\implies n \mid a(X - Y) \implies n \mid (X - Y) \implies X \equiv Y \pmod{n}$$

The proof in \implies) direction is left as an exercise.

QED.

■ Chinese Remainder Theorem.

Suppose we want to solve a system of linear congruences

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 2 \pmod{7} \end{cases}$$

Does a solution exist? Is it unique?

Let us build a solution. We solve the first equation $x \equiv 2 \pmod{3}$

$$y_1 \equiv 2 \pmod{3}$$

assuming that the solution is 0 modulo 5 and 7

$$y_1 \equiv 0 \pmod{5}$$

$$y_1 \equiv 0 \pmod{7}$$

The straightforward answer is defined by

$$5 * 7 * z \equiv 2 \pmod{3}$$

Therefore,

$$z = 1 \implies y_1 = 35$$

Next we solve the second equation $x \equiv 3 \pmod{5}$, again assuming that the solution is 0 modulo 3 and 7.

$$3 * 7 * z \equiv 3 \pmod{5}$$

It follows

$$z = 3 \implies y_2 = 63$$

In the same manner, solving the last equation $x \equiv 2 \pmod{7}$ yields

$$3 * 5 * z \equiv 2 \pmod{7}$$

$$z = 2 \implies y_3 = 30$$

What is the general solution?

$$y_1 + y_2 + y_3 = 35 + 63 + 30 = 128$$

Clearly, this is the solution modulo $3*5*7$

$$x \equiv 128 \pmod{105}$$

In the parametric form

$$x = 23 + 105 * k, \quad k \in \mathbb{Z}$$

Theorem. (Chinese Remainder Theorem) Let $m_1, m_2, \dots, m_n \in \mathbb{Z}^+$ be pairwise relatively prime. The system

$$x \equiv a_k \pmod{m_k}, \quad k = 1, 2, \dots, n$$

has a unique solution modulo $M = m_1 * m_2 * \dots * m_n$.

Proof.

First we prove that the system has a solution. Proceeding as in the above example, we define solution to each equation as

$$y_k \equiv a_k \pmod{m_k}$$

$$y_k \equiv 0 \pmod{m_j}, \quad k \neq j$$

Combining them together yields the equation

$$\frac{M}{m_k} z \equiv a_k \pmod{m_k}$$

which definitely has a solution (why-?) with respect to z . Thus

$$y_k = \frac{M}{m_k} z$$

Next we define the general solution by

$$y_1 + y_2 + \dots + y_n$$

Clearly this is a solution to a given system by modulo M .

Uniqueness.

Suppose there are two solutions, say X , and Y . Then we would have

$$X \equiv_{m_k} Y \quad \text{for all } k$$

Therefore, they are the same modulo M

$$X \equiv_M Y$$

■ Arithmetic with Large Integers

Problem. Compute $7!$ without ever encountering a number larger than 2^8 .

First, we find moduli m_k relatively prime and $m_1 m_2 \dots m_n > 7!$

We choose 13, 11, 9 and 7.

Next we do all computations using these moduli

$$2 * 5 = \{10, 10, 1, 3\}$$

$$3 * 4 = \{12, 1, 3, 5\}$$

$$5 != \{120, 10, 3, 15\} = \{3, 10, 3, 1\}$$

$$6 != \{18, 60, 18, 6\} = \{5, 5, 0, 6\}$$

$$7 != \{35, 35, 0, 42\} = \{9, 2, 0, 0\}$$

The last step we recover $7!$ by applying the CRT

$$\begin{cases} x \equiv 9 \pmod{13} \\ x \equiv 2 \pmod{11} \\ x \equiv 0 \pmod{9} \\ x \equiv 0 \pmod{7} \end{cases}$$

We find such a number x that $0 < x < 13 * 11 * 9 * 7$.

Note. Solving the above system, you should not generate numbers bigger than 2^8 .

Here we outline another method of solving the system of congruences. We denote the list of moduli and the list of remainders by

$$m = \{13, 11, 9, 7\}$$

$$r = \{9, 2, 0, 0\}$$

respectively. On the first step, we find the inverses of each modulo with respect to each later modulo in the list. This can be done by the extended Euclidean algorithm (see the proof of the theorem for solving linear congruence.)

$$13^{-1} \equiv 6 \pmod{11}, \quad 13^{-1} \equiv 7 \pmod{9}, \quad 13^{-1} \equiv 6 \pmod{7}$$

$$11^{-1} \equiv 5 \pmod{9}, \quad 11^{-1} \equiv 2 \pmod{7}$$

$$9^{-1} \equiv 4 \pmod{7}$$

Let us denote the list of inverse (padded with zeros on the left) by

$$\text{inv} = \{\{0, 6, 7, 6\}, \{0, 0, 5, 2\}, \{0, 0, 0, 4\}\};$$

Then

$$\text{inv}_{j,k} * m_j \equiv 1 \pmod{m_k}$$

Next, starting with a given list of remainders we do

```

for(j = 1, j < n, j++,
  for(k = j+1, k <= n, k++,
    r[k] = (r[k]-r[j])*inv[j][k]) % m[k];
  ])

```

Finally, the solution is built up by

$$x = \sum_{k=1}^R \text{rem}_k \prod_{j=1}^{k-1} m_j$$

One can check that $0 \leq x < M = m_1 m_2 \dots, m_n$. In case of four moduli the solution is

$$x \equiv r_1 + m_1 v_1 + m_1 m_2 v_2 + m_1 m_2 m_3 v_3 \pmod{M}$$

where

$$v_1 = (r_2 - r_1) \text{inv}_{1,2} \pmod{m_2}$$

$$v_2 = ((r_3 - r_1) \text{inv}_{1,3} - v_1) \text{inv}_{2,3} \pmod{m_3}$$

$$v_3 = (((r_4 - r_1) \text{inv}_{1,4} - v_1) \text{inv}_{2,4} - v_2) \text{inv}_{3,4} \pmod{m_4}$$

To verify that this is indeed a solution, we compute $x \pmod{m_k}$, $k = 1, 2, 3, 4$. Obviously

$$x \equiv r_1 \pmod{m_1}$$

since the each term of x is divisible by m_1 except the first one. Next, we compute $x \pmod{m_2}$. Applying $\pmod{m_2}$, we obtain

$$x \equiv r_1 + m_1 v_1 \pmod{m_2}$$

Replacing v_1

$$x \equiv r_1 + (r_2 - r_1) m_1 \text{inv}_{1,2} \pmod{m_2}$$

and using

$$m_1 * \text{inv}_{1,2} \equiv 1 \pmod{m_2}$$

yeilds

$$x \equiv r_2 \pmod{m_2}$$

In the same manner we prove other two cases.

Here is the *Mathematica* code

```

n = 4;
m = {13, 11, 9, 7};
rem = {9, 2, 0, 0};
inv = {{0, 6, 7, 6}, {0, 0, 5, 2}, {0, 0, 0, 4}};
For[j = 1, j < n, j++,
  For[k = j + 1, k ≤ n, k++,
    rem[[k]] = Mod[(rem[[k]] - rem[[j]]) inv[[j, k]], m[[k]] ]
  ]];
Sum[rem[[k]] Product[m[[j]], {j, 1, k - 1}],
{k, 1, n}]
5040

```

Generally, choosing the following five relatively prime numbers as moduli

$$\{2^{35} - 1, 2^{34} - 1, 2^{33} - 1, 2^{29} - 1, 2^{23} - 1\}$$

we can add and multiply positive integers up to the size of 2^{154}

$$M = \prod_{k=1}^5 m_k > 2^{154}$$

Compare this with the size of the largest integer $2^{32} - 1$ on a 32-bit CPU.