Solving Congruences

Section 4.4

Linear Congruences

Definition: A congruence of the form
$$ax \equiv b \pmod{m},$$
where m is a positive integer, a and b are integers, and x is a variable, is called a *linear congruence*.

- The solutions to a linear congruence $ax \equiv b \pmod{m}$ are all integers x that satisfy the congruence.

Definition: An integer $\bar{a}$ such that $\bar{a}a \equiv 1 \pmod{m}$ is said to be an *inverse* of a modulo m.

Example: 5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 \pmod 7$

- One method of solving linear congruences makes use of an inverse $\bar{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\bar{a}$ to solve for x. Indeed,

$$ax \equiv b \pmod{m} \quad \rightarrow \quad \bar{a}ax \equiv \bar{a}b \pmod{m} \quad \rightarrow \quad x \equiv \bar{a}b \pmod{m}$$

Th. 5 (sec 4.1)   since $\bar{a}a = 1 + km$ (Th. 4 in sec 4.1)

Inverse of *a* modulo *m*

- The following theorem guarantees that an inverse of *a* modulo *m* exists whenever *a* and *m* are relatively prime, that is when $\gcd(a,m) = 1$.

Theorem 1: If *a* and *m* are relatively prime integers and $m > 1$, then an inverse of *a* modulo *m* exists. Furthermore, this inverse is unique modulo *m* (that is, there is a unique positive integer $\bar{a}$ less than *m* that is an inverse of *a* modulo *m* and every other inverse of *a* modulo *m* is congruent to $\bar{a}$ modulo *m*).

Proof: Since $\gcd(a,m) = 1$, by Bezout's Theorem, there are integers *s* and *t* such that $sa + tm = 1$.

- Hence, $tm = 1 - sa$.
- Therefore, *m* divides $1 - sa$
- According to the definition of congruence, $sa \equiv 1 \pmod{m}$
- Consequently, *s* is an inverse of *a* modulo *m*.
- The uniqueness of the inverse is Exercise 7.

◀

Finding Inverses

- The (extended) Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

  Example: Find an inverse of 3 modulo 7.

  Solution: Because gcd(3,7) = 1, by Theorem 1, an inverse of 3 modulo 7 exists.

  - Using the Euclidian algorithm: 7 = 2·3 + 1.
  - From this equation, we get −2·3 + 1·7 = 1.
    (That is, −2 and 1 are Bézout coefficients of 3 and 7.)
  - Hence, −2·3 ≡ 1 (mod 7) and −2 is an inverse of 3 modulo 7.
  - Also every integer congruent to −2 modulo 7 is an inverse of 3 modulo 7, i.e., 5, −9, 12, etc.

Finding Inverses

Example: Find an inverse of 101 modulo 4620.

Solution: First use the Euclidian algorithm to show that gcd(101,4620) = 1.

Working Backwards to find Bézout coefficients

$4620 = 45 \cdot 101 + 75$
$101 = 1 \cdot 75 + 26$
$75 = 2 \cdot 26 + 23$
$26 = 1 \cdot 23 + 3$
$23 = 7 \cdot 3 + 2$
$3 = 1 \cdot 2 + 1$
$2 = 2 \cdot 1$

$1 = 3 - 1 \cdot 2$
$1 = 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3$
$1 = -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23$
$1 = 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = 26 \cdot 26 - 9 \cdot 75$
$1 = 26 \cdot (101 - 1 \cdot 75) - 9 \cdot 75$
$\qquad = 26 \cdot 101 - 35 \cdot 75$
$1 = 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101)$
$\qquad = -35 \cdot 4620 + 1601 \cdot 101$

Since the last nonzero remainder is 1,
gcd(101,4260) = 1

Bézout coefficients for 4620 and 101 are:
-35 and 1601

1601 is an inverse of 101 modulo 4620

Also, -35 is an inverse of 4620 modulo 101

Using Inverses to Solve Congruences

- We can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

 Example: What are the solutions of the congruence $3x \equiv 4 \pmod 7$?

 Solution: First, $\gcd(3,7) = 1$ and we found that -2 is an inverse of 3 modulo 7 (two slides back). We multiply both sides of the congruence by -2 giving

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod 7.$$

 Because $-6 \equiv 1 \pmod 7$, it follows that if x is a solution then

$$x \equiv -8 \pmod 7 \quad \text{or} \quad \boxed{x \equiv 6 \pmod 7} \qquad \text{since } 6 \equiv -8 \pmod 7$$

 To verify this solution, assume arbitrary x s.t. $x \equiv 6 \pmod 7$. By Theorem 5 of Section 4.1, it follows that $3x \equiv 3 \cdot 6 \equiv 18 \equiv 4 \pmod 7$ which shows that all such x satisfy the congruence above.

 The solutions are the integers x such that $x \equiv 6 \pmod 7$, namely, $6, 13, 20 \ldots$ and $-1, -8, -15 \ldots$

System of Linear Congruences

- *The Chinese Remainder Theorem: let $m_1, m_2, ..., m_n$ be pairwise relatively prime integers greater than one and $a_1, a_2, ..., a_n$ be arbitrary integers. Then, system*

$$x \equiv a_1 \ (\mathbf{mod}\ m_1)$$

$$x \equiv a_2 \ (\mathbf{mod}\ m_2)$$

$$......$$

$$x \equiv a_n \ (\mathbf{mod}\ m_n)$$

has a unique solution modulo $m = m_1 \cdot m_2 \cdot \cdot \cdot m_n$

easy to solve (see text)

Nonlinear Congruences

- *Discrete logarithm* of y modulo p to the base r: find all x such that $r^x \equiv y \pmod{p}$

hard to solve

Applications of Congruences

Section 4.5

Section Summary

- Hashing Functions
- Pseudorandom Numbers
- Check Digits

Hashing Functions

Definition: A *hashing function h* assigns memory location $h(k)$ to the record that has k as its key.

- A common hashing function is $h(k) = k \textbf{ mod } m$, where m is the number of memory locations.
- Because this hashing function is onto, all memory locations are possible.

Example: Let $h(k) = k \textbf{ mod } 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$h(064212848) = 064212848 \textbf{ mod } 111 = 14$
$h(037149212) = 037149212 \textbf{ mod } 111 = 65$

$h(107405723) = 107405723 \textbf{ mod } 111 = 14$, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

- The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we say a *collision* occurs. Here a collision has been resolved by assigning the record to the first free location.

- For collision resolution, we can use a *linear probing function*:
  $$h(k,i) = (h(k) + i) \textbf{ mod } m, \text{ where } i \text{ runs from 0 to } m - 1.$$
- There are many other methods of handling with collisions (later CS course).

Pseudorandom Numbers

- Randomly chosen numbers are needed for many purposes, including computer simulations.

- *Pseudorandom numbers* are not truly random since they are generated by systematic methods.

- The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.

- Four integers are needed: the *modulus m*, the *multiplier a*, the *increment c*, and *seed x_0*, with $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$.

- We generate a sequence of pseudorandom numbers $\{x_n\}$ with $0 \leq x_n < m$ for all n, by successively using the recursive function

$$x_{n+1} = (ax_n + c) \bmod m.$$

Pseudorandom Numbers

- **Example**: Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.
- **Solution**: Compute the terms of the sequence by successively using the congruence
$$x_{n+1} = (7x_n + 4) \bmod 9 \qquad \text{with } x_0 = 3.$$

$$x_1 = 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7,$$
$$x_2 = 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8,$$
$$x_3 = 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6,$$
$$x_4 = 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1,$$
$$x_5 = 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2,$$
$$x_6 = 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0,$$
$$x_7 = 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = 4 \bmod 9 = 4,$$
$$x_8 = 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5,$$
$$x_9 = 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3.$$

The sequence generated is 3,7,8,6,1,2,0,4,5,3,7,8,6,1,2,0,4,5,3,…
It repeats after generating 9 terms.

- Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator*. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16,807$ generates $2^{31} - 2$ numbers before repeating.

Check Digits: UPCs

- A common method of detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function. If the final digit is not correct, then the string is assumed not to be correct.

  Example: Retail products are identified by their *Universal Product Codes* (*UPCs*). Usually these have 12 decimal digits, the last one being the check digit. The check digit is determined by the congruence:

  $$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}.$$

  a. Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?
  b. Is 041331021641 a valid UPC?

Solution:

a. $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$
    $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$
    $98 + x_{12} \equiv 0 \pmod{10}$
    $x_{12} \equiv 0 \pmod{10}$ So, the check digit is 2.

b. $3 \cdot 0 + 4 + 3 \cdot 1 + 3 + 3 \cdot 3 + 1 + 3 \cdot 0 + 2 + 3 \cdot 1 + 6 + 3 \cdot 4 + 1 \equiv 0 \pmod{10}$
    $0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \not\equiv 0 \pmod{10}$
    Hence, 041331021641 is not a valid UPC.

Check Digits: ISBNs

Books are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code

$$x_1, x_2, x_3, \ldots, x_9, \boxed{x_{10}}$$

The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \ (\mathrm{mod}\ 11).$$

Since $\quad 11 x_{10} \equiv 0 \ (\mathrm{mod}\ 11) \quad$ **and** $\quad x_{10} + 10 x_{10} \equiv \sum_{i=1}^{10} i x_i \ (\mathrm{mod}\ 11)$

it is easy to show that the validity of an ISBN-10 number can be equivalently evaluated by checking $\quad \sum_{i=1}^{10} i x_i \equiv 0 \ (\mathrm{mod}\ 11).$

Check Digits: ISBNs

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11} \quad \Longleftrightarrow \quad \sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}.$$

a. Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
b. Is 084930149X a valid ISBN10?

Solution:

a. $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \pmod{11}$.
    $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$.
    $x_{10} \equiv 189 \equiv 2 \pmod{11}$.   Hence, $x_{10} = 2$.

    X is used as the digit 10.

b. $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10 =$
    $0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$
    Hence, 084930149X is not a valid ISBN-10.

- A *single error* is an error in one digit of an identification number and a *transposition error* is the accidental interchanging of two digits. Both of these kinds of errors can be detected by the check digit for ISBN-10.
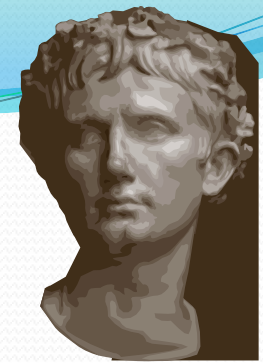
Cryptography

Section 4.6

Section Summary

- Classical cryptography
- Public Key cryptography
- RSA cryptosystem (overview)

Caesar Cipher

Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A. This process of making a message secret is an example of *encryption*.

Here is how the encryption process works:

- Replace each letter by an integer from $\mathbf{Z}_{26}$, that is an integer from 0 to 25 representing one less than its position in the alphabet.
- The encryption function is $f(p) = (p + 3) \bmod 26$. It replaces each integer p in the set $\{0,1,2,...,25\}$ by $f(p)$ in the set $\{0,1,2,...,25\}$.
- Replace each integer p by the letter with the position $p + 1$ in the alphabet.

Example: Encrypt the message "MEET YOU IN THE PARK" using the Caesar cipher.

Solution: Write with numbers in $\mathbf{Z}_{26}$: 12 4 4 19   24 14 20   8 13   19 7 4   15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3) \bmod 26$.

15 7 7 22   1 17 23   11 16   22 10 7   18 3 20 13.

Translating the numbers back to letters produces the encrypted message

"PHHW BRX LQ WKH SDUN."

Caesar Cipher

- To recover the original message, use $f^{-1}(p) = (p-3) \bmod 26$. So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters. This process of recovering the original message from the encrypted message is called *decryption*.

- The Caesar cipher is one of a family of ciphers called *shift ciphers.* Letters can be shifted by an integer *k,* with 3 being just one possibility. The <u>encryption function</u> is

    $f(p) = (p + k) \bmod 26$

  and the <u>decryption function</u> is

    $f^{-1}(p) = (p-k) \bmod 26$

  The integer k is called a *key.*

Shift Cipher

Example 1: Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with $k = 11$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

18 19 14 15   6 11 14 1 0 11   22 0 17 12 8 13 6.

Apply the shift $f(p) = (p + 11)$ **mod** 26, yielding

3 4 25 0   17 22 25 12 11 22   7 11 2 23 19 24 17.

Translating the numbers back to letters produces the ciphertext

"DEZA RWZMLW HLCXTYR."

Shift Cipher

Example 2: Decrypt the message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted using the shift cipher with $k = 7$.

Solution: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0   0 11 7 9 14 11 24.

Shift each of the numbers by $-k = -7$ modulo 26, yielding

4 23 15 4 17 8 4 13 2 4 8 18   0   6 17 4 0 19   19 4 0 2 7 4 17.

Translating the numbers back to letters produces the decrypted message

"EXPERIENCE IS A GREAT TEACHER."

Affine Ciphers

- Shift ciphers are a special case of *affine ciphers* which use functions of the form
$$f(p) = (ap + b) \bmod 26,$$
where a and b are integers, chosen so that f is a bijection.

  Note: this function is a bijection if and only if $\gcd(a,26) = 1$. (exercise: prove this)

- **Example**: What letter replaces the letter K when the function $f(p) = (7p + 3) \bmod 26$ is used for encryption.
  Solution: Since 10 represents K,
$$f(10) = (7{\cdot}10 + 3) \bmod 26 = 21,$$
  which corresponds to letter V.

Affine Ciphers

To decrypt a message encrypted by a shift cipher, the congruence $c \equiv ap + b \pmod{26}$ needs to be solved for p.

- Subtract b from both sides to obtain

$$ap \equiv c - b \pmod{26}.$$

- Multiply both sides by the inverse $\bar{a}$ of a modulo 26, which exists since $\gcd(a, 26) = 1$

$$\bar{a}ap \equiv \bar{a}(c - b) \pmod{26},$$

which simplifies to

$$p \equiv \bar{a}(c - b) \pmod{26}.$$

determining plain text p in $\mathbf{Z}_{26}$ given a, b and cryptotext c.

Example

- What is the decryption function for an affine cipher
$$f(x) \equiv 3x + 7 \pmod{26} \quad ?$$

- Decrypt the following message encrypted by the above

"UTTQ CTOA"

Note: 9 is inverse of 3 modulo 26   and $-9 \cdot 7 = -63 \equiv 15 \pmod{26}$

Solution: $f(x) \equiv 9x + 15 \pmod{26}$
and the plain text is   "NEED HELP"

Public Key Cryptography

- All classical ciphers, including shift and affine ciphers, are *private key cryptosystems*. Knowing the encryption key allows one to quickly determine the decryption key.

- All parties who wish to communicate using a private key cryptosystem must share the key and keep it a secret.

- In public key cryptosystems, first invented in the 1970s, knowing how to encrypt a message does not help one to decrypt the message. Therefore, everyone can have a publicly known encryption key. The only key that needs to be kept secret is the decryption key.

The RSA Cryptosystem

Clifford Cocks
(Born 1950)
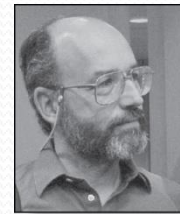
- A public key cryptosystem, now known as the RSA system was introduced in 1976 by three researchers at MIT.

Ronald Rivest
(Born 1948)

Adi Shamir
(Born 1952)

Leonard Adelman
(Born 1945)

It is now known that the method was discovered earlier by Clifford Cocks, working secretly for the UK government.

- The public encryption key is a pair (n,e) where the modulus n is the product of two large (200 digits) primes p and q and exponent e is relatively prime to $(p-1)(q-1)$.

- Factorization $n = p \cdot q$ is kept private! With approximately 400 digits, n cannot be factored in a reasonable length of time.

RSA Encryption (overview)

- To encrypt a message using RSA using a public key (n,e) :
  i. Translate the *plaintext message M* into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
  ii. Concatenate the two digit integers into strings of digits.
  iii. Divide this string into equally sized blocks of $2N$ digits where $2N$ is the largest even number 2525…25 with $2N$ digits that does not exceed n.
  iv. The plaintext message M is now a sequence of integers $m_1, m_2, …, m_k$.
  v. Each block (an integer) is encrypted using <u>modular exponentiation</u> function (efficiently computable, see Chapter 4.2, p.253) that gives *ciphertext message C*:

$$C = M^e \bmod n$$

RSA Decryption (overview)

- Decryption $C \rightarrow M$ requires known <u>exponentiation inverse</u> d of e modulo n
$$C^d = (M^e)^d \equiv M \pmod{n}$$
Modular exponentiation is a *one-way function* : it is easy to compute, but hard to invert. In general, finding modular exponential inverse d is believed to be very difficult (as difficult as finding primal factorization of modulus n).

- RSA assumes "privately" known factorization $n = p \cdot q$ where p and q are prime. **In this case**, the decryption key d can be obtained as a <u>multiplicative inverse</u> of e modulo $(p-1)(q-1)$, which is easy to compute (via Euclidean algorithm for Bezout coefficients) assuming relative primality $\gcd(e,(p-1)(q-1)) = 1$. It can be shown that such (privately known) key d allows to decrypt ciphertext message C with the simple computation
$$M = C^d \bmod p \cdot q \qquad (\textit{see text for the proof}).$$

- RSA works as a public key system since the only known method of finding d is based on a factorization of n into primes. There is currently no known feasible method for factoring large numbers into primes.