Lets continue on **Matrices**

Definition 1 A *matrix* is a rectangular array of numbers. A matrix with m rows and n columns is called an $m \times n$ matrix. The plural of matrix is *matrices*. A matrix with the same number of rows as columns is called *square*. Two matrices are *equal* if they have the same number of rows and the same number of columns and the corresponding entries in every position are equal.

EXAMPLE 1 The matrix $\begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$ is a 3×2 matrix.

◀

Matrices

Definition 2 Let m and n be positive integers and let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

The ith *row* of $\mathbf{A}$ is the $1 \times n$ matrix $[a_{i1}, a_{i2}, \dots, a_{in}]$. The jth *column* of $\mathbf{A}$ is the $m \times 1$ matrix

$$\begin{bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{mj} \end{bmatrix}.$$

The (i, j)th *element* or *entry* of $\mathbf{A}$ is the element a_{ij}, that is, the number in the ith row and jth column of $\mathbf{A}$. A convenient shorthand notation for expressing the matrix $\mathbf{A}$ is to write $\mathbf{A} = [a_{ij}]$, which indicates that $\mathbf{A}$ is the matrix with its (i, j)th element equal to a_{ij}.

Matrix Arithmetic

The basic operations of matrix arithmetic will now be discussed, beginning with a definition of matrix addition.

Definition 3 Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ be $m \times n$ matrices. The *sum* of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} + \mathbf{B}$, is the $m \times n$ matrix that has $a_{ij} + b_{ij}$ as its (i, j)th element. In other words, $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]$.

The sum of two matrices of the same size is obtained by adding elements in the corresponding positions. Matrices of different sizes cannot be added, because such matrices will not both have entries in some of their positions.

EXAMPLE 2

We have $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & -3 \\ 3 & 4 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 & -1 \\ 1 & -3 & 0 \\ -1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -2 \\ 3 & -1 & -3 \\ 2 & 5 & 2 \end{bmatrix}.$ ◀

We now discuss matrix products. A product of two matrices is defined only when the number of columns in the first matrix equals the number of rows of the second matrix.

Definition 4 Let $\mathbf{A}$ be an $m \times k$ matrix and $\mathbf{B}$ be a $k \times n$ matrix. The *product* of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{AB}$, is the $m \times n$ matrix with its (i, j)th entry equal to the sum of the products of the corresponding elements from the ith row of $\mathbf{A}$ and the jth column of $\mathbf{B}$. In other words, if $\mathbf{AB} = [c_{ij}]$, then

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}.$$

Matrix Arithmetic

EXAMPLE 3 Let

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 4 \\ 2 & 1 & 1 \\ 3 & 1 & 0 \\ 0 & 2 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 2 & 4 \\ 1 & 1 \\ 3 & 0 \end{bmatrix}.$$

Find **AB** if it is defined.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ik} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mk} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ b_{k1} & b_{k2} & \dots & b_{kj} & \dots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & c_{ij} & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

FIGURE 1 **The product of $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$.**

Matrix Arithmetic

Extra Examples

Solution: Because **A** is a 4×3 matrix and **B** is a 3×2 matrix, the product **AB** is defined and is a 4×2 matrix. To find the elements of **AB**, the corresponding elements of the rows of **A** and the columns of **B** are first multiplied and then these products are added. For instance, the element in the (3, 1)th position of **AB** is the sum of the products of the corresponding elements of the third row of **A** and the first column of **B**; namely, $3 \cdot 2 + 1 \cdot 1 + 0 \cdot 3 = 7$. When all the elements of **AB** are computed, we see that

$$\mathbf{AB} = \begin{bmatrix} 14 & 4 \\ 8 & 9 \\ 7 & 13 \\ 8 & 2 \end{bmatrix}.$$

EXAMPLE 4 Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Does $\mathbf{AB} = \mathbf{BA}$?

Solution: We find that

$$\mathbf{AB} = \begin{bmatrix} 3 & 2 \\ 5 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{BA} = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}.$$

Hence, $\mathbf{AB} \neq \mathbf{BA}$.

Transposes and Powers of Matrices

Definition 5 The *identity matrix of order n* is the $n \times n$ matrix $\mathbf{I}_n = [\delta_{ij}]$, (the *Kronecker delta*) where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Hence,

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

Multiplying a matrix by an appropriately sized identity matrix does not change this matrix. In other words, when $\mathbf{A}$ is an $m \times n$ matrix, we have

$$\mathbf{A}\mathbf{I}_n = \mathbf{I}_m\mathbf{A} = \mathbf{A}.$$

Powers of square matrices can be defined because matrix multiplication is associative. When $\mathbf{A}$ is an $n \times n$ matrix, we have

$$\mathbf{A}^0 = \mathbf{I}_n, \qquad \mathbf{A}^r = \underbrace{\mathbf{A}\mathbf{A}\mathbf{A}\cdots\mathbf{A}}_{r \text{ times}}.$$

The operation of interchanging the rows and columns of a square matrix arises in many contexts.

Transposes and Powers of Matrices

Definition 6 Let $\mathbf{A} = [a_{ij}]$ be an $m \times n$ matrix. The *transpose* of $\mathbf{A}$, denoted by $\mathbf{A}^t$, is the $n \times m$ matrix obtained by interchanging the rows and columns of $\mathbf{A}$. In other words, if $\mathbf{A}^t = [b_{ij}]$, then $b_{ij} = a_{ji}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

EXAMPLE 5 The transpose of the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ is the matrix $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$.

Definition 7 A square matrix $\mathbf{A}$ is called *symmetric* if $\mathbf{A} = \mathbf{A}^t$. Thus, $\mathbf{A} = [a_{ij}]$ is symmetric if $a_{ij} = a_{ji}$ for all i and j with $1 \leq i \leq n$ and $1 \leq j \leq n$.

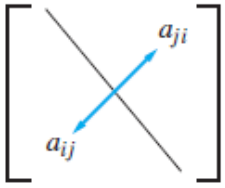


FIGURE 2 **A symmetric matrix.**

Note that a matrix is symmetric if and only if it is square and it is symmetric with respect to its main diagonal (which consists of entries that are in the ith row and ith column for some i). This symmetry is displayed in Figure 2.

EXAMPLE 6 The matrix $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ is symmetric.

Transposes and Powers of Matrices

Definition 9 Let $\mathbf{A} = [a_{ij}]$ be an $m \times k$ zero–one matrix and $\mathbf{B} = [b_{ij}]$ be a $k \times n$ zero–one matrix. Then the *Boolean product* of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} \odot \mathbf{B}$, is the $m \times n$ matrix with (i, j)th entry c_{ij} where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdots \vee (a_{ik} \wedge b_{kj}).$$

Note that the Boolean product of $\mathbf{A}$ and $\mathbf{B}$ is obtained in an analogous way to the ordinary product of these matrices, but with addition replaced with the operation $\vee$ and with multiplication replaced with the operation $\wedge$. We give an example of the Boolean products of matrices.

EXAMPLE 8 Find the Boolean product of $\mathbf{A}$ and $\mathbf{B}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Solution: The Boolean product $\mathbf{A} \odot \mathbf{B}$ is given by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) \\ (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \end{bmatrix}$$

$$= \begin{bmatrix} 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \\ 0 \vee 0 & 0 \vee 1 & 0 \vee 1 \\ 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Enough Mathematical Appetizers!

Let us look at something more interesting:

Algorithms

Algorithms

What is an algorithm?

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

This is a rather vague definition. You will get to know a more precise and mathematically useful definition when you attend CMSC441.

But this one is good enough for now…

Algorithms

Properties of algorithms:

- **Input** from a specified set,
- **Output** from a specified set (solution),
- **Definiteness** of every step in the computation,
- **Correctness** of output for every possible input,
- **Finiteness** of the number of calculation steps,
- **Effectiveness** of each calculation step and
- **Generality** for a class of problems.

Algorithm Examples

We will use a pseudocode to specify algorithms, which slightly reminds us of Basic and Pascal.

Example: an algorithm that finds the maximum element in a finite sequence

procedure max(a_1, a_2, …, a_n: integers)
max := a_1
for i := 2 **to** n
    if max < a_i **then** max := a_i
{max is the largest element}

Algorithm Examples

Another example: a linear search algorithm, that is, an algorithm that linearly searches a sequence for a particular element.

procedure linear_search(x: integer; a_1, a_2, …, a_n: integers)

i := 1

while (i ≤ n and x ≠ a_i)

    i := i + 1

if i ≤ n **then** location := i

else location := 0

{location is the subscript of the term that equals x, or is zero if x is not found}

Algorithm Examples

If the terms in a sequence are ordered, a binary search algorithm is more efficient than linear search.

The binary search algorithm iteratively restricts the relevant search interval until it closes in on the position of the element to be located.

Algorithm Examples

binary search for the letter ‘j’

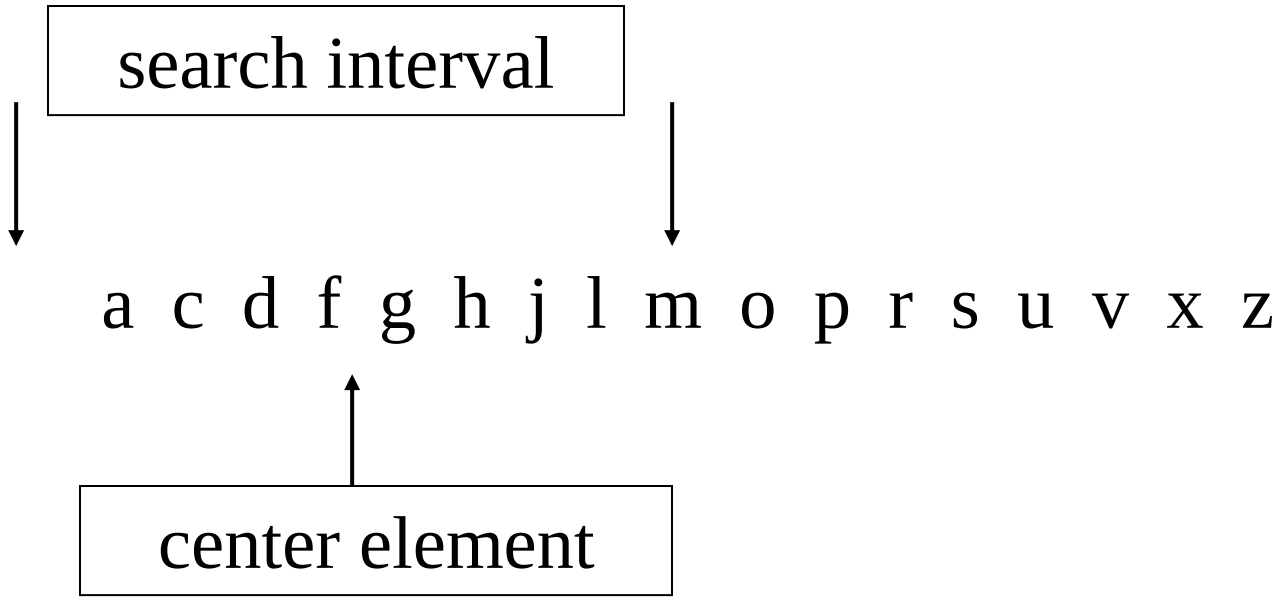
search interval

a c d f g h j l m o p r s u v x z

center element

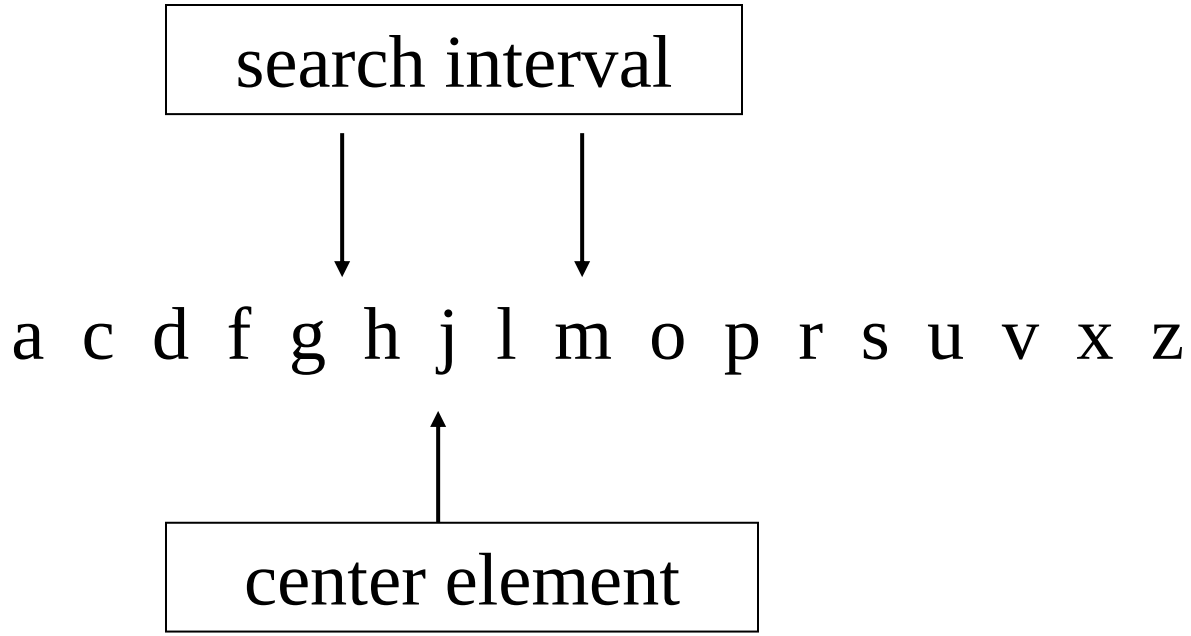
Algorithm Examples

binary search for the letter ‘j’



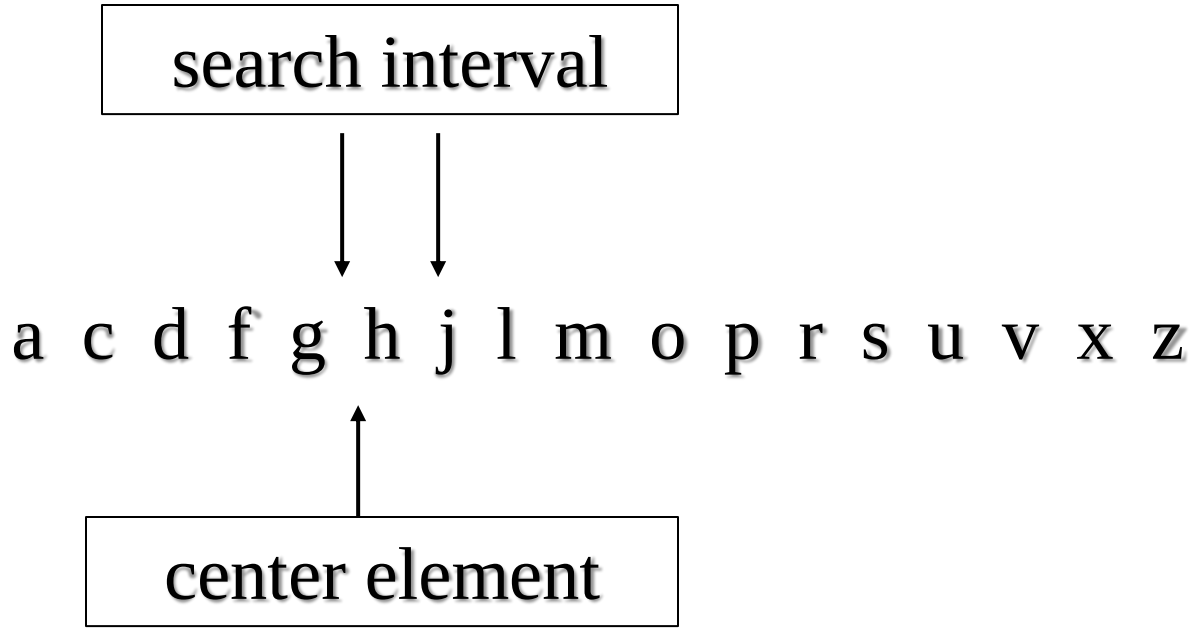
Algorithm Examples

binary search for the letter ‘j’



Algorithm Examples

binary search for the letter ‘j’



Algorithm Examples

binary search for the letter ‘j’

search interval

a c d f g h j l m o p r s u v x z

found !

Algorithm Examples

procedure binary_search(x: integer; a_1, a_2, …, a_n: integers)

i := 1 {i is left endpoint of search interval}

j := n {j 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 that equals x, or is zero if x is not found}

Let us get into…

Number Theory

Introduction to Number Theory

Number theory is about **integers** and their properties.

We will start with the basic principles of

- divisibility,
- greatest common divisors,
- least common multiples, and
- modular arithmetic

and look at some relevant algorithms.

Division

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

When a divides b we say that a is a **factor** of b and that b is a **multiple** of a.

The notation **a | b** means that a divides b.

We write **a χ b** when a does not divide b
(see book for correct symbol).

Divisibility Theorems

For integers a, b, and c it is true that

- if a | b and a | c, then a | (b + c)
  Example: 3 | 6 and 3 | 9, so 3 | 15.

- if a | b, then a | bc for all integers c
  Example: 5 | 10, so 5 | 20, 5 | 30, 5 | 40, …

- if a | b and b | c, then a | c
  Example: 4 | 8 and 8 | 24, so 4 | 24.

Primes

A positive integer p greater than 1 is called prime if the only positive factors of p are 1 and p.
Note: 1 is not a prime

A positive integer that is greater than 1 and is not prime is called composite.

The fundamental theorem of arithmetic:

Every positive integer can be written **uniquely** as the **product of primes**, where the prime factors are written in order of increasing size.

Primes

Examples:

$15 = 3 \cdot 5$

$48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3$

$17 = 17$

$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$

$512 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^9$

$515 = 5 \cdot 103$

$28 = 2 \cdot 2 \cdot 7$

Primes

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

This is easy to see: if n is a composite integer, it must have at least two prime divisors. Let the largest two be p_1 and p_2. Then $p_1 \cdot p_2 <= n$.

p_1 and p_2 cannot both be greater than $\sqrt{n}$, because then $p_1 \cdot p_2 > n$.

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**.

In the above equation,

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

The Division Algorithm

Example:

When we divide 17 by 5, we have

$17 = 5 \cdot 3 + 2.$

- 17 is the dividend,
- 5 is the divisor,
- 3 is called the quotient, and
- 2 is called the remainder.

The Division Algorithm

Another example:

What happens when we divide -11 by 3 ?

Note that the remainder cannot be negative.

$-11 = 3 \cdot (-4) + 1.$

- -11 is the dividend,
- 3 is the divisor,
- -4 is called the quotient, and
- 1 is called the remainder.

Greatest Common Divisors

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.

The greatest common divisor of a and b is denoted by gcd(a, b).

Example 1: What is gcd(48, 72) ?

The positive common divisors of 48 and 72 are 1, 2, 3, 4, 6, 8, 12, 16, and 24, so gcd(48, 72) = 24.

Example 2: What is gcd(19, 72) ?

The only positive common divisor of 19 and 72 is 1, so gcd(19, 72) = 1.

Greatest Common Divisors

Using prime factorizations:

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$,
where $p_1 < p_2 < \ldots < p_n$ and $a_i, b_i \in \mathbf{N}$ for $1 \leq i \leq n$

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

Example:

$a = 60 = 2^2 3^1 5^1$

$b = 54 = 2^1 3^3 5^0$

$\gcd(a, b) = 2^1 3^1 5^0 = 6$

Relatively Prime Integers

Definition:

Two integers a and b are **relatively prime** if gcd(a, b) = 1.

Examples:

Are 15 and 28 relatively prime?

Yes, gcd(15, 28) = 1.

Are 55 and 28 relatively prime?

Yes, gcd(55, 28) = 1.

Are 35 and 28 relatively prime?

No, gcd(35, 28) = 7.

Relatively Prime Integers

Definition:

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

Examples:

Are 15, 17, and 27 pairwise relatively prime?
No, because gcd(15, 27) = 3.

Are 15, 17, and 28 pairwise relatively prime?
Yes, because gcd(15, 17) = 1, gcd(15, 28) = 1 and gcd(17, 28) = 1.

Least Common Multiples

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.

We denote the least common multiple of a and b by lcm(a, b).

Examples:

lcm(3, 7) = 21

lcm(4, 6) = 12

lcm(5, 10) = 10

Least Common Multiples

Using prime factorizations:

$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$,
where $p_1 < p_2 < \ldots < p_n$ and $a_i, b_i \in \mathbf{N}$ for $1 \leq i \leq n$

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

Example:

$a = 60 = 2^2 3^1 5^1$

$b = 54 = 2^1 3^3 5^0$

$\text{lcm}(a, b) = 2^2 3^3 5^1 = 4 \cdot 27 \cdot 5 = 540$

GCD and LCM

$a = 60 = 2^2\ 3^1\ 5^1$

$b = 54 = 2^1\ 3^3\ 5^0$

$\gcd(a, b) = 2^1\ 3^1\ 5^0 = 6$

$\text{lcm}(a, b) = 2^2\ 3^3\ 5^1 = 540$

Theorem: $a \cdot b = \gcd(a,b) \cdot \text{lcm}(a,b)$

Modular Arithmetic

Let a be an integer and m be a positive integer.
We denote by **a mod m** the remainder when a is divided by m.

Examples:

9 mod 4 = 1

9 mod 3 = 0

9 mod 10 = 9

-13 mod 4 = 3

Congruences

Let a and b be integers and m be a positive integer. We say that **a is congruent to b modulo m** if
m divides a – b.

We use the notation **$a \equiv b \pmod{m}$** to indicate that a is congruent to b modulo m.

In other words:
$a \equiv b \pmod{m}$ if and only if **a mod m = b mod m**.

Congruences

Examples:

Is it true that $46 \equiv 68 \pmod{11}$?
Yes, because $11 \mid (46 - 68)$.

Is it true that $46 \equiv 68 \pmod{22}$?
Yes, because $22 \mid (46 - 68)$.

For which integers z is it true that $z \equiv 12 \pmod{10}$?
It is true for any $z \in \{\ldots, -28, -18, -8, 2, 12, 22, 32, \ldots\}$

Theorem: 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 $a = b + km$.

Congruences

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

Proof:
We know that $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ implies that there are integers s and t with
$b = a + sm$ and $d = c + tm$.

Therefore,
$b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$ and
$bd = (a + sm)(c + tm) = ac + m(at + cs + stm)$.

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

The Euclidean Algorithm

The **Euclidean Algorithm** finds the **greatest common divisor** of two integers a and b.

For example, if we want to find gcd(287, 91), we **divide** 287 by 91:

$287 = 91 \cdot 3 + 14$

We know that for integers a, b and c,
if a | b and a | c, then a | (b + c).

Therefore, any divisor (including their gcd) of 287 and 91 must also be a divisor of $287 - 91 \cdot 3 = 14$.

Consequently, gcd(287, 91) = gcd(14, 91).

The Euclidean Algorithm

In the next step, we divide 91 by 14:

$91 = 14 \cdot 6 + 7$

This means that gcd(14, 91) = gcd(14, 7).

So we divide 14 by 7:

$14 = 7 \cdot 2 + 0$

We find that 7 | 14, and thus gcd(14, 7) = 7.

Therefore, gcd(287, 91) = 7.

The Euclidean Algorithm

In **pseudocode**, the algorithm can be implemented as follows:

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

Representations of Integers

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

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

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

Example for b=10:

$$859 = 8 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0$$

Representations of Integers

Example for b=2 (binary expansion):

$(10110)_2 = 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 = (22)_{10}$

Example for b=16 (hexadecimal expansion):

(we use letters A to F to indicate numbers 10 to 15)

$(3A0F)_{16} = 3 \cdot 16^3 + 10 \cdot 16^2 + 15 \cdot 16^0 = (14863)_{10}$

Representations of Integers

How can we construct the base b expansion of an integer n?

First, divide n by b to obtain a quotient q_0 and remainder a_0, that is,

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

The remainder a_0 is the rightmost digit in the base b expansion of n.

Next, divide q_0 by b to obtain:

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

a_1 is the second digit from the right in the base b expansion of n. Continue this process until you obtain a quotient equal to zero.

Representations of Integers

Example:
What is the base 8 expansion of $(12345)_{10}$?

First, divide 12345 by 8:
$12345 = 8 \cdot 1543 + 1$

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

The result is: $(12345)_{10} = (30071)_{8}$.

Representations of Integers

procedure base_b_expansion(n, b: positive integers)
q := n
k := 0
while $q \neq 0$
begin
    $a_k := q \bmod b$
    $q := \lfloor q/b \rfloor$
    k := k + 1
end
{the base b expansion of n is $(a_{k-1} \ldots a_1 a_0)_b$ }

Addition of Integers

How do we (humans) add two integers?

Example:

$$\begin{array}{rl} & 1\,1\,1 \quad \text{carry} \\ & 7583 \\ + & 4932 \\ \hline & 12515 \end{array}$$

Binary expansions:

$$\begin{array}{rl} & 1\;\;1 \quad \text{carry} \\ & (1011)_2 \\ + & (1010)_2 \\ \hline & (10101)_2 \end{array}$$

Addition of Integers

Let $a = (a_{n-1}a_{n-2}\ldots a_1a_0)_2$, $b = (b_{n-1}b_{n-2}\ldots b_1b_0)_2$.

How can we **algorithmically** add these two binary numbers?

First, add their rightmost bits:

$a_0 + b_0 = c_0 \cdot 2 + s_0$,

where s_0 is the **rightmost bit** in the binary expansion of a + b, and c_0 is the **carry**.

Then, add the next pair of bits and the carry:

$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1$,

where s_1 is the **next bit** in the binary expansion of a + b, and c_1 is the carry.

Addition of Integers

Continue this process until you obtain c_{n-1}.

The leading bit of the sum is $s_n = c_{n-1}$.

The result is:

$a + b = (s_n s_{n-1} \ldots s_1 s_0)_2$

Addition of Integers

Example:

Add $a = (1110)_2$ and $b = (1011)_2$.

$a_0 + b_0 = 0 + 1 = 0 \cdot 2 + 1$, so that $c_0 = 0$ and $s_0 = 1$.

$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 \cdot 2 + 0$, so $c_1 = 1$ and $s_1 = 0$.

$a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 \cdot 2 + 0$, so $c_2 = 1$ and $s_2 = 0$.

$a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 \cdot 2 + 1$, so $c_3 = 1$ and $s_3 = 1$.

$s_4 = c_3 = 1$.

Therefore, $s = a + b = (11001)_2$.

Addition of Integers

procedure add(a, b: positive integers)
c := 0
for j := 0 to n-1
begin
    d := $\lfloor (a_j + b_j + c)/2 \rfloor$
    $s_j := a_j + b_j + c - 2d$
    c := d
end
s_n := c
{the binary expansion of the sum is $(s_n s_{n-1} \ldots s_1 s_0)_2$}