Heaven's light is our guide"

Rajshahi University of Engineering & Technology Department of Computer Science & Engineering

Network Security

Course No.: 305

Chapter 3: The Fundamentals: Algorithms, the Integers and

Matrices

Prepared By: Julia Rahman



Definition 1:

An algorithm is a finite sequence of precise instructions for performing a computation or for solving a problem.

Example 1: Describe an algorithm for finding the largest value in a finite sequence of integers.

Solution: We perform the following steps:

- 1. Set the temporary maximum equal to the first integer in the sequence
- 2. Compare the next integer in the sequence to the temporary maximum, and if it is larger that the temporary maximum, set the temporary maximum equal to this integer
- 3. Repeat the previous step if there are more integers in the sequence
- 4. Stop when there are no integers left in the sequence. The temporary maximum at this point is the largest integer in the sequence

Pseudocode: intermediate step between an English language description of an algorithm and an implementation of this algorithm in a programming language

Algorithm: Finding the maximum element in a finite sequence

```
Procedure max(a_1, a_2, ..., a_n): integer)

max := a_1

For i := 2 to n

If max < a_i then max := a_i

{max is the largest element}
```

Properties of an algorithm:

Input: An algorithm has input values from a specified set

Output: For each set of input an algorithm produces output from a specified set.

The output values are the solution to the problem

Definiteness: The steps of an algorithm must be defined precisely

Correctness: Algorithm should produce the correct output for each set of input.

Finiteness: Algorithm should produce the desired output after a finite number of steps for input in the set

Effectiveness: It must be possible to perform each step of an algorithm exactly and in a finite amount of time

Generality: Should be applicable for all problems of the desired form not just for a particular set of input values.

Searching Algorithms:

Problem: "Locate an element x in a list of distinct elements $a_1, a_2, ..., a_n$, or determine that it is not in the list."

We should provide as a solution to this search problem the location of the term in the list that equals x.

The linear search

Algorithm: The linear search algorithm

```
Procedure linear search(x: integer, a_1, a_2, ..., a_n: distinct integers)
i := 1
while (i \le n \text{ and } x \ne a_i)
i := i + 1
if i \le n then location := i
else location := 0
{location is the subscript of the term that equals x, or is 0 if x is not found}
```

The binary search:

- Constraint: can be used when the list has terms occurring in order of increasing size (words listed in lexicographic order)
- Methodology: Compare the element to be located to the middle term of the list
- **Example:** Search 19 in the list 1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22
 - ✓ split the list into 2 subsets with 8 terms each

1 2 3 5 6 7 8 10

12 13 15 16 18 19 20 22

✓ Compare 19 with the largest element of the first set

 $10 < 19 \Rightarrow$ search 19 in the second set

✓ Split the second subset into 2 smaller subsets

12 13 15 16

18 19 20 22

✓ Compare 19 with 16

 $16 < 19 \Rightarrow$ search 19 in the second set

- ✓ Split the second subset as:
- 18 19
- 20 22
- ✓ Compare 19 > 19 is false \Rightarrow search 19 in 18 19
- ✓ Split the subset as :

18

- 19
- ✓ Since $18 < 19 \Rightarrow$ search restricted to the second list
- ✓ Finally 19 is located at the 14th element of the original list Julia Rahman, Dept. CSE, RUET

Algorithm: the binary search algorithm

```
Procedure binary search (x: integer, a_1, a_2, ..., a_n: increasing integers)
i := 1 \ \{i \text{ is left endpoint of search interval}\}
j := n \ \{j \text{ is right endpoint of search interval}\}

While i < j

Begin

m := \lfloor (i+j)/2 \rfloor

If x > a_m then i := m+1

else j := m

End

If x := a_i then location := i

Else location := 0

{location is the subscript of the term equal to x, or 0 if x is not found}
```

The Big-O Notation

4Definition 1:

Let f and g be functions from integers or real numbers to real numbers. We say that f(x) is O(g(x)) if there are constants C and k such that $|f(x)| \le C|g(x)|$ whenever x > k.

- \checkmark "f(x) is big-oh of g(x)"
- ✓ witnesses: C, k

Another Definition

✓ Let f and g be functions from N to R.

Then g asymptotically dominates f, denoted f is O(g) or 'f is big-O of g,' or 'f is order g,' iff

$$\exists k \ \exists C \ \forall n \ [n > k \rightarrow |f(n)| \le C \ |g(n)|]$$

Note:

Choose k

Choose C; it may depend on your choice of k

Once you choose k and C, you must prove the truth of the implication (often by induction)

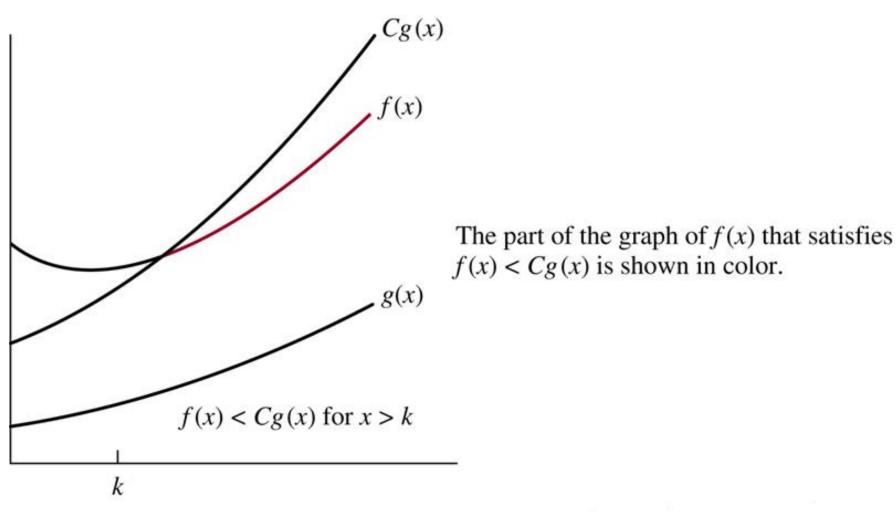


FIGURE 2 The Function f(x) is O(g(x)).

Example 1: Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$

Solution: Since

$$x^2+2x+1 \le x^2+2x^2+x^2 = 4x^2$$

whenever x > 1, it follows that f(x) is $O(x^2)$

take
$$C = 4$$
 and $k = 1$

Another:

If x > 2, we see that

$$x^2 + 2x + 1 \le x^2 + x^2 + x^2 = 3x^2$$

take
$$C = 3$$
 and $k = 2$

Example 2: Show that $7x^2$ is $O(x^3)$.

Solution: When x > 7,

then $7x^2 < x^3$ (multiplying both sides of x > 7 by x^2 .)

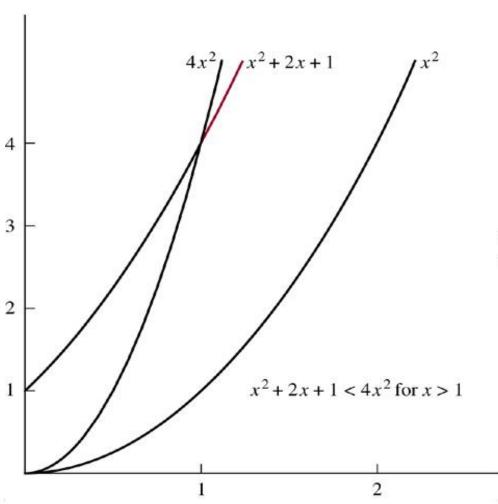
take C = 1 and k = 7 as witnesses to establish the relationship $7x^2$ is O (x^3) .

Alternatively,

when x > 1, we have $7x^2 < 7x^3$,

so that C = 7 and k = 1 are also witnesses to the relationship $7x^2$ is $O(x^3)$.

© The McGraw-Hill Companies, Inc. all rights reserved.



The part of the graph of $f(x) = x^2 + 2x + 1$ that satisfies $f(x) < 4x^2$ is shown in color.

FIGURE 1 The Function $x^2 + 2x + 1$ is $O(x^2)$.

Example 3: show that n^2 is not O(n).

Solution: To show that n^2 is not O (n),

- ✓ Must show no pair of constants C and k exist such that $n^2 \le Cn$ whenever n > k.
- ✓ To see that there are no such constants, when n > 0 and $n^2 \le Cn$ equivalent $n \le C$ (dividing both sides by n).
- ✓ Now see that no matter what C and k are, the inequality $n \le C$ cannot hold for all n with n > k.
- ✓ In particular, once set a value of k, see that when n is larger than the maximum of k and C, it is not true that $n \le C$ even though n > k.
- **Example 4:** Is it also true that x^3 is $O(7x^2)$?

Solution: To determine whether x^3 is $O(7x^2)$,

- ✓ Need to determine constants C and k such that $x^3 \le C$ (7x²) where x > k.
- ✓ $x^3 \le C(7x^2)$ is equivalent to $x \le 7C$, (dividing the inequality by x^2).
- ✓ No C exists for which $x \le 7C$ for all x > k, because x can be made arbitrarily large.
- ✓ It follows no witnesses C and k exist for this proposed big-O relationship. Hence, x^3 is not O(7 x^2). _{Julia Rahman, Dept. CSE, RUET} 13

Properties of Big-O:

- ✓ f is O(g) if and only if $O(f) \subseteq O(g)$
- ✓ If f is O(g) and g is O(f) then O(f) = O(g)
- \checkmark The set O(g) is closed under addition :

If f is
$$O(g)$$
 and h is $O(g)$ then $f + h$ is $O(g)$

- \checkmark The set O(g) is closed under multiplication by a scalar a (real number):
 - If f is O(g) then a*f is O(g)
 - that is,

O(g) is a vector space.

- \checkmark if f is O(g) and g is O(h), then f is O(h).
 - ✓ In particular

$$O(f) \subseteq O(g) \subseteq O(h)$$

Some Important Big-O Results

Theorem 1: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$, where $a_0, a_1, ..., a_{n-1}, a_n$ are real numbers. Then, f(x) is $O(x^n)$.

Proof: Using the triangle inequality, if x > 1 we have

$$|f(x)| = |a_{n}x^{n} + a_{n-1}x^{n-1} + \dots + a_{1}x + a_{0}|$$

$$\leq |a_{n}|x^{n} + |a_{n-1}|x^{n-1} + \dots + |a_{1}|x + |a_{0}|$$

$$= x^{n} (|a_{n}| + |a_{n-1}|/x + \dots + |a_{1}|/x^{n-1} + |a_{0}|/x^{n})$$

$$\leq x^{n} (|a_{n}| + |a_{n-1}| + \dots + |a_{1}| + |a_{0}|)$$

This shows that $|f(x)| \le Cx^n$

where $C = |a_n| + |a_{n-1}| + ... + |a_1| + |a_0|$ whenever x > 1. Hence, the witnesses $C = |a_n| + |a_{n-1}| + ... + |a_1| + |a_0|$ and k = 1 show that f(x) is $O(x^n)$.

Example 5: How can big-O notation be used to estimate the sum of the first n positive integers?

Solution: Because each of the integers in the sum of the first n positive integers does not exceed n, it follows that

$$1 + 2 + \ldots + n \le n + n + \ldots + n = n^2$$

From this inequality, it follows that 1 + 2 + 3 + ... + n is $O(n^2)$, taking C = 1 and k = 1 as witnesses.

Julia Rahman, Dept. CSE, RUET

15

The Growth of Combinations of Functions

© The McGraw-Hill Companies, Inc. all rights reserved.

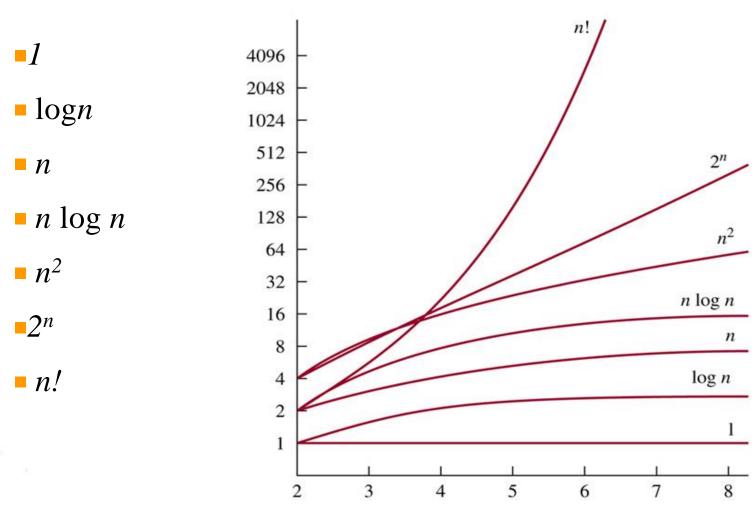


FIGURE 3 A Display of the Growth of Functions Commonly Used in Big-O Estimates.

Julia Rahman, Dept. CSE, RUET

16

Suppose that $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$. From the definition of big- O notation, there are constants C_1 , C_2 , k_1 and k_2 such that

$$|f_1(x)| \le C_1 |g_1(x)|$$
 when $x > k_1$ and $|f_2(x)| \le C_2 |g_2(x)|$ when $x > k_2$ and

To estimate the sum of $f_1(x)$ and $f_2(x)$, note that

$$| (f_1 + f_2)(x) | = | f_1(x) + f_2(x) |$$

$$\leq |f_1(x)| + |f_2(x)|$$
 using the triangle inequality $|a+b| \leq |a| + |b|$

When x i s greater than both k_1 and k_2 , it follows from the inequalities for $f_1(x)$ and $f_2(x)$ that

$$|f_1(x)| + |f_2(x)| \le C_1 |g_1(x)| + C_2 |g_2(x)|$$

 $\le C_1 |g(x)| + C_2 |g(x)|$
 $= (C_1 + C_2) |g(x)|$
 $= C|g(x)|$
where $C = C_1 + C_2$ and $g(x) = \max(|g_1(x)|, |g_2(x)|)$.

- **Theorem 2:** Suppose that $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$. Then $(f_1+f_2)(x)$ is $O(max(|g_1(x)|, |g_2(x)|))$
- **Corollary 1:** Suppose that $f_1(x)$ and $f_2(x)$ are both O(g(x)). Then $(f_1+f_2)(x)$ is

In a similar way big- O estimates can be derived for the product of the functions $f_1(x)$ and $f_2(x)$. When x is greater than $\max(k_1, k_2)$ it follows that

$$| (f_1 f_2) (x) | = |f_1(x)| |f_2(x)|$$

$$\leq C_1 |g_1(x)| |C_2 |g_2(x)|$$

$$\leq (C_1 C_2) |(g_1 g_2)(x)|$$

$$\leq C |(g_1 g_2)(x)|$$

where $C = C_1 C_2$. From this inequality, it follows that $f_1(x)f_2(x)$ is $O(g_1g_2)$, because there are constants C and k, namely, $C = C_1 C_2$ and $k = \max(k_1, k_2)$.

Theorem 3: Suppose that $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$. Then $(f_1f_2)(x)$ is $O(g_1(x)g_2(x))$

Big-Omega and Big-Theta Notation

- \triangleright Big-O notation does not provide a lower bound for the size of f(x) for large x.
- \triangleright For this, big-Omega (big- Ω) notation is used.
- When we want to give both an upper and a lower bound on the size of a function f(x), relative to a reference function g(x), we use big-Theta (big- Θ) notation.

Definition 2:

Let f and g be functions from integers or real numbers to real numbers. We say that f(x) is $\Omega(g(x))$ if there are positive constants C and k such that $|f(x)| \ge C|g(x)|$

whenever x > k.

"f(x) is big-Omega of g(x)"

Example 10 : The function $f(x) = 8x^3 + 5x^2 + 7$ is $\Omega(g(x))$, where g(x) is the function $g(x) = x^3$.

This is easy to see because $f(x) = 8x^3 + 5x^2 + 7 \ge x^3$ for all positive real numbers x. This is equivalent to saying that $g(x) = x^3$ is $O(8x^3 + 5x^2 + 7)$, which can be established directly by turning the inequality around.

Definition 3:

Let f and g be functions from integers or real numbers to real numbers. We say that f(x) is $\Theta(g(x))$ if f(x) is O(g(x)) and f(x) is $\Omega(g(x))$.

- "f(x) is big-Theta of g(x)"
- "f(x) is of order g(x)
- f(X) is $\Theta(g(x))$, then g(x) is $\Theta(f(x))$
- **Example 12:** Show that $3x^2 + 8x(\log x)$ is $\Theta(x^2)$.

Solution: Because $0 \le 8x \log x \le 8x^2$, it follows that $3x^2 + 8x \log x \le 11x^2$ for x > 1. Consequently, $3x^2 + 8x \log x$ is $O(x^2)$. Clearly, x^2 is $O(3x^2 + 8x \log x)$. Consequently, $3x^2 + 8x \log x$ is $O(x^2)$.

Theorem 4: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$, where $a_0, a_1, ..., a_{n-1}, a_n$ are real numbers with $a_n \neq 0$. Then, f(x) is of order x^n .

3.3 Complexity of Algorithms

3.3 Complexity of Algorithms

- **Time Complexity:** Determine the approximate number of operations required to solve a problem of size n.
- **♣ Space Complexity:** Determine the approximate memory required to solve a problem of size n.
- **☐** Time Complexity
 - ✓ Use the Big-O notation
 - ✓ Ignore house keeping
 - ✓ Count the <u>expensive</u> operations only
 - ✓ Basic operations:
 - searching algorithms key comparisons
 - sorting algorithms list component comparisons
 - numerical algorithms floating point ops. (flops) multiplications/divisions and/or additions/subtractions

Worst Case: maximum number of operations

Average Case: mean number of operations assuming an input probability distribution

3.3 Complexity of Algorithms

Examples:

Multiply an n x n matrix A by a scalar c to produce the matrix B:

```
\label{eq:procedure} \begin{split} \textbf{procedure} & (n, \, c, \, A, \, B) \\ \textbf{for} & i \, \textbf{from} \, 1 \, to \, n \, do \\ \textbf{for} & j \, \textbf{from} \, 1 \, to \, n \, do \\ B(i, \, j) &= cA(i, \, j) \\ \textbf{end} & \textbf{do} \\ \textbf{end} & \textbf{do} \end{split}
```

Analysis (worst case):

```
Count the number of floating point multiplications. n^2 elements requires n^2 multiplications. time complexity is O(n^2) or quadratic complexity.
```

4 Three applications of modular arithmetic:

- 1) Generating pseudorandom numbers,
- 2) Assigning computer memory locations to files, and
- 3) Encrypting and decrypting messages.

Division

Definition 1:

If a and b are integers with $a\neq 0$, we say that a divides b (a|b) if there is an integer c such that b=ac.

- a is a factor of b
- b is a multiple of a
- a b if a does not divide b
- **EXAMPLE 1:** Determine whether 3 | 7 and whether 3 | 12.

4 Theorem 1:

Let a, b, c be integers. Then

- 1) if a|b and a|c, then a|(b+c)
- 2) if a|b, then a|bc for all integers c
- 3) if a|b and b|c, then a|c

Proof:

Proof of (1).

- ✓ Suppose that a | b and a | c.
- \checkmark From the definition of divisibility, it follows that there are integers s and t with b = as and c = at.
- \checkmark Hence, b + c = as + at = a(s + t).
- \checkmark Therefore, a divides b + c. This establishes part (i) of the theorem.
- Proof of 2 and 3 is homework

Corollary 1:

If a, b, and c are integers such that a|b and a|c, then a|mb+nc whenever m and n are integers.

Proof: By part (ii) of Theorem 1 it follows that a | mb and a | nc whenever m and n are integers. By part (i) of Theorem 1 it follows that a | mb + nc.

The Division Algorithm

Theorem 2: (The Division Algorithm)

Let a be an integer and d a positive integer. Then there are unique integers q and r, with 0 <= r < d, such that a = dq + r.

Lesson 4 Definition 2:

In the equality given in the division algorithm,

d is called the divisor,

a is called the dividend,

q is called the quotient, and

r is called the remainder.

This notation is used to express the quotient and remainder:

$$q = a \operatorname{div} d$$
, $r = a \operatorname{mod} d$.

EXAMPLE 3: What are the quotient and remainder when 101 is divided by 11? **Solution:** We have $101 = 11 \cdot 9 + 2$.

Hence, the quotient when 101 is divided by 11 is 9 = 101 div 11, and the remainder is 2 = 101 mod 11.

EXAMPLE 4: What are the quotient and remainder when - 11 is divided by 3? **Solution:** We have -11 = 3(-4) + 1.

Hence, the quotient when - 11 is divided by 3 is -4 = - 11 div 3, and the remainder is $1 = -11 \mod 3$.

Note that the remainder cannot be negative. Consequently, the remainder is not - 2, even though

$$-11 = 3(-3) - 2$$
,

because r = -2 does not satisfy $0 \le r < 3$.

Modular Arithmetic

4 Definition 3:

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.

$$a \equiv b \pmod{m}$$

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 $a \mod m = b \mod m$.

EXAMPLE 5: Determine whether 1 7 is congruent to 5 modulo 6 and whether 24 and 1 4 are congruent modulo 6.

Solution: Because 6 divides 17 - 5 = 12, we see that $17 \equiv 5 \pmod{6}$. However, because 24 - 14 = 10 is not divisible by 6, we see that $24 \not\equiv 1 \pmod{6}$.

4 Theorem 4:

Let m be a positive integer. The integers a and b are congruent modulo m iff there is an integer k such that a=b+km.

Proof:

- ✓ If $a \equiv b \pmod{m}$, then $m \mid (a b)$.
- ✓ There is an integer k such that a b = km, so that a = b + km
- \checkmark Conversely, if there is an integer k such that a = b + km, then km = a b.
- ✓ Hence, m divides a b, so that $a \equiv b \pmod{m}$.
- ❖ The set of all integers congruent to an integer a modulo m is called the congruence class of a modulo m.

Theorem 5:

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a+c \equiv b+d \pmod{m}$ m) and $ac \equiv bd \pmod{m}$.

Proof:

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

EXAMPLE 6:

Because $7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$, it follows from Theorem 5 that $18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$ and that $77 = 7 \cdot 11 == 2 \cdot 1 = 2 \pmod{5}$.

Corollary 2:

Let m be a positive integer and let a and b be integers. Then $(a+b) \mod m = ((a \mod m) + (b \mod m)) \mod m$ and $ab \mod m = ((a \mod m)(b \mod m)) \mod m.$

Proof: By the definitions mod m and the definition of congruence modulo m, we know that $a \equiv (a \mod m) \pmod m$ and $b \equiv (b \mod m) \pmod m$. Hence, Theorem 5 tell us that $a + b \equiv (a \mod m) + (b \mod m) \pmod m$ and $ab \equiv (a \mod m) + (b \mod m)$ $mod\ m$)(b $mod\ m$) (mod m) . $_{Julia\ Rahman,\ Dept.\ CSE,\ RUET}$

Applications of Congruences:

✓ Hashing functions

$$h(k) = k \mod m$$

✓ Pseudorandom numbers

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

✓ Cryptology

Caesar cipher:
$$f(p) = (p+k) \mod 26$$

Primes

Definition 1:

- ✓ A positive integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p.
- ✓ Integer n is *composite* if and only if there exists an integer a such that a|n and 1<a<n.

EXAMPLE 1:

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

Theorem 1: (Fundamental Theorem of Arithmetic):

Every positive integer greater than 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 nondecreasing size.

EXAMPLE 2:

The prime factorizations of 1 00, 64 1, 999, and 1 024 are given by

4 Theorem 2:

If n is a composite integer, then n has a prime divisor less than or equal to \sqrt{n} .

EXAMPLE 3: Show that 101 is prime.

Solution: The only primes not exceeding $\sqrt{101}$ are 2, 3, 5, and 7. Because 101 is not divisible by 2, 3, 5, or 7 (the quotient of 101 and each of these integers is not an integer), it follows that 101 is prime.

EXAMPLE 4: Find the prime factorization of 7007.

Solution:

- ✓ To find the prime factorization of 7007, first perform divisions of 7007 by successive primes, beginning with 2.
- ✓ 7 divides 7007, with 7007/7 = 1001.
- ✓ 1001/7 = 143 and 143/11 = 13.
- ✓ It follows that the prime factorization of 7007 is $7*7*11*13 = 7^2*11*13$.

Theorem 3:

There are infinitely many primes.

Mersenne primes - $2^p - 1$, where p is also prime

- **EXAMPLE 5:** The numbers $2^2 1 = 3$, $2^3 1 = 7$, and $2^5 1 = 31$ are Mersenne primes, while $2^{11} 1 = 2047$ is not a Mersenne prime because $2047 = 23 \cdot 89$.
- **Theorem 4:** (Prime Number Theorem)

 The ratio of the number of primes not exceeding x and x/ln x approaches 1 as x grows without bound.
- **Example 6:** $f(n)=n^2-n+41$, for n not exceeding 40 For every polynomial f(n) with integer coefficients, there is a positive integer y such that f(y) is composite.
- **Example 7:** Goldbach's Conjecture (1742): every even integer n, n>2, is the sum of two primes.
- **Example 8:** There are infinitely many primes of the form n^2+1 , where n is a positive integer.
- **4** Example 9:
 - ✓ Twin Prime Conjecture: There are infinitely many twin primes.
 - \checkmark Twin primes are primes that differ by 2.
 - ✓ such as 3 and 5, 5 and 7, 1 1 and 1 3, 1 7 and 1 9, and 4967 and 4969.

Definition 2:

Let a and b be integers, not both zero. The largest integer d such that d|a and d|b is called the *greatest common divisor* of a and b. (or gcd(a, b))

- **EXAMPLE 10:** What is the greatest common divisor of 24 and 36? **Solution:** The positive common divisors of 24 and 36 are 1, 2, 3, 4, 6, and 12. Hence, gcd(24, 36) = 12.
- **EXAMPLE 11:** What is the greatest common divisor of 17 and 22? **Solution:** The integers 17 and 22 have no positive common divisors other than 1, so that gcd(17, 22) = 1.
- **◆ Definition 3:** The integers a and b are *relatively prime* if their gcd is 1.
- **EXAMPLE 12:** That the integers 17 and 22 are relatively prime, because gcd(17, 22) = 1.
- Definition 4:

The integers a_1 , ..., a_{n-1} , a_n are pairwise relatively prime if $gcd(a_i, a_j)=1$ whenever $1 \le i < j \le n$.

3.5 Primes and Greatest Common Divisors

EXAMPLE 13: Determine whether the integers 10, 17, and 21 are pairwise relatively prime and whether the integers 10, 19, and 24 are pairwise relatively prime.

Solution: Because gcd(10, 17) = 1, gcd(10, 21) = 1, and gcd(17, 21) = 1, we conclude that 10, 17, and 21 are pairwise relatively prime. Because gcd(10, 24) = 2 > 1, we see that 10, 19, and 24 are not pairwise relatively prime.

- **EXAMPLE 14:** Because the prime factorizations of 120 and 500 are $120 = 2^3*3*5$ and $500 = 2^2*5^3$, the greatest common divisor is $gcd(120, 500) = 2^{min(3,2)}*3^{min(1,0)}*5^{min(1,3)} = 2^2*3^0*5^1 = 20$

4 Definition 5:

The *least common multiple* of positive integers a and b is the smallest positive integer that is divisible by both a and b. (or lcm(a, b)) $lcm(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \dots p_n^{\max(a_n, b_n)}$

3.5 Primes and Greatest Common Divisors

- **EXAMPLE 15:** What is the least common multiple of $2^3 3^5 7^2$ and $2^4 3^3$? **Solution:** $lcm(2^3 3^5 7^2, 2^4 3^3) = 2^{max(3,4)} 3^{max(5,3)} 7^{max(2,0)} = 2^4 3^5 7^2$
- **♣** Theorem 5:

Let a and b be positive integers. Then ab=gcd(a,b)*lcm(a,b)

3.6 Integers and Algorithms

3.6 Integers and Algorithms

✓ Representation of Integers

Decimal, binary, octal, hexadecimal

Theorem 1:

Let b be a positive integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form

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

where k is a nonnegative integer, a_0 , a_1 , ..., a_k are nonnegative integers less than b, and $a_k \neq 0$.

- \triangleright Base b expansion of n: $(a_k a_{k-1} ... a_1 a_0)_b$
- ➤ Binary expansion, hexadecimal expansion
- ➤ Base conversion
- **Algorithm 1:** Constructing Base b Expansions

Procedure base b expansion(n: positive integer)

3.6 Integers and Algorithms

✓ Computing the greatest common divisor using prime factorizations of integers is inefficient because of time consuming to find prime factorizations. More efficient method of finding the greatest common divisor, called the Euclidean algorithm.

Euclidean Algorithm:

```
Procedure gcd(a, b: positive integers) gcd (6,12):

x:=a

y:=b

while y\neq0

while y\neq0

begin

r:=x mod y

x:=y

y:=r

end {gcd(a,b) is x}

gcd (6,12):

x = 6

while y \neq 0

x = 12

while y \neq 0

r = 12 m

x = 6

while y = 0
```

```
gcd (6,12):

x = 6 y = 12

while y \neq 0

r = 6 \mod 12 = 6

x = 12 y = 6

while y \neq 0

r = 12 \mod 6 = 0

x = 6 y = 0

while y = 0, end. \therefore gcd (6,12) = 6
```

Example: Find gcd of 414 and 662 using the Euclidean algorithm.

```
Solution: 662 = 414 * 1 + 248

414 = 248 * 1 + 166

248 = 166 * 1 + 82

166 = 82 * 2 + 2

82 = 2 * 41
```

Hence, gcd(414, 662) = 2, because 2 is the last nonzero remainder.

ightharpoonup gcd(a,b) can be expressed as a linear combination with integer coefficients of a and b.

♣ Theorem 1:

If a and b are positive integers, then there exist integers s and t such that gcd(a,b) = sa+tb.

Example 1 Express gcd(252, 198) = 18 as a linear combination of 252 and 198.

Solution:

Exercise: 1(g)

Linear Congruences

- A congruence of the form $ax \equiv b \pmod{m}$ where m is a positive integer, a and b are integers, and x is variable, is called a **linear congruence.**
- ✓ Such congruences arise throughout number theory and its applications.
- ✓ How can we solve the linear congruence $ax \equiv b \pmod{m}$? That is, find the x that satisfy this congruence.
- ✓ One method that we will describe uses an integer \bar{a} such that $a\bar{a}\equiv 1 \pmod{m}$, if such an integer exist.
- ✓ Such an integer \bar{a} is said to be an **inverse** of a modulo m.
- ✓ Theorem 3 guarantees that an inverse of a modulo m exists whenever a and m are relatively prime.

4 Theorem 3:

If a and m are relatively prime integers and m>1, then an inverse of modulo m exist. Furthermore, this inverse is unique modulo m. (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.)

When we have an inverse of a modulo m, $ax \equiv 1 \pmod{m}$, we solve the congruence $ax \equiv b \pmod{m}$.

Proof: By Theorem 1, because gcd(a, m) = 1, there are integers s and t such that

sa + tm = 1.

This implies that

 $sa + tm \equiv 1 \pmod{m}$.

Because $tm \equiv 0 \pmod{m}$, it follows that

 $sa \equiv 1 \pmod{m}$.

Consequently, s is an inverse of a modulo m.

EXAMPLE 3: Find an inverse of 3 modulo 7.

Solution: Because gcd(3, 7) = 1, Theorem 3 tells us that an inverse of 3 modulo 7 exists. Using Euclidean algorithm to find greatest common divisor of 3 and 7: $7 = 2 \cdot 3 + 1$.

From this equation we see that

$$-2 \cdot 3 + 1 \cdot 7 = 1$$
.

- ♣ This shows that -2 is an inverse of 3 modulo 7. (Note that every integer congruent to -2 modulo 7 is also an inverse of 3, such as 5, -9, 12, and so on.)
- **EXAMPLE 4:** What are the solutions of the linear congruence $3x \equiv 4 \pmod{7}$? **Solution:** By Example 3 know that -2 is an inverse of 3 modulo 7. $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$ [Multiplying both sides of the congruence by -2] Because $-6 \equiv 1 \pmod{7}$ and $-8 \equiv 6 \pmod{7}$, it follows that if x is a solution, then $x \equiv -8 \equiv 6 \pmod{7}$.

We need to determine whether every x with $x \equiv 6 \pmod{7}$ is a solution. Assume that $x \equiv 6 \pmod{7}$. Then $3x \equiv 3 \cdot 6 = 1 \ 8 \equiv 4 \pmod{7}$, which shows that all such x satisfy the congruence. We conclude that the solutions to the congruence are the integers x such that $x \equiv 6 \pmod{7}$, namely, 6, 1 3, 20, . . . and - 1, -8,-15,

Pseudoprimes

- **4** Theorem 5: 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

$$a^p \equiv a \pmod{p}$$

- ✓ Unfortunately, there are composite integer n, such that $2^{n-l} \equiv 1 \pmod{p}$. Such integers are called pseudoprimes to the base 2.
- **EXAMPLE 9:** The integer 341 is a pseudoprime to the base 2 because it is composite (341 = 11 * 31) and as $2^{340} \equiv 1 \pmod{341}$

Definition 1:

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

Definition 2:

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

- **Example 10:** The integer 561 is a Carmichael number.
 - ✓ To see this, first note that 561 is composite because 561 = 3*11*17.
 - ✓ Next, note that if gcd(b, 561) = 1, then gcd(b, 3) = gcd(b, 11) = gcd(b, 17) = 1.
 - ✓ Using Fermat's Little Theorem we find that $b^2 \equiv 1 \pmod{3}, b^{10} \equiv 1 \pmod{11}, \text{ and } b^{16} \equiv 1 \pmod{17}.$
 - ✓ It follows that

$$b^{560} = (b^2)^{280} \equiv 1 \pmod{3},$$

 $b^{560} = (b^{10})^{56} \equiv 1 \pmod{11},$
 $b^{560} = (b^{16})^{35} \equiv 1 \pmod{17}.$

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

Exercise 18. Find all solutions to the system of congruences $x \equiv 2 \pmod{3}$ $x \equiv 1 \pmod{4}$ $x \equiv 3 \pmod{5}$

Sol:

$$a_1=2$$
, $a_2=1$, $a_3=3$,
 $m_1=3$, $m_2=4$, $m_3=5$ $m=3\times4\times5=60$
 $M_1=20$, $M_2=15$, $M_3=12$
 $20\equiv 2 \pmod{3} \Rightarrow 20\times2\equiv 1 \pmod{3}$
 $15\equiv 3 \pmod{4} \Rightarrow 15\times3\equiv 1 \pmod{4}$
 $12\equiv 2 \pmod{5} \Rightarrow 12\times3\equiv 1 \pmod{5}$
 $\therefore x = 2\times20\times2+1\times15\times3+3\times12\times3$
 $= 80+45+108=233\equiv 53 \pmod{60}$

Ex 21. Find all solutions, if any, to the system of congruences.

$$x\equiv 7 \pmod{9}$$

 $x\equiv 4 \pmod{12}$
 $x\equiv 16 \pmod{21}$

Sol. Rewrite the system as the following:

```
x\equiv 7 \pmod{9}

x\equiv 1 \pmod{3} x\equiv 0 \pmod{4}

x\equiv 1 \pmod{3} x\equiv 2 \pmod{7}

i.e.,

x\equiv 7 \pmod{9} x\equiv 0 \pmod{4} x\equiv 0 \pmod{4} ...

x\equiv 2 \pmod{7} x\equiv 2 \pmod{7}
```