Chapter 3. The fundamentals: algorithms, the integers, and matrices

3.1 Algorithms

*algorithm*: a finite set of precise instructions for solving a problem.

example, an algorithm for finding the maximum value in a finite list of integers

pseudo-code, Algorithm 1 p169

Properties that algorithms share:

input
output
definiteness
correctness
finiteness
effectiveness
generality
Searching:

General searching problem: locate an element \( x \) in a list of distinct elements \( a_1, \ldots, a_n \) or determine that it is not in the list.

Linear search, Algorithm 2 p170

Binary search, Algorithm 3 p172

which one is better? why?
Sorting:

Bubble sort, Algorithm 4 p 173.

Insertion sort, Algorithm 5 p 174.

which one is better? why?
3.2 Growth of functions

*Big-O notation*

\( f(x) \) is \( O(g(x)) \) if there are constants \( C \) and \( k \) such that
\[ f(x) \leq Cg(x) \text{ for } x > k. \]

*important results:*

– about polynomial functions

– relationships among logarithm, linear, polynomial, and exponential functions

*growth of combinations of functions*

– \((f_1 + f_2)(x)\)

– \((f_1f_2)(x)\)

– \((g(f(x))) \)

*Big-Omega and Big-Theta notation*
3.3 Complexity of algorithms

time complexity
space complexity

worst case complexity

average case complexity

table 1 on p196 commonly used terminology for complexity

solvable/not solvable problems

tractable/intractable problems
3.4 The integers and division

\textit{division:}

\textit{a divides b, written as } a|b, \textit{ if } \exists c( ac = b )

how many integers \( \leq n \) are divisible by \( d \)?

\textbf{Theorem 1:}

1. \( (a|b \land a|c) \rightarrow a|(b + c) \)
2. \( a|b \rightarrow \forall c(a|bc) \)
3. \( (a|b \land b|c) \rightarrow a|c \)

\textbf{Corollary 1:}

\( (a|b \land a|c) \rightarrow \forall n \forall m a|(mb + nc) \)
3.5 Primes and gcds

*primes:*

A positive integer $p > 1$ is called *prime* if only 1 and $p$ can divide $p$. A non-prime positive integer is called *composite*.

Theorem 2: (The fundamental theorem of arithmetic)
For every $n \geq 1$, $n = p_1^{c_1}p_2^{c_2} \ldots p_r^{c_r}$ uniquely where $p_1 < p_2 < \ldots < p_r$ are primes and $c_i > 0, i = 1, \ldots, r$

factorization was hard .......

Theorem 3: If $n$ is composite, then $n$ has a prime divisor $\leq \sqrt{n}$
The infinitude of primes

Theorem 4: there are infinitely many primes

The distribution of primes

Theorem 5: (The prime number theorem)
The number of primes not exceeding \( x \) is
\[ \approx \frac{x}{\ln x}. \]

The division algorithm

Theorem 6: Let \( a \) be an integer and \( d \) be a positive integer. Then there exist two unique integers \( q \) and \( r \), \( 0 \leq r < d \), such that
\[ a = dq + r \]

Proof:
\textit{gcd} and \textit{lcm}

\textit{gcd}: greatest common divisor of \(a\) and \(b\): the largest \(d\) such that \(d|a\) and \(d|b\)

\[ \text{gcd}(24, 36) = 12 \]

\(a\) and \(b\) are \textit{relatively prime} if \(\text{gcd}(a, b) = 1\)

\(a_1, \ldots, a_n\) are \textit{pairwise relatively prime} if \(\text{gcd}(a_i, a_j) = 1\) whenever \(1 \leq i < j \leq n\).

use factorizations of \(a\) and \(b\) to find the gcd.

\textit{lcm}: least common multiple of \(a\) and \(b\): the smallest \(m\) such that \(a|m\) and \(b|m\)

use factorization of \(a\) and \(b\) to find the lcm.

\textbf{Theorem 7}: \(ab = \text{gcd}(a, b) \cdot \text{lcm}(a, b)\)
*modular arithmetic*

$a$ is *congruent to* $b$ *modulo* $m$
if $m | (a - b)$, denoted as $a \equiv b \pmod{m}$

Theorem 8: $a \equiv b \pmod{m}$ if and only if $a \mod m = b \mod m$

Theorem 9: $a \equiv b \pmod{m}$ if and only if there is an integer $k$ such that $a = b + km$.

*applications of congruences:*

hashing functions: $h(k) = k \mod m$
pseudorandom numbers: $x_{n+1} = (ax_n + c) \mod m$
cryptology: $f(p) = (p + k) \mod 26$
3.6 Integers and Algorithms

representations of integers

Theorem 1: (base b expansion)
\[ n = a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0, \] where \( a_i < b \) is non-negative and \( a_k \neq 0 \)

denoted as \((a_k \ldots a_1 a_0)_b\)

decimal expansion
binary expansion
hexadecimal expansion
octal expansion
**algorithms for integers**

Algorithm 1 Constructing Base $b$ Expansion

Algorithm 2 Addition of Integers

Algorithm 3 Multiplying Integers

Algorithm 4 Computing div and mod

Algorithm 5 Modular Exponentiation

Algorithm 6 Euclidean Algorithm

Lemma 1 Let $a = bq + r$. Then $gcd(a, b) = gcd(b, r)$.

Proof:
3.8 Matrices

matrix: definition, row, column vectors, dimensions.

addition of matrices

multiplication of matrices

\[ A = [a_{ij}]_{m \times n} \text{ and } B = [b_{ij}]_{n \times r} \]
\[ A \times B = AB = [c_{ij}]_{m \times r} \text{ where} \]
\[ c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} \]

\[ AB \neq BA \]

Algorithm 1 for matrix multiplication p249.

\[ A^r = A \times A \times \ldots \times A \ (r \text{ times}) \]
0-1 matrices

logic operations
$A \lor B$ and $A \land B$ similar to addition

$A \odot B = [c_{ij}]$ where

$$c_{ij} = (a_{i1} \land b_{1j}) \lor (a_{i2} \land b_{2j}) \lor \ldots \lor (a_{in} \land b_{nj})$$

Algorithm 2 for boolean matrix production

boolean matrices representing graphs
determining connectivity