# Discrete Math for Computer Science

Peter Schaefer

Freshman - Kutztown

## Contents

# 1 Integer Properties

## 1.1 The Division Algorithm

In **integer division**, the input and output values must always be integers. For example, when 9 is divided by 4, the answer is 2 with a remainder of 1, instead of 2.25.

### Divides

Let $x$ and $y$ be two integers. Then $x$ *divides* $y$, $x \mid y$, if and only if $x \neq 0$ and there is an integer $k$ such that $y = kx$. If there is no such integer or if $x = 0$, then $x$ does not divide $y$, $x \nmid y$. If $x \mid y$, then $y$ is said to be a *multiple* of $x$, and $x$ is a *factor* or *divisor* of $y$.

### Theorem: Divisibility and linear combinations

Let $x, y,$ and $z$ be integers. If $x \mid y$ and $x \mid z$, then $x \mid (sy + tz)$ for any integers $s$ and $t$.

*Proof.* Since $x \mid y$, then $y = kx$ for some integer $k$. Similarly, since $x \mid z$, then $z = jx$ for some integer $j$. A linear combination of $y$ and $z$ can be expressed as:

$$sy + tz = s(kx) + t(jx) = (sk + tj)x.$$

For some integers $s$ and $t$. Since $sy + tz$ is an integer multiple of $x$, then $x \mid (sy + tz)$. □

### Quotients and remainders

If $x \nmid y$, then there is s non-zero remainder when $x$ is divided into $y$. The **Division Algorithm**, states that the result of the division and the remainder are unique.

### Theorem: The Division Algorithm

Let $n$ be an integer and let $d$ be a positive integer. Then, there are unique integers $q$ and $r$, with $0 \leq r \leq d-1$, such that $n = qd + r$.

### Integer division definitions

In the Division Algorithm, $q$ is called the **quotient** and $r$ is called the **remainder**. The operations **div** and **mod** produce the quotient and the remainder as a function of $n$ and $d$.

$$\begin{aligned} q &= n \operatorname{div} d \\ r &= n \bmod d \end{aligned}$$

Here is some examples of computing div and mod:

$$\begin{aligned} 15 \bmod 6 &= 3 & \qquad -11 \bmod 4 &= 1 \\ 15 - 2 \cdot 6 &= 3 & \qquad -11 - (-3) \cdot 4 &= 1 \end{aligned}$$

$$\begin{aligned} 15 \operatorname{div} 6 &= 2 & \qquad -11 \operatorname{div} 4 &= -3 \\ \frac{15 - 3}{2} &= 2 & \qquad \frac{-11 - 1}{4} &= -3 \end{aligned}$$

## 1.2 Modular arithmetic

Given a finite set of integers, we can define addition and multiplication on the elements in the set such that after every operation, we apply a modular function equal to the cardinality of the set.

- **addition mod** $m$

  the operation defined by adding two numbers and applying $\bmod m$ to the result

- **multiplication mod** $m$

  the operation defined by multiplying two numbers and applying $\bmod m$ to the result

The set $\{0, 1, 2, \ldots, m-1\}$ along with addition and multiplication mod $m$ defines a closed mathematical system with $m$ elements called a **ring**. The ring $\{0, 1, 2, \ldots, m-1\}$ with addition and multiplication mod $m$ is denoted by $\mathbb{Z}_m$.

### Applications

A common way to organize data is to maintain an array called a **hash table** which is slightly larger than the number of data items to be stored. A bldhash function is used to map each data item to a location in the array. Modulus is used to keep the results from a hash function in the range of the hash table.

Computers use functions called bldpseudo-random number generators that produce numbers having many of the statistical properties of random numbers but are in fact deterministically generated. Modulus is used to keep these pseudo-random number generators in a certain range when used.

### Congruence mod $m$

Let $m \in \mathbb{Z} > 1$. Let $x$ and $y$ be any two integers. Then bld$x$ is congruent to $y \bmod m$ if $x \bmod m = y \bmod m$. The fact that $x$ is congruent to $y \bmod m$ is denoted

$$x \equiv y \pmod{m}.$$

### Theorem: Alternate characterization of congruence mod $m$

Let $m \in \mathbb{Z} > 1$. Let $x$ and $y$ be any two integers. Then $x \equiv y \pmod{m}$ if and only if $m \mid (x - y)$.

*Proof.* First suppose that $x \equiv y \pmod{m}$. By definition $xtmodm = y \bmod m$. We define the variable $r$ to be the value of $x \bmod m = y \bmod m$. Therefore, $x = r + km$ for some integer $k$ and $y = r + jm$ for some integer $j$. Then

$$x - y = (r + km) - (r + jm) = (k - j)m.$$

Since $(k - j)$ is an integer, $m \mid (x - y)$.
Now suppose that $m \mid (x - y)$. Then $(x - y) = tm$ for some integer $t$. Let $r$ be the value of $x \bmod m$. Then $x = r + km$ for some integer $k$. The integer $y$ can be expressed as

$$y = x - (x - y) = (r + km) - tm = r + (k - t)m.$$

Since $r$ is an integer in the range from 0 to $m - 1$, $r$ is the unique remainder when $y$ is divided by $m$. Therefore $r = y \bmod m = x \bmod m$, and by definition $x \equiv y \pmod{m}$. □

### Precedence of the mod operation

$$6 + 2 \bmod y = 6 + (2 \bmod 7) = 8$$
$$6 \cdot 2 \bmod 7 = (6 \cdot 2) \bmod 7 = 5$$

However, in general it is best to just use parentheses in order to clarify which operations should be performed first.

**Theorem: Computing arithmetic operations mod $m$**

Let $m$ be an integer larger than 1. Let $x$ and $y$ be any integers. Then

$$\begin{aligned}[(x \bmod m) + (y \bmod m)] \bmod m &= [x + y] \bmod m \\ [(x \bmod m)(y \bmod m)] \bmod m &= [x + y] \bmod m\end{aligned}$$

## 1.3 Prime factorizations

A number $p$ is **prime** if it is an integer greater than 1 and its only factors are 1 and $p$. A positive integer is **composite** if it has a factor other than 1 or itself. Every integer greater than 1 is either prime of composite. Every positive integer greater than one can be expressed as a product of primes called its **prime factorization**. Moreover, the prime factorization is unique up to ordering of the factors.

**Theorem: The Fundamental Theorem of Arithmetic**

Every positive integer other than 1 can be expressed uniquely as a product of prime numbers where the primes factors are written in non-decreasing order.

Examples of prime factorizations in non-decreasing order

$$\begin{aligned}112 &= 2^4 \cdot 7 \\ 612 &= 2^2 \cdot 3^3 \cdot 17 \\ 243 &= 3^5 \\ 17 &= 17\end{aligned}$$

**Greater common divisors and least common multiples**

- The **greatest common divisor (gcd)** of integers $x$ and $y$ that are not both zero is the largest integer that is a factor of both $x$ and $y$.
- The **least common multiples (lcm)** of non-zero integers $x$ and $y$ is the smallest positive integer that is an integer multiple of both $x$ and $y$.

Two numbers are **relatively prime** if their greatest common divisor is 1.

**Theorem: GCD and LCM from prime factorizations**

Let $x$ and $y$ be two positive integers with prime factorizations expressed using a common set of primes as:

$$\begin{aligned}x &= p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_r^{\alpha_r} \\ y &= p_1^{\beta_1} \cdot p_2^{\beta_2} \cdots p_r^{\beta_r}\end{aligned}$$

The $p_i$'s are all distinct prime numbers. The exponents $\alpha_i$'s and $\beta_i$'s are non-negative integers. Then:

- $x \mid y$ if and only if $\alpha_i \leq \beta_i$ for all $1 \leq i \leq r$
- $\gcd(x, y) = p_1^{\min(\alpha_1,\beta_1)} \cdot p_2^{\min(\alpha_2,\beta_2)} \cdots p_r^{\min(\alpha_r,\beta_r)}$
- $\text{lcm}(x, y) = p_1^{\max(\alpha_1,\beta_1)} \cdot p_2^{\max(\alpha_2,\beta_2)} \cdots p_r^{\max(\alpha_r,\beta_r)}$

## 1.4 Factoring and primality testing

A **brute force algorithm** solves a problem by exhaustively searching all positive solutions without using an understanding of the mathematical structure in the problem to eliminate steps.

### Theorem: Small Factors

If $N$ is a composite number, then $N$ has a factor greater than 1 and at most $\sqrt{N}$

### Theorem: Infinite number of primes

There are an infinite number of primes.

*Proof.* Suppose that there are a finite number of primes. Since there are only a finite number, they can be listed:

$$p_1, p_2, \ldots, p_k$$

Take the product of all the primes and add 1. Call the resulting number $N$:

$$N = (p_1 \cdot p_2 \cdots p_k) + 1$$

The number $N$ is larger than all of the primes numbers that were listed, so it must not be prime. Since $N$ is a composite number, it is the product of at least two primes by the Fundamental Theorem of Arithmetic. There $N$ is divisible by some prime $p_j$. Let

$$\frac{N}{p_j} = \frac{(p_1 \cdot p_2 \cdots p_k)}{p_j} + \frac{1}{p_j}$$

Note that $p_j$ is one of the prime factors in $(p_1 \cdot p_2 \cdots p_k)$, so $(p_1 \cdot p_2 \cdots p_k)/p_j$ is an integer. However, $1/p_j$ is not an integer. Since $N/p_j$ is the sum of two terms, one of which is an integer and the other of which is not an integer, then $N/p_j$ is not an integer. This contradicts the fact that $p_j$ evenly divides $N$. □

### The Prime Number Theorem

Let $\pi(x)$ be the number of prime numbers in the range from 2 through $x$. Then

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln x} = 1.$$

Another way to interpret the Prime Number Theorem is that if a random number is selected from the range 2 to $x$, then the likelihood that the selected number is prime is roughly $1/\ln x$.

## 1.5 Greatest common factor divisor and Euclid's algorithm

There is an efficient way to compute the gcd of two numbers without finding their prime factorizations. The algorithm presented in this subsection is in wide use today and is attributed to the Greek mathematician Euclid who lived around 300 B.C. The basis of the algorithm is the following theorem:

### GCD Theorem

Let $x$ and $y$ be two positive integers. Then $\gcd(x, y) = \gcd(y \bmod x, x)$.

### Euclid's Algorithm for finding the greatest common divisor

```
Input: Two positive integers, x and y.
Output: gcd(x, y)

If(y < x)
  Swap x and y.
r := y mod x

While(r != 0)
  y := x
```

```
    x := r
    r := y mod x
End-While

Return(x)
```

Sample execution of Euclid's algorithm for $\gcd(675, 210)$:

$$\begin{array}{cccccc}
675 & 210 & 675 \bmod 210 & & & \\
 & 210 & 45 & 210 \bmod 45 & & \\
 & & 45 & 30 & 45 \bmod 30 & \\
 & & & 30 & 15 & 30 \bmod 15 \\
 & & & & \mathbf{15} & 0
\end{array}$$

The last non-zero number was 15, so $\gcd(675, 210) = 15$.

**Expressing $\gcd(x, y)$ as a linear combination of $x$ and $y$**

Let $x$ and $y$ be integers, then there are integers $s$ and $t$ such that

$$\gcd(x, y) = sx + ty.$$

The values for $s$ and $t$ in the theorem above can be found by a series of substitutions using the equation from each iteration. The algorithm used to find the coefficient, $s$ and $t$, such that $\gcd(x, y) = sx + ty$, is called the **Extended Euclidean Algorithm**.

**The Extended Euclidean Algorithm**

$$\begin{array}{ccccl}
 & & y & x & r \\
675 & 210 & 45 & 30 & 15 \\
 & & & & r = y \bmod x \\
 & & & & r = y - (y \operatorname{div} x) \cdot x \\
 & & & & 15 = 45 - (45 \operatorname{div} 30) \cdot 30 \\
 & & & & \mathbf{15 = 45 - 1 \cdot 30} \\
 & & & 30 & = 210 - (210 \operatorname{div} 45) \cdot 45 \\
 & & & \mathbf{30} & \mathbf{= 210 - 4 \cdot 45} \\
 & & 45 & = & 675 - (675 \operatorname{div} 210) \cdot 210 \\
 & & \mathbf{45} & = & \mathbf{675 - 3 \cdot 210}
\end{array}$$

We can use the bolded equations to solve for $15 = c \cdot 210 + d \cdot 675$.

$$\begin{aligned}
15 &= 45 - 30 \\
&= 45 - (210 - 4 \cdot 45) \\
&= 5 \cdot 45 - 210 \\
&= 5 \cdot (675 - 3 \cdot 210) - 210 \\
&= 5 \cdot 675 - 16 \cdot 210
\end{aligned}$$

Now we have the full answer and expansion for $\gcd(675, 210)$.

$$\gcd(675, 210) = 15 = 5 \cdot 675 - 16 \cdot 210.$$

**The Multiplicative Inverse mod $n$**

A **multiplicative inverse mod** $n$, or just **inverse mod** $n$, of an integer $x$, is an integer $s \in \{1, 2, \ldots, n-1\}$ such that $sx \bmod n = 1$.

For example, 3 is an inverse of 7 mod 10 because $3 \cdot 7 \bmod 10 = 1$. The number 7 is an inverse of 5 mod 17 because $7 \cdot 5 \bmod 17 = 1$. It is possible for a number to be its own multiplicative inverse mod $n$. For example, 7 is the inverse of $7 \bmod 8$ because $7 \cdot 7 \bmod 8 = 1$.

Not every number has an inverse mod $n$. For example, 4 does not have an inverse mod 6. The condition is that $x$ has an inverse $\bmod\, n$ if and only if $x$ and $n$ are relatively prime.

The Extended Euclidean Algorithm can be used to find the multiplicative inverse of $x \bmod n$ when is exists.

- If $\gcd(x, n) \neq 1$, then $x$ does not have a multiplicative inverse $\bmod\, n$.
- If $x$ and $n$ are relatively prime, then the Extended Euclidean Algorithm finds integers $s$ and $t$ such that $1 = sx + tn$.
- $sx - 1 = -tn$. Therefore, $(sx \bmod n) = (1 \bmod n)$. If $A - B$ is a multiple of $n$ then $(A \bmod n) = (B \bmod n)$.
- $(s \bmod n)$ is the unique multiplicative inverse of $x$ in $\{0, 1, \ldots, n-1\}$.

For example, suppose that Euclid's Algorithm returns

$$\gcd(31, 43) = 1 = 13 \cdot 34 - 18 \cdot 31$$

The coefficient of 31 is -18. Therefore, the multiplicative inverse of $31 \bmod 43$ is $(-18 \bmod 43) = 25$.

## 1.6 Number representation

## 1.7 Fast exponentiation

## 1.8 Introduction to cryptography

## 1.9 The RSA cryptosystem

# 2 Introduction to Counting

## 2.1 Sum and Product Rules

## 2.2 The Bijection Rules

## 2.3 The generalized product rule

## 2.4 Counting permutations

## 2.5 Counting subsets

## 2.6 Subset and permutation examples

## 2.7 Counting by complement

## 2.8 Permutations with repetitions

## 2.9 Counting multisets

## 2.10 Assignment problems: Balls in bins

## 2.11 Inclusion-exclusion principle

# 3 Advanced Counting

## 3.1 Generating permutations

## 3.2 Binomial coefficients and combinatorial identities

## 3.3 The pigeonhole principle

## 3.4 Generating functions

# 4 Discrete Probability

## 4.1 Probability of an event

## 4.2 Unions and complements of events

## 4.3 Conditional probability and independence

## 4.4 Bayes' Theorem

## 4.5 Random variables

## 4.6 Expectation of random variables

## 4.7 Linearity of expectations

## 4.8 Bernoulli trials and the binomial distribution

# 5 Graphs

## 5.1 Introduction to Graphs

## 5.2 Graph representations

## 5.3 Graph isomorphism

## 5.4 Walks, trails, circuits, paths, and cycles

## 5.5 Graph connectivity

## 5.6 Euler circuits and trails

## 5.7 Hamiltonian cycles and paths

## 5.8 Planar coloring

## 5.9 Graph coloring

# 6 Trees

## 6.1 Introduction to trees

## 6.2 Tree application examples

## 6.3 Properties of trees

## 6.4 Tree traversals

## 6.5 Spanning trees and graph traversals

## 6.6 Minimum spanning trees