Discrete Structures for Computing

CSCE 222

Sandeep Kumar

Many slides based on [Lee19], [Rog21], [GK22]

Number Theory and Cryptography

Chapter 4

Chapter Motivation

Number theory is the part of mathematics devoted to the study of the integers and their properties.

- Key ideas in number theory include divisibility and the primality of integers.
- Representations of integers, including binary and hexadecimal representations, are part of number theory.
- Number theory has long been studied because of the beauty of its ideas, its accessibility, and its wealth of open questions.
- We'll use many ideas developed in Chapter 1 about proof methods and proof strategy in our exploration of number theory.
- Mathematicians have long considered number theory to be pure mathematics, but it has important applications to computer science and cryptography studied in Sections 4.5 and 4.6.

Chapter Summary

- Divisibility and Modular Arithmetic
- Integer Representations and Algorithms
- Primes and Greatest Common Divisors
- Solving Congruences
- Applications of Congruences
- Cryptography

Section Summary

Divisibility and Modular Arithmetic

- Division
- Division Algorithm
- Modular Arithmetic

Division

Definition: If a and b are integers with $a \neq 0$, then a *divides* b if there exists an integer c such that $b = ac$.

- When a divides b, we say that a is a *factor* or *divisor* of b and that b is a multiple of a.
- The notation $a \mid b$ denotes that a divides b.
- If $a \mid b$, then b/a is an integer.
- If a does not divide b, we write $a \nmid b$.

Determine whether $3 \mid 7$, and whether $3 \mid 12$.

Properties of Divisibility

Theorem 1: Let a, b, c be integers, where $a \neq 0$.

1. If $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;
2. If $a \mid b$, then $a \mid bc$ for all integers c;
3. If $a \mid b$ and $b \mid c$, then $a \mid c$.

Proof: (1) Suppose $a \mid b$ and $a \mid c$, then it follows that there are integers s and t with $b = as$ and $c = at$. Hence,

$$b + c = as + at = a(s + t)$$

Hence, $a \mid (b + c)$. Exercises 3 and 4 ask for proofs of parts (2) and (3).

Corollary: If a, b, c are integers, where $a \neq 0$, such that $a \mid b$ and $a \mid c$, then $a \mid mb + nc$ whenever m and n are integers.

Can you show how it follows easily from (1) and (2) of Theorem 1?

Division Algorithm

When an integer is divided by a positive integer, there is a quotient and a remainder. This is traditionally called the "Division Algorithm", but is really a theorem.

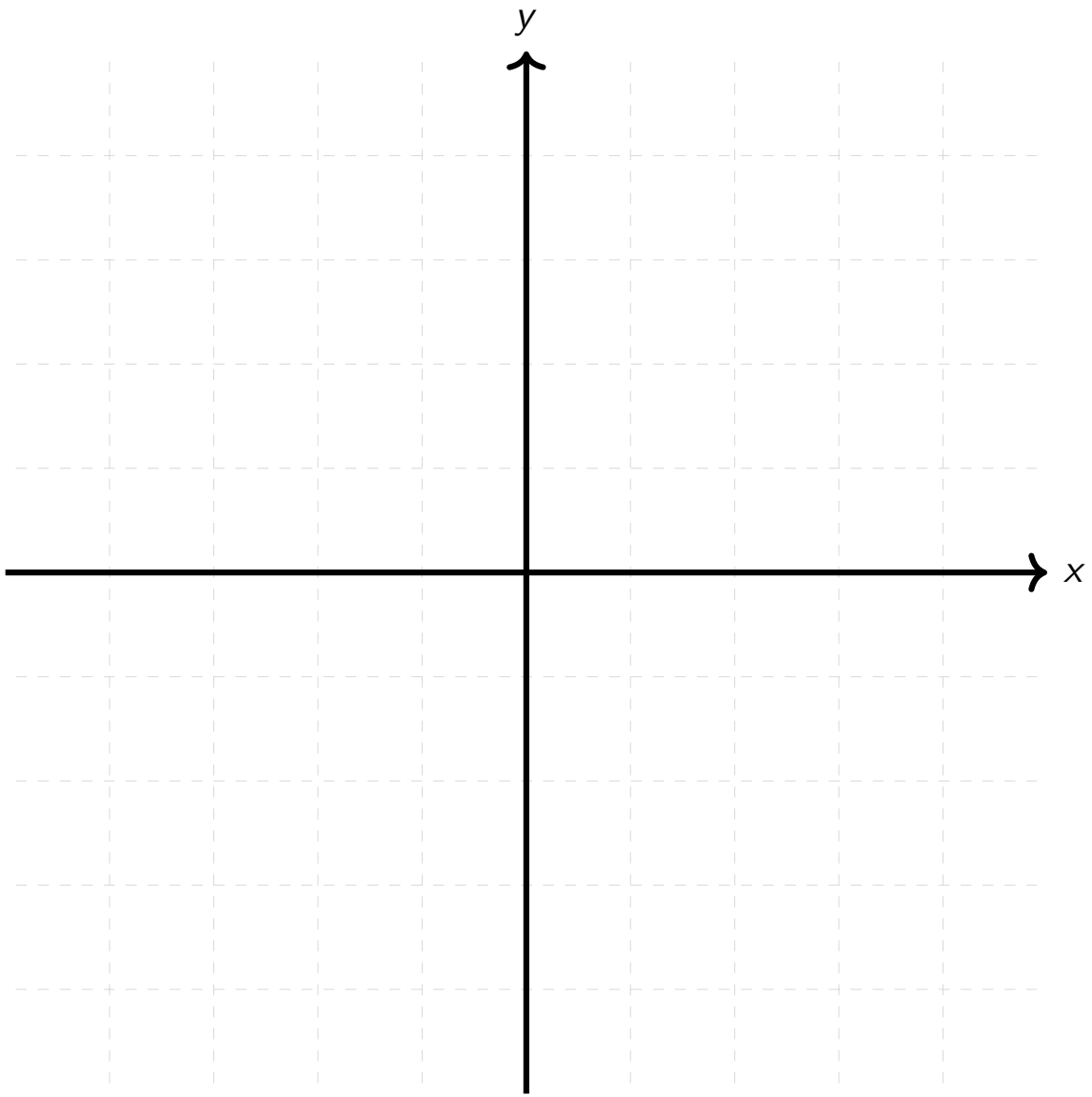
Division Algorithm: If a is an integer and d a positive integer, then there are unique integers q and r, with $0 \leq r < d$, such that $a = dq + r$ (proved in Section 5.2).

- d is called the *divisor*, a is called the *dividend*.
- q is called the *quotient*, r is called the *remainder*.

Assume non-uniqueness and show contradiction for

$$a = dq + r = dq' + r'$$
$$d(q - q') = (r' - r)$$

Division Algorithm...



Division Algorithm ✗

- What are the quotient and remainder when 101 is divided by 11?
  - The quotient when 101 is divided by 11 is $9 = 101$ **div** 11, and the remainder is $2 = 101 \bmod 11$.
- What are the quotient and remainder when -11 is divided by 3?
  - The quotient when -11 is divided by 3 is $-4 = -11$ **div** 3, and the remainder is $1 = 11 \bmod 3$.

Congruence Relation

Definition: If a and b are integers and m is a positive integer, then a is *congruent* to b *modulo* m if m divides $a - b$.

- The notation $a \equiv b \pmod{m}$ says that a is congruent to b modulo m.
- We say that $a \equiv b \pmod{m}$ is a congruence and that m is its modulus.
- Two integers are congruent mod m if and only if they have the same remainder when divided by m.
- If a is not congruent to b modulo m, we write $a \not\equiv b \pmod{m}$.

Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

- $17 \equiv 5 \pmod{6}$ because 6 divides $17 - 5 = 12$.
- $24 \not\equiv 14 \pmod{6}$ since $24 - 14 = 10$ is not divisible by 6.

More on Congruences

Theorem 4: Let m be a positive integer. The integers a and b are congruent modulo m if and only if there is an integer k such that $a = b + km$.

Proof:

- If $a \equiv b \pmod{m}$, then by the definition of congruence, $m \mid a - b$. Hence, there is an integer k such that $a - b = km$ and equivalently $a = b + km$.
- Conversely, if there is an integer k such that $a = b + km$, then $km = a - b$. Hence, $m \mid a - b$ and $a \equiv b \pmod{m}$.

The Relationship between $(\mod m)$ and $\bmod m$ notations

The use of "mod" in $a \equiv b \pmod{m}$ and $a \bmod m = b$ are different.

- $a \equiv b \pmod{m}$ is a relation on the set of integers.
- In $a \bmod m = b$, the notation **mod** denotes a function.

The relationship between these notations is made clear in this theorem.

Theorem 3: Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if

$$a \bmod m = b \bmod m$$

Proof sketch.

$$[a \equiv b \pmod{m}] \rightarrow [a \bmod m = b \bmod m] \ \wedge$$
$$[a \bmod m = b \bmod m] \rightarrow [a \equiv b \pmod{m}]$$

- $a = b + km$, or $(\overbrace{r + pm}^{a} = \overbrace{r' + qm + km}^{a})$. By uniqueness of representation, $r = r'$.
- $a = r + pm, b = r + qm$. So $a - b = km$.

Congruences of Sums and Products

Theorem 5: Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then

$$\boxed{a + c \equiv b + d \pmod{m}, \text{ and } ac \equiv bd \pmod{m}}$$

Proof:

- Because $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, by Theorem 4 there are integers s and t with $b = a + sm$ and $d = c + tm$.
- Therefore,
  - $b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$
  - $bd = (a + sm)(c + tm) = ac + m(at + cs + stm)$
- Hence, $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

Example: $\because 7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$, it follows from above that

- $18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$
- $77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod{5}$

Algebraic Manipulation of Congruences

Skip for now

- Multiplying both sides of a valid congruence by an integer preserves validity.
  - If $a \equiv b \pmod{m}$ holds then $c \cdot a \equiv c \cdot b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.
- Adding an integer to both sides of a valid congruence preserves validity.
  - If $a \equiv b \pmod{m}$ holds then $c + a \equiv c + b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.
- Dividing a congruence by an integer does not always produce a valid congruence. Explain using this.
  - The congruence $14 \equiv 8 \pmod{6}$ holds. But dividing both sides by 2 does not produce a valid congruence since $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \mod 6$.
  - See Section 4.3 Theorem 7 for conditions when division is ok.

Computing the (mod m) function of Products and Sums

We use the following corollary to Theorem 5 to compute the remainder of the product or sum of two integers when divided by m from the remainders when each is divided by m.

Corollary: Let m be a positive integer and let a and b be integers. Then,

$$(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$$
$$ab \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$$

Why? $\because a \equiv a \pmod{m}, b \equiv b \pmod{m}$. Apply Theorem 5.

Arithmetic Modulo m

Skip for now

Let Z_m be the set of nonnegative integers less than m: $\{0, 1, \ldots, m-1\}$

- The operation $+_m$ is defined as $a +_m b = (a + b) \bmod m$. This is *addition modulo m*.
- The operation $\cdot_m$ is defined as $a \cdot_m b = (a \cdot b) \bmod m$. This is *multiplication modulo m*.
- Using these operations is said to be doing *arithmetic modulo m*.

Example: Find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

- $7 +_{11} 9 = (7 + 9) \bmod 11 = 16 \bmod 11 = 5$
- $7 \cdot_{11} 9 = (7 \cdot 9) \bmod 11 = 63 \bmod 11 = 8$

Arithmetic Modulo m . . .

Skip for now

The operations $+_m$ and $\cdot_m$ satisfy many of the same properties as ordinary addition and multiplication.

- *Closure*: If a and $b \in Z_m$, then $a +_m b$ and $a \cdot_m b \in Z_m$.
- *Associativity*: If a, b, c belong to Z_m , then

$$\begin{aligned} (a +_m b) +_m c &= a +_m (b +_m c), \text{ and} \\ (a \cdot_m b) \cdot_m c &= a \cdot_m (b \cdot_m c) \end{aligned}$$

- *Commutativity*: If a and b belong to Z_m , then $a +_m b = b +_m a$ and $a \cdot_m b = b \cdot_m a$.
- *Identity elements*: The elements 0 and 1 are identity elements for addition and multiplication modulo m, respectively.

  If $a \in Z_m$, then $a +_m 0 = a$, and $a \cdot_m 1 = a$.

Arithmetic Modulo m . . .

Skip for now

- *Additive inverses*: If $a \neq 0$ belongs to Z_m , then $m - a$ is the additive inverse of a modulo m and 0 is its own additive inverse.
  - $a +_m (m - a) = 0$ and $0 +_m 0 = 0$
- *Distributivity*: If a, b, c belong to Z_m , then
  - $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$, and
  - $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

Exercises 42-44 ask for proofs of these properties.

Multiplicatative inverses have not been included since they do not always exist. For example, there is no multiplicative inverse of 2 modulo 6.

Using the terminology of abstract algebra, Z_m with $+_m$ is a commutative group and Z_m with $+_m$ and $\cdot_m$ is a commutative ring.

Properties of modular arithmetic

For integers a and b and $k > 0$:

$$\begin{aligned} k \bmod k &= 0 \\ (a+b) \bmod k &= [(a \bmod k) + (b \bmod k)] \bmod k \\ ab \bmod k &= [(a \bmod k) \cdot (b \bmod k)] \bmod k \\ a^b \bmod k &= [(a \bmod k)^b] \bmod k \end{aligned}$$

$ab \bmod k \neq (a \bmod k) \cdot (b \bmod k)$ in general.

- $14 \bmod 6 = 2$, and $5 \bmod 6 = 5$, but
- $(14 \cdot 5) \bmod 6 = 4 \neq 2 \cdot 5$.

Subsection 1

Integer Representations and Algorithms

Section Summary

Integer Representations

- Base b Expansions
- Binary Expansions
- Octal Expansions
- Hexadecimal Expansions

Base Conversion Algorithm

Algorithms for Integer Operations

Representations of Integers

- In the modern world, we use *decimal*, or base 10, notation to represent integers. For example when we write 965, we mean

$$9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$$

- We can represent numbers using any base b, where $b > 1$.
- The bases $b = 2$ (binary), $b = 8$ (octal), and $b = 16$ (hexadecimal) are important for computing and communications
- The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base b Representations

We can use positive integer $b > 1$ as a base, because of this theorem:

Theorem: Let b be a positive integer > 1. Then if n is a positive integer, it can be expressed *uniquely* in the form:

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

where k is a nonnegative integer, $a_0, a_1, \ldots a_k$ are nonnegative integers less than b, and $a_k \neq 0$. The $a_j, j = 0, \ldots, k$ are called the base-b digits of the representation.

The representation of n given in the Theorem above is called the *base* b expansion of n and is denoted by $(a_k a_{k-1} \ldots a_1 a_0)_b$.

We usually omit the subscript 10 for base 10 expansions.

Binary Expansions

Most computers represent integers and do arithmetic with binary (base 2) expansions of integers. In these expansions, the only digits used are $0, 1$.

- Decimal expansion of $(1\ 0101\ 1111)_2$?

$$1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351$$

- What is the decimal expansion of the integer that has $(11011)_2$ as its binary expansion?

$$(11011)_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 27$$

Octal Expansions

The octal expansion (base 8) uses the digits $\{0, 1, 2, 3, 4, 5, 6, 7\}$.

- What is the decimal expansion of the number with octal expansion $(7016)_8$?

$$7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 = 3598$$

- What is the decimal expansion of the number with octal

$$1 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 = 64 + 8 + 1 = 73$$

Hexadecimal Expansions

The hexadecimal expansion needs 16 digits, but our decimal system provides only 10. So letters are used for the additional symbols. The hexadecimal system uses the digits

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

The letters A through F represent the decimal numbers 10 through 15.

- What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

$$2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 175627$$

- What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?

$$14 \cdot 16^1 + 5 \cdot 16^0 = 224 + 5 = 229$$

Base Conversion

To construct the base b expansion of an integer n:

- Divide n by b to obtain a quotient and remainder.

$$n = bq_0 + a_0, 0 \leq a_0 < b$$

- The remainder, a_0, is the rightmost digit in the base b expansion of n. Next, divide q_0 by b.

$$q_0 = bq_1 + a_1, 0 \leq a_1 < b$$

- The remainder, a_1, is the second digit from the right in the base b expansion of n.
- Continue by successively dividing the quotients by b, obtaining the additional base b digits as the remainder. The process terminates when the quotient is 0.

Proof of Base Conversion

```
procedure base_b_expansion(n, b: positive integers with b > 1)
q := n
k := 0
while (q ≠ 0)
    a_k := q mod b
    q := q div b
    k := k + 1
return (a_{k-1}, ..., a_1, a_0) {base b expansion of n}
```

- q represents the quotient obtained by successive divisions by b, starting with $q = n$.
- The digits in the base b expansion are the remainders of the division given by q mod b.
- The algorithm terminates when $q = 0$ is reached.

Base Conversion

Find the octal expansion of $(12345)_{10}$

Solution: Successively dividing by 8 gives:

- $12345 = 8 \cdot 1543 + 1$
- $1543 = 8 \cdot 192 + 7$
- $192 = 8 \cdot 24 + 0$
- $24 = 8 \cdot 3 + 0$
- $3 = 8 \cdot 0 + 3$

The remainders are the digits from right to left yielding $(30071)_8$.

Comparison of Hexadecimal, Octal, and Binary Representations

Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15																
Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Oct	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Bin	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

- Initial 0s are not shown.
- Each octal digit corresponds to a block of 3 binary digits.
- Each hexadecimal digit corresponds to a block of 4 binary digits.
- So, conversion between binary, octal, and hexadecimal is easy.

Conversion Between Binary, Octal, and Hexadecimal Expansions

Find the octal and hexadecimal expansions of $(11111010111100)_2$.

- To convert to octal, we group the digits into blocks of three

$$(011\ 111\ 010\ 111\ 100)_2$$
$$=(011)\cdot 8^4+(111)\cdot 8^3+(010)\cdot 8^2+(111)\cdot 8^1+(100)\cdot 8^0$$

  adding initial 0s as needed. The blocks from left to right correspond to the digits $3, 7, 2, 7, 4$. Hence, the solution is $(37274)_8$.

- To convert to hexadecimal, we group the digits into blocks of four

$$(0011\ 1110\ 1011\ 1100)_2$$

  adding initial 0s as needed. The blocks from left to right correspond to the digits $3, E, B, C$. Hence, the solution is $(3EBC)_{16}$.

Binary Addition of Integers

Algorithms for performing operations with integers using their binary expansions are important as computer chips work with binary numbers. Each digit is called a bit.

```
procedure add(a, b: positive integers)
{binary expansions of a and b are (a_{n-1}, a_{n-2}, ... a_0)_2 and
                    (b_{n-1}, b_{n-2}, ..., b_0)_2, respectively}
c := 0
for j := 0 to n-1
    d := ⌊(a_j + b_j + c)/2⌋ carry when sum ≥ 2
    s_j := a_j + b_j + c − 2d 0 when ≤ 2, 1 when > 2
    c := d
s_n := c
return(s_0, s_1, ... s_n) {the binary expansion of the sum is (s_n, s_{n-1}, ..., s_0)_2}
```

The number of additions of bits used by the algorithm to add two n-bit integers is $O(n)$.

Binary Multiplication of Integers

Algorithm for computing the product of two n bit integers.

```
procedure multiply(a, b: positive integers)
{binary expansions of a and b are (a_{n-1}, a_{n-2}, ... a_0)_2 and
                                  (b_{n-1}, b_{n-2}, ..., b_0)_2, respectively}
for j := 0 to n-1
    if b_j = 1 then
        c_j = a shifted j places
    else
        c_j := 0
{c_0, c_1, ..., c_{n-1} are the partial products}
p := 0
for j := 0 to n - 1
    p := p + c_j
return p {p is the value of ab}
```

The number of additions of bits used by the algorithm to multiply two n-bit integers is $O(n^2)$.

Modular Exponentiation

FYIO

Say you want to compute $6^{469} \bmod 7$. You could compute

$$6 \times 6 \times 6 \times \cdots\cdots 469 \text{ times}$$

Or, observe that

$$469 = 111010101_2 = 2^8 + 2^7 + 2^6 + 2^4 + 2^2 + 2^0$$

That is

$$6^{469} = 6^{2^8} \times 6^{2^7} \times 6^{2^6} \times 6^{2^4} \times 6^{2^2} \times 6^{2^0}$$

So instead compute each term individually with one multiply each. That is, compute $6^2, 6^4, 6^8, 6^{16}, 6^{32}, 6^{64}, 6^{128}, 6^{256}$ by repeated squaring.

Binary Modular Exponentiation

FYIO

In cryptography, it is important to be able to find $b^n \pmod m$ efficiently, where b, n, m are large integers.

Use the binary expansion of n, $n = (a_{k-1}, \ldots, a_1, a_0)_2$, to compute b_n .

Note that:

$$b^n = b^{a_{k-1} \cdot 2^{k-1} + \cdots + a_1 \cdot 2 + a_0}$$

Therefore, to compute b^n, we need only compute the values of

$$b, b^2, (b^2)^2 = b^4, (b^4)^2 = b^8, \ldots$$

and the multiply the terms in this list, where $a_j = 1$.

Example: Compute 3^{11} using this method.

Note that $11 = (1011)_2$ so that

$$3^{11} = 3^8 3^2 3^1 = ((3^2)^2)^2 3^2 3^1 = (9^2)^2 \cdot 9 \cdot 3 = (81)^2 \cdot 9 \cdot 3 = 6561 \cdot 9 \cdot 3 = 117,147$$

Binary Modular Exponentiation Algorithm

FYIO

```
procedure modular_exponentiation(b: integer, n = (a_{k-1}a_{k-2}a_1a_0)_2,
                                 m: positive integers)
x := 1
power := b mod m
for i := 0 to k-1
    if a_i = 1 then
        x := x · power
    power := (power · power) mod m
return x {x ≡ b^n mod m}
```

$O((\log m)^2 \log n)$ bit operations are used to find $b^n \bmod m$.

Subsection 2

Primes and Greatest Common Divisors

Section Summary

- Prime Numbers and their Properties
- Conjectures and Open Problems About Primes
- Greatest Common Divisors and Least Common Multiples
- The Euclidian Algorithm
- GCDs as Linear Combinations

Primes

Definition: A positive integer $p > 1$ is called *prime* if the only positive factors of p are 1 and p. A positive integer that is greater than 1 and is not prime is called *composite*.

Example: The integer 7 is prime because its only positive factors are 1 and 7, but 9 is composite because it is divisible by 3.

The Fundamental Theorem of Arithmetic

Theorem: Every positive integer > 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of non-decreasing size.

Examples:

- $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$.
- $641 = 641$.
- $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$.
- $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$.

- Let n be the *smallest* non-prime that cannot be represented as a product of primes. Then n is a product of composites, which are product of primes!
- Uniqueness shown in a later slide.

The Sieve of Erastosthenes

The *Sieve of Erastosthenes* can be used to find all primes not exceeding a specified positive integer. For example, begin with the list of integers between 1 and 100.

- Delete all the integers, other than 2, divisible by 2.
- Delete all the integers, other than 3, divisible by 3.
- Next, delete all the integers, other than 5, divisible by 5.
- Next, delete all the integers, other than 7, divisible by 7.
- Since all the remaining integers are not divisible by any of the previous integers, other than 1, the primes are:

$$\{2, 3, 5, 7, 11, 15, 1719, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97\}$$

The Sieve of Erastosthenes

- If an integer n is composite, then it has a prime divisor less than or equal to $\sqrt{n}$.
- To see this, note that if $n = ab$, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.
  - By contradiction, what would happen if both $a, b > \sqrt{n}$?
- Trial division, a very inefficient method of determining if a number n is prime, is to try every integer $i \leq n$ and see if n is divisible by i.

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

Infinitude of Primes ✗

Theorem: There are infinitely many primes. (Euclid)

Proof: Assume finitely many primes: $p_1, p_2, \ldots, p_n$

- Let $q = p_1 p_2 \cdots p_n + 1$.
- Either q is prime or by the fundamental theorem of arithmetic it is a product of primes.
  - But none of the primes p_j divides q since if $p_j \mid q$, then p_j divides $q - p_1 p_2 \cdots p_n = 1$.
  - Hence, there is a prime not on the list $p_1, p_2, \ldots, p_n$. It is either q, or if q is composite, it is a prime factor of q. This contradicts the assumption that $p_1, p_2, \ldots, p_n$ are all the primes.
- Consequently, there are infinitely many primes.

This proof was given by Euclid in The Elements. The proof is considered to be one of the most beautiful in all mathematics. It is the first proof in The Book, inspired by the famous mathematician Paul Erdös imagined collection of perfect proofs maintained by God.

Representing Functions

FYIO

Definition: Prime numbers of the form $2^p - 1$, where p is prime, are called Mersenne primes.

- $2^2 - 1 = 3, 2^3 - 1 = 7, 2^5 - 1 = 37,$ and $2^7 - 1 = 127$ are Mersenne primes.
- $2^{11} - 1 = 2047$ is not a Mersenne prime since $2047 = 23 \times 89$.
- There is an efficient test for determining if $2^p - 1$ is prime.
- The largest known prime numbers are Mersenne primes.
- As of mid 2011, 47 Mersenne primes were known, the largest is $2^{43,112,609} - 1$, which has nearly 13 million decimal digits.
- The *Great Internet Mersenne Prime Search (GIMPS)* is a distributed computing project to search for new Mersenne Primes.

  See http://www.mersenne.org/.

Distribution of Primes

FYIO

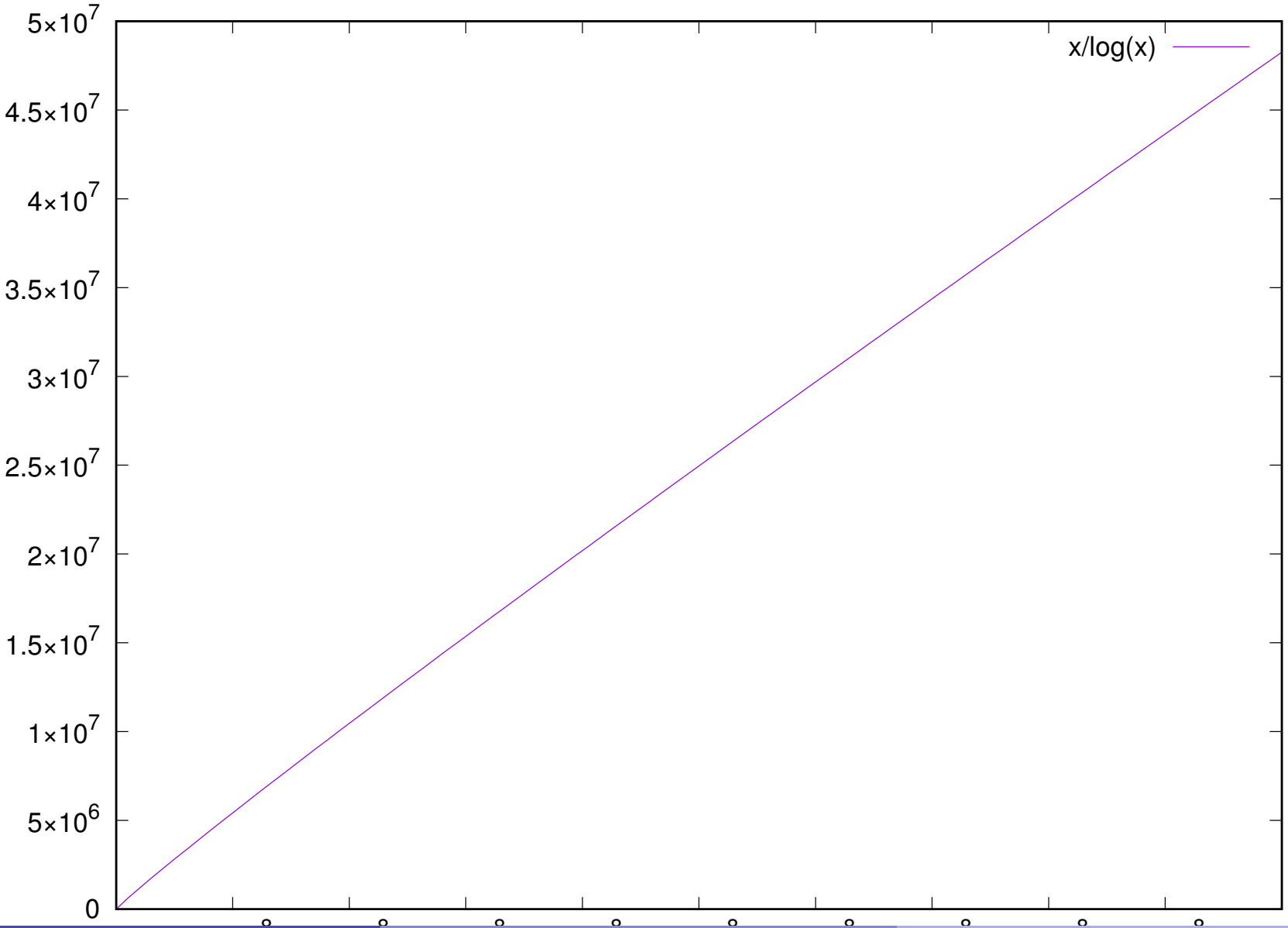
Mathematicians have been interested in the distribution of prime numbers among the positive integers. In the nineteenth century, the *prime number theorem* was proved which gives an asymptotic estimate for the number of primes not exceeding x.

Prime Number Theorem: The ratio of the number of primes not exceeding x and $x/\ln x$ approaches 1 as x grows without bound.

- The theorem tells us that the number of primes not exceeding x, can be approximated by $x/\ln x$.
- The odds that a randomly selected positive integer $< n$ is prime are approximately $(n/\ln n)/n = 1/\ln n$.

Distribution of Primes

FYIO



Primes and Arithmetic Progressions

FYIO

Euclid's proof that there are infinitely many primes can be easily adapted to show that there are infinitely many primes in the following $4k + 3, k = 1, 2, \ldots$ (See Exercise 55)

In the 19th century G. Lejuenne Dirichlet showed that every arithmetic progression $ka + b, k = 1, 2, \ldots$ where a and b have no common factor greater than 1 contains infinitely many primes. (The proof is beyond the scope of the text.)

Are there long arithmetic progressions made up entirely of primes?

- $5, 11, 17, 23, 29$ is an arithmetic progression of five primes.
- $199, 409, 619, 829, 1039, 1249, 1459, 1669, 1879, 2089$ is an arithmetic progression of ten primes.

In the 1930s, Paul Erdös conjectured that for every positive integer $n > 1$, there is an arithmetic progression of length n made up entirely of primes. This was proven in 2006, by Ben Green and Terrence Tao.

Generating Primes

FYIO

The problem of generating large primes is of both theoretical and practical interest. We will see (in Section 4.6) that finding large primes with hundreds of digits is important in cryptography.

So far, no useful closed formula that always produces primes has been found. There is no simple function $f(n)$ such that $f(n)$ is prime for all positive integers n.

But $f(n) = n^2 - n + 41$ is prime for all integers $1, 2, \ldots, 40$. Because of this, we might conjecture that $f(n)$ is prime for all positive integers n. But $f(41) = 41^2$ is not prime.

More generally, there is no polynomial with integer coefficients such that $f(n)$ is prime for all positive integers n. (See supplementary Exercise 23.)

Fortunately, we can generate large integers which are almost certainly primes. See Chapter 7.

Conjectures about Primes

FYIO

Even though primes have been studied extensively for centuries, many conjectures about them are unresolved, including:

- *Goldbach's Conjecture*: Every even integer $n, n > 2$, is the sum of two primes. It has been verified by computer for all positive even integers up to 1.6^{1018}. The conjecture is believed to be true by most mathematicians.
- There are infinitely many primes of the form $n^2 + 1$, where n is a positive integer. But it has been shown that there are infinitely many primes of the form $n^2 + 1$, where n is a positive integer or the product of at most two primes.
- *The Twin Prime Conjecture*: The twin prime conjecture is that there are infinitely many pairs of twin primes. Twin primes are pairs of primes that differ by 2. Examples are 3 and 5, 5 and 7, 11 and 13, etc. The current world's record for twin primes (as of mid 2011) consists of numbers $65,516,468,355^{2333,333} \pm 1$, which have $100,355$ decimal digits.

Greatest Common Divisor

Definition: Let a and b be integers, not both zero. The largest integer d such that $d \mid a$ and also $d \mid b$ is called the greatest common divisor of a and b. The greatest common divisor of a and b is denoted by $gcd(a, b)$.

One can find greatest common divisors of small numbers by inspection.

- What is the greatest common divisor of 24 and 36?
  - $gcd(24, 36) = 12$
- What is the greatest common divisor of 17 and 22?
  - $gcd(17, 22) = 1$

Greatest Common Divisor

Definition: The integers a and b are relatively prime if their greatest common divisor is 1.

Example: 17 and 22.

Definition: The integers $a_1, a_2, \ldots, a_n$ are pairwise relatively prime if $gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

- Determine whether the integers $10, 17, 21$ are pairwise relatively prime.
- $gcd(10, 17) = 1$, $gcd(10, 21) = 1$, and $gcd(17, 21) = 1$. So 10, 17, and 21 are pairwise relatively prime.
- Determine whether the integers $10, 19, 24$ are pairwise relatively prime.
- Because $gcd(10, 24) = 2$, $\therefore$ 10, 19, and 24 are not pairwise relatively prime.

Finding the Greatest Common Divisor Using Prime Factorizations

Suppose the prime factorizations of a and b are:

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}, \quad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

where each exponent is ≥ 0, and where all primes occurring in either prime factorization are included in both. Then,

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \dots p_n^{\min(a_n, b_n)}$$

This formula is valid since the integer on the right (of the equals sign) divides both a and b. No larger integer can divide both a and b.

Example: $120 = 2^3 \cdot 3 \cdot 5 \quad 500 = 2^2 \cdot 5^3$

$$\gcd(120, 500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$$

Finding the gcd of two positive integers using their prime factorizations is not efficient because there is no efficient algorithm for finding the prime factorization of a positive integer.

Least Common Multiple

Definition: The least common multiple of the positive integers a and b is the smallest positive integer that is divisible by both a and b. It is denoted by (a, b).

The least common multiple can also be computed from the prime factorizations.

$$\text{lcm}(a, b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \dots p_n^{\max(a_n,b_n)}$$

This number is divided by both a and b and no smaller number is divided by a and b.

Example: $lcm(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)} 3^{\max(5,3)} 7^{\max(2,0)} = 2^4 3^5 7^2$

The greatest common divisor and the least common multiple of two integers are related by:

Theorem 5: Let a and b be positive integers. Then

$$ab = gcd(a, b) \times lcm(a, b)$$

Proof is Exercise 31, page 289.

The Euclidean Algorithm for finding GCDs

The Euclidian algorithm is an efficient method for computing the greatest common divisor of two integers. It is based on the idea that

$$gcd(a, b) = gcd(b \bmod a, a), a < b$$

Example: Find $gcd(91, 287)$.

$$\begin{aligned} 287 &= 91 \cdot 3 + 14 \\ 91 &= 14 \cdot 6 + 7 \\ 14 &= 7 \cdot 2 + 0 \end{aligned}$$

- $287 = 91 \cdot 3 + 14$
- $91 = 14 \cdot 6 + 7$
- $14 = 7 \cdot 2 + 0$

Stopping condition: remainder is 0.

$$gcd(287, 91) = gcd(91, 14) = gcd(14, 7) = 7$$

Euclidean Algorithm

```
procedure gcd(a, b: positive integers)
    x := a
    y := b
    while y ≠ 0
        r := x mod y
        x := y
        y := r
    return x {gcd(a,b) is x}
```

In Section 5.3, we'll see that the time complexity of the algorithm is $O(\log b)$, where $a > b$.

Correctness of Euclidean Algorithm

Lemma 1: Let $a = bq + r$, where a, b, q, and r are integers. Then

$$gcd(a, b) = gcd(b, r)$$

Proof:

- Suppose that d divides both a and b.
  - Then d also divides $a - bq = r$ (by Theorem 1 of Section 4.1).
  - Hence, any common divisor of a and b must also be any common divisor of b and r.
- Suppose that d divides both b and r.
  - Then d also divides $bq + r = a$.
  - Hence, any common divisor of b and r must also be a common divisor of a and b.
- Therefore, $gcd(a, b) = gcd(b, r)$.

gcds as Linear Combinations

B?zout's Theorem: If a and b are positive integers, then there exist integers s and t such that

$$gcd(a, b) = sa + tb$$

Proof in exercises of Section 5.2. Illustration on the next slide.

EGCD

- The Extended Euclidean Algorithm $EGCD(a, b)$ permits one to find

$$b^{-1} \pmod{a} \text{ and } a^{-1} \pmod{b}$$

  provided that $GCD(a, b) = 1$, in addition to $GCD(a, b)$.
- Start with the vectors

x_1	x_2	x_3
1	0	a
0	1	b

  and reduce one vector by subtracting a multiple of the other from it until the result has the third component 1.
- Both vectors maintain the invariant $ax_1 + bx_2 = x_3$.
- Eventually, you get an equation of the form $ax_1 + bx_2 = 1$.

  This gives $x_2 = b^{-1} \pmod{a}$ and $x_1 = a^{-1} \pmod{b}$.
- *Demo*.

Modular Division

Proposition [TW02, Page 68].

Let a, b, c, n be integers with $n \neq 0$ and with $\text{GCD}(a, n) = 1$. If $ab \equiv ac \pmod{n}$ then,

$$b \equiv c \pmod{n}$$

- Example: $2 \times 1 \equiv 2 \times 4 \pmod{6}$, but $1 \not\equiv 4 \pmod{6}$.
- Solving $ax \equiv c \pmod{n}$ when $\text{GCD}(a, n) = 1$ is now easy.
- Dividing a congruence by an integer does *not* always produce a valid congruence. Explain using *this demo*.
  - The congruence $14 \equiv 8 \pmod{6}$ holds.
  - But dividing both sides by 2 does not produce a valid congruence, since
  - $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \mod 6$.

Consequences of B?zout's Theorem

FYIO

Lemma 2: If a, b, and c are positive integers such that $gcd(a, b) = 1$ and $a \mid bc$, then $a \mid c$.

Proof: Assume $gcd(a, b) = 1$ and $a \mid bc$.

- Since $gcd(a, b) = 1$, by B?zout's Theorem there are integers s and t such that
$$sa + tb = 1$$
- Multiplying both sides of the equation by c, yields $sac + tbc = c$.
- From Theorem 1 of Section 4.1: $a \mid tbc$ (part ii) and a divides $sac + tbc$ since $a \mid sac$ and $a \mid tbc$.
- We conclude $a \mid c$, since $sac + tbc = c$.

Lemma 3: If p is prime and $p \mid a_1 a_2 \cdots a_n$, then $p \mid a_i$ for some i. Proof uses mathematical induction—see Exercise 64 of Section 5.1)

Lemma 3 is crucial in the proof of the uniqueness of prime factorizations.

Uniqueness of Prime Factorization

We will prove that a prime factorization of a positive integer where the primes are in nondecreasing order is unique.

Proof: (*By contradiction*) Suppose that the positive integer n can be written as a product of primes in two distinct ways:

$$n = p_1^{a_1} p_2^{a_2} \cdots p_s^{a_s} = q_1^{b_1} q_2^{b_2} \cdots q_t^{b_t}$$

- Remove all common primes from the factorizations to get,
  - $a_i, b_j > 0$.
  - No $p_i = q_j$.
- But that is not possible.
  - If p_j divides LHS, then it must divide one of the $q_i^{b_i}$ by Lemma 3.

Subsection 3

Solving Congruences

Section Summary

- Linear Congruences
- The Chinese Remainder Theorem
- Computer Arithmetic with Large Integers
- Fermat's Little Theorem
- Pseudoprimes
- Primitive Roots and Discrete Logarithms

The Chinese Remainder Theorem

In the first century, the Chinese mathematician Sun-Tsu asked:

> There are certain things whose number is unknown. When divided by 3, the remainder is 2; when divided by 5, the remainder is 3; when divided by 7, the remainder is 2. What will be the number of things?

This puzzle can be translated into the solution of the system of congruences:

$$\begin{aligned} x &\equiv 2 \pmod 3, \\ x &\equiv 3 \pmod 5, \\ x &\equiv 2 \pmod 7. \end{aligned}$$

We'll see how the theorem that is known as the *Chinese Remainder Theorem* can be used to solve Sun-Tsu's problem.

ChatGPT: Is the attribution of the Chinese Remainder Theorem to Sun-Tsu apocryphal?

The Chinese Remainder Theorem

Theorem 2: (The Chinese Remainder Theorem) Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers > 1 and $a_1, a_2, \ldots, a_n$ arbitrary integers. Then the system

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

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

That is, there is a solution x with $0 \leq x < m$ and all other solutions are congruent modulo m to this solution.

Proof: We'll show that a solution exists by describing a way to construct the solution. Showing that the solution is unique modulo m is Exercise 30. Show the isomorphism here.

The Chinese Remainder Theorem

To construct a solution first let $M_k = m/m_k$ for $k = 1, 2, \ldots, n$ and let $m = m_1 m_2 \cdots m_n$.

Since $gcd(m_k, M_k) = 1$, by Theorem 1, there is an integer y_k, an inverse of M_k modulo m_k, such that

$$M_k y_k \equiv 1 \pmod{m_k}$$

Form the sum

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n$$

Note that because $M_j \equiv 0 \pmod{m_k}$ whenever $j \neq k$, all terms except the kth term in this sum are congruent to 0 modulo m_k.

Because $M_k y_k \equiv 1 \pmod{m_k}$, we see that

$$x \equiv a_k M_k y_k \equiv a_k \pmod{m_k}, \text{ for } k = 1, 2, \ldots, n$$

The Chinese Remainder Theorem

Consider the 3 congruences from Sun-Tsu's problem:

$$x \equiv 2 \pmod 3, x \equiv 3 \pmod 5, x \equiv 2 \pmod 7$$

- Let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$, $M_3 = m/7 = 15$.
- We see that
  - 2 is an inverse of $M_1 = 35$ modulo 3 since $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod 3$.
  - 1 is an inverse of $M_2 = 21$ modulo 5 since $21 \equiv 1 \pmod 5$.
  - 1 is an inverse of $M_3 = 15$ modulo 7 since $15 \equiv 1 \pmod 7$.
- Hence,

$$\begin{aligned} x &= a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 \\ &= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \\ &= 233 \equiv 23 \pmod{105} \end{aligned}$$

- We have shown that 23 is the smallest positive integer that is a simultaneous solution. Check it!

Fermat's Little Theorem

Theorem 3: (Fermat's Little Theorem) If p is prime and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$.

- Furthermore, for every integer a we have $a^p \equiv a \pmod{p}$. Proof outlined in Exercise 19.
- Fermat's little theorem is useful in computing the remainders modulo p of large powers of integers.

. . . . . .

Find $7^{222} \pmod{11}$.

By Fermat's little theorem, we know that $7^{10} \equiv 1 \pmod{11}$, and so $(7^{10})^k \equiv 1 \pmod{11}$, for every positive integer k. Therefore,

$$7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} \cdot 7^2 \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}$$

Hence, $7^{222} \pmod{11} = 5$.

Fermat's Little Theorem—Proof Sketch

- Consider

$$P = 1a \cdot 2a \cdot 3a \cdots (p-2)a \cdot (p-1)a = a^{p-1}(p-1)!$$

- $1 \cdot a \neq 2 \cdot a \neq 3 \cdot a \neq \cdots \neq (p-1) \cdot a$ because the residue system mod p is a field and a has an inverse in it.
- Thus $1a, 2a, \ldots$ merely enumerate the numbers $1 \ldots (p-1)$ in some order.
- Canceling out $(p-1)!$ from both sides [because $(p-1)!$ is coprime to p] of the equation we get $a^{p-1} = 1$.

Pseudoprimes

FYIO

By Fermat's little theorem, for $n > 2$ prime,

$$2^{n-1} \equiv q \pmod{n}$$

But if this congruence holds, n may not be prime. Composite integers n such that $2^{n-1} \equiv 1 \pmod{n}$ are called pseudoprimes to the base 2.

Example: The integer 341 is a pseudoprime to the base 2.

- $341 = 11 \cdot 31$
- $2^{340} \equiv 1 \pmod{341}$ (see in Exercise 37)
- We can replace 2 by any integer $b > 2$.

Definition: Let b be a positive integer. If n is a composite integer, and $b^{n-1} \equiv 1 \pmod{n}$, then n is called a pseudoprime to the base b.

Pseudoprimes

FYIO

Given a positive integer n, such that $2^{n-1} \equiv 1 \pmod{n}$:

- If n does not satisfy the congruence, it is composite.
- If n does satisfy the congruence, it is either prime or a pseudoprime to the base 2.

Doing similar tests with additional bases b, provides more evidence as to whether n is prime.

Among the positive integers not exceeding a positive real number x, compared to primes, there are relatively few pseudoprimes to the base b.

- For example, among the positive integers less than 10^{10} there are $455,052,512$ primes, but only $14,884$ pseudoprimes to the base 2.

Carmichael Numbers I

FYIO

There are composite integers n that pass all tests with bases b such that $gcd(b, n) = 1$.

Definition: A composite integer n that satisfies the congruence $b^{n-1} \equiv 1 \pmod{n}$ for all positive integers b with $gcd(b, n) = 1$ is called a Carmichael number.

Example: The integer 561 is a Carmichael number. To see this:

- 561 is composite, since $561 = 3 \cdot 11 \cdot 13$.
- If $gcd(b, 561) = 1$, then $gcd(b, 3) = 1$, then $gcd(b, 11) = gcd(b, 17) = 1$.
- Using Fermat's Little Theorem: $b^2 \equiv 1 \pmod{3}$, $b^{10} \equiv 1 \pmod{11}$, $b^{16} \equiv 1 \pmod{17}$.

Carmichael Numbers II

FYIO

- Then,

$$b^{560} = \left(b^2\right)^{280} \equiv 1 (\mathrm{mod} 3),$$
$$b^{560} = \left(b^{10}\right)^{56} \equiv 1 (\mathrm{mod} 11),$$
$$b^{560} = \left(b^{16}\right)^{35} \equiv 1 (\mathrm{mod} 17)$$

- It follows (see Exercise 29) that $b^{560} \equiv 1 \pmod{561}$ for all positive integers b with $gcd(b, 561) = 1$. Hence, 561 is a Carmichael number.

Even though there are infinitely many Carmichael numbers, there are other tests (described in the exercises) that form the basis for efficient probabilistic primality testing. (see Chapter 7)

Primitive Roots

FYIO

Definition: A primitive root modulo a prime p is an integer r in Z_p such that every nonzero element of Z_p is a power of r.

Example: Since every element of Z_{11} is a power of 2, 2 is a primitive root of 11.

Powers of 2 modulo 11: $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 5$, $2^5 = 10$, $2^6 = 9$, $2^7 = 7$, $2^8 = 3$, $2^{10} = 2$.

Example: Since not all elements of Z_{11} are powers of 3, 3 is not a primitive root of 11.

Powers of 3 modulo 11: $3^1 = 3$, $3^2 = 9$, $3^3 = 5$, $3^4 = 4$, $3^5 = 1$, and the pattern repeats for higher powers.

Important Fact: There is a primitive root modulo p for every prime number p.

Discrete Logarithms

FYIO

Suppose p is prime and r is a primitive root modulo p. If a is an integer between 1 and $p-1$, that is an element of Z_p, there is a unique exponent e such that $r^e \pmod{p} = a$ in Zp, that is, $r^e \pmod{p} = a$.

Definition: Suppose that p is prime, r is a primitive root modulo p, and a is an integer between 1 and $p-1$, inclusive. If $r^e \pmod{p} = a$ and $1 \leq e \leq p-1$, we say that e is the discrete logarithm of a modulo p to the base r and we write $\log_r a = e$ (where the prime p is understood).

Example 1: We write $\log_2 3 = 8$ since the discrete logarithm of 3 modulo 11 to the base 2 is 8 as $2^8 = 3$ modulo 11.

Example 2: We write $\log_2 5 = 4$ since the discrete logarithm of 5 modulo 11 to the base 2 is 4 as $2^4 = 5$ modulo 11.

There is no known polynomial time algorithm for computing the discrete logarithm of a modulo p to the base r (when given the prime p, a root r modulo p, and a positive integer $a \in Z_p$). The problem plays a role in cryptography as will be discussed in Section 4.6.

Subsection 4

Applications of Congruences

FYIO—the entire section

- Hashing Functions
- Pseudorandom Numbers
- Check Digits

Hashing Functions I

FYIO

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

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

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

- $h(064212848) = 064212848 \bmod 111 = 14$
- $h(037149212) = 037149212 \bmod 111 = 65$

Hashing Functions II

FYIO

- $h(107405723) = 107405723 \bmod 111 = 14$, 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*:

$$h(k, i) = (h(k) + i) \bmod m, 0 \leq i < m$$

There are many other methods of handling with collisions. You may cover these in a later CS course.

Pseudorandom Numbers I

FYIO

- 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 recursively defined function

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

(an example of a recursive definition, discussed in Section 5.3)

Pseudorandom Numbers II

FYIO

- If psuedorandom numbers between 0 and 1 are needed, then the generated numbers are divided by the modulus, x_n/m.

Pseudorandom Numbers I

FYIO

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

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

Pseudorandom Numbers II

FYIO

The sequence generated is $3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$

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

FYIO

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:

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

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

Check Digits: UPCs...

FYIO

Solution:

$3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$

$21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$

$98 + x_{12} \equiv 0 \pmod{10}$

$x_{12} \equiv 2 \pmod{10}$. So, the check digit is 2.

$3 \cdot 0 + 4 + 3 \cdot 1 + 3 + 3 \cdot 3 + 1 + 3 \cdot 0 + 2 + 3 \cdot 1 + 6 + 3 \cdot 4 + 1 \equiv 0 \pmod{10}$

$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv 0 \pmod{10}$

Hence, 041331021641 is not a valid UPC.

See Wikipedia.

Check Digits: ISBNs

FYIO

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

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

The validity of an ISBN-10 number can be evaluated with the equivalent

- Suppose that the first 9 digits of the ISBN-10 are 007288008. What is the check digit?
- Is 084930149X[1] a valid ISBN10?

[1] X is used for the digit 10.

Check Digits: ISBNs...

FYIO

Solution:

- $X_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \pmod{11}$

  $X_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$

  $X_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $X_{10} = 2$.

- $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10 =$

  $0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \equiv 0 \pmod{11}$

  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. (see text for more details)

Subsection 5

Cryptography

Section Summary

- Classical Cryptography
- Cryptosystems
- Public Key Cryptography
- RSA Cryptosystem
- Cryptographic Protocols
- Primitive Roots and Discrete Logarithms

The RSA Cryptosystem

A public key cryptosystem, now known as the RSA system was introduced in 1976 by three researchers at MIT.

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

- The public encryption key is (n, e), where $n = pq$ (the modulus) is the product of two large (200 digits) primes p and q,
- An exponent e that is relatively prime to $(p-1)(q-1)$.
- The two large primes can be quickly found using probabilistic primality tests, discussed earlier.
- But $n = pq$, with approximately 400 digits, cannot be factored in a reasonable length of time.

RSA Encryption

To encrypt a message using RSA using a key (n, e):

- Translate the plaintext message M into sequences of two digit integers representing the letters. Use 00 for A, 01 for B, etc.
- Concatenate the two digit integers into strings of digits.
- Divide this string into equally sized blocks of $2N$ digits where $2N$ is the largest even number $2525\ldots25$ with $2N$ digits that does not exceed n.
- The plaintext message M is now a sequence of integers $m_1, m_2, \ldots, m_k$.
- Each block (an integer) is encrypted using $C = M^e \pmod{n}$.

RSA Encryption Example

Encrypt the message "STOP" using the RSA cryptosystem with key $(2537, 13)$.

- $2537 = 43 \cdot 59$, $p = 43$ and $q = 59$ are primes and
- $gcd(e, (p-1)(q-1)) = gcd(13, 42 \cdot 58) = 1$.

Translate the letters in "STOP" to their numerical equivalents 18 19 14 15.

- Divide into blocks of four digits (because $2525 < 2537 < 252525$) to obtain 1819 1415.
- Encrypt each block using the mapping $C = M^{13} \pmod{2537}$.
- Since $1819^{13} \pmod{2537} = 2081$ and $1415^{13} \pmod{2537} = 2182$, the encrypted message is 2081 2182.

RSA Decryption

- To decrypt an RSA ciphertext message, the decryption key d, an inverse of e modulo $(p-1)(q-1)$, is needed.
- The inverse exists since $gcd(e,(p-1)(q-1)) = 1$.
- With the decryption key d, we can decrypt each block with the computation $M = C^d \pmod{n}$.
- 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.

RSA Decryption Example

The message 0981 0461 is received. What is the decrypted message if it was encrypted using the RSA cipher from the previous example?

The message was encrypted with $n = 43 \cdot 59$ and exponent 13. An inverse of 13 modulo $42 \cdot 58 = 2436$ is $d = 937$.

- To decrypt a block C, $M = C^{937} \pmod{2537}$.
- Since $0981^{937} \pmod{2537} = 0704$ and $0461^{937} \pmod{2537} = 1115$,
  - the decrypted message is 0704 1115.
  - Translating back to English letters, the message is HELP.

Bibliography I


Ashutosh Gupta and S. Krishna.
Cs 228: Logic for computer science 2022.
https://www.cse.iitb.ac.in/ akg/courses/2022-logic/, January 2022.


Hyunyoung Lee.
Discrete structures for computing.
Class slides for TAMU CSCE 222, 2019.


Phillip Rogaway.
Ecs20 fall 2021 lecture notes, Fall 2021.


Wade Trappe and Lawrence Washington.
Introduction to Cryptography with Coding Theory.
Prentice Hall, 2002.