CHAPTER

4 Number Theory and Cryptography

The part of mathematics devoted to the study of the set of integers and their properties is known as number theory. In this chapter we will develop some of the important concepts of number theory including many of those used in computer science. As we develop number theory, we will use the proof methods developed in Chapter 1 to prove many theorems.

We will first introduce the notion of divisibility of integers, which we use to introduce modular, or clock, arithmetic. Modular arithmetic operates with the remainders of integers when they are divided by a fixed positive integer, called the modulus. We will prove many important results about modular arithmetic which we will use extensively in this chapter.

Integers can be represented with any positive integer b greater than 1 as a base. In this chapter we discuss base b representations of integers and give an algorithm for finding them. In particular, we will discuss binary, octal, and hexadecimal (base 2, 8, and 16) representations. We will describe algorithms for carrying out arithmetic using these representations and study their complexity. These algorithms were the first procedures called algorithms.

We will discuss prime numbers, the positive integers that have only 1 and themselves as positive divisors. We will prove that there are infinitely many primes; the proof we give is considered to be one of the most beautiful proofs in mathematics. We will discuss the distribution of primes and many famous open questions concerning primes. We will introduce the concept of greatest common divisors and study the Euclidean algorithm for computing them. This algorithm was first described thousands of years ago. We will introduce the fundamental theorem of arithmetic, a key result which tells us that every positive integer has a unique factorization into primes.

We will explain how to solve linear congruences, as well as systems of linear congruences, which we solve using the famous Chinese remainder theorem. We will introduce the notion of pseudoprimes, which are composite integers masquerading as primes, and show how this notion can help us rapidly generate prime numbers.

This chapter introduces several important applications of number theory. In particular, we will use number theory to generate pseudorandom numbers, to assign memory locations to computer files, and to find check digits used to detect errors in various kinds of identification numbers. We also introduce the subject of cryptography. Number theory plays an essential role both in classical cryptography, first used thousands of years ago, and modern cryptography, which plays an essential role in electronic communication. We will show how the ideas we develop can be used in cryptographical protocols, introducing protocols for sharing keys and for sending signed messages. Number theory, once considered the purest of subjects, has become an essential tool in providing computer and Internet security.

Finally, it should be noted that this chapter is designed to introduce some key aspects of number theory. As with all the topics covered in this book, there is a great deal more to learn. Interested students can consult [Ro11], the author's number theory text, to explore this fascinating subject more fully.

4.1 Divisibility and Modular Arithmetic

4.1.1 Introduction

The ideas that we will develop in this section are based on the notion of divisibility. Division of an integer by a positive integer produces a quotient and a remainder. Working with these remainders leads to modular arithmetic, which plays an important role in mathematics and which

is used throughout computer science. We will discuss some important applications of modular arithmetic later in this chapter, including generating pseudorandom numbers, assigning computer memory locations to files, constructing check digits, and encrypting messages.

4.1.2 Division

When one integer is divided by a second nonzero integer, the quotient may or may not be an integer. For example, $12/3 = 4$ is an integer, whereas $11/4 = 2.75$ is not. This leads to Definition 1.

Definition 1 If a and b are integers with $a \neq 0$, we say that a *divides* b if there is an integer c such that $b = ac$ (or equivalently, if $\frac{b}{a}$ is an integer). 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. We write $a \nmid b$ when a does not divide b.

Remark: We can express $a \mid b$ using quantifiers as $\exists c(ac = b)$, where the universe of discourse is the set of integers.

In Figure 1 a number line indicates which integers are divisible by the positive integer d.

EXAMPLE 1 Determine whether $3 \mid 7$ and whether $3 \mid 12$.

Solution: We see that $3 \nmid 7$, because $7/3$ is not an integer. On the other hand, $3 \mid 12$ because $12/3 = 4$. ◀

EXAMPLE 2 Let n and d be positive integers. How many positive integers not exceeding n are divisible by d?

Extra Examples

Solution: The positive integers divisible by d are all the integers of the form dk, where k is a positive integer. Hence, the number of positive integers divisible by d that do not exceed n equals the number of integers k with $0 < dk \leq n$, or with $0 < k \leq n/d$. Therefore, there are $\lfloor n/d \rfloor$ positive integers not exceeding n that are divisible by d. ◀

Some of the basic properties of divisibility of integers are given in Theorem 1.

THEOREM 1 Let a, b, and c be integers, where $a \neq 0$. Then

(*i*) if $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;
(*ii*) if $a \mid b$, then $a \mid bc$ for all integers c;
(*iii*) if $a \mid b$ and $b \mid c$, then $a \mid c$.

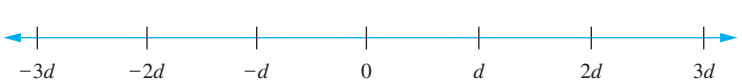


FIGURE 1 **Integers divisible by the positive integer *d*.**

Proof: We will give a direct proof of (*i*). Suppose that $a \mid b$ and $a \mid c$. Then, from the definition of divisibility, it follows that there are integers s and t with $b = as$ and $c = at$. Hence,

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

Therefore, a divides $b + c$. This establishes part (*i*) of the theorem. The proofs of parts (*ii*) and (*iii*) are left as Exercises 3 and 4. ◁

Theorem 1 has this useful consequence.

COROLLARY 1 If a, b, and 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.

Proof: We will give a direct proof. By part (*ii*) of Theorem 1 we see that $a \mid mb$ and $a \mid nc$ whenever m and n are integers. By part (*i*) of Theorem 1 it follows that $a \mid mb + nc$. ◁

4.1.3 The Division Algorithm

When an integer is divided by a positive integer, there is a quotient and a remainder, as the division algorithm shows.

THEOREM 2 **THE DIVISION ALGORITHM** Let a be an integer and d a positive integer. Then there are unique integers q and r, with $0 \leq r < d$, such that $a = dq + r$.

We defer the proof of the division algorithm to Section 5.2. (See Example 5 and Exercise 37 in that section.)

Remark: Theorem 2 is not really an algorithm. (Why not?) Nevertheless, we use its traditional name.

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:

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

Remark: Note that both a **div** d and a **mod** d for a fixed d are functions on the set of integers. Furthermore, when a is an integer and d is a positive integer, we have a **div** $d = \lfloor a/d \rfloor$ and a **mod** $d = a - d$. (See Exercise 24.)

Examples 3 and 4 illustrate the division algorithm.

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?

Extra Examples

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 $0 \leq r < 3$. ◀

Note that the integer a is divisible by the integer d if and only if the remainder is zero when a is divided by d.

Remark: A programming language may have one, or possibly two, operators for modular arithmetic, denoted by mod (in BASIC, Maple, Mathematica, EXCEL, and SQL), % (in C, C++, Java, and Python), rem (in Ada and Lisp), or something else. Be careful when using them, because for $a < 0$, some of these operators return $a - m\lceil a/m \rceil$ instead of a **mod** $m = a - m\lfloor a/m \rfloor$ (as shown in Exercise 24). Also, unlike a **mod** m, some of these operators are defined when $m < 0$, and even when $m = 0$.

4.1.4 Modular Arithmetic

In some situations we care only about the remainder of an integer when it is divided by some specified positive integer. For instance, when we ask what time it will be (on a 24-hour clock) 50 hours from now, we care only about the remainder when 50 plus the current hour is divided by 24. Because we are often interested only in remainders, we have special notations for them. We have already introduced the notation a **mod** m to represent the remainder when an integer a is divided by the positive integer m. We now introduce a different, but related, notation that indicates that two integers have the same remainder when they are divided by the positive integer m.

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 $a - b$. We use the notation $a \equiv b \pmod{m}$ to indicate 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** (plural **moduli**). If a and b are not congruent modulo m, we write $a \not\equiv b \pmod{m}$.

Although both notations $a \equiv b \pmod{m}$ and a **mod** $m = b$ include "mod," they represent fundamentally different concepts. The first represents a relation on the set of integers, whereas the second represents a function. However, the relation $a \equiv b \pmod{m}$ and the **mod** m function are closely related, as described in Theorem 3.

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.

The proof of Theorem 3 is left as Exercises 21 and 22. Recall that a **mod** m and b **mod** m are the remainders when a and b are divided by m, respectively. Consequently, Theorem 3 also says that $a \equiv b \pmod{m}$ if and only if a and b have the same remainder when divided by m.

EXAMPLE 5 Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 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 $24 \not\equiv 14 \pmod{6}$. ◀

The great German mathematician Karl Friedrich Gauss developed the concept of congruences at the end of the eighteenth century. The notion of congruences has played an important role in the development of number theory.

Theorem 4 provides a useful way to work with 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 $a \equiv b \pmod{m}$, by the definition of congruence (Definition 3), we know that $m \mid (a - b)$. This means that there is an integer k such that $a - b = km$, so that $a = b + km$. 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. In Chapter 9 we will show that there are m pairwise disjoint equivalence classes modulo m and that the union of these equivalence classes is the set of integers.

Theorem 5 shows that additions and multiplications preserve congruences.

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

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

Links

©Hulton Archive/Getty Images

KARL FRIEDRICH GAUSS (1777–1855) Karl Friedrich Gauss, the son of a bricklayer, was a child prodigy. He demonstrated his potential at the age of 10, when he quickly solved a problem assigned by a teacher to keep the class busy. The teacher asked the students to find the sum of the first 100 positive integers. Gauss realized that this sum could be found by forming 50 pairs, each with the sum 101: $1 + 100, 2 + 99, \ldots, 50 + 51$. This brilliance attracted the sponsorship of patrons, including Duke Ferdinand of Brunswick, who made it possible for Gauss to attend Caroline College and the University of Göttingen. While a student, he invented the method of least squares, which is used to estimate the most likely value of a variable from experimental results. In 1796 Gauss made a fundamental discovery in geometry, advancing a subject that had not advanced since ancient times. He showed that a 17-sided regular polygon could be drawn using just a ruler and compass.

In 1799 Gauss presented the first rigorous proof of the fundamental theorem of algebra, which states that a polynomial of degree n has exactly n roots in the complex numbers (counting multiplicities). Gauss achieved worldwide fame when he successfully calculated the orbit of the first asteroid discovered, Ceres, using scanty data.

Gauss was called the Prince of Mathematics by his contemporary mathematicians. Although Gauss is noted for his many discoveries in geometry, algebra, analysis, astronomy, and physics, he had a special interest in number theory, which can be seen from his statement "Mathematics is the queen of the sciences, and the theory of numbers is the queen of mathematics." Gauss laid the foundations for modern number theory with the publication of his book *Disquisitiones Arithmeticae* in 1801.

Proof: We use a direct 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 $d = c + tm$. Hence,

$$b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$$

and

$$bd = (a + sm)(c + tm) = ac + m(at + cs + stm).$$

Hence,

$$a + c \equiv b + d \pmod{m} \qquad \text{and} \qquad ac \equiv bd \pmod{m}.$$ ◁

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

$$77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod{5}.$$ ◀

You cannot always divide both sides of a congruence by the same number!

We must be careful working with congruences. Some properties we may expect to be true are not valid. For example, if $ac \equiv bc \pmod{m}$, the congruence $a \equiv b \pmod{m}$ may be false. Similarly, if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, the congruence $a^c \equiv b^d \pmod{m}$ may be false. (See Exercise 43.)

Corollary 2 shows how to find the values of the **mod** m function at the sum and product of two integers using the values of this function at each of these integers. We will use this result in Section 5.4.

COROLLARY 2 Let m be a positive integer and let a and b be integers. Then

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

and

$$ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m.$$

Proof: By the definitions of **mod** m and of congruence modulo m, we know that $a \equiv (a \bmod m) \pmod{m}$ and $b \equiv (b \bmod m) \pmod{m}$. Hence, Theorem 5 tells us that

$$a + b \equiv (a \bmod m) + (b \bmod m) \pmod{m}$$

and

$$ab \equiv (a \bmod m)(b \bmod m) \pmod{m}.$$

The equalities in this corollary follow from these last two congruences by Theorem 3. ◁

In Section 4.6 we will carry out a variety of computations using the **mod** function when we study cryptography. Example 7 illustrates a type of computation involving the **mod** function that we will encounter.

EXAMPLE 7 Find the value of $(19^3 \bmod 31)^4 \bmod 23$.

Solution: To compute $(19^3 \bmod 31)^4 \bmod 23$, we will first evaluate $19^3 \bmod 31$. Because $19^3 = 6859$ and $6859 = 221 \cdot 31 + 8$, we have $19^3 \bmod 31 = 6859 \bmod 31 = 8$. So, $(19^3 \bmod 31)^4 \bmod 23 = 8^4 \bmod 23$.

Next, note that $8^4 = 4096$. Because $4096 = 178 \cdot 23 + 2$, we have $4096 \bmod 23 = 2$. Hence, $(19^3 \bmod 31)^4 \bmod 23 = 2$. ◀

4.1.5 Arithmetic Modulo *m*

We can define arithmetic operations on $\mathbf{Z}_m$, the set of nonnegative integers less than m, that is, the set $\{0, 1, \ldots, m-1\}$. In particular, we define addition of these integers, denoted by $+_m$ by

$$a +_m b = (a + b) \bmod m,$$

where the addition on the right-hand side of this equation is the ordinary addition of integers, and we define multiplication of these integers, denoted by $\cdot_m$ by

$$a \cdot_m b = (a \cdot b) \bmod m,$$

where the multiplication on the right-hand side of this equation is the ordinary multiplication of integers. The operations $+_m$ and $\cdot_m$ are called addition and multiplication modulo m and when we use these operations, we are said to be doing **arithmetic modulo** m.

EXAMPLE 8 Use the definition of addition and multiplication in $\mathbf{Z}_m$ to find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definition of addition modulo 11, we find that

$$7 +_{11} 9 = (7 + 9) \bmod 11 = 16 \bmod 11 = 5,$$

and

$$7 \cdot_{11} 9 = (7 \cdot 9) \bmod 11 = 63 \bmod 11 = 8.$$

Hence, $7 +_{11} 9 = 5$ and $7 \cdot_{11} 9 = 8$. ◀

The operations $+_m$ and $\cdot_m$ satisfy many of the same properties of ordinary addition and multiplication of integers. In particular, they satisfy these properties:

Closure If a and b belong to $\mathbf{Z}_m$, then $a +_m b$ and $a \cdot_m b$ belong to $\mathbf{Z}_m$.

Associativity If a, b, and c belong to $\mathbf{Z}_m$, then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.

Commutativity If a and b belong to $\mathbf{Z}_m$, 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. That is, if a belongs to $\mathbf{Z}_m$, then $a +_m 0 = 0 +_m a = a$ and $a \cdot_m 1 = 1 \cdot_m a = a$.

Additive inverses If $a \neq 0$ belongs to $\mathbf{Z}_m$, then $m - a$ is an additive inverse of a modulo m and 0 is its own additive inverse. That is, $a +_m (m - a) = 0$ and $0 +_m 0 = 0$.

Distributivity If a, b, and c belong to $\mathbf{Z}_m$, then $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ and $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

These properties follow from the properties we have developed for congruences and remainders modulo m, together with the properties of integers; we leave their proofs as Exercises 48–50. Note that we have listed the property that every element of $\mathbf{Z}_m$ has an additive inverse, but no analogous property for multiplicative inverses has been included. This is because multiplicative inverses do not always exist modulo m. For instance, there is no multiplicative inverse of 2 modulo 6, as the reader can verify. We will return to the question of when an integer has a multiplicative inverse modulo m later in this chapter.

Remark: Because $\mathbf{Z}_m$ with the operations of addition and multiplication modulo m satisfies the properties listed, $\mathbf{Z}_m$ with modular addition is said to be a **commutative group** and $\mathbf{Z}_m$ with both of these operations is said to be a **commutative ring**. Note that the set of integers with ordinary addition and multiplication also forms a commutative ring. Groups and rings are studied in courses that cover abstract algebra.

Remark: In Exercise 36, and in later sections, we will use the notations $+$ and $\cdot$ for $+_m$ and $\cdot_m$ without the subscript m on the symbol for the operator whenever we work with $\mathbf{Z}_m$.

Exercises

1. Does 17 divide each of these numbers?
a) 68 **b)** 84 **c)** 357 **d)** 1001

2. Prove that if a is an integer other than 0, then
a) 1 divides a. **b)** a divides 0.

3. Prove that part (*ii*) of Theorem 1 is true.

4. Prove that part (*iii*) of Theorem 1 is true.

5. Show that if $a \mid b$ and $b \mid a$, where a and b are integers, then $a = b$ or $a = -b$.

6. Show that if a, b, c, and d are integers, where $a \neq 0$, such that $a \mid c$ and $b \mid d$, then $ab \mid cd$.

7. Show that if a, b, and c are integers, where $a \neq 0$ and $c \neq 0$, such that $ac \mid bc$, then $a \mid b$.

8. Prove or disprove that if $a \mid bc$, where a, b, and c are positive integers and $a \neq 0$, then $a \mid b$ or $a \mid c$.

9. Prove that if a and b are integers and a divides b, then a is odd or b is even.

10. Prove that if a and b are nonzero integers, a divides b, and $a + b$ is odd, then a is odd.

11. Prove that if a is an integer that is not divisible by 3, then $(a + 1)(a + 2)$ is divisible by 3.

12. Prove that if a is a positive integer, then 4 does not divide $a^2 + 2$.

13. What are the quotient and remainder when
a) 19 is divided by 7? **b)** -111 is divided by 11?
c) 789 is divided by 23? **d)** 1001 is divided by 13?
e) 0 is divided by 19? **f)** 3 is divided by 5?
g) -1 is divided by 3? **h)** 4 is divided by 1?

14. What are the quotient and remainder when
a) 44 is divided by 8?
b) 777 is divided by 21?
c) -123 is divided by 19?
d) -1 is divided by 23?
e) -2002 is divided by 87?
f) 0 is divided by 17?
g) 1,234,567 is divided by 1001?
h) -100 is divided by 101?

15. What time does a 12-hour clock read
a) 80 hours after it reads 11:00?
b) 40 hours before it reads 12:00?
c) 100 hours after it reads 6:00?

16. What time does a 24-hour clock read
a) 100 hours after it reads 2:00?
b) 45 hours before it reads 12:00?
c) 168 hours after it reads 19:00?

17. Suppose that a and b are integers, $a \equiv 4 \pmod{13}$, and $b \equiv 9 \pmod{13}$. Find the integer c with $0 \leq c \leq 12$ such that
a) $c \equiv 9a \pmod{13}$.
b) $c \equiv 11b \pmod{13}$.
c) $c \equiv a + b \pmod{13}$.
d) $c \equiv 2a + 3b \pmod{13}$.
e) $c \equiv a^2 + b^2 \pmod{13}$.
f) $c \equiv a^3 - b^3 \pmod{13}$.

18. Suppose that a and b are integers, $a \equiv 11 \pmod{19}$, and $b \equiv 3 \pmod{19}$. Find the integer c with $0 \leq c \leq 18$ such that
a) $c \equiv 13a \pmod{19}$.
b) $c \equiv 8b \pmod{19}$.
c) $c \equiv a - b \pmod{19}$.
d) $c \equiv 7a + 3b \pmod{19}$.
e) $c \equiv 2a^2 + 3b^2 \pmod{19}$.
f) $c \equiv a^3 + 4b^3 \pmod{19}$.

19. Show that if a and d are positive integers, then $(-a)\,\mathbf{div}\,d = -a\,\mathbf{div}\,d$ if and only if d divides a.

20. Prove or disprove that if a, b, and d are integers with $d > 0$, then $(a + b)\,\mathbf{div}\,d = a\,\mathbf{div}\,d + b\,\mathbf{div}\,d$.

21. Let m be a positive integer. Show that $a \equiv b \pmod{m}$ if $a\ \mathbf{mod}\ m = b\ \mathbf{mod}\ m$.

22. Let m be a positive integer. Show that $a\ \mathbf{mod}\ m = b\ \mathbf{mod}\ m$ if $a \equiv b \pmod{m}$.

23. Show that if n and k are positive integers, then $\lceil n/k \rceil = \lfloor (n-1)/k \rfloor + 1$.

24. Show that if a is an integer and d is an integer greater than 1, then the quotient and remainder obtained when a is divided by d are $\lfloor a/d \rfloor$ and $a - d\lfloor a/d \rfloor$, respectively.

25. Find a formula for the integer with smallest absolute value that is congruent to an integer a modulo m, where m is a positive integer.

26. Evaluate these quantities.

a) $-17\ \mathbf{mod}\ 2$ **b)** $144\ \mathbf{mod}\ 7$
c) $-101\ \mathbf{mod}\ 13$ **d)** $199\ \mathbf{mod}\ 19$

27. Evaluate these quantities.

a) $13\ \mathbf{mod}\ 3$ **b)** $-97\ \mathbf{mod}\ 11$
c) $155\ \mathbf{mod}\ 19$ **d)** $-221\ \mathbf{mod}\ 23$

28. Find $a\ \mathbf{div}\ m$ and $a\ \mathbf{mod}\ m$ when

a) $a = -111$, $m = 99$.
b) $a = -9999$, $m = 101$.
c) $a = 10299$, $m = 999$.
d) $a = 123456$, $m = 1001$.

29. Find $a\ \mathbf{div}\ m$ and $a\ \mathbf{mod}\ m$ when

a) $a = 228$, $m = 119$.
b) $a = 9009$, $m = 223$.
c) $a = -10101$, $m = 333$.
d) $a = -765432$, $m = 38271$.

30. Find the integer a such that

a) $a \equiv 43 \pmod{23}$ and $-22 \le a \le 0$.
b) $a \equiv 17 \pmod{29}$ and $-14 \le a \le 14$.
c) $a \equiv -11 \pmod{21}$ and $90 \le a \le 110$.

31. Find the integer a such that

a) $a \equiv -15 \pmod{27}$ and $-26 \le a \le 0$.
b) $a \equiv 24 \pmod{31}$ and $-15 \le a \le 15$.
c) $a \equiv 99 \pmod{41}$ and $100 \le a \le 140$.

32. List five integers that are congruent to 4 modulo 12.

33. List all integers between -100 and 100 that are congruent to -1 modulo 25.

34. Decide whether each of these integers is congruent to 3 modulo 7.

a) 37 **b)** 66
c) -17 **d)** -67

35. Decide whether each of these integers is congruent to 5 modulo 17.

a) 80 **b)** 103
c) -29 **d)** -122

36. Find each of these values.

a) $(177\ \mathbf{mod}\ 31 + 270\ \mathbf{mod}\ 31)\ \mathbf{mod}\ 31$
b) $(177\ \mathbf{mod}\ 31 \cdot 270\ \mathbf{mod}\ 31)\ \mathbf{mod}\ 31$

37. Find each of these values.

a) $(-133\ \mathbf{mod}\ 23 + 261\ \mathbf{mod}\ 23)\ \mathbf{mod}\ 23$
b) $(457\ \mathbf{mod}\ 23 \cdot 182\ \mathbf{mod}\ 23)\ \mathbf{mod}\ 23$

38. Find each of these values.

a) $(19^2\ \mathbf{mod}\ 41)\ \mathbf{mod}\ 9$
b) $(32^3\ \mathbf{mod}\ 13)^2\ \mathbf{mod}\ 11$
c) $(7^3\ \mathbf{mod}\ 23)^2\ \mathbf{mod}\ 31$
d) $(21^2\ \mathbf{mod}\ 15)^3\ \mathbf{mod}\ 22$

39. Find each of these values.

a) $(99^2\ \mathbf{mod}\ 32)^3\ \mathbf{mod}\ 15$
b) $(3^4\ \mathbf{mod}\ 17)^2\ \mathbf{mod}\ 11$
c) $(19^3\ \mathbf{mod}\ 23)^2\ \mathbf{mod}\ 31$
d) $(89^3\ \mathbf{mod}\ 79)^4\ \mathbf{mod}\ 26$

40. Show that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, where a, b, c, d, and m are integers with $m \ge 2$, then $a - c \equiv b - d \pmod{m}$.

41. Show that if $n \mid m$, where n and m are integers greater than 1, and if $a \equiv b \pmod{m}$, where a and b are integers, then $a \equiv b \pmod{n}$.

☞ **42.** Show that if a, b, c, and m are integers such that $m \ge 2$, $c > 0$, and $a \equiv b \pmod{m}$, then $ac \equiv bc \pmod{mc}$.

43. Find counterexamples to each of these statements about congruences.

a) If $ac \equiv bc \pmod{m}$, where a, b, c, and m are integers with $m \ge 2$, then $a \equiv b \pmod{m}$.
b) If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, where a, b, c, d, and m are integers with c and d positive and $m \ge 2$, then $a^c \equiv b^d \pmod{m}$.

44. Show that if n is an integer then $n^2 \equiv 0$ or $1 \pmod{4}$.

45. Use Exercise 44 to show that if m is a positive integer of the form $4k + 3$ for some nonnegative integer k, then m is not the sum of the squares of two integers.

46. Prove that if n is an odd positive integer, then $n^2 \equiv 1 \pmod{8}$.

47. Show that if a, b, k, and m are integers such that $k \ge 1$, $m \ge 2$, and $a \equiv b \pmod{m}$, then $a^k \equiv b^k \pmod{m}$.

48. Show that $\mathbf{Z}_m$ with addition modulo m, where $m \ge 2$ is an integer, satisfies the closure, associative, and commutative properties, 0 is an additive identity, and for every nonzero $a \in \mathbf{Z}_m$, $m - a$ is an inverse of a modulo m.

49. Show that $\mathbf{Z}_m$ with multiplication modulo m, where $m \ge 2$ is an integer, satisfies the closure, associative, and commutativity properties, and 1 is a multiplicative identity.

50. Show that the distributive property of multiplication over addition holds for $\mathbf{Z}_m$, where $m \ge 2$ is an integer.

51. Write out the addition and multiplication tables for $\mathbf{Z}_5$ (where by addition and multiplication we mean $+_5$ and $\cdot_5$).

52. Write out the addition and multiplication tables for $\mathbf{Z}_6$ (where by addition and multiplication we mean $+_6$ and $\cdot_6$).

53. Determine whether each of the functions $f(a) = a\ \mathbf{div}\ d$ and $g(a) = a\ \mathbf{mod}\ d$, where d is a fixed positive integer, from the set of integers to the set of integers, is one-to-one, and determine whether each of these functions is onto.

4.2 Integer Representations and Algorithms

4.2.1 Introduction

Integers can be expressed using any integer greater than one as a base, as we will show in this section. Although we commonly use decimal (base 10), representations, binary (base 2), octal (base 8), and hexadecimal (base 16) representations are often used, especially in computer science. Given a base b and an integer n, we will show how to construct the base b representation of this integer. We will also explain how to quickly convert between binary and octal and between binary and hexadecimal notations.

As mentioned in Section 3.1, the term *algorithm* originally referred to procedures for performing arithmetic operations using the decimal representations of integers. These algorithms, adapted for use with binary representations, are the basis for computer arithmetic. They provide good illustrations of the concept of an algorithm and the complexity of algorithms. For these reasons, they will be discussed in this section.

We will also introduce an algorithm for finding a **div** d and a **mod** d where a and d are integers with $d > 1$. Finally, we will describe an efficient algorithm for modular exponentiation, which is a particularly important algorithm for cryptography, as we will see in Section 4.6.

4.2.2 Representations of Integers

In everyday life we use decimal notation to express integers. In decimal notation, an integer n is written as a sum of the form $a_k 10^k + a_{k-1} 10^{k-1} + \cdots + a_1 10 + a_0$, where a_j is an integer with $0 \leq a_j \leq 9$ for $j = 0, 1, \ldots, k$. For example, 965 is used to denote $9 \cdot 10^2 + 6 \cdot 10 + 5$. However, it is often convenient to use bases other than 10. In particular, computers usually use binary notation (with 2 as the base) when carrying out arithmetic, and octal (base 8) or hexadecimal (base 16) notation when expressing characters, such as letters or digits. In fact, we can use any integer greater than 1 as the base when expressing integers. This is stated in Theorem 1.

THEOREM 1

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

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

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

A proof of this theorem can be constructed using mathematical induction, a proof method that is discussed in Section 5.1. It can also be found in [Ro10]. The representation of n given in Theorem 1 is called the **base *b* expansion of *n***. The base b expansion of n is denoted by $(a_k a_{k-1} \ldots a_1 a_0)_b$. For instance, $(245)_8$ represents $2 \cdot 8^2 + 4 \cdot 8 + 5 = 165$. Typically, the subscript 10 is omitted for base 10 expansions of integers because base 10, or **decimal expansions**, are commonly used to represent integers.

BINARY EXPANSIONS Choosing 2 as the base gives **binary expansions** of integers. In binary notation each digit is either a 0 or a 1. In other words, the binary expansion of an integer is just a bit string. Binary expansions (and related expansions that are variants of binary expansions) are used by computers to represent and do arithmetic with integers.

EXAMPLE 1 What is the decimal expansion of the integer that has $(1\ 0101\ 1111)_2$ as its binary expansion?

Solution: We have

$$(1\ 0101\ 1111)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351.$$ ◀

OCTAL AND HEXADECIMAL EXPANSIONS Among the most important bases in computer science are base 2, base 8, and base 16. Base 8 expansions are called **octal** expansions and base 16 expansions are **hexadecimal** expansions.

EXAMPLE 2 What is the decimal expansion of the number with octal expansion $(7016)_8$?

Solution: Using the definition of a base b expansion with $b = 8$ tells us that

$$(7016)_8 = 7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8 + 6 = 3598.$$ ◀

Sixteen different digits are required for hexadecimal expansions. Usually, the hexadecimal digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F, where the letters A through F represent the digits corresponding to the numbers 10 through 15 (in decimal notation).

EXAMPLE 3 What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

Solution: Using the definition of a base b expansion with $b = 16$ tells us that

$$(2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16 + 11 = 175627.$$ ◀

Each hexadecimal digit can be represented using four bits. For instance, we see that $(1110\ \ 0101)_2 = (E5)_{16}$ because $(1110)_2 = (E)_{16}$ and $(0101)_2 = (5)_{16}$. **Bytes**, which are bit strings of length eight, can be represented by two hexadecimal digits.

BASE CONVERSION We will now describe an algorithm for constructing the base b expansion of an integer n. First, divide n by b to obtain a quotient and remainder, that is,

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

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

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

We see that a_1 is the second digit from the right in the base b expansion of n. Continue this process, successively dividing the quotients by b, obtaining additional base b digits as the remainders. This process terminates when we obtain a quotient equal to zero. It produces the base b digits of n from the right to the left.

EXAMPLE 4 Find the octal expansion of $(12345)_{10}$.

Extra Examples

Solution: First, divide 12345 by 8 to obtain

$$12345 = 8 \cdot 1543 + 1.$$

Successively dividing quotients by 8 gives

$$\begin{aligned} 1543 &= 8 \cdot 192 + 7, \\ 192 &= 8 \cdot 24 + 0, \\ 24 &= 8 \cdot 3 + 0, \\ 3 &= 8 \cdot 0 + 3. \end{aligned}$$

The successive remainders that we have found, 1, 7, 0, 0, and 3, are the digits from the right to the left of 12345 in base 8. Hence,

$$(12345)_{10} = (30071)_8.$$ ◀

EXAMPLE 5 Find the hexadecimal expansion of $(177130)_{10}$.

Solution: First divide 177130 by 16 to obtain

$$177130 = 16 \cdot 11070 + 10.$$

Successively dividing quotients by 16 gives

$$\begin{aligned} 11070 &= 16 \cdot 691 + 14, \\ 691 &= 16 \cdot 43 + 3, \\ 43 &= 16 \cdot 2 + 11, \\ 2 &= 16 \cdot 0 + 2. \end{aligned}$$

The successive remainders that we have found, 10, 14, 3, 11, 2, give us the digits from the right to the left of 177130 in the hexadecimal (base 16) expansion of $(177130)_{10}$. It follows that

$$(177130)_{10} = (2B3EA)_{16}.$$

(Recall that the integers 10, 11, and 14 correspond to the hexadecimal digits A, B, and E, respectively.) ◀

EXAMPLE 6 Find the binary expansion of $(241)_{10}$.

Solution: First divide 241 by 2 to obtain

$$241 = 2 \cdot 120 + 1.$$

Successively dividing quotients by 2 gives

$$\begin{aligned} 120 &= 2 \cdot 60 + 0, \\ 60 &= 2 \cdot 30 + 0, \\ 30 &= 2 \cdot 15 + 0, \\ 15 &= 2 \cdot 7 + 1, \\ 7 &= 2 \cdot 3 + 1, \\ 3 &= 2 \cdot 1 + 1, \\ 1 &= 2 \cdot 0 + 1. \end{aligned}$$

The successive remainders that we have found, 1, 0, 0, 0, 1, 1, 1, 1, are the digits from the right to the left in the binary (base 2) expansion of $(241)_{10}$. Hence,

$$(241)_{10} = (1111\ 0001)_2.$$ ◀

The pseudocode given in Algorithm 1 finds the base b expansion $(a_{k-1} \ldots a_1 a_0)_b$ of the integer n.

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

ALGORITHM 1 Constructing Base b Expansions.

procedure *base b expansion*(n, b: positive integers with $b > 1$)
$q := n$
$k := 0$
while $q \neq 0$
    $a_k := q$ **mod** b
    $q := q$ **div** b
    $k := k + 1$
return $(a_{k-1}, \ldots, a_1, a_0)$ {$(a_{k-1} \ldots a_1 a_0)_b$ is the base b expansion of n}

In Algorithm 1, 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 these divisions and are given by q **mod** b. The algorithm terminates when a quotient $q = 0$ is reached.

Remark: Note that Algorithm 1 can be thought of as a greedy algorithm, because the base b digits are taken as large as possible in each step.

CONVERSION BETWEEN BINARY, OCTAL, AND HEXADECIMAL EXPANSIONS

Conversion between binary and octal and between binary and hexadecimal expansions is extremely easy because each octal digit corresponds to a block of three binary digits and each hexadecimal digit corresponds to a block of four binary digits, with these correspondences shown in Table 1 without initial 0s shown. (We leave it as Exercises 13–16 to show that this is the case.) This conversion is illustrated in Example 7.

EXAMPLE 7 Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$ and the binary expansions of $(765)_8$ and $(A8D)_{16}$.

Solution: To convert $(11\ 1110\ 1011\ 1100)_2$ into octal notation we group the binary digits into blocks of three, adding initial zeros at the start of the leftmost block if necessary. These blocks, from left to right, are 011, 111, 010, 111, and 100, corresponding to 3, 7, 2, 7, and 4, respectively. Consequently, $(11\ 1110\ 1011\ 1100)_2 = (37274)_8$. To convert $(11\ 1110\ 1011\ 1100)_2$ into hexadecimal notation we group the binary digits into blocks of four, adding initial zeros at the start of the leftmost block if necessary. These blocks, from left to right, are 0011, 1110, 1011, and 1100, corresponding to the hexadecimal digits 3, E, B, and C, respectively. Consequently, $(11\ 1110\ 1011\ 1100)_2 = (3EBC)_{16}$.

To convert $(765)_8$ into binary notation, we replace each octal digit by a block of three binary digits. These blocks are 111, 110, and 101. Hence, $(765)_8 = (1\ 1111\ 0101)_2$. To convert $(A8D)_{16}$ into binary notation, we replace each hexadecimal digit by a block of four binary digits. These blocks are 1010, 1000, and 1101. Hence, $(A8D)_{16} = (1010\ 1000\ 1101)_2$. ◀

4.2.3 Algorithms for Integer Operations

The algorithms for performing operations with integers using their binary expansions are extremely important in computer arithmetic. We will describe algorithms for the addition and the multiplication of two integers expressed in binary notation. We will also analyze the computational complexity of these algorithms, in terms of the actual number of bit operations used. Throughout this discussion, suppose that the binary expansions of a and b are

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

so that a and b each have n bits (putting bits equal to 0 at the beginning of one of these expansions if necessary).

We will measure the complexity of algorithms for integer arithmetic in terms of the number of bits in these numbers.

ADDITION ALGORITHM Consider the problem of adding two integers in binary notation. A procedure to perform addition can be based on the usual method for adding numbers with pencil and paper. This method proceeds by adding pairs of binary digits together with carries, when they occur, to compute the sum of two integers. This procedure will now be specified in detail.

To add a and b, first add their rightmost bits. This gives

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

where s_0 is the rightmost bit in the binary expansion of $a + b$ and c_0 is the **carry**, which is either 0 or 1. 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 (from the right) in the binary expansion of $a + b$, and c_1 is the carry. Continue this process, adding the corresponding bits in the two binary expansions and the carry, to determine the next bit from the right in the binary expansion of $a + b$. At the last stage, add a_{n-1}, b_{n-1}, and c_{n-2} to obtain $c_{n-1} \cdot 2 + s_{n-1}$. The leading bit of the sum is $s_n = c_{n-1}$. This procedure produces the binary expansion of the sum, namely, $a + b = (s_ns_{n-1}s_{n-2} \dots s_1s_0)_2$.

EXAMPLE 8 Add $a = (1110)_2$ and $b = (1011)_2$.

Solution: Following the procedure specified in the algorithm, first note that

$$a_0 + b_0 = 0 + 1 = 0 \cdot 2 + 1,$$

so that $c_0 = 0$ and $s_0 = 1$. Then, because

$$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 \cdot 2 + 0,$$

it follows that $c_1 = 1$ and $s_1 = 0$. Continuing,

$$a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 \cdot 2 + 0,$$

so that $c_2 = 1$ and $s_2 = 0$. Finally, because

$$a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 \cdot 2 + 1,$$

follows that $c_3 = 1$ and $s_3 = 1$. This means that $s_4 = c_3 = 1$. Therefore, $s = a + b = (1\ 1001)_2$. This addition is displayed in Figure 1, where carries are shown in color. ◀

```
 1 1 1
   1 1 1 0
+  1 0 1 1
-----------
 1 1 0 0 1
```

FIGURE 1 **Adding $(1110)_2$ and $(1011)_2$.**

The algorithm for addition can be described using pseudocode as follows.

ALGORITHM 2 Addition of Integers.

procedure *add*(a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1}a_{n-2} \ldots a_1a_0)_2$
and $(b_{n-1}b_{n-2} \ldots b_1b_0)_2$, respectively}
$c := 0$
for $j := 0$ **to** $n - 1$
$d := \lfloor (a_j + b_j + c)/2 \rfloor$
$s_j := a_j + b_j + c - 2d$
$c := d$
$s_n := c$
return $(s_0, s_1, \ldots, s_n)$ {the binary expansion of the sum is $(s_ns_{n-1} \ldots s_0)_2$}

Next, the number of additions of bits used by Algorithm 2 will be analyzed.

EXAMPLE 9 How many additions of bits are required to use Algorithm 2 to add two integers with n bits (or less) in their binary representations?

Solution: Two integers are added by successively adding pairs of bits and, when it occurs, a carry. Adding each pair of bits and the carry requires two additions of bits. Thus, the total number of additions of bits used is less than twice the number of bits in the expansion. Hence, the number of additions of bits used by Algorithm 2 to add two n-bit integers is $O(n)$. ◀

MULTIPLICATION ALGORITHM Next, consider the multiplication of two n-bit integers a and b. The conventional algorithm (used when multiplying with pencil and paper) works as follows. Using the distributive law, we see that

$$\begin{aligned} ab &= a(b_0 2^0 + b_1 2^1 + \cdots + b_{n-1} 2^{n-1}) \\ &= a(b_0 2^0) + a(b_1 2^1) + \cdots + a(b_{n-1} 2^{n-1}). \end{aligned}$$

We can compute ab using this equation. We first note that $ab_j = a$ if $b_j = 1$ and $ab_j = 0$ if $b_j = 0$. Each time we multiply a term by 2, we shift its binary expansion one place to the left and add a zero at the tail end of the expansion. Consequently, we can obtain $(ab_j)2^j$ by **shifting** the binary expansion of ab_j j places to the left, adding j zero bits at the tail end of this binary expansion. Finally, we obtain ab by adding the n integers $ab_j 2^j$, $j = 0, 1, 2, \ldots, n - 1$.

Algorithm 3 displays this procedure for multiplication.

ALGORITHM 3 Multiplication of Integers.

procedure *multiply*(a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1}a_{n-2} \ldots a_1a_0)_2$
and $(b_{n-1}b_{n-2} \ldots b_1b_0)_2$, respectively}
for $j := 0$ **to** $n - 1$
    if $b_j = 1$ **then** $c_j :=$ a shifted j places
    else $c_j := 0$
{$c_0, c_1, \ldots, c_{n-1}$ are the partial products}
$p := 0$
for $j := 0$ **to** $n - 1$
    $p := add(p, c_j)$
return p {p is the value of ab}

Example 10 illustrates the use of this algorithm.

EXAMPLE 10 Find the product of $a = (110)_2$ and $b = (101)_2$.

Solution: First note that

$$ab_0 \cdot 2^0 = (110)_2 \cdot 1 \cdot 2^0 = (110)_2,$$

$$ab_1 \cdot 2^1 = (110)_2 \cdot 0 \cdot 2^1 = (0000)_2,$$

and

$$ab_2 \cdot 2^2 = (110)_2 \cdot 1 \cdot 2^2 = (11000)_2.$$

To find the product, add $(110)_2$, $(0000)_2$, and $(11000)_2$. Carrying out these additions (using Algorithm 2, including initial zero bits when necessary) shows that $ab = (1\ 1110)_2$. This multiplication is displayed in Figure 2. ◀

```
  1 1 0
× 1 0 1
-------
  1 1 0
0 0 0
1 1 0
-------
1 1 1 1 0
```

FIGURE 2 Multiplying $(110)_2$ and $(101)_2$.

Next, we determine the number of additions of bits and shifts of bits used by Algorithm 3 to multiply two integers.

EXAMPLE 11 How many additions of bits and shifts of bits are used to multiply a and b using Algorithm 3?

Solution: Algorithm 3 computes the products of a and b by adding the partial products $c_0, c_1, c_2, \ldots,$ and c_{n-1}. When $b_j = 1$, we compute the partial product c_j by shifting the binary expansion of a by j bits. When $b_j = 0$, no shifts are required because $c_j = 0$. Hence, to find all n of the integers $ab_j2^j, j = 0, 1, \ldots, n - 1$, requires at most

$$0 + 1 + 2 + \cdots + n - 1$$

shifts. Hence, by Example 5 in Section 3.2 the number of shifts required is $O(n^2)$.

To add the integers ab_j from $j = 0$ to $j = n - 1$ requires the addition of an n-bit integer, an $(n + 1)$-bit integer, … , and a $(2n)$-bit integer. We know from Example 9 that each of these additions requires $O(n)$ additions of bits. Consequently, a total of $O(n^2)$ additions of bits are required for all n additions. ◀

Surprisingly, there are more efficient algorithms than the conventional algorithm for multiplying integers. One such algorithm, which uses $O(n^{1.585})$ bit operations to multiply n-bit numbers, will be described in Section 8.3.

ALGORITHM FOR div AND mod Given integers a and d, $d > 0$, we can find $q = a$ **div** d and $r = a$ **mod** d using Algorithm 4. In this brute-force algorithm, when a is positive we subtract d from a as many times as necessary until what is left is less than d. The number of times we perform this subtraction is the quotient and what is left over after all these subtractions is the remainder. Algorithm 4 also covers the case where a is negative. This algorithm finds the quotient q and remainder r when $|a|$ is divided by d. Then, when $a < 0$ and $r > 0$, it uses these to find the quotient $-(q + 1)$ and remainder $d - r$ when a is divided by d. We leave it to the reader (Exercise 65) to show that, assuming that $a > d$, this algorithm uses $O(q \log a)$ bit operations.

ALGORITHM 4 Computing div and mod.

procedure *division algorithm*(a: integer, d: positive integer)
$q := 0$
$r := |a|$
while $r \geq d$
    $r := r - d$
    $q := q + 1$
if $a < 0$ and $r > 0$ **then**
    $r := d - r$
    $q := -(q + 1)$
return (q, r) {$q = a$ **div** d is the quotient, $r = a$ **mod** d is the remainder}

There are more efficient algorithms than Algorithm 4 for determining the quotient $q = a$ **div** d and the remainder $r = a$ **mod** d when a positive integer a is divided by a positive integer d (see [Kn98] for details). These algorithms require $O(\log a \cdot \log d)$ bit operations. If both of the binary expansions of a and d contain n or fewer bits, then we can replace $\log a \cdot \log d$ by n^2. This means that we need $O(n^2)$ bit operations to find the quotient and remainder when a is divided by d.

4.2.4 Modular Exponentiation

In cryptography it is important to be able to find b^n **mod** m efficiently without using an excessive amount of memory, where b, n, and m are large integers. It is impractical to first compute b^n and then find its remainder when divided by m, because b^n can be a huge number and we will need a huge amount of computer memory to store such numbers. Instead, we can avoid time and memory problems by using an algorithm that employs the binary expansion of the exponent n.

Before we present an algorithm for fast modular exponentiation based on the binary expansion of the exponent, first observe that we can avoid using large amount of memory if we compute b^n **mod** m by successively computing b^k **mod** m for $k = 1, 2, \ldots, n$ using the fact that b^{k+1} **mod** $m = b(b^k$ **mod** $m)$ **mod** m (by Corollary 2 of Theorem 5 of Section 4.1). (Recall that $1 \leq b < m$.) However, this approach is impractical because it requires $n - 1$ multiplications of integers and n might be huge.

To motivate the fast modular exponentiation algorithm, we illustrate its basic idea. We will explain how to use the binary expansion of n, say $n = (a_{k-1} \ldots a_1 a_0)_2$, to compute b^n. First, note that

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

If you are rusty with the laws for exponents, this is the time to review them! See Theorem 1 in Appendix 2.

This shows that to compute b^n, we need only compute the values of b, b^2, $(b^2)^2 = b^4$, $(b^4)^2 = b^8, \ldots, b^{2^k}$. Once we have these values, we multiply the terms b^{2^j} in this list, where $a_j = 1$. (For efficiency and to reduce space requirements, after multiplying by each term, we reduce the result modulo m.)

Note that $(b^{2^n})^2 = b^{2^{n+1}}$ when n is a nonnegative integer.

This gives us b^n. For example, to compute 3^{11} we first note that $11 = (1011)_2$, so that $3^{11} = 3^8 3^2 3^1$. By successively squaring, we find that $3^2 = 9$, $3^4 = 9^2 = 81$, and $3^8 = (81)^2 = 6561$. Consequently, $3^{11} = 3^8 3^2 3^1 = 6561 \cdot 9 \cdot 3 = 177{,}147$.

Be sure to reduce modulo m after each multiplication!

The algorithm successively finds b **mod** m, b^2 **mod** m, b^4 **mod** $m, \ldots, b^{2^{k-1}}$ **mod** m and multiplies together those terms b^{2^j} **mod** m where $a_j = 1$, finding the remainder of the product when divided by m after each multiplication. Note that we need only perform $O(\log_2(n))$ multiplications. Pseudocode for this algorithm is shown in Algorithm 5. Note that in Algorithm 5 we can use the most efficient algorithm available to compute values of the **mod** function, not necessarily Algorithm 4.

ALGORITHM 5 Fast Modular Exponentiation.

procedure *modular exponentiation*(b: integer, $n = (a_{k-1}a_{k-2} \ldots a_1 a_0)_2$,
m: positive integers)
$x := 1$
$power := b$ **mod** m
for $i := 0$ **to** $k - 1$
  if $a_i = 1$ **then** $x := (x \cdot power)$ **mod** m
  $power := (power \cdot power)$ **mod** m
return x {x equals b^n **mod** m}

We illustrate how Algorithm 5 works in Example 12.

EXAMPLE 12 Use Algorithm 5 to find 3^{644} **mod** 645.

Solution: Algorithm 5 initially sets $x = 1$ and $power = 3$ **mod** $645 = 3$. In the computation of 3^{644} **mod** 645, this algorithm determines 3^{2^j} **mod** 645 for $j = 1, 2, \ldots, 9$ by successively squaring and reducing modulo 645. If $a_j = 1$ (where a_j is the bit in the jth position in the binary expansion of 644, which is $(1010000100)_2$), it multiplies the current value of x by 3^{2^j} **mod** 645 and reduces the result modulo 645. Here are the steps used:

$i = 0$: Because $a_0 = 0$, we have $x = 1$ and $power = 3^2$ **mod** $645 = 9$ **mod** $645 = 9$;
$i = 1$: Because $a_1 = 0$, we have $x = 1$ and $power = 9^2$ **mod** $645 = 81$ **mod** $645 = 81$;
$i = 2$: Because $a_2 = 1$, we have $x = 1 \cdot 81$ **mod** $645 = 81$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;
$i = 3$: Because $a_3 = 0$, we have $x = 81$ and $power = 111^2$ **mod** $645 = 12{,}321$ **mod** $645 = 66$;
$i = 4$: Because $a_4 = 0$, we have $x = 81$ and $power = 66^2$ **mod** $645 = 4356$ **mod** $645 = 486$;
$i = 5$: Because $a_5 = 0$, we have $x = 81$ and $power = 486^2$ **mod** $645 = 236{,}196$ **mod** $645 = 126$;
$i = 6$: Because $a_6 = 0$, we have $x = 81$ and $power = 126^2$ **mod** $645 = 15{,}876$ **mod** $645 = 396$;
$i = 7$: Because $a_7 = 1$, we find that $x = (81 \cdot 396)$ **mod** $645 = 471$ and $power = 396^2$ **mod** $645 = 156{,}816$ **mod** $645 = 81$;
$i = 8$: Because $a_8 = 0$, we have $x = 471$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;
$i = 9$: Because $a_9 = 1$, we find that $x = (471 \cdot 111)$ **mod** $645 = 36$.

This shows that following the steps of Algorithm 5 produces the result 3^{644} **mod** $645 = 36$. ◀

Algorithm 5 is quite efficient; it uses $O((\log m)^2 \log n)$ bit operations to find b^n **mod** m (see Exercise 64).

Exercises

1. Convert the decimal expansion of each of these integers to a binary expansion.
a) 231 **b)** 4532 **c)** 97644

2. Convert the decimal expansion of each of these integers to a binary expansion.
a) 321 **b)** 1023 **c)** 100632

3. Convert the binary expansion of each of these integers to a decimal expansion.
a) $(1\ 1111)_2$ **b)** $(10\ 0000\ 0001)_2$
c) $(1\ 0101\ 0101)_2$ **d)** $(110\ 1001\ 0001\ 0000)_2$

4. Convert the binary expansion of each of these integers to a decimal expansion.
a) $(1\ 1011)_2$ **b)** $(10\ 1011\ 0101)_2$
c) $(11\ 1011\ 1110)_2$ **d)** $(111\ 1100\ 0001\ 1111)_2$

5. Convert the octal expansion of each of these integers to a binary expansion.
a) $(572)_8$ **b)** $(1604)_8$
c) $(423)_8$ **d)** $(2417)_8$

6. Convert the binary expansion of each of these integers to an octal expansion.
a) $(1111\ 0111)_2$
b) $(1010\ 1010\ 1010)_2$
c) $(111\ 0111\ 0111\ 0111)_2$
d) $(101\ 0101\ 0101\ 0101)_2$

7. Convert the hexadecimal expansion of each of these integers to a binary expansion.
a) $(80E)_{16}$ **b)** $(135AB)_{16}$
c) $(ABBA)_{16}$ **d)** $(DEFACED)_{16}$

8. Convert $(BADFACED)_{16}$ from its hexadecimal expansion to its binary expansion.

9. Convert $(ABCDEF)_{16}$ from its hexadecimal expansion to its binary expansion.

10. Convert each of the integers in Exercise 6 from a binary expansion to a hexadecimal expansion.

11. Convert $(1011\ 0111\ 1011)_2$ from its binary expansion to its hexadecimal expansion.

12. Convert $(1\ 1000\ 0110\ 0011)_2$ from its binary expansion to its hexadecimal expansion.

13. Show that the hexadecimal expansion of a positive integer can be obtained from its binary expansion by grouping together blocks of four binary digits, adding initial zeros if necessary, and translating each block of four binary digits into a single hexadecimal digit.

14. Show that the binary expansion of a positive integer can be obtained from its hexadecimal expansion by translating each hexadecimal digit into a block of four binary digits.

15. Show that the octal expansion of a positive integer can be obtained from its binary expansion by grouping together blocks of three binary digits, adding initial zeros if necessary, and translating each block of three binary digits into a single octal digit.

16. Show that the binary expansion of a positive integer can be obtained from its octal expansion by translating each octal digit into a block of three binary digits.

17. Convert $(7345321)_8$ to its binary expansion and $(10\ 1011\ 1011)_2$ to its octal expansion.

18. Give a procedure for converting from the hexadecimal expansion of an integer to its octal expansion using binary notation as an intermediate step.

19. Give a procedure for converting from the octal expansion of an integer to its hexadecimal expansion using binary notation as an intermediate step.

20. Explain how to convert from binary to base 64 expansions and from base 64 expansions to binary expansions and from octal to base 64 expansions and from base 64 expansions to octal expansions.

21. Find the sum and the product of each of these pairs of numbers. Express your answers as a binary expansion.
a) $(100\ 0111)_2$, $(111\ 0111)_2$
b) $(1110\ 1111)_2$, $(1011\ 1101)_2$
c) $(10\ 1010\ 1010)_2$, $(1\ 1111\ 0000)_2$
d) $(10\ 0000\ 0001)_2$, $(11\ 1111\ 1111)_2$

22. Find the sum and product of each of these pairs of numbers. Express your answers as a base 3 expansion.
a) $(112)_3$, $(210)_3$
b) $(2112)_3$, $(12021)_3$
c) $(20001)_3$, $(1111)_3$
d) $(120021)_3$, $(2002)_3$

23. Find the sum and product of each of these pairs of numbers. Express your answers as an octal expansion.
a) $(763)_8$, $(147)_8$
b) $(6001)_8$, $(272)_8$
c) $(1111)_8$, $(777)_8$
d) $(54321)_8$, $(3456)_8$

24. Find the sum and product of each of these pairs of numbers. Express your answers as a hexadecimal expansion.
a) $(1AE)_{16}$, $(BBC)_{16}$
b) $(20CBA)_{16}$, $(A01)_{16}$
c) $(ABCDE)_{16}$, $(1111)_{16}$
d) $(E0000E)_{16}$, $(BAAA)_{16}$

25. Use Algorithm 5 to find 7^{644} **mod** 645.

26. Use Algorithm 5 to find 11^{644} **mod** 645.

27. Use Algorithm 5 to find 3^{2003} **mod** 99.

28. Use Algorithm 5 to find 123^{1001} **mod** 101.

29. Show that every positive integer can be represented uniquely as the sum of distinct powers of 2. [*Hint:* Consider binary expansions of integers.]

30. It can be shown that every integer can be uniquely represented in the form

$$e_k 3^k + e_{k-1} 3^{k-1} + \cdots + e_1 3 + e_0,$$

where $e_j = -1, 0,$ or 1 for $j = 0, 1, 2, \ldots, k$. Expansions of this type are called **balanced ternary expansions**. Find the balanced ternary expansions of
a) 5. **b)** 13. **c)** 37. **d)** 79.

31. Show that a positive integer is divisible by 3 if and only if the sum of its decimal digits is divisible by 3.

32. Show that a positive integer is divisible by 11 if and only if the difference of the sum of its decimal digits in even-numbered positions and the sum of its decimal digits in odd-numbered positions is divisible by 11.

33. Show that a positive integer is divisible by 3 if and only if the difference of the sum of its binary digits in even-numbered positions and the sum of its binary digits in odd-numbered positions is divisible by 3.

34. Determine how we can use the decimal expansion of an integer n to determine whether n is divisible by

a) 2 **b)** 5 **c)** 10

35. Determine how we can use the decimal expansion of an integer n to determine whether n is divisible by

a) 4 **b)** 25 **c)** 20

36. Suppose that n and b are positive integers with $b \geq 2$ and the base b expansion of n is $n = (a_m a_{m-1} \dots a_1 a_0)_b$. Find the base b expansion of

a) bn. **b)** $b^2 n$;

c) $\lfloor n/b \rfloor$, **d)** $\lfloor n/b^2 \rfloor$.

37. Prove that if n and b are positive integers with $b \geq 2$ the base b representation of n has $\lfloor \log_b n \rfloor + 1$ digits.

38. Find the decimal expansion of the number with the n-digit base seven expansion $(111 \dots 111)_7$ (with n 1's). [*Hint*: Use the formula for the sum of the terms of a geometric progression.]

39. Find the decimal expansion of the number with the $3n$ bit binary expansion $(101101....101101)_2$ (so that the binary expansion is made of n copies of 101). [*Hint*: Use the formula for the sum of the terms of a geometric progression.]

One's complement representations of integers are used to simplify computer arithmetic. To represent positive and negative integers with absolute value less than 2^{n-1}, a total of n bits is used. The leftmost bit is used to represent the sign. A 0 bit in this position is used for positive integers, and a 1 bit in this position is used for negative integers. For positive integers, the remaining bits are identical to the binary expansion of the integer. For negative integers, the remaining bits are obtained by first finding the binary expansion of the absolute value of the integer, and then taking the complement of each of these bits, where the complement of a 1 is a 0 and the complement of a 0 is a 1.

40. Find the one's complement representations, using bit strings of length six, of the following integers.

a) 22 **b)** 31 **c)** -7 **d)** -19

41. What integer does each of the following one's complement representations of length five represent?

a) 11001 **b)** 01101

c) 10001 **d)** 11111

42. If m is a positive integer less than 2^{n-1}, how is the one's complement representation of $-m$ obtained from the one's complement of m, when bit strings of length n are used?

43. How is the one's complement representation of the sum of two integers obtained from the one's complement representations of these integers?

44. How is the one's complement representation of the difference of two integers obtained from the one's complement representations of these integers?

45. Show that the integer m with one's complement representation $(a_{n-1}a_{n-2} \dots a_1 a_0)$ can be found using the equation $m = -a_{n-1}(2^{n-1} - 1) + a_{n-2}2^{n-2} + \cdots + a_1 \cdot 2 + a_0$.

Two's complement representations of integers are also used to simplify computer arithmetic and are used more commonly than one's complement representations. To represent an integer x with $-2^{n-1} \leq x \leq 2^{n-1} - 1$ for a specified positive integer n, a total of n bits is used. The leftmost bit is used to represent the sign. A 0 bit in this position is used for positive integers, and a 1 bit in this position is used for negative integers, just as in one's complement expansions. For a positive integer, the remaining bits are identical to the binary expansion of the integer. For a negative integer, the remaining bits are the bits of the binary expansion of $2^{n-1} - |x|$. Two's complement expansions of integers are often used by computers because addition and subtraction of integers can be performed easily using these expansions, where these integers can be either positive or negative.

46. Answer Exercise 40, but this time find the two's complement expansion using bit strings of length six.

47. Answer Exercise 41 if each expansion is a two's complement expansion of length five.

48. Answer Exercise 42 for two's complement expansions.

49. Answer Exercise 43 for two's complement expansions.

50. Answer Exercise 44 for two's complement expansions.

51. Show that the integer m with two's complement representation $(a_{n-1}a_{n-2} \dots a_1 a_0)$ can be found using the equation $m = -a_{n-1} \cdot 2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 \cdot 2 + a_0$.

52. Give a simple algorithm for forming the two's complement representation of an integer from its one's complement representation.

53. Sometimes integers are encoded by using four-digit binary expansions to represent each decimal digit. This produces the **binary coded decimal** form of the integer. For instance, 791 is encoded in this way by 011110010001. How many bits are required to represent a number with n decimal digits using this type of encoding?

A **Cantor expansion** is a sum of the form

$$a_n n! + a_{n-1}(n-1)! + \cdots + a_2 2! + a_1 1!,$$

where a_i is an integer with $0 \leq a_i \leq i$ for $i = 1, 2, \dots, n$.

54. Find the Cantor expansions of

a) 2. **b)** 7.

c) 19. **d)** 87.

e) 1000. **f)** 1,000,000.

*55. Describe an algorithm that finds the Cantor expansion of an integer.

*56. Describe an algorithm to add two integers from their Cantor expansions.

57. Add $(10111)_2$ and $(11010)_2$ by working through each step of the algorithm for addition given in the text.

58. Multiply $(1110)_2$ and $(1010)_2$ by working through each step of the algorithm for multiplication given in the text.

59. Describe an algorithm for finding the difference of two binary expansions.

60. Estimate the number of bit operations used to subtract two binary expansions.

61. Devise an algorithm that, given the binary expansions of the integers a and b, determines whether $a > b$, $a = b$, or $a < b$.

62. How many bit operations does the comparison algorithm from Exercise 61 use when the larger of a and b has n bits in its binary expansion?

63. Estimate the complexity of Algorithm 1 for finding the base b expansion of an integer n in terms of the number of divisions used.

*64. Show that Algorithm 5 uses $O((\log m)^2 \log n)$ bit operations to find b^n **mod** m.

65. Show that Algorithm 4 uses $O(q \log a)$ bit operations, assuming that $a > d$.

4.3 Primes and Greatest Common Divisors

4.3.1 Introduction

In Section 4.1 we studied the concept of divisibility of integers. One important concept based on divisibility is that of a prime number. A prime is an integer greater than 1 that is divisible by no positive integers other than 1 and itself. The study of prime numbers goes back to ancient times. Thousands of years ago it was known that there are infinitely many primes; the proof of this fact, found in the works of Euclid, is famous for its elegance and beauty.

We will discuss the distribution of primes among the integers. We will describe some of the results about primes found by mathematicians in the last 400 years. In particular, we will introduce an important theorem, the fundamental theorem of arithmetic. This theorem, which asserts that every positive integer can be written uniquely as the product of primes in nondecreasing order, has many interesting consequences. We will also discuss some of the many old conjectures about primes that remain unsettled today.

Primes have become essential in modern cryptographic systems, and we will develop some of their properties important in cryptography. For example, finding large primes is essential in modern cryptography. The length of time required to factor large integers into their prime factors is the basis for the strength of some important modern cryptographic systems.

In this section we will also study the greatest common divisor of two integers, as well as the least common multiple of two integers. We will develop an important algorithm for computing greatest common divisors, called the Euclidean algorithm.

4.3.2 Primes

Every integer greater than 1 is divisible by at least two integers, because a positive integer is divisible by 1 and by itself. Positive integers that have exactly two different positive integer factors are called **primes**.

Definition 1

An integer p greater than 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*.

Remark: The integer 1 is not prime, because it has only one positive factor. Note also that an integer n is composite if and only if there exists an integer a such that $a \mid n$ 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. ◀

The primes are the building blocks of positive integers, as the fundamental theorem of arithmetic shows. The proof will be given in Section 5.2.

THEOREM 1 **THE FUNDAMENTAL THEOREM OF ARITHMETIC** Every 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 gives some prime factorizations of integers.

EXAMPLE 2 The prime factorizations of 100, 641, 999, and 1024 are given by

Extra Examples ❯

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

4.3.3 Trial Division

It is often important to show that a given integer is prime. For instance, in cryptology, large primes are used in some methods for making messages secret. One procedure for showing that an integer is prime is based on the following observation.

THEOREM 2 If n is a composite integer, then n has a prime divisor less than or equal to $\sqrt{n}$.

Proof: If n is composite, by the definition of a composite integer, we know that it has a factor a with $1 < a < n$. Hence, by the definition of a factor of a positive integer, we have $n = ab$, where b is a positive integer greater than 1. We will show that $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$. If $a > \sqrt{n}$ and $b > \sqrt{n}$, then $ab > \sqrt{n} \cdot \sqrt{n} = n$, which is a contradiction. Consequently, $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$. Because both a and b are divisors of n, we see that n has a positive divisor not exceeding $\sqrt{n}$. This divisor is either prime or, by the fundamental theorem of arithmetic, has a prime divisor less than itself. In either case, n has a prime divisor less than or equal to $\sqrt{n}$. ◁

From Theorem 2, it follows that an integer is prime if it is not divisible by any prime less than or equal to its square root. This leads to the brute-force algorithm known as **trial division**. To use trial division we divide n by all primes not exceeding $\sqrt{n}$ and conclude that n is prime if it is not divisible by any of these primes. In Example 3 we use trial division to show that 101 is prime.

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. ◀

Because every integer has a prime factorization, it would be useful to have a procedure for finding this prime factorization. Consider the problem of finding the prime factorization of n. Begin by dividing n by successive primes, starting with the smallest prime, 2. If n has a prime factor, then by Theorem 3 a prime factor p not exceeding $\sqrt{n}$ will be found. So, if no prime factor not exceeding $\sqrt{n}$ is found, then n is prime. Otherwise, if a prime factor p is found, continue by factoring n/p. Note that n/p has no prime factors less than p. Again, if n/p has no prime factor greater than or equal to p and not exceeding its square root, then it is prime. Otherwise, if it has a prime factor q, continue by factoring $n/(pq)$. This procedure is continued until the factorization has been reduced to a prime. This procedure is illustrated in Example 4.

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. None of the primes 2, 3, and 5 divides 7007. However, 7 divides 7007, with $7007/7 = 1001$. Next, divide 1001 by successive primes, beginning with 7. It is immediately seen that 7 also divides 1001, because $1001/7 = 143$. Continue by dividing 143 by successive primes, beginning with 7. Although 7 does not divide 143, 11 does divide 143, and $143/11 = 13$. Because 13 is prime, the procedure is completed. It follows that $7007 = 7 \cdot 1001 = 7 \cdot 7 \cdot 143 = 7 \cdot 7 \cdot 11 \cdot 13$. Consequently, the prime factorization of 7007 is $7 \cdot 7 \cdot 11 \cdot 13 = 7^2 \cdot 11 \cdot 13$. ◀

Links

Prime numbers were studied in ancient times for philosophical reasons. Today, there are highly practical reasons for their study. In particular, large primes play a crucial role in cryptography, as we will see in Section 4.6.

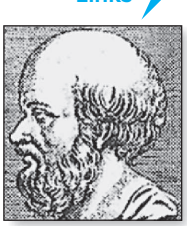
4.3.4 The Sieve of Eratosthenes

Note that composite integers not exceeding 100 must have a prime factor not exceeding 10. Because the only primes less than 10 are 2, 3, 5, and 7, the primes not exceeding 100 are these four primes and those positive integers greater than 1 and not exceeding 100 that are divisible by none of 2, 3, 5, or 7.

Links

The **sieve of Eratosthenes** is used to find all primes not exceeding a specified positive integer. For instance, the following procedure is used to find the primes not exceeding 100. We begin with the list of all integers between 1 and 100. To begin the sieving process, the integers that are divisible by 2, other than 2, are deleted. Because 3 is the first integer greater than 2 that is left, all those integers divisible by 3, other than 3, are deleted. Because 5 is the next integer left after 3, those integers divisible by 5, other than 5, are deleted. The next integer left is 7,

Links



Source: Math Tutor Archive

ERATOSTHENES (276 B.C.E.–194 B.C.E.) It is known that Eratosthenes was born in Cyrene, a Greek colony west of Egypt, and spent time studying at Plato's Academy in Athens. We also know that King Ptolemy II invited Eratosthenes to Alexandria to tutor his son and that later Eratosthenes became chief librarian at the famous library at Alexandria, a central repository of ancient wisdom. Eratosthenes was an extremely versatile scholar, writing on mathematics, geography, astronomy, history, philosophy, and literary criticism. Besides his work in mathematics, he is most noted for his chronology of ancient history and for his famous measurement of the size of the earth.

TABLE 1 The Sieve of Eratosthenes.

Integers divisible by 2 other than 2 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 3 other than 3 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 5 other than 5 receive an underline.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Integers divisible by 7 other than 7 receive an underline; integers in color are prime.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

so those integers divisible by 7, other than 7, are deleted. Because all composite integers not exceeding 100 are divisible by 2, 3, 5, or 7, all remaining integers except 1 are prime. In Table 1, the panels display those integers deleted at each stage, where each integer divisible by 2, other than 2, is underlined in the first panel, each integer divisible by 3, other than 3, is underlined in the second panel, each integer divisible by 5, other than 5, is underlined in the third panel, and each integer divisible by 7, other than 7, is underlined in the fourth panel. The integers not underlined are the primes not exceeding 100. We conclude that the primes less than 100 are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, and 97.

THE INFINITUDE OF PRIMES It has long been known that there are infinitely many primes. This means that whenever $p_1, p_2, \ldots, p_n$ are the n smallest primes, we know there is a larger prime not listed. We will prove this fact using a proof given by Euclid in his famous mathematics text, *The Elements*. This simple, yet elegant, proof is considered by many mathematicians to be among the most beautiful proofs in mathematics. It is the first proof presented in the book *Proofs from THE BOOK* (AiZi[14]), where THE BOOK refers to the imagined collection of perfect proofs that the legendary mathematician Paul Erdős claimed is maintained by God. By the way, there are a vast number of different proofs that there are an infinitude of primes, and new ones are published surprisingly frequently.

THEOREM 3 There are infinitely many primes.

Proof: We will prove this theorem using a proof by contradiction. We assume that there are only finitely many primes, $p_1, p_2, \ldots, p_n$. Let

$$Q = p_1 p_2 \cdots p_n + 1.$$

By the fundamental theorem of arithmetic, Q is prime or else it can be written as the product of two or more primes. However, none of the primes p_j divides Q, for if $p_j \mid Q$, then p_j divides $Q - p_1 p_2 \cdots p_n = 1$. Hence, there is a prime not in the list $p_1, p_2, \ldots, p_n$. This prime is either Q, if it is prime, or a prime factor of Q. This is a contradiction because we assumed that we have listed all the primes. Consequently, there are infinitely many primes. ◁

Remark: Note that in this proof we do *not* state that Q is prime! Furthermore, in this proof, we have given a nonconstructive existence proof that given any n primes, there is a prime not in this list. For this proof to be constructive, we would have had to explicitly give a prime not in our original list of n primes.

Because there are infinitely many primes, given any positive integer there are primes greater than this integer. There is an ongoing quest to discover larger and larger prime numbers; for almost all the last 300 years, the largest prime known has been an integer of the special form $2^p - 1$, where p is also prime. (Note that $2^n - 1$ cannot be prime when n is not prime; see Exercise 9.) Such primes are called **Mersenne primes**, after the French monk Marin Mersenne, who studied them in the seventeenth century. The reason that the largest known prime has usually been a Mersenne prime is that there is an extremely efficient test, known as the Lucas–Lehmer test, for determining whether $2^p - 1$ is prime. Furthermore, it is not currently possible to test numbers not of this or certain other special forms anywhere near as quickly to determine whether they are prime.

EXAMPLE 5 The numbers $2^2 - 1 = 3, 2^3 - 1 = 7, 2^5 - 1 = 31$ and $2^7 - 1 = 127$ are Mersenne primes, while $2^{11} - 1 = 2047$ is not a Mersenne prime because $2047 = 23 \cdot 89$. ◀

Progress in finding Mersenne primes has been steady since computers were invented. As of early 2018, 50 Mersenne primes were known, with 19 found since 1990. The largest Mersenne prime known (again as of early 2018) is $2^{77,232,917} - 1$, a number with 23,249,425 decimal

Links

©Apic/Getty Images

MARIN MERSENNE (1588–1648) Mersenne was born in Maine, France, into a family of laborers and attended the College of Mans and the Jesuit College at La Flèche. He continued his education at the Sorbonne, studying theology from 1609 to 1611. He joined the religious order of the Minims in 1611, a group whose name comes from the word *minimi* (the members of this group were extremely humble; they considered themselves the least of all religious orders). Besides prayer, the members of this group devoted their energy to scholarship and study. In 1612 he became a priest at the Place Royale in Paris; between 1614 and 1618 he taught philosophy at the Minim Convent at Nevers. He returned to Paris in 1619, where his cell in the Minims de l'Annociade became a place for meetings of French scientists, philosophers, and mathematicians, including Fermat and Pascal. Mersenne corresponded extensively with scholars throughout Europe, serving as a clearinghouse for mathematical and scientific knowledge, a function later served by mathematical journals (and today also by the Internet). Mersenne books covering mechanics, wrote mathematical physics, mathematics, music, and acoustics. He studied prime numbers and tried unsuccessfully to construct a formula representing all primes. In 1644 Mersenne claimed that $2^p - 1$ is prime for $p = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$ but is composite for all other primes less than 257. It took over 300 years to determine that Mersenne's claim was wrong five times. Specifically, $2^p - 1$ is not prime for $p = 67$ and $p = 257$ but is prime for $p = 61$, $p = 87$, and $p = 107$. It is also noteworthy that Mersenne defended two of the most famous men of his time, Descartes and Galileo, from religious critics. He also helped expose alchemists and astrologers as frauds.

Links

digits, which was shown to be prime in December, 2017. A communal effort, the Great Internet Mersenne Prime Search (GIMPS), is devoted to the search for new Mersenne primes. You can join this search, and if you are lucky, find a new Mersenne prime and possibly even win a cash prize. By the way, even the search for Mersenne primes has practical implications. A commonly used quality control test for supercomputers is to replicate the Lucas–Lehmer test that establishes the primality of a large Mersenne prime. Also, in January, 2016, it was reported that a bug in the Intel Skylake processor was found when GIMPS software was run. (See [Ro10] for more information about the quest for finding Mersenne primes.)

THE DISTRIBUTION OF PRIMES Theorem 3 tells us that there are infinitely many primes. However, how many primes are less than a positive number x? This question interested mathematicians for many years; in the late eighteenth century, mathematicians produced large tables of prime numbers to gather evidence concerning the distribution of primes. Using this evidence, the great mathematicians of the day, including Gauss and Legendre, conjectured, but did not prove, Theorem 4.

THEOREM 4

THE PRIME NUMBER THEOREM The ratio of $\pi(x)$, the number of primes not exceeding x, and $x/\ln x$ approaches 1 as x grows without bound. (Here $\ln x$ is the natural logarithm of x.)

Links

The prime number theorem was first proved in 1896 by the French mathematician Jacques Hadamard and the Belgian mathematician Charles-Jean-Gustave-Nicholas de la Vallée-Poussin using the theory of complex variables. Although proofs not using complex variables have been found, all known proofs of the prime number theorem are quite complicated. Many refinements of the prime number theorem have been proved, with many addressing the error made by estimating $\pi(x)$ with $x/\ln x$, and by estimating $\pi(x)$ with other functions. Many unsolved questions remain in this area of study.

Table 2 displays $\pi(x)$, $x/\ln x$, and their ratio, for $x = 10^n$ where $3 \leq n \leq 10$. A tremendous amount of effort has been devoted to computing $\pi(x)$ for progressively larger values of x. As of late 2017, the number of primes less than or equal to 10^n has been determined for all positive integers n with $n \leq 26$. In particular, it is known that

$$\pi(10^{26}) = 1{,}699{,}246{,}750{,}872{,}437{,}141{,}327{,}603,$$

to the nearest integer

$$\pi(10^{26}) - (10^{26}/\ln 10^{26}) = 28{,}883{,}358{,}936{,}853{,}188{,}823{,}261,$$

TABLE 2 Approximating $\pi(x)$ by $x/\ln x$.

x	$\pi(x)$	$x/\ln x$	$\pi(x)/(x/\ln x)$
10^3	168	144.8	1.161
10^4	1229	1085.7	1.132
10^5	9592	8685.9	1.104
10^6	78,498	72,382.4	1.084
10^7	664,579	620,420.7	1.071
10^8	5,761,455	5,428,681.0	1.061
10^9	50,847,534	48,254,942.4	1.054
10^{10}	455,052,512	434,294,481.9	1.048

and up to six decimal places

$$\pi(10^{26})/(10^{26}/\ln(10^{26})) = 1.01729.$$

Links

You can find a great deal of computational data relating to $\pi(x)$ and functions that estimate $\pi(x)$ using the web.

We can use the prime number theorem to estimate the probability that a randomly chosen number is prime. (See Chapter 7 to learn the basics of probability theory.) The prime number theorem tells us that the number of primes not exceeding x can be approximated by $x/\ln x$. Consequently, the odds that a randomly selected positive integer less than n is prime are approximately $(n/\ln n)/n = 1/\ln n$. Sometimes we need to find a prime with a particular number of digits. We would like an estimate of how many integers with a particular number of digits we need to select before we encounter a prime. Using the prime number theorem and calculus, it can be shown that the probability that an integer n is prime is also approximately $1/\ln n$. For example, the odds that an integer near 10^{1000} is prime are approximately $1/\ln 10^{1000}$, which is approximately $1/2300$. (Note that if we choose only odd numbers, we double our chances of finding a prime.)

Using trial division with Theorem 2 gives procedures for factoring and for primality testing. However, these procedures are not efficient algorithms; many much more practical and efficient algorithms for these tasks have been developed. Factoring and primality testing have become important in the applications of number theory to cryptography. This has led to a great interest in developing efficient algorithms for both tasks. Clever procedures have been devised in the last 30 years for efficiently generating large primes. Moreover, in 2002, an important theoretical discovery was made by Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. They showed there is a polynomial-time algorithm in the number of bits in the binary expansion of an integer for determining whether a positive integer is prime. Algorithms based on their work use $O((\log n)^6)$ bit operations to determine whether a positive integer n is prime.

However, even though powerful new factorization methods have been developed in the same time frame, factoring large numbers remains extraordinarily more time-consuming than primality testing. No polynomial-time algorithm for factoring integers is known. Nevertheless, the challenge of factoring large numbers interests many people. There is a communal effort on the Internet to factor large numbers, especially those of the special form $k^n \pm 1$, where k is a small positive integer and n is a large positive integer (such numbers are called *Cunningham numbers*). At any given time, there is a list of the "Ten Most Wanted" large numbers of this type awaiting factorization.

Links

©Paul Fearn/Alamy Stock Photo

CHARLES-JEAN-GUSTAVE-NICHOLAS DE LA VALLÉE-POUSSIN (1866–1962) De la Vallée-Poussin, the son of a professor of geology, was born in Louvain, Belgium. He attended the Jesuit College at Mons, first studying philosophy, but then turning to engineering. After graduating, he devoted himself to mathematics instead of engineering. His most important contribution to mathematics was his proof of the prime number theorem. He also established results about the distribution of primes in arithmetic progressions and refined the prime number theorem to include error estimates. De la Vallée-Poussin made important contributions to differential equations, analysis, and approximation theory. He also wrote a textbook, *Cours d'analyse*, which had significant impact on mathematical thought in the first half of the twentieth century.

©bpk/Salomon/ullstein bild via Getty Images

JACQUES HADAMARD (1865–1963) Hadamard, whose father was a Latin teacher and mother a distinguished piano teacher, was born in Versailles, France. After graduating from college, he taught at a secondary school in Paris. After receiving his Ph.D. in 1892, he was a lecturer at the Faculté des Sciences of Bordeaux. Later, he served on the faculties of the Sorbonne, the Collège de France, the École Polytéchnique, and the École Centrale des Arts et Manufacturers. Hadamard made significant contributions to complex analysis, functional analysis, and mathematical physics. He was recognized as an innovative teacher, writing many articles about elementary mathematics that were used in French schools and a widely used elementary geometry book.

PRIMES AND ARITHMETIC PROGRESSIONS Every odd integer is in one of the two arithmetic progressions $4k + 1$ or $4k + 3$, $k = 1, 2, \ldots$. Because we know that there are infinitely many primes, we can ask whether there are infinitely many primes in both of these arithmetic progressions. The primes 5, 13, 17, 29, 37, 41, … are in the arithmetic progression $4k + 1$; the primes 3, 7, 11, 19, 23, 31, 43, … are in the arithmetic progression $4k + 3$. Looking at the evidence hints that there may be infinitely many primes in both progressions. What about other arithmetic progressions $ak + b$, $k = 1, 2, \ldots$, where no integer greater than one divides both a and b? Do they contain infinitely many primes? The answer was provided by the German mathematician G. Lejeune Dirichlet, who proved that every such arithmetic progression contains infinitely many primes. His proof, and all proofs found later, are beyond the scope of this book. However, it is possible to prove special cases of Dirichlet's theorem using the ideas developed in this book. For example, Exercises 54 and 55 ask for proofs that there are infinitely many primes in the arithmetic progressions $3k + 2$ and $4k + 3$, where k is a positive integer. (The hint for each of these exercises supplies the basic idea needed for the proof.)

We have explained that every arithmetic progression $ak + b$, $k = 1, 2, \ldots$, where a and b have no common factor greater than one, contains infinitely many primes. But are there long arithmetic progressions made up of just primes? For example, some exploration shows that 5, 11, 17, 23, 29 is an arithmetic progression of five primes and 199, 409, 619, 829, 1039, 1249, 1459, 1669, 1879, 2089 is an arithmetic progression of ten primes. In the 1930s, the legendary and prolific mathematician Paul Erdős conjectured that for every positive integer n greater than two, there is an arithmetic progression of length n made up entirely of primes. In 2006, Ben Green and Terence Tao were able to prove this conjecture. Their proof, considered to be a mathematical tour de force, is a nonconstructive proof that combines powerful ideas from several advanced areas of mathematics.

4.3.5 Conjectures and Open Problems About Primes

Number theory is noted as a subject for which it is easy to formulate conjectures, some of which are difficult to prove and others that remained open problems for many years. We will describe some conjectures in number theory and discuss their status in Examples 6–9.

EXAMPLE 6 *Extra Examples*

It would be useful to have a function $f(n)$ such that $f(n)$ is prime for all positive integers n. If we had such a function, we could find large primes for use in cryptography and other applications. Looking for such a function, we might check out different polynomial functions, as some mathematicians did several hundred years ago. After a lot of computation we may encounter

Links

Courtesy of Reed Hutchinson/UCLA

TERENCE TAO (BORN 1975) Tao was born in Australia. His father is a pediatrician and his mother taught mathematics at a Hong Kong secondary school. Tao was a child prodigy, teaching himself arithmetic at the age of two. At 10, he became the youngest contestant at the International Mathematical Olympiad (IMO); he won an IMO gold medal at 13. Tao received his bachelor's and master's degrees when he was 17, and began graduate studies at Princeton, receiving his Ph.D. in three years. In 1996 he became a faculty member at UCLA, where he continues to work.

Tao is extremely versatile; he enjoys working on problems in diverse areas, including harmonic analysis, partial differential equations, number theory, and combinatorics. You can follow his work by reading his blog, where he discusses progress on various problems. His most famous result is the Green-Tao theorem, which says that there are arbitrarily long arithmetic progressions of primes. Tao has made important contributions to the applications of mathematics, such as developing a method for reconstructing digital images using the least possible amount of information.

Tao has an amazing reputation among mathematicians; he has become a Mr. Fix-It for researchers in mathematics. The well-known mathematician Charles Fefferman, himself a child prodigy, has said that "if you're stuck on a problem, then one way out is to interest Terence Tao." Tao maintains a popular blog that describes his research work and many mathematical problems in great detail. In 2006 Tao was awarded a Fields Medal, the most prestigious award for mathematicians under the age of 40. He was also awarded a MacArthur Fellowship in 2006, and in 2008, he received the Allan T. Waterman award, which came with a $500,000 cash prize to support research work of scientists early in their careers. Tao's wife Laura is an engineer at the Jet Propulsion Laboratory.

the polynomial $f(n) = n^2 - n + 41$. This polynomial has the interesting property that $f(n)$ is prime for all positive integers n not exceeding 40. [We have $f(1) = 41$, $f(2) = 43$, $f(3) = 47$, $f(4) = 53$, and so on.] This can lead us to the conjecture that $f(n)$ is prime for all positive integers n. Can we settle this conjecture?

Solution: Perhaps not surprisingly, this conjecture turns out to be false; we do not have to look far to find a positive integer n for which $f(n)$ is composite, because $f(41) = 41^2 - 41 + 41 = 41^2$. Because $f(n) = n^2 - n + 41$ is prime for all positive integers n with $1 \leq n \leq 40$, we might be tempted to find a different polynomial with the property that $f(n)$ is prime for *all* positive integers n. However, there is no such polynomial. It can be shown that for every polynomial $f(n)$ with integer coefficients, there is a positive integer y such that $f(y)$ is composite. (See Exercise 23 in the Supplementary Exercises.) ◀

Many famous problems about primes still await ultimate resolution by clever people. We describe a few of the most accessible and better known of these open problems in Examples 7–9. Number theory is noted for its wealth of easy-to-understand conjectures that resist attack by all but the most sophisticated techniques, or simply resist all attacks. We present these conjectures to show that many questions that seem relatively simple remain unsettled even in the twenty-first century.

EXAMPLE 7 **Goldbach's Conjecture** In 1742, Christian Goldbach, in a letter to Leonhard Euler, conjectured that every odd integer n, $n > 5$, is the sum of three primes. Euler replied that this conjecture is equivalent to the conjecture that every even integer n, $n > 2$, is the sum of two primes (see Exercise 21 in the Supplementary Exercises). The conjecture that every even integer n, $n > 2$, is the sum of two primes is now called **Goldbach's conjecture**. We can check this conjecture for small even numbers. For example, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 5 + 3$, $10 = 7 + 3$, $12 = 7 + 5$, and so on. Goldbach's conjecture was verified by hand calculations for numbers up to the millions prior to the advent of computers. With computers it can be checked for extremely large numbers. As of early 2018, the conjecture has been checked for all positive even integers up to $4 \cdot 10^{18}$.

Links

Although no proof of Goldbach's conjecture has been found, most mathematicians believe it is true. Several theorems have been proved, using complicated methods from analytic number theory far beyond the scope of this book, establishing results weaker than Goldbach's conjecture. Among these are the result that every even integer greater than 2 is the sum of at most six primes (proved in 1995 by O. Ramaré) and that every sufficiently large positive integer is the sum of a prime and a number that is either prime or the product of two primes (proved in 1966 by J. R. Chen). Perhaps Goldbach's conjecture will be settled in the not too distant future. ◀

EXAMPLE 8 There are many conjectures asserting that there are infinitely many primes of certain special forms. A conjecture of this sort is the conjecture that there are infinitely many primes of the form $n^2 + 1$, where n is a positive integer. For example, $5 = 2^2 + 1$, $17 = 4^2 + 1$, $37 = 6^2 + 1$, and so on. The best result currently known is that there are infinitely many positive integers n such that $n^2 + 1$ is prime or the product of at most two primes (proved by Henryk Iwaniec in 1973 using advanced techniques from analytic number theory, far beyond the scope of this book). ◀

Links

EXAMPLE 9 **The Twin Prime Conjecture** **Twin primes** are pairs of primes that differ by 2, such as 3 and 5, 5 and 7, 11 and 13, 17 and 19, and 4967 and 4969. The twin prime conjecture asserts that

Links

CHRISTIAN GOLDBACH (1690–1764) Christian Goldbach was born in Königsberg, Prussia, the city noted for its famous bridge problem (which will be studied in Section 10.5). He became professor of mathematics at the Academy in St. Petersburg in 1725. In 1728 Goldbach went to Moscow to tutor the son of the Tsar. He entered the world of politics when, in 1742, he became a staff member in the Russian Ministry of Foreign Affairs. Goldbach is best known for his correspondence with eminent mathematicians, including Euler and Bernoulli, for his intriguing conjectures in number theory, and for several contributions to analysis.

Links

there are infinitely many twin primes. The strongest result proved concerning twin primes is that there are infinitely many pairs p and $p + 2$, where p is prime and $p + 2$ is prime or the product of two primes (proved by J. R. Chen in 1966).

The world's record for twin primes, as of early 2018, consists of the numbers $2{,}996{,}863{,}034{,}895 \cdot 2^{1{,}290{,}000} \pm 1$, which have 388,342 decimal digits.

Let $P(n)$ be the statement that there are infinitely many pairs of primes that differ by exactly n. The twin prime conjecture is the statement that $P(2)$ is true. Mathematicians working on the twin prime conjecture formulated a weaker conjecture, known as the *bounded gap conjecture*, which asserts that there is an integer N for which $P(N)$ is true. The mathematical community was surprised when Yitang Zhang, a 50-year-old professor at the University of New Hampshire, who had not published a paper since 2001, proved the bounded gap conjecture in 2013. In particular, he showed that there is an integer $N < 70{,}000{,}000$ such that $P(N)$ is true. A team of mathematicians, including Terrance Tao, lowered the Zhang's bound by showing that there is an integer $N \leq 246$ for which $P(N)$ is true. Furthermore, they showed that if a certain conjecture was true, it could be shown that $N \leq 6$ and that this is the best possible estimate that could be proved using the methods introduced by Zhang. ◀

4.3.6 Greatest Common Divisors and Least Common Multiples

The largest integer that divides both of two integers is called the **greatest common divisor** of these integers.

Definition 2

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

The greatest common divisor of two integers, not both zero, exists because the set of common divisors of these integers is nonempty and finite. One way to find the greatest common divisor of two integers is to find all the positive common divisors of both integers and then take the largest divisor. This is done in Examples 10 and 11. Later, a more efficient method of finding greatest common divisors will be given.

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, $\gcd(24, 36) = 12$. ◀

Links

Source: John D. & Catherine T. MacArthur Foundation

YITANG ZHANG (BORN 1955) Yitang Zhang was born in Shanghai, China, in 1955. When he was ten years old, he learned about famous conjectures, including Fermat's last theorem and the Goldbach conjecture. During the Cultural Revolution he spent ten years working in the fields instead of attending school. However, once this period was over, he was able to attend Peking University, receiving his bachelor's and master's degree in 1982 and 1984, respectively. He moved to the United States, attending Purdue University and completing the work for his Ph.D. in 1991.

After receiving his Ph.D., Zhang could not find an academic position because of the poor job market and disagreements with his thesis advisor. Instead he did accounting work and delivered food for a Queens, New York restaurant; he later worked in Kentucky at Subway restaurants owned by a friend. He even lived in his car while looking for work, but was finally able to obtain an academic job as a lecturer at the University of New Hampshire. He held this position from 1999 until early 2014. From 2009 to 2013, he worked on the bounded gap conjecture seven days a week, about ten hours a day, until he made his key discovery. His success led the University of New Hampshire to promote him to full professorship. In 2015, however, he accepted the offer of a full professorship at the University of California, Santa Barbara. Zhang was a awarded a MacArthur Fellowship, also known as a Genius Award, in 2014.

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 $\gcd(17, 22) = 1$. ◀

Because it is often important to specify that two integers have no common positive divisor other than 1, we have Definition 3.

Definition 3 The integers a and b are *relatively prime* if their greatest common divisor is 1.

EXAMPLE 12 By Example 11 it follows that the integers 17 and 22 are relatively prime, because $\gcd(17, 22) = 1$. ◀

Because we often need to specify that no two integers in a set of integers have a common positive divisor greater than 1, we make Definition 4.

Definition 4 The integers $a_1, a_2, \ldots, a_n$ are *pairwise relatively prime* if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

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. ◀

Another way to find the greatest common divisor of two positive integers is to use the prime factorizations of these integers. Suppose that the prime factorizations of the positive integers a and b are

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

where each exponent is a nonnegative integer, and where all primes occurring in the prime factorization of either a or b are included in both factorizations, with zero exponents if necessary. Then $\gcd(a, b)$ is given by

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

where $\min(x, y)$ represents the minimum of the two numbers x and y. To show that this formula for $\gcd(a, b)$ is valid, we must show that the integer on the right-hand side divides both a and b, and that no larger integer also does. This integer does divide both a and b, because the power of each prime in the factorization does not exceed the power of this prime in either the factorization of a or that of b. Further, no larger integer can divide both a and b, because the exponents of the primes in this factorization cannot be increased, and no other primes can be included.

EXAMPLE 14 Because the prime factorizations of 120 and 500 are $120 = 2^3 \cdot 3 \cdot 5$ and $500 = 2^2 \cdot 5^3$, the greatest common divisor is

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

Prime factorizations can also be used to find the **least common multiple** of two integers.

Definition 5 The *least common multiple* of the positive integers a and b is the smallest positive integer that is divisible by both a and b. The least common multiple of a and b is denoted by $\text{lcm}(a, b)$.

The least common multiple exists because the set of integers divisible by both a and b is nonempty (because ab belongs to this set, for instance), and every nonempty set of positive integers has a least element (by the well-ordering property, which will be discussed in Section 5.2). Suppose that the prime factorizations of a and b are as before. Then the least common multiple of a and b is given by

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

where $\max(x, y)$ denotes the maximum of the two numbers x and y. This formula is valid because a common multiple of a and b has at least $\max(a_i, b_i)$ factors of p_i in its prime factorization, and the least common multiple has no other prime factors besides those in a and b.

EXAMPLE 15 What is the least common multiple of $2^3 3^5 7^2$ and $2^4 3^3$?

Solution: We have

$$\text{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3,4)}3^{\max(5,3)}7^{\max(2,0)} = 2^4 3^5 7^2.$$ ◀

Theorem 5 gives the relationship between the greatest common divisor and least common multiple of two integers. It can be proved using the formulae we have derived for these quantities. The proof of this theorem is left as Exercise 31.

THEOREM 5 Let a and b be positive integers. Then

$$ab = \gcd(a, b) \cdot \text{lcm}(a, b).$$

4.3.7 The Euclidean Algorithm

Links

Computing the greatest common divisor of two integers directly from the prime factorizations of these integers is inefficient. The reason is that it is time-consuming to find prime factorizations. We will give a more efficient method of finding the greatest common divisor, called the **Euclidean algorithm**. This algorithm has been known since ancient times. It is named after the

ancient Greek mathematician Euclid, who included a description of this algorithm in his book *The Elements*.

Before describing the Euclidean algorithm, we will show how it is used to find gcd(91, 287). First, divide 287, the larger of the two integers, by 91, the smaller, to obtain

$$287 = 91 \cdot 3 + 14.$$

Any divisor of 91 and 287 must also be a divisor of $287 - 91 \cdot 3 = 14$. Also, any divisor of 91 and 14 must also be a divisor of $287 = 91 \cdot 3 + 14$. Hence, the greatest common divisor of 91 and 287 is the same as the greatest common divisor of 91 and 14. This means that the problem of finding gcd(91, 287) has been reduced to the problem of finding gcd(91, 14).

Next, divide 91 by 14 to obtain

$$91 = 14 \cdot 6 + 7.$$

Because any common divisor of 91 and 14 also divides $91 - 14 \cdot 6 = 7$ and any common divisor of 14 and 7 divides 91, it follows that gcd(91, 14) = gcd(14, 7).

Continue by dividing 14 by 7, to obtain

$$14 = 7 \cdot 2.$$

Because 7 divides 14, it follows that gcd(14, 7) = 7. Furthermore, because gcd(287, 91) = gcd(91, 14) = gcd(14, 7) = 7, the original problem has been solved.

We now describe how the Euclidean algorithm works in generality. We will use successive divisions to reduce the problem of finding the greatest common divisor of two positive integers to the same problem with smaller integers, until one of the integers is zero.

The Euclidean algorithm is based on the following result about greatest common divisors and the division algorithm.

LEMMA 1 Let $a = bq + r$, where a, b, q, and r are integers. Then $\gcd(a, b) = \gcd(b, r)$.

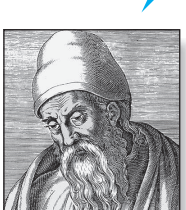
Proof: If we can show that the common divisors of a and b are the same as the common divisors of b and r, we will have shown that $\gcd(a, b) = \gcd(b, r)$, because both pairs must have the same *greatest* common divisor.

So suppose that d divides both a and b. Then it follows that d also divides $a - bq = r$ (from Theorem 1 of Section 4.1). Hence, any common divisor of a and b is also a common divisor of b and r.

Likewise, suppose that d divides both b and r. Then d also divides $bq + r = a$. Hence, any common divisor of b and r is also a common divisor of a and b.

Consequently, $\gcd(a, b) = \gcd(b, r)$. ◁

Links



©bilwissedition Ltd. & Co. KG/Alamy Stock Photo

EUCLID (325 B.C.E.–265 B.C.E.) Euclid was the author of the most successful mathematics book ever written, *The Elements*, which appeared in over 1000 different editions from ancient to modern times. Little is known about Euclid's life, other than that he taught at the famous academy at Alexandria in Egypt. Apparently, Euclid did not stress applications. When a student asked what he would get by learning geometry, Euclid explained that knowledge was worth acquiring for its own sake and told his servant to give the student a coin "because he must make a profit from what he learns."

Suppose that a and b are positive integers with $a \geq b$. Let $r_0 = a$ and $r_1 = b$. When we successively apply the division algorithm, we obtain

$$\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \leq r_2 < r_1, \\
r_1 &= r_2 q_2 + r_3 && 0 \leq r_3 < r_2, \\
&\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \leq r_n < r_{n-1}, \\
r_{n-1} &= r_n q_n.
\end{aligned}$$

Eventually a remainder of zero occurs in this sequence of successive divisions, because the sequence of remainders $a = r_0 > r_1 > r_2 > \cdots \geq 0$ cannot contain more than a terms. Furthermore, it follows from Lemma 1 that

$$\begin{aligned}
\gcd(a, b) &= \gcd(r_0, r_1) = \gcd(r_1, r_2) = \cdots = \gcd(r_{n-2}, r_{n-1}) \\
&= \gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n.
\end{aligned}$$

Hence, the greatest common divisor is the last nonzero remainder in the sequence of divisions.

EXAMPLE 16 Find the greatest common divisor of 414 and 662 using the Euclidean algorithm.

Solution: Successive uses of the division algorithm give:

$$\begin{aligned}
662 &= 414 \cdot 1 + 248 \\
414 &= 248 \cdot 1 + 166 \\
248 &= 166 \cdot 1 + 82 \\
166 &= 82 \cdot 2 + 2 \\
82 &= 2 \cdot 41.
\end{aligned}$$

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

We can summarize these steps in tabular form.

j	r_j	r_{j+1}	q_{j+1}	r_{j+2}
0	662	414	1	248
1	414	248	1	166
2	248	166	1	82
3	166	82	2	2
4	82	2	41	0

◀

The Euclidean algorithm is expressed in pseudocode in Algorithm 1.

ALGORITHM 1 The Euclidean Algorithm.

procedure *gcd*(a, b: positive integers)
$x := a$
$y := b$
while $y \neq 0$
  $r := x$ **mod** y
  $x := y$
  $y := r$
return x {gcd(a, b) is x}

In Algorithm 1, the initial values of x and y are a and b, respectively. At each stage of the procedure, x is replaced by y, and y is replaced by x **mod** y, which is the remainder when x is divided by y. This process is repeated as long as $y \neq 0$. The algorithm terminates when $y = 0$, and the value of x at that point, the last nonzero remainder in the procedure, is the greatest common divisor of a and b.

We will study the time complexity of the Euclidean algorithm in Section 5.3, where we will show that the number of divisions required to find the greatest common divisor of a and b, where $a \geq b$, is $O(\log b)$.

4.3.8 gcds as Linear Combinations

An important result we will use throughout the remainder of this section is that the greatest common divisor of two integers a and b can be expressed in the form

$$sa + tb,$$

where s and t are integers. In other words, $\gcd(a, b)$ can be expressed as a **linear combination** with integer coefficients of a and b. For example, $\gcd(6, 14) = 2$, and $2 = (-2) \cdot 6 + 1 \cdot 14$. We state this fact as Theorem 6.

THEOREM 6 **BÉZOUT'S THEOREM** If a and b are positive integers, then there exist integers s and t such that $\gcd(a, b) = sa + tb$.

Definition 6 If a and b are positive integers, then integers s and t such that $\gcd(a, b) = sa + tb$ are called *Bézout coefficients* of a and b (after Étienne Bézout, a French mathematician of the eighteenth century). Also, the equation $\gcd(a, b) = sa + tb$ is called *Bézout's identity*.

We will not give a formal proof of Theorem 6 here (see Exercise 36 in Section 5.2 and [Ro10] for proofs). We will present two different methods that can be used to find a linear combination of two integers equal to their greatest common divisor. (In this section, we will assume that a linear combination has integer coefficients.)

The first method proceeds by working backward through the divisions of the Euclidean algorithm, so this method requires a forward pass and a backward pass through the steps of the Euclidean algorithm. We will illustrate how this method works with an example. The main

Links

©Chronicle/Alamy Stock Photo

ÉTIENNE BÉZOUT (1730–1783) Bézout was born in Nemours, France, where his father was a magistrate. Reading the writings of the great mathematician Leonhard Euler enticed him to become a mathematician. In 1758 he was appointed to a position at the Académie des Sciences in Paris; in 1763 he was appointed examiner of the Gardes de la Marine, where he was assigned the task of writing mathematics textbooks. This assignment led to a four-volume textbook completed in 1767. Bézout is well known for his six-volume comprehensive textbook on mathematics. His textbooks were extremely popular and were studied by many generations of students hoping to enter the École Polytechnique, the highly regarded engineering and science school. His books were translated into English and used in North America, including at Harvard.

His most important original work was published in 1779 in the book *Théorie générale des équations algébriques*, where he introduced important methods for solving simultaneous polynomial equations in many unknowns. The most well-known result in this book is now called *Bézout's theorem*, which in its general form tells us that the number of common points on two plane algebraic curves equals the product of the degrees of these curves. Bézout is also credited with inventing the determinant (which was called the Bézoutian by the noted English mathematician James Joseph Sylvester). He was considered to be a kind person with a warm heart, although he had a reserved and somber personality. He was happily married and a father.

Links

advantage of the second method, known as the **extended Euclidean algorithm**, is that it uses one pass through the steps of the Euclidean algorithm to find Bézout coefficients of a and b, unlike the first method, which uses two passes. To run this extended Euclidean algorithm we set $s_0 = 1$, $s_1 = 0$, $t_0 = 0$, and $t_1 = 1$ and let

$$s_j = s_{j-2} - q_{j-1}s_{j-1} \text{ and } t_j = t_{j-2} - q_{j-1}t_{j-1}$$

for $j = 2, 3, \ldots, n$, where the q_j are the quotients in the divisions used when the Euclidean algorithm finds $\gcd(a, b)$, as shown in the text. We can prove by strong induction (see Exercise 44 in Section 5.2, or see [Ro10]) that $\gcd(a, b) = s_n a + t_n b$.

EXAMPLE 17 Express $\gcd(252, 198) = 18$ as a linear combination of 252 and 198 by working backwards through the steps of the Euclidean algorithm.

Solution: To show that $\gcd(252, 198) = 18$, the Euclidean algorithm uses these divisions:

$$252 = 198 \cdot 1 + 54$$
$$198 = 54 \cdot 3 + 36$$
$$54 = 36 \cdot 1 + 18$$
$$36 = 18 \cdot 2 + 0.$$

We summarize these steps in tabular form:

j	r_j	r_{j+1}	q_{j+1}	r_{j+2}
0	252	198	1	54
1	198	54	3	36
2	54	36	1	18
3	36	18	2	0

Using the next-to-last division (the third division), we can express $\gcd(252, 198) = 18$ as a linear combination of 54 and 36. We find that

$$18 = 54 - 1 \cdot 36.$$

The second division tells us that

$$36 = 198 - 3 \cdot 54.$$

Substituting this expression for 36 into the previous equation, we can express 18 as a linear combination of 54 and 198. We have

$$18 = 54 - 1 \cdot 36 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198.$$

The first division tells us that

$$54 = 252 - 1 \cdot 198.$$

Substituting this expression for 54 into the previous equation, we can express 18 as a linear combination of 252 and 198. We conclude that

$$18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198,$$

completing the solution. ◀

The next example shows how to solve the same problem posed in the previous example using the extended Euclidean algorithm.

EXAMPLE 18 Express $\gcd(252, 198) = 18$ as a linear combination of 252 and 198 using the extended Euclidean algorithm.

Solution: Example 17 displays the steps the Euclidean algorithm uses to find $\gcd(252, 198) = 18$. The quotients are $q_1 = 1$, $q_2 = 3$, $q_3 = 1$, and $q_4 = 2$. The desired Bézout coefficients are the values of s_4 and t_4 generated by the extended Euclidean algorithm, where $s_0 = 1$, $s_1 = 0$, $t_0 = 0$, and $t_1 = 1$, and

$$s_j = s_{j-2} - q_{j-1}s_{j-1} \qquad \text{and} \qquad t_j = t_{j-2} - q_{j-1}t_{j-1}$$

for $j = 2, 3, 4$. We find that

$$s_2 = s_0 - s_1q_1 = 1 - 0 \cdot 1 = 1,\ t_2 = t_0 - t_1q_1 = 0 - 1 \cdot 1 = -1,$$
$$s_3 = s_1 - s_2q_2 = 0 - 1 \cdot 3 = -3,\ t_3 = t_1 - t_2q_2 = 1 - (-1)3 = 4,$$
$$s_4 = s_2 - s_3q_3 = 1 - (-3) \cdot 1 = 4,\ t_4 = t_2 - t_3q_3 = -1 - 4 \cdot 1 = -5.$$

Because $s_4 = 4$ and $t_4 = -5$, we see that $18 = \gcd(252, 198) = 4 \cdot 252 - 5 \cdot 198$.

We summarize the steps of the extended Euclidean algorithm in a table:

j	r_j	r_{j+1}	q_{j+1}	r_{j+2}	s_j	t_j
0	252	198	1	54	1	0
1	198	54	3	36	0	1
2	54	36	1	18	1	−1
3	36	18	2	0	−3	4
4					4	−5

◀

We will use Theorem 6 to develop several useful results. One of our goals will be to prove the part of the fundamental theorem of arithmetic asserting that a positive integer has at most one prime factorization. We will show that if a positive integer has a factorization into primes, where the primes are written in nondecreasing order, then this factorization is unique.

First, we need to develop some results about divisibility.

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: Because $\gcd(a, b) = 1$, by Bézout's theorem there are integers s and t such that

$$sa + tb = 1.$$

Multiplying both sides of this equation by c, we obtain

$$sac + tbc = c.$$

We can now use Theorem 1 of Section 4.1 to show that $a \mid c$. By part (ii) of that theorem, $a \mid tbc$. Because $a \mid sac$ and $a \mid tbc$, by part (i) of that theorem, we conclude that a divides $sac + tbc$. Because $sac + tbc = c$, we conclude that $a \mid c$, completing the proof. ◁

We will use the following generalization of Lemma 2 in the proof of uniqueness of prime factorizations. (The proof of Lemma 3 is left as Exercise 64 in Section 5.1, because it can be most easily carried out using the method of mathematical induction, covered in that section.)

LEMMA 3 If p is a prime and $p \mid a_1a_2 \cdots a_n$, where each a_i is an integer, then $p \mid a_i$ for some i.

We can now show that a factorization of an integer into primes is unique. That is, we will show that every integer can be written as the product of primes in nondecreasing order in at most one way. This is part of the fundamental theorem of arithmetic. We will prove the other part, that every integer has a factorization into primes, in Section 5.2.

Proof (of the uniqueness of the prime factorization of a positive integer): We will use a proof by contradiction. Suppose that the positive integer n can be written as the product of primes in two different ways, say, $n = p_1 p_2 \cdots p_s$ and $n = q_1 q_2 \cdots q_t$, where each p_i and q_j is prime such that $p_1 \le p_2 \le \cdots \le p_s$ and $q_1 \le q_2 \le \cdots \le q_t$.

When we remove all common primes from the two factorizations, we have

$$p_{i_1} p_{i_2} \cdots p_{i_u} = q_{j_1} q_{j_2} \cdots q_{j_v},$$

where no prime occurs on both sides of this equation and u and v are positive integers. By Lemma 3 it follows that p_{i_1} divides q_{j_k} for some k. Because no prime divides another prime, this is impossible. Consequently, there can be at most one factorization of n into primes in nondecreasing order. ◁

Lemma 2 can also be used to prove a result about dividing both sides of a congruence by the same integer. We have shown (Theorem 5 in Section 4.1) that we can multiply both sides of a congruence by the same integer. However, dividing both sides of a congruence by an integer does not always produce a valid congruence, as Example 19 shows.

EXAMPLE 19 The congruence $14 \equiv 8 \pmod 6$ holds, but both sides of this congruence cannot be divided by 2 to produce a valid congruence because $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \pmod 6$. ◀

Although we cannot divide both sides of a congruence by any integer to produce a valid congruence, we can if this integer is relatively prime to the modulus. Theorem 7 establishes this important fact. We use Lemma 2 in the proof.

THEOREM 7 Let m be a positive integer and let a, b, and c be integers. If $ac \equiv bc \pmod m$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod m$.

Proof: Because $ac \equiv bc \pmod m$, $m \mid ac - bc = c(a - b)$. By Lemma 2, because $\gcd(c, m) = 1$, it follows that $m \mid a - b$. We conclude that $a \equiv b \pmod m$. ◁

Exercises

1. Determine whether each of these integers is prime.
a) 21 **b)** 29
c) 71 **d)** 97
e) 111 **f)** 143

2. Determine whether each of these integers is prime.
a) 19 **b)** 27
c) 93 **d)** 101
e) 107 **f)** 113

3. Find the prime factorization of each of these integers.
a) 88 **b)** 126 **c)** 729
d) 1001 **e)** 1111 **f)** 909,090

4. Find the prime factorization of each of these integers.
a) 39 **b)** 81 **c)** 101
d) 143 **e)** 289 **f)** 899

5. Find the prime factorization of 10!.

***6.** How many zeros are there at the end of 100!?

7. Express in pseudocode the trial division algorithm for determining whether an integer is prime.

8. Express in pseudocode the algorithm described in the text for finding the prime factorization of an integer.

9. Show that $a^m + 1$ is composite if a and m are integers greater than 1 and m is odd. [*Hint:* Show that $x + 1$ is a factor of the polynomial $x^m + 1$ if m is odd.]

10. Show that if $2^m + 1$ is an odd prime, then $m = 2^n$ for some nonnegative integer n. [*Hint:* First show that the polynomial identity $x^m + 1 = (x^k + 1)(x^{k(t-1)} - x^{k(t-2)} + \cdots - x^k + 1)$ holds, where $m = kt$ and t is odd.]

***11.** Show that $\log_2 3$ is an irrational number. Recall that an irrational number is a real number x that cannot be written as the ratio of two integers.

12. Prove that for every positive integer n, there are n consecutive composite integers. [*Hint:* Consider the n consecutive integers starting with $(n + 1)! + 2$.]

***13.** Prove or disprove that there are three consecutive odd positive integers that are primes, that is, odd primes of the form p, $p + 2$, and $p + 4$.

14. Which positive integers less than 12 are relatively prime to 12?

15. Which positive integers less than 30 are relatively prime to 30?

16. Determine whether the integers in each of these sets are pairwise relatively prime.

a) 21, 34, 55 **b)** 14, 17, 85
c) 25, 41, 49, 64 **d)** 17, 18, 19, 23

17. Determine whether the integers in each of these sets are pairwise relatively prime.

a) 11, 15, 19 **b)** 14, 15, 21
c) 12, 17, 31, 37 **d)** 7, 8, 9, 11

18. We call a positive integer **perfect** if it equals the sum of its positive divisors other than itself.

a) Show that 6 and 28 are perfect.
b) Show that $2^{p-1}(2^p - 1)$ is a perfect number when $2^p - 1$ is prime.

19. Show that if $2^n - 1$ is prime, then n is prime. [*Hint:* Use the identity $2^{ab} - 1 = (2^a - 1) \cdot (2^{a(b-1)} + 2^{a(b-2)} + \cdots + 2^a + 1)$.]

20. Determine whether each of these integers is prime, verifying some of Mersenne's claims.

a) $2^7 - 1$ **b)** $2^9 - 1$
c) $2^{11} - 1$ **d)** $2^{13} - 1$

The value of the **Euler ϕ-function** at the positive integer n is defined to be the number of positive integers less than or equal to n that are relatively prime to n. For instance, $\phi(6) = 2$ because of the positive integers less or equal to 6, only 1 and 5 are relatively prime to 6. [*Note:* ϕ is the Greek letter phi.]

21. Find these values of the Euler ϕ-function.

a) $\phi(4)$ **b)** $\phi(10)$ **c)** $\phi(13)$

22. Show that n is prime if and only if $\phi(n) = n - 1$.

23. What is the value of $\phi(p^k)$ when p is prime and k is a positive integer?

24. What are the greatest common divisors of these pairs of integers?

a) $2^2 \cdot 3^3 \cdot 5^5$, $2^5 \cdot 3^3 \cdot 5^2$
b) $2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$, $2^{11} \cdot 3^9 \cdot 11 \cdot 17^{14}$
c) 17, 17^{17} **d)** $2^2 \cdot 7$, $5^3 \cdot 13$
e) 0, 5 **f)** $2 \cdot 3 \cdot 5 \cdot 7$, $2 \cdot 3 \cdot 5 \cdot 7$

25. What are the greatest common divisors of these pairs of integers?

a) $3^7 \cdot 5^3 \cdot 7^3$, $2^{11} \cdot 3^5 \cdot 5^9$
b) $11 \cdot 13 \cdot 17$, $2^9 \cdot 3^7 \cdot 5^5 \cdot 7^3$
c) 23^{31}, 23^{17}
d) $41 \cdot 43 \cdot 53$, $41 \cdot 43 \cdot 53$
e) $3^{13} \cdot 5^{17}$, $2^{12} \cdot 7^{21}$
f) 1111, 0

26. What is the least common multiple of each pair in Exercise 24?

27. What is the least common multiple of each pair in Exercise 25?

28. Find gcd(1000, 625) and lcm(1000, 625) and verify that gcd(1000, 625) · lcm(1000, 625) = 1000 · 625.

29. Find gcd(92928, 123552) and lcm(92928, 123552), and verify that gcd(92928, 123552) · lcm(92928, 123552) = 92928 · 123552. [*Hint:* First find the prime factorizations of 92928 and 123552.]

30. If the product of two integers is $2^7 3^8 5^2 7^{11}$ and their greatest common divisor is $2^3 3^4 5$, what is their least common multiple?

31. Show that if a and b are positive integers, then $ab = \gcd(a, b) \cdot \text{lcm}(a, b)$. [*Hint:* Use the prime factorizations of a and b and the formulae for $\gcd(a, b)$ and $\text{lcm}(a, b)$ in terms of these factorizations.]

32. Use the Euclidean algorithm to find

a) gcd(1, 5). **b)** gcd(100, 101).
c) gcd(123, 277). **d)** gcd(1529, 14039).
e) gcd(1529, 14038). **f)** gcd(11111, 111111).

33. Use the Euclidean algorithm to find

a) gcd(12, 18). **b)** gcd(111, 201).
c) gcd(1001, 1331). **d)** gcd(12345, 54321).
e) gcd(1000, 5040). **f)** gcd(9888, 6060).

34. How many divisions are required to find gcd(21, 34) using the Euclidean algorithm?

35. How many divisions are required to find gcd(34, 55) using the Euclidean algorithm?

***36.** Show that if a and b are both positive integers, then $(2^a - 1) \bmod (2^b - 1) = 2^{a \bmod b} - 1$.

☞ ***37.** Use Exercise 36 to show that if a and b are positive integers, then $\gcd(2^a - 1, 2^b - 1) = 2^{\gcd(a, b)} - 1$. [*Hint:* Show that the remainders obtained when the Euclidean algorithm is used to compute $\gcd(2^a - 1, 2^b - 1)$ are of the form $2^r - 1$, where r is a remainder arising when the Euclidean algorithm is used to find $\gcd(a, b)$.]

38. Use Exercise 37 to show that the integers $2^{35} - 1$, $2^{34} - 1$, $2^{33} - 1$, $2^{31} - 1$, $2^{29} - 1$, and $2^{23} - 1$ are pairwise relatively prime.

39. Using the method followed in Example 17, express the greatest common divisor of each of these pairs of integers as a linear combination of these integers.

a) 10, 11 **b)** 21, 44 **c)** 36, 48
d) 34, 55 **e)** 117, 213 **f)** 0, 223
g) 123, 2347 **h)** 3454, 4666 **i)** 9999, 11111

40. Using the method followed in Example 17, express the greatest common divisor of each of these pairs of integers as a linear combination of these integers.

a) 9, 11 **b)** 33, 44 **c)** 35, 78
d) 21, 55 **e)** 101, 203 **f)** 124, 323
g) 2002, 2339 **h)** 3457, 4669 **i)** 10001, 13422

41. Use the extended Euclidean algorithm to express gcd(26, 91) as a linear combination of 26 and 91.

42. Use the extended Euclidean algorithm to express gcd(252, 356) as a linear combination of 252 and 356.

43. Use the extended Euclidean algorithm to express gcd(144, 89) as a linear combination of 144 and 89.

44. Use the extended Euclidean algorithm to express gcd(1001, 100001) as a linear combination of 1001 and 100001.

45. Describe the extended Euclidean algorithm using pseudocode.

46. Find the smallest positive integer with exactly n different positive factors when n is

a) 3. **b)** 4. **c)** 5.
d) 6. **e)** 10.

47. Can you find a formula or rule for the nth term of a sequence related to the prime numbers or prime factorizations so that the initial terms of the sequence have these values?

a) 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, …
b) 1, 2, 3, 2, 5, 2, 7, 2, 3, 2, 11, 2, 13, 2, …
c) 1, 2, 2, 3, 2, 4, 2, 4, 3, 4, 2, 6, 2, 4, …
d) 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, …
e) 1, 2, 3, 3, 5, 5, 7, 7, 7, 7, 11, 11, 13, 13, …
f) 1, 2, 6, 30, 210, 2310, 30030, 510510, 9699690, 223092870, …

48. Can you find a formula or rule for the nth term of a sequence related to the prime numbers or prime factorizations so that the initial terms of the sequence have these values?

a) 2, 2, 3, 5, 5, 7, 7, 11, 11, 11, 11, 13, 13, …
b) 0, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, …
c) 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, …
d) 1, −1, −1, 0, −1, 1, −1, 0, 0, 1, −1, 0, −1, 1, 1, …
e) 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, …
f) 4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, …

49. Prove that the product of any three consecutive integers is divisible by 6.

50. Show that if a, b, and m are integers such that $m \geq 2$ and $a \equiv b \pmod{m}$, then $\gcd(a, m) = \gcd(b, m)$.

***51.** Prove or disprove that $n^2 - 79n + 1601$ is prime whenever n is a positive integer.

52. Prove or disprove that $p_1 p_2 \cdots p_n + 1$ is prime for every positive integer n, where $p_1, p_2, \ldots, p_n$ are the n smallest prime numbers.

53. Show that there is a composite integer in every arithmetic progression $ak + b$, $k = 1, 2, \ldots$, where a and b are positive integers.

54. Adapt the proof in the text that there are infinitely many primes to prove that there are infinitely many primes of the form $3k + 2$, where k is a nonnegative integer. [*Hint:* Suppose that there are only finitely many such primes $q_1, q_2, \ldots, q_n$, and consider the number $3q_1 q_2 \cdots q_n - 1$.]

55. Adapt the proof in the text that there are infinitely many primes to prove that there are infinitely many primes of the form $4k + 3$, where k is a nonnegative integer. [*Hint:* Suppose that there are only finitely many such primes $q_1, q_2, \ldots, q_n$, and consider the number $4q_1 q_2 \cdots q_n - 1$.]

***56.** Prove that the set of positive rational numbers is countable by setting up a function that assigns to a rational number p/q with $\gcd(p, q) = 1$ the base 11 number formed by the decimal representation of p followed by the base 11 digit A, which corresponds to the decimal number 10, followed by the decimal representation of q.

***57.** Prove that the set of positive rational numbers is countable by showing that the function K is a one-to-one correspondence between the set of positive rational numbers and the set of positive integers if $K(m/n) = p_1^{2a_1} p_2^{2a_2} \cdot \cdots \cdot p_s^{2a_s} q_1^{2b_1 - 1} q_2^{2b_2 - 1} \cdot \cdots \cdot q_t^{2b_t - 1}$, where $\gcd(m, n) = 1$ and the prime-power factorizations of m and n are $m = p_1^{a_1} p_2^{a_2} \cdot \cdots \cdot p_s^{a_s}$ and $n = q_1^{b_1} q_2^{b_2} \cdots q_t^{b_t}$.

4.4 Solving Congruences

4.4.1 Introduction

Solving linear congruences, which have the form $ax \equiv b \pmod{m}$, is an essential task in the study of number theory and its applications, just as solving linear equations plays an important role in calculus and linear algebra. To solve linear congruences, we employ inverses modulo m. We explain how to work backwards through the steps of the Euclidean algorithm to find inverses modulo m. Once we have found an inverse of a modulo m, we solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides of the congruence by this inverse.

Simultaneous systems of linear congruence have been studied since ancient times. For example, the Chinese mathematician Sun-Tsu studied them in the first century. We will show how to solve systems of linear congruences modulo pairwise relatively prime moduli. The result we will prove is called the Chinese remainder theorem, and our proof will give a method to find all solutions of such systems of congruences. We will also show how to use the Chinese remainder theorem as a basis for performing arithmetic with large integers.

We will introduce a useful result of Fermat, known as Fermat's little theorem, which states that if p is prime and p does not divide a, then $a^{p-1} \equiv 1 \pmod{p}$. We will examine the converse of this statement, which will lead us to the concept of a pseudoprime. A pseudoprime m to the base a is a composite integer m that masquerades as a prime by satisfying the congruence $a^{m-1} \equiv 1 \pmod{m}$. We will also give an example of a Carmichael number, which is a composite integer that is a pseudoprime to all bases a relatively prime to it.

We also introduce the notion of discrete logarithms, which are analogous to ordinary logarithms. To define discrete logarithms we must first define primitive roots. A primitive root of a prime p is an integer r such that every integer not divisible by p is congruent to a power of r modulo p. If r is a primitive root of p and $r^e \equiv a \pmod{p}$, then e is the discrete logarithm of a modulo p to the base r. Finding discrete logarithms turns out to be an extremely difficult problem in general. The difficulty of this problem is the basis for the security of many cryptographic systems.

4.4.2 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 a 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, how can we find all integers x that satisfy this congruence? One method that we will describe uses an integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$, if such an integer exists. Such an integer $\overline{a}$ is said to be an **inverse** of a modulo m. Theorem 1 guarantees that an inverse of a modulo m exists whenever a and m are relatively prime.

THEOREM 1 If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m. (That is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to $\overline{a}$ modulo m.)

Proof: By Theorem 6 of Section 4.3, 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. That this inverse is unique modulo m is left as Exercise 7. ◁

Using inspection to find an inverse of a modulo m is easy when m is small. To find this inverse, we look for a multiple of a that exceeds a multiple of m by 1. For example, to find an inverse of 3 modulo 7, we can find $j \cdot 3$ for $j = 1, 2, \ldots, 6$, stopping when we find a multiple of 3 that is one more than a multiple of 7. We can speed this approach up if we note that $2 \cdot 3 \equiv -1 \pmod 7$. This means that $(-2) \cdot 3 \equiv 1 \pmod 7$. Hence, $5 \cdot 3 \equiv 1 \pmod 7$, so 5 is an inverse of 3 modulo 7.

We can design a more efficient algorithm than brute force to find an inverse of a modulo m when $\gcd(a, m) = 1$ using the steps of the Euclidean algorithm. By reversing these steps as in Example 17 of Section 4.3, we can find a linear combination $sa + tm = 1$, where s and t are integers. Reducing both sides of this equation modulo m tells us that s is an inverse of a modulo m. We illustrate this procedure in Example 1.

EXAMPLE 1 Find an inverse of 3 modulo 7 by first finding Bézout coefficients of 3 and 7. (Note that we have already shown that 5 is an inverse of 3 modulo 7 by inspection.)

Solution: Because $\gcd(3, 7) = 1$, Theorem 1 tells us that an inverse of 3 modulo 7 exists. The Euclidean algorithm ends quickly when used to find the 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 and 1 are Bézout coefficients of 3 and 7. We see 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 2 Find an inverse of 101 modulo 4620.

Solution: For completeness, we present all steps used to compute an inverse of 101 modulo 4620. (Only the last step goes beyond methods developed in Section 4.3 and illustrated in Example 17 in that section.) First, we use the Euclidean algorithm to show that $\gcd(101, 4620) = 1$. Then we will reverse the steps to find Bézout coefficients a and b such that $101a + 4620b = 1$. It will then follow that a is an inverse of 101 modulo 4620. The steps used by the Euclidean algorithm to find $\gcd(101, 4620)$ are

$$4620 = 45 \cdot 101 + 75$$
$$101 = 1 \cdot 75 + 26$$
$$75 = 2 \cdot 26 + 23$$
$$26 = 1 \cdot 23 + 3$$
$$23 = 7 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$
$$2 = 2 \cdot 1.$$

Because the last nonzero remainder is 1, we know that $\gcd(101, 4620) = 1$. We can now find the Bézout coefficients for 101 and 4620 by working backwards through these steps, expressing

$\gcd(101, 4620) = 1$ in terms of each successive pair of remainders. In each step we eliminate the remainder by expressing it as a linear combination of the divisor and the dividend. We obtain

$$\begin{aligned}
1 &= 3 - 1 \cdot 2 \\
&= 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3 \\
&= -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23 \\
&= 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = -9 \cdot 75 + 26 \cdot 26 \\
&= -9 \cdot 75 + 26 \cdot (101 - 1 \cdot 75) = 26 \cdot 101 - 35 \cdot 75 \\
&= 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101) = -35 \cdot 4620 + 1601 \cdot 101.
\end{aligned}$$

That $-35 \cdot 4620 + 1601 \cdot 101 = 1$ tells us that -35 and 1601 are Bézout coefficients of 4620 and 101, and 1601 is an inverse of 101 modulo 4620. ◀

Once we have an inverse $\overline{a}$ of a modulo m, we can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides of the linear congruence by $\overline{a}$, as Example 3 illustrates.

EXAMPLE 3 What are the solutions of the linear congruence $3x \equiv 4 \pmod{7}$?

Solution: By Example 1 we know that -2 is an inverse of 3 modulo 7. Multiplying both sides of the congruence by -2 shows that

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}.$$

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, by Theorem 5 of Section 4.1, it follows that

$$3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7},$$

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, $6, 13, 20, \ldots$ and $-1, -8, -15, \ldots$. ◀

4.4.3 The Chinese Remainder Theorem

Systems of linear congruences arise in many contexts. For example, as we will see later, they are the basis for a method that can be used to perform arithmetic with large integers. Such systems can even be found as word puzzles in the writings of ancient Chinese and Hindu mathematicians, such as that given in Example 4.

Links

EXAMPLE 4 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; and when divided by 7, the remainder is 2. What will be the number of things?

This puzzle can be translated into the following question: What are the solutions of the systems of congruences

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

We will solve this system, and with it Sun-Tsu's puzzle, later in this section. ◀

The *Chinese remainder theorem*, named after the Chinese heritage of problems involving systems of linear congruences, states that when the moduli of a system of linear congruences are pairwise relatively prime, there is a unique solution of the system modulo the product of the moduli.

THEOREM 2

THE CHINESE REMAINDER THEOREM Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers greater than one 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: To establish this theorem, we need to show that a solution exists and that it is unique modulo m. We will show that a solution exists by describing a way to construct this solution; showing that the solution is unique modulo m is Exercise 30.

To construct a simultaneous solution, first let

$$M_k = m/m_k$$

for $k = 1, 2, \ldots, n$. That is, M_k is the product of the moduli except for m_k. Because m_i and m_k have no common factors greater than 1 when $i \neq k$, it follows that $\gcd(m_k, M_k) = 1$. Consequently, by Theorem 1, we know that there is an integer y_k, an inverse of M_k modulo m_k, such that

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

To construct a simultaneous solution, form the sum

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

We will now show that x is a simultaneous solution. First, 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

$$x \equiv a_k M_k y_k \equiv a_k \pmod{m_k},$$

for $k = 1, 2, \ldots, n$. We have shown that x is a simultaneous solution to the n congruences. ◁

Example 5 illustrates how to use the construction given in our proof of the Chinese remainder theorem to solve a system of congruences. We will solve the system given in Example 4, which arises in Sun-Tsu's puzzle.

EXAMPLE 5 To solve the system of congruences in Example 4, first let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$, and $M_3 = m/7 = 15$. We see that 2 is an inverse of $M_1 = 35$ modulo 3, because $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod 3$; 1 is an inverse of $M_2 = 21$ modulo 5, because $21 \equiv 1 \pmod 5$; and 1 is an inverse of $M_3 = 15 \pmod 7$, because $15 \equiv 1 \pmod 7$. The solutions to this system are those x such that

$$\begin{aligned} x \equiv a_1M_1y_1 + a_2M_2y_2 + a_3M_3y_3 &= 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \\ &= 233 \equiv 23 \pmod{105}. \end{aligned}$$

It follows that 23 is the smallest positive integer that is a simultaneous solution. We conclude that 23 is the smallest positive integer that leaves a remainder of 2 when divided by 3, a remainder of 3 when divided by 5, and a remainder of 2 when divided by 7. ◀

Although the construction in Theorem 2 provides a general method for solving systems of linear congruences with pairwise relatively prime moduli, it can be easier to solve a system using a different method. Example 6 illustrates the use of a method known as **back substitution**.

EXAMPLE 6 Use the method of back substitution to find all integers x such that $x \equiv 1 \pmod 5$, $x \equiv 2 \pmod 6$, and $x \equiv 3 \pmod 7$.

Solution: By Theorem 4 in Section 4.1, the first congruence can be rewritten as an equality, $x = 5t + 1$, where t is an integer. Substituting this expression for x into the second congruence tells us that

$$5t + 1 \equiv 2 \pmod 6,$$

which can be solved to show that $t \equiv 5 \pmod 6$ (as the reader should verify). Using Theorem 4 in Section 4.1 again, we see that $t = 6u + 5$, where u is an integer. Substituting this expression for t back into the equation $x = 5t + 1$ tells us that $x = 5(6u + 5) + 1 = 30u + 26$. We insert this into the third equation to obtain

$$30u + 26 \equiv 3 \pmod 7.$$

Solving this congruence tells us that $u \equiv 6 \pmod 7$ (as the reader should verify). Hence, Theorem 4 in Section 4.1 tells us that $u = 7v + 6$, where v is an integer. Substituting this expression for u into the equation $x = 30u + 26$ tells us that $x = 30(7v + 6) + 26 = 210u + 206$. Translating this back into a congruence, we find the solution to the simultaneous congruences,

$$x \equiv 206 \pmod{210}.$$ ◀

4.4.4 Computer Arithmetic with Large Integers

Suppose that $m_1, m_2, \ldots, m_n$ are pairwise relatively prime moduli and let m be their product. By the Chinese remainder theorem, we can show (see Exercise 28) that an integer a with $0 \leq a < m$ can be uniquely represented by the n-tuple consisting of its remainders upon division by m_i, $i = 1, 2, \ldots, n$. That is, we can uniquely represent a by

$$(a \bmod m_1, a \bmod m_2, \ldots, a \bmod m_n).$$

EXAMPLE 7 What are the pairs used to represent the nonnegative integers less than 12 when they are represented by the ordered pair where the first component is the remainder of the integer upon division by 3 and the second component is the remainder of the integer upon division by 4?

Solution: We have the following representations, obtained by finding the remainder of each integer when it is divided by 3 and by 4:

$$\begin{array}{lll} 0 = (0, 0) & 4 = (1, 0) & 8 = (2, 0) \\ 1 = (1, 1) & 5 = (2, 1) & 9 = (0, 1) \\ 2 = (2, 2) & 6 = (0, 2) & 10 = (1, 2) \\ 3 = (0, 3) & 7 = (1, 3) & 11 = (2, 3). \end{array}$$

◀

To perform arithmetic with large integers, we select moduli $m_1, m_2, \ldots, m_n$, where each m_i is an integer greater than 2, $\gcd(m_i, m_j) = 1$ whenever $i \neq j$, and $m = m_1 m_2 \cdots m_n$ is greater than the results of the arithmetic operations we want to carry out.

Once we have selected our moduli, we carry out arithmetic operations with large integers by performing componentwise operations on the n-tuples representing these integers using their remainders upon division by m_i, $i = 1, 2, \ldots, n$. Once we have computed the value of each component in the result, we recover its value by solving a system of n congruences modulo m_i, $i = 1, 2, \ldots, n$. This method of performing arithmetic with large integers has several valuable features. First, it can be used to perform arithmetic with integers larger than can ordinarily be carried out on a computer. Second, computations with respect to the different moduli can be done in parallel, speeding up the arithmetic.

EXAMPLE 8 Suppose that performing arithmetic with integers less than 100 on a certain processor is much quicker than doing arithmetic with larger integers. We can restrict almost all our computations to integers less than 100 if we represent integers using their remainders modulo pairwise relatively prime integers less than 100. For example, we can use the moduli of 99, 98, 97, and 95. (These integers are relatively prime pairwise, because no two have a common factor greater than 1.)

By the Chinese remainder theorem, every nonnegative integer less than $99 \cdot 98 \cdot 97 \cdot 95 = 89{,}403{,}930$ can be represented uniquely by its remainders when divided by these four moduli. For example, we represent 123,684 as (33, 8, 9, 89), because 123,684 **mod** 99 = 33; 123,684 **mod** 98 = 8; 123,684 **mod** 97 = 9; and 123,684 **mod** 95 = 89. Similarly, we represent 413,456 as (32, 92, 42, 16).

To find the sum of 123,684 and 413,456, we work with these 4-tuples instead of these two integers directly. We add the 4-tuples componentwise and reduce each component with respect to the appropriate modulus. This yields

$$\begin{aligned} &(33, 8, 9, 89) + (32, 92, 42, 16) \\ &\qquad = (65 \textbf{ mod } 99, 100 \textbf{ mod } 98, 51 \textbf{ mod } 97, 105 \textbf{ mod } 95) \\ &\qquad = (65, 2, 51, 10). \end{aligned}$$

To find the sum, that is, the integer represented by (65, 2, 51, 10), we need to solve the system of congruences

$$\begin{aligned} x &\equiv 65 \pmod{99}, \\ x &\equiv \ \ 2 \pmod{98}, \\ x &\equiv 51 \pmod{97}, \\ x &\equiv 10 \pmod{95}. \end{aligned}$$

It can be shown (see Exercise 53) that 537,140 is the unique nonnegative solution of this system less than 89,403,930. Consequently, 537,140 is the sum. Note that it is only when we

have to recover the integer represented by (65, 2, 51, 10) that we have to do arithmetic with integers larger than 100. ◀

Particularly good choices for moduli for arithmetic with large integers are sets of integers of the form $2^k - 1$, where k is a positive integer, because it is easy to do binary arithmetic modulo such integers, and because it is easy to find sets of such integers that are pairwise relatively prime. [The second reason is a consequence of the fact that $\gcd(2^a - 1, 2^b - 1) = 2^{\gcd(a, b)} - 1$, as Exercise 37 in Section 4.3 shows.] Suppose, for instance, that we can do arithmetic with integers less than 2^{35} easily on our computer, but that working with larger integers requires special procedures. We can use pairwise relatively prime moduli less than 2^{35} to perform arithmetic with integers as large as their product. For example, as Exercise 38 in Section 4.3 shows, the integers $2^{35} - 1$, $2^{34} - 1$, $2^{33} - 1$, $2^{31} - 1$, $2^{29} - 1$, and $2^{23} - 1$ are pairwise relatively prime. Because the product of these six moduli exceeds 2^{184}, we can perform arithmetic with integers as large as 2^{184} (as long as the results do not exceed this number) by doing arithmetic modulo each of these six moduli, none of which exceeds 2^{35}.

4.4.5 Fermat's Little Theorem

The French mathematician Pierre de Fermat, one of the leading mathematicians of the first half of the 17th century, made many important discoveries in number theory. One of the most useful of these states that p divides $a^{p-1} - 1$ whenever p is prime and a is an integer not divisible by p. Fermat announced this result in a letter to one of his correspondents. However, he did not include a proof in the letter, stating that he feared the proof would be too long. Although Fermat never published a proof of this fact, there is little doubt that he knew how to prove it, unlike the result known as Fermat's last theorem. The first published proof is credited to Leonhard Euler. We now state this theorem in terms of congruences.

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

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

Remark: Fermat's little theorem tells us that if $a \in \mathbf{Z}_p$, then $a^{p-1} = 1$ in $\mathbf{Z}_p$.

The proof of Theorem 3 is outlined in Exercise 19.

Fermat's little theorem is extremely useful in computing the remainders modulo p of large powers of integers, as Example 9 illustrates.

EXAMPLE 9 Find 7^{222} **mod** 11.

Solution: We can use Fermat's little theorem to evaluate 7^{222} **mod** 11 rather than using the fast modular exponentiation algorithm. By Fermat's little theorem we know that $7^{10} \equiv 1 \pmod{11}$, so $(7^{10})^k \equiv 1 \pmod{11}$ for every positive integer k. To take advantage of this last congruence, we divide the exponent 222 by 10, finding that $222 = 22 \cdot 10 + 2$. We now see that

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

It follows that 7^{222} **mod** $11 = 5$. ◀

Example 9 illustrated how we can use Fermat's little theorem to compute a^n **mod** p, where p is prime and $p \nmid a$. First, we use the division algorithm to find the quotient q and remainder r when n is divided by $p - 1$, so that $n = q(p - 1) + r$, where $0 \leq r < p - 1$. It follows that $a^n = a^{q(p-1)+r} = (a^{p-1})^q a^r \equiv 1^q a^r \equiv a^r \pmod{p}$. Hence, to find a^n **mod** p, we only need to compute a^r **mod** p. We will take advantage of this simplification many times in our study of number theory.

4.4.6 Pseudoprimes

In Section 4.2 we showed that an integer n is prime when it is not divisible by any prime p with $p \leq \sqrt{n}$. Unfortunately, using this criterion to show that a given integer is prime is inefficient. It requires that we find all primes not exceeding $\sqrt{n}$ and that we carry out trial division by each such prime to see whether it divides n.

Are there more efficient ways to determine whether an integer is prime? According to some sources, ancient Chinese mathematicians believed that n was an odd prime if and only if

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

If this were true, it would provide an efficient primality test. Why did they believe this congruence could be used to determine whether an integer $n > 2$ is prime? First, they observed that the congruence holds whenever n is an odd prime. For example, 5 is prime and

$$2^{5-1} = 2^4 = 16 \equiv 1 \pmod{5}.$$

By Fermat's little theorem, we know that this observation was correct, that is, $2^{n-1} \equiv 1 \pmod{n}$ whenever n is an odd prime. Second, they never found a composite integer n for which the congruence holds. However, the ancient Chinese were only partially correct. They were correct in thinking that the congruence holds whenever n is prime, but they were incorrect in concluding that n is necessarily prime if the congruence holds.

Unfortunately, there are composite integers n such that $2^{n-1} \equiv 1 \pmod{n}$. Such integers are called **pseudoprimes** to the base 2.

EXAMPLE 10 The integer 341 is a pseudoprime to the base 2 because it is composite ($341 = 11 \cdot 31$) and as Exercise 37 shows

$$2^{340} \equiv 1 \pmod{341}.$$ ◀

We can use an integer other than 2 as the base when we study pseudoprimes.

Links

©PHOTOS.com/Getty Images

PIERRE DE FERMAT (1601–1665) Pierre de Fermat, one of the most important mathematicians of the seventeenth century, was a lawyer by profession. He is the most famous amateur mathematician in history. Fermat published little of his mathematical discoveries. It is through his correspondence with other mathematicians that we know of his work. Fermat was one of the inventors of analytic geometry and developed some of the fundamental ideas of calculus. Fermat, along with Pascal, gave probability theory a mathematical basis. Fermat formulated what was the most famous unsolved problem in mathematics. He asserted that the equation $x^n + y^n = z^n$ has no nontrivial positive integer solutions when n is an integer greater than 2. For more than 300 years, no proof (or counterexample) was found. In his copy of the works of the ancient Greek mathematician Diophantus, Fermat wrote that he had a proof but that it would not fit in the margin. Because the first proof, found by Andrew Wiles in 1994, relies on sophisticated, modern mathematics, most people think that Fermat thought he had a proof, but that the proof was incorrect. However, he may have been tempting others to look for a proof, not being able to find one himself.

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*.

Given a positive integer n, determining whether $2^{n-1} \equiv 1 \pmod{n}$ is a useful test that provides some evidence concerning whether n is prime. In particular, if n satisfies this congruence, then it is either prime or a pseudoprime to the base 2; if n does not satisfy this congruence, it is composite. We can perform similar tests using bases b other than 2 and obtain more evidence as to whether n is prime. If n passes all such tests, it is either prime or a pseudoprime to all the bases b we have chosen. Furthermore, among the positive integers not exceeding x, where x is a positive real number, compared to primes there are relatively few pseudoprimes to the base b, where b is a positive integer. 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. Unfortunately, we cannot distinguish between primes and pseudoprimes just by choosing sufficiently many bases, because there are composite integers n that pass all tests with bases b such that $\gcd(b, n) = 1$. This leads to Definition 2.

Definition 2 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*. (These numbers are named after Robert Carmichael, who studied them in the early twentieth century.)

EXAMPLE 11 The integer 561 is a Carmichael number. To see this, first note that 561 is composite because $561 = 3 \cdot 11 \cdot 17$. 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

$$b^{560} = (b^2)^{280} \equiv 1 \pmod{3},$$
$$b^{560} = (b^{10})^{56} \equiv 1 \pmod{11},$$
$$b^{560} = (b^{16})^{35} \equiv 1 \pmod{17}.$$

By Exercise 29, 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. ◀

Although there are infinitely many Carmichael numbers, more delicate tests, described in the exercise set, can be devised that can be used as the basis for efficient probabilistic primality tests. Such tests can be used to quickly show that it is almost certainly the case that a given

Links

ROBERT DANIEL CARMICHAEL (1879–1967) Robert Daniel Carmichael was born in Alabama. He received his undergraduate degree from Lineville College in 1898 and his Ph.D. in 1911 from Princeton. Carmichael held positions at Indiana University from 1911 until 1915 and at the University of Illinois from 1915 until 1947. Carmichael was an active researcher in a wide variety of areas, including number theory, real analysis, differential equations, mathematical physics, and group theory. His Ph.D. thesis, written under the direction of G. D. Birkhoff, is considered the first significant American contribution to the subject of differential equations.

integer is prime. More precisely, if an integer is not prime, then the probability that it passes a series of tests is close to 0. We will describe such a test in Chapter 7 and discuss the notions from probability theory that this test relies on. These probabilistic primality tests can be used, and are used, to find large primes extremely rapidly on computers.

4.4.7 Primitive Roots and Discrete Logarithms

In the set of positive real numbers, if $b > 1$, and $x = b^y$, we say that y is the logarithm of x to the base b. Here, we will show that we can also define the concept of logarithms modulo p of positive integers, where p is a prime. Before we do so, we need a definition.

Definition 3

A *primitive root* modulo a prime p is an integer r in $\mathbf{Z}_p$ such that every nonzero element of $\mathbf{Z}_p$ is a power of r.

EXAMPLE 12 Determine whether 2 and 3 are primitive roots modulo 11.

Solution: When we compute the powers of 2 in $\mathbf{Z}_{11}$, we obtain $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 5$, $2^5 = 10$, $2^6 = 9$, $2^7 = 7$, $2^8 = 3$, $2^9 = 6$, $2^{10} = 1$. Because every nonzero element of $\mathbf{Z}_{11}$ is a power of 2, 2 is a primitive root of 11.

When we compute the powers of 3 modulo 11, we obtain $3^1 = 3$, $3^2 = 9$, $3^3 = 5$, $3^4 = 4$, $3^5 = 1$. This pattern repeats when we compute higher powers of 3. Because not all nonzero elements of $\mathbf{Z}_{11}$ are powers of 3, we conclude that 3 is not a primitive root of 11. ◀

An important fact in number theory is that there is a primitive root modulo p for every prime p. We refer the reader to [Ro10] for a proof of this fact. Suppose that p is prime and r is a primitive root modulo p. If a is an integer between 1 and $p - 1$, that is, a nonzero element of $\mathbf{Z}_p$, we know that there is an unique exponent e such that $r^e = a$ in $\mathbf{Z}_p$, that is, $r^e \bmod p = a$.

Definition 4

Suppose that p is a prime, r is a primitive root modulo p, and a is an integer between 1 and $p - 1$ inclusive. If $r^e \bmod p = a$ and $0 \leq e \leq p - 1$, 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 13 Find the discrete logarithms of 3 and 5 modulo 11 to the base 2.

Solution: When we computed the powers of 2 modulo 11 in Example 12, we found that $2^8 = 3$ and $2^4 = 5$ in $\mathbf{Z}_{11}$. Hence, the discrete logarithms of 3 and 5 modulo 11 to the base 2 are 8 and 4, respectively. (These are the powers of 2 that equal 3 and 5, respectively, in $\mathbf{Z}_{11}$.) We write $\log_2 3 = 8$ and $\log_2 5 = 4$ (where the modulus 11 is understood and not explicitly noted in the notation). ◀

The discrete logarithm problem is hard!

The **discrete logarithm problem** takes as input a prime p, a primitive root r modulo p, and a positive integer $a \in \mathbf{Z}_p$; its output is the discrete logarithm of a modulo p to the base r. Although this problem might seem not to be that difficult, it turns out that no polynomial time algorithm is known for solving it. The difficulty of this problem plays an important role in cryptography, as we will see in Section 4.6.

Exercises

1. Show that 15 is an inverse of 7 modulo 26.

☞ **2.** Show that 937 is an inverse of 13 modulo 2436.

3. By inspection (as discussed prior to Example 1), find an inverse of 4 modulo 9.

4. By inspection (as discussed prior to Example 1), find an inverse of 2 modulo 17.

5. Find an inverse of a modulo m for each of these pairs of relatively prime integers using the method followed in Example 2.

a) $a = 4, m = 9$
b) $a = 19, m = 141$
c) $a = 55, m = 89$
d) $a = 89, m = 232$

6. Find an inverse of a modulo m for each of these pairs of relatively prime integers using the method followed in Example 2.

a) $a = 2, m = 17$
b) $a = 34, m = 89$
c) $a = 144, m = 233$
d) $a = 200, m = 1001$

***7.** Show that if a and m are relatively prime positive integers, then the inverse of a modulo m is unique modulo m. [*Hint:* Assume that there are two solutions b and c of the congruence $ax \equiv 1 \pmod{m}$. Use Theorem 7 of Section 4.3 to show that $b \equiv c \pmod{m}$.]

8. Show that an inverse of a modulo m, where a is an integer and $m > 2$ is a positive integer, does not exist if $\gcd(a, m) > 1$.

9. Solve the congruence $4x \equiv 5 \pmod{9}$ using the inverse of 4 modulo 9 found in part (a) of Exercise 5.

10. Solve the congruence $2x \equiv 7 \pmod{17}$ using the inverse of 2 modulo 17 found in part (a) of Exercise 6.

11. Solve each of these congruences using the modular inverses found in parts (b), (c), and (d) of Exercise 5.

a) $19x \equiv 4 \pmod{141}$
b) $55x \equiv 34 \pmod{89}$
c) $89x \equiv 2 \pmod{232}$

12. Solve each of these congruences using the modular inverses found in parts (b), (c), and (d) of Exercise 6.

a) $34x \equiv 77 \pmod{89}$
b) $144x \equiv 4 \pmod{233}$
c) $200x \equiv 13 \pmod{1001}$

13. Find the solutions of the congruence $15x^2 + 19x \equiv 5 \pmod{11}$. [*Hint:* Show the congruence is equivalent to the congruence $15x^2 + 19x + 6 \equiv 0 \pmod{11}$. Factor the left-hand side of the congruence; show that a solution of the quadratic congruence is a solution of one of the two different linear congruences.]

14. Find the solutions of the congruence $12x^2 + 25x \equiv 10 \pmod{11}$. [*Hint:* Show the congruence is equivalence to the congruence $12x^2 + 25x + 12 \equiv 0 \pmod{11}$. Factor the left-hand side of the congruence; show that a solution of the quadratic congruence is a solution of one of two different linear congruences.]

***15.** Show that if m is an integer greater than 1 and $ac \equiv bc \pmod{m}$, then $a \equiv b \pmod{m/\gcd(c, m)}$.

16. a) Show that the positive integers less than 11, except 1 and 10, can be split into pairs of integers such that each pair consists of integers that are inverses of each other modulo 11.

b) Use part (a) to show that $10! \equiv -1 \pmod{11}$.

17. Show that if p is prime, the only solutions of $x^2 \equiv 1 \pmod{p}$ are integers x such that $x \equiv 1 \pmod{p}$ or $x \equiv -1 \pmod{p}$.

***18. a)** Generalize the result in part (a) of Exercise 16; that is, show that if p is a prime, the positive integers less than p, except 1 and $p - 1$, can be split into $(p - 3)/2$ pairs of integers such that each pair consists of integers that are inverses of each other. [*Hint:* Use the result of Exercise 17.]

b) From part (a) conclude that $(p - 1)! \equiv -1 \pmod{p}$ whenever p is prime. This result is known as **Wilson's theorem**.

c) What can we conclude if n is a positive integer such that $(n - 1)! \not\equiv -1 \pmod{n}$?

***19.** This exercise outlines a proof of Fermat's little theorem.

a) Suppose that a is not divisible by the prime p. Show that no two of the integers $1 \cdot a, 2 \cdot a, \ldots, (p - 1)a$ are congruent modulo p.

b) Conclude from part (a) that the product of $1, 2, \ldots, p - 1$ is congruent modulo p to the product of $a, 2a, \ldots, (p - 1)a$. Use this to show that

$$(p - 1)! \equiv a^{p-1}(p - 1)! \pmod{p}.$$

c) Use Theorem 7 of Section 4.3 to show from part (b) that $a^{p-1} \equiv 1 \pmod{p}$ if $p \nmid a$. [*Hint:* Use Lemma 3 of Section 4.3 to show that p does not divide $(p - 1)!$ and then use Theorem 7 of Section 4.3. Alternatively, use Wilson's theorem from Exercise 18(b).]

d) Use part (c) to show that $a^p \equiv a \pmod{p}$ for all integers a.

20. Use the construction in the proof of the Chinese remainder theorem to find all solutions to the system of congruences $x \equiv 2 \pmod{3}$, $x \equiv 1 \pmod{4}$, and $x \equiv 3 \pmod{5}$.

21. Use the construction in the proof of the Chinese remainder theorem to find all solutions to the system of congruences $x \equiv 1 \pmod{2}$, $x \equiv 2 \pmod{3}$, $x \equiv 3 \pmod{5}$, and $x \equiv 4 \pmod{11}$.

22. Solve the system of congruence $x \equiv 3 \pmod{6}$ and $x \equiv 4 \pmod{7}$ using the method of back substitution.

23. Solve the system of congruences in Exercise 20 using the method of back substitution.

24. Solve the system of congruences in Exercise 21 using the method of back substitution.

25. Write out in pseudocode an algorithm for solving a simultaneous system of linear congruences based on the construction in the proof of the Chinese remainder theorem.

***26.** Find all solutions, if any, to the system of congruences $x \equiv 5 \pmod 6$, $x \equiv 3 \pmod{10}$, and $x \equiv 8 \pmod{15}$.

***27.** Find all solutions, if any, to the system of congruences $x \equiv 7 \pmod 9$, $x \equiv 4 \pmod{12}$, and $x \equiv 16 \pmod{21}$.

28. Use the Chinese remainder theorem to show that an integer a, with $0 \leq a < m = m_1 m_2 \cdots m_n$, where the positive integers $m_1, m_2, \ldots, m_n$ are pairwise relatively prime, can be represented uniquely by the n-tuple (a **mod** m_1, a **mod** $m_2, \ldots, a$ **mod** m_n).

***29.** Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime integers greater than or equal to 2. Show that if $a \equiv b \pmod{m_i}$ for $i = 1, 2, \ldots, n$, then $a \equiv b \pmod m$, where $m = m_1 m_2 \cdots m_n$. (This result will be used in Exercise 30 to prove the Chinese remainder theorem. Consequently, do not use the Chinese remainder theorem to prove it.)

***30.** Complete the proof of the Chinese remainder theorem by showing that the simultaneous solution of a system of linear congruences modulo pairwise relatively prime moduli is unique modulo the product of these moduli. [*Hint:* Assume that x and y are two simultaneous solutions. Show that $m_i \mid x - y$ for all i. Using Exercise 29, conclude that $m = m_1 m_2 \cdots m_n \mid x - y$.]

31. Which integers leave a remainder of 1 when divided by 2 and also leave a remainder of 1 when divided by 3?

32. Which integers are divisible by 5 but leave a remainder of 1 when divided by 3?

33. Use Fermat's little theorem to find 7^{121} **mod** 13.

34. Use Fermat's little theorem to find 23^{1002} **mod** 41.

35. Use Fermat's little theorem to show that if p is prime and $p \nmid a$, then a^{p-2} is an inverse of a modulo p.

36. Use Exercise 35 to find an inverse of 5 modulo 41.

37. a) Show that $2^{340} \equiv 1 \pmod{11}$ by Fermat's little theorem and noting that $2^{340} = (2^{10})^{34}$.
b) Show that $2^{340} \equiv 1 \pmod{31}$ using the fact that $2^{340} = (2^5)^{68} = 32^{68}$.
c) Conclude from parts (a) and (b) that $2^{340} \equiv 1 \pmod{341}$.

38. a) Use Fermat's little theorem to compute 3^{302} **mod** 5, 3^{302} **mod** 7, and 3^{302} **mod** 11.
b) Use your results from part (a) and the Chinese remainder theorem to find 3^{302} **mod** 385. (Note that $385 = 5 \cdot 7 \cdot 11$.)

39. a) Use Fermat's little theorem to compute 5^{2003} **mod** 7, 5^{2003} **mod** 11, and 5^{2003} **mod** 13.
b) Use your results from part (a) and the Chinese remainder theorem to find 5^{2003} **mod** 1001. (Note that $1001 = 7 \cdot 11 \cdot 13$.)

40. Show with the help of Fermat's little theorem that if n is a positive integer, then 42 divides $n^7 - n$.

41. Show that if p is an odd prime, then every divisor of the Mersenne number $2^p - 1$ is of the form $2kp + 1$, where k is a nonnegative integer. [*Hint:* Use Fermat's little theorem and Exercise 37 of Section 4.3.]

42. Use Exercise 41 to determine whether $M_{13} = 2^{13} - 1 = 8191$ and $M_{23} = 2^{23} - 1 = 8{,}388{,}607$ are prime.

43. Use Exercise 41 to determine whether $M_{11} = 2^{11} - 1 = 2047$ and $M_{17} = 2^{17} - 1 = 131{,}071$ are prime.

☞ Let n be a positive integer and let $n - 1 = 2^s t$, where s is a nonnegative integer and t is an odd positive integer. We say that n passes **Miller's test for the base *b*** if either $b^t \equiv 1 \pmod n$ or $b^{2^j t} \equiv -1 \pmod n$ for some j with $0 \leq j \leq s - 1$. It can be shown (see [Ro10]) that a composite integer n passes Miller's test for fewer than $n/4$ bases b with $1 < b < n$. A composite positive integer n that passes Miller's test to the base b is called a **strong pseudoprime to the base *b***.

***44.** Show that if n is prime and b is a positive integer with $n \nmid b$, then n passes Miller's test to the base b.

45. Show that 2047 is a strong pseudoprime to the base 2 by showing that it passes Miller's test to the base 2, but is composite.

46. Show that 1729 is a Carmichael number.

47. Show that 2821 is a Carmichael number.

***48.** Show that if $n = p_1 p_2 \cdots p_k$, where $p_1, p_2, \ldots, p_k$ are distinct primes that satisfy $p_j - 1 \mid n - 1$ for $j = 1, 2, \ldots, k$, then n is a Carmichael number.

49. a) Use Exercise 48 to show that every integer of the form $(6m + 1)(12m + 1)(18m + 1)$, where m is a positive integer and $6m + 1$, $12m + 1$, and $18m + 1$ are all primes, is a Carmichael number.
b) Use part (a) to show that 172,947,529 is a Carmichael number.

50. Find the nonnegative integer a less than 28 represented by each of these pairs, where each pair represents (a **mod** 4, a **mod** 7).

a) (0, 0)	**b)** (1, 0)	**c)** (1, 1)
d) (2, 1)	**e)** (2, 2)	**f)** (0, 3)
g) (2, 0)	**h)** (3, 5)	**i)** (3, 6)

51. Express each nonnegative integer a less than 15 as a pair (a **mod** 3, a **mod** 5).

52. Explain how to use the pairs found in Exercise 51 to add 4 and 7.

53. Solve the system of congruences that arises in Example 8.

54. Show that 2 is a primitive root of 19.

55. Find the discrete logarithms of 5 and 6 to the base 2 modulo 19.

56. Let p be an odd prime and r a primitive root of p. Show that if a and b are positive integers in $\mathbf{Z}_p$, then $\log_r(ab) \equiv \log_r a + \log_r b \pmod{p - 1}$.

57. Write out a table of discrete logarithms modulo 17 with respect to the primitive root 3.

If m is a positive integer, the integer a is a **quadratic residue** of m if $\gcd(a, m) = 1$ and the congruence $x^2 \equiv a \pmod m$ has a solution. In other words, a quadratic residue of m is an integer relatively prime to m that is a perfect square modulo m. If a is not a quadratic residue of m and $\gcd(a, m) = 1$, we say that it is a **quadratic nonresidue** of m. For example, 2 is a quadratic residue of 7 because $\gcd(2, 7) = 1$ and $3^2 \equiv 2 \pmod 7$ and 3 is a quadratic nonresidue of 7 because $\gcd(3, 7) = 1$ and $x^2 \equiv 3 \pmod 7$ has no solution.

58. Which integers are quadratic residues of 11?

59. Show that if p is an odd prime and a is an integer not divisible by p, then the congruence $x^2 \equiv a \pmod{p}$ has either no solutions or exactly two incongruent solutions modulo p.

60. Show that if p is an odd prime, then there are exactly $(p-1)/2$ quadratic residues of p among the integers $1, 2, \ldots, p-1$.

If p is an odd prime and a is an integer not divisible by p, the **Legendre symbol** $\left(\frac{a}{p}\right)$ is defined to be 1 if a is a quadratic residue of p and -1 otherwise.

61. Show that if p is an odd prime and a and b are integers with $a \equiv b \pmod{p}$, then

$$\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right).$$

62. Prove **Euler's criterion**, which states that if p is an odd prime and a is a positive integer not divisible by p, then

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

[*Hint:* If a is a quadratic residue modulo p, apply Fermat's little theorem; otherwise, apply Wilson's theorem, given in Exercise 18(b).]

63. Use Exercise 62 to show that if p is an odd prime and a and b are integers not divisible by p, then

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$$

64. Show that if p is an odd prime, then -1 is a quadratic residue of p if $p \equiv 1 \pmod{4}$, and -1 is not a quadratic residue of p if $p \equiv 3 \pmod{4}$. [*Hint:* Use Exercise 62.]

65. Find all solutions of the congruence $x^2 \equiv 29 \pmod{35}$. [*Hint:* Find the solutions of this congruence modulo 5 and modulo 7, and then use the Chinese remainder theorem.]

66. Find all solutions of the congruence $x^2 \equiv 16 \pmod{105}$. [*Hint:* Find the solutions of this congruence modulo 3, modulo 5, and modulo 7, and then use the Chinese remainder theorem.]

67. Describe a brute-force algorithm for solving the discrete logarithm problem and find the worst-case and average-case time complexity of this algorithm.

4.5 Applications of Congruences

Congruences have many applications to discrete mathematics, computer science, and many other disciplines. We will introduce three applications in this section: the use of congruences to assign memory locations to computer files, the generation of pseudorandom numbers, and check digits.

Suppose that a customer identification number is ten digits long. To retrieve customer files quickly, we do not want to assign a memory location to a customer record using the ten-digit identification number. Instead, we want to use a smaller integer associated to the identification number. This can be done using what is known as a hashing function. In this section we will show how we can use modular arithmetic to do hashing.

Constructing sequences of random numbers is important for randomized algorithms, for simulations, and for many other purposes. Constructing a sequence of truly random numbers is extremely difficult, or perhaps impossible, because any method for generating what are supposed to be random numbers may generate numbers with hidden patterns. As a consequence, methods have been developed for finding sequences of numbers that have many desirable properties of random numbers, and which can be used for various purposes in place of random numbers. In this section we will show how to use congruences to generate sequences of pseudorandom numbers. The advantage is that the pseudorandom numbers so generated are constructed quickly; the disadvantage is that they have too much predictability to be used for many tasks.

Congruences also can be used to produce check digits for identification numbers of various kinds, such as code numbers used to identify retail products, numbers used to identify books, airline ticket numbers, and so on. We will explain how to construct check digits using congruences for a variety of types of identification numbers. We will show that these check digits can be used to detect certain kinds of common errors made when identification numbers are printed.

4.5.1 Hashing Functions

Links

The central computer at an insurance company maintains records for each of its customers. How can memory locations be assigned so that customer records can be retrieved quickly? The solution to this problem is to use a suitably chosen **hashing function**. Records are identified using a **key**, which uniquely identifies each customer's records. For instance, customer records are often identified using the Social Security number of the customer as the key. A hashing function h assigns memory location $h(k)$ to the record that has k as its key.

In practice, many different hashing functions are used. One of the most common is the function

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

where m is the number of available memory locations.

Hashing functions should be easily evaluated so that files can be quickly located. The hashing function $h(k) = k \bmod m$ meets this requirement; to find $h(k)$, we need only compute the remainder when k is divided by m. Furthermore, the hashing function should be onto, so that all memory locations are possible. The function $h(k) = k \bmod m$ also satisfies this property.

EXAMPLE 1 Find the memory locations assigned by the hashing function $h(k) = k \bmod 111$ to the records of customers with Social Security numbers 064212848 and 037149212.

Solution: The record of the customer with Social Security number 064212848 is assigned to memory location 14, because

$$h(064212848) = 064212848 \bmod 111 = 14.$$

Similarly, because

$$h(037149212) = 037149212 \bmod 111 = 65,$$

the record of the customer with Social Security number 037149212 is assigned to memory location 65. ◀

Because a hashing function is not one-to-one (because there are more possible keys than memory locations), more than one file may be assigned to a memory location. When this happens, we say that a **collision** occurs. One way to resolve a collision is to assign the first free location following the occupied memory location assigned by the hashing function.

EXAMPLE 2 After making the assignments of records to memory locations in Example 1, assign a memory location to the record of the customer with Social Security number 107405723.

Solution: First note that the hashing function $h(k) = k \bmod 111$ maps the Social Security number 107405723 to location 14, because

$$h(107405723) = 107405723 \bmod 111 = 14.$$

However, this location is already occupied (by the file of the customer with Social Security number 064212848). But, because memory location 15, the first location following memory location 14, is free, we assign the record of the customer with Social Security number 107405723 to this location. ◀

In Example 1 we used a **linear probing function**, namely, $h(k, i) = h(k) + i \bmod m$, to look for the first free memory location, where i runs from 0 to $m - 1$. There are many other ways to resolve collisions that are discussed in the references on hashing functions given at the end of the book.

4.5.2 Pseudorandom Numbers

Randomly chosen numbers are often needed for computer simulations. Different methods have been devised for generating numbers that have properties of randomly chosen numbers. Because numbers generated by systematic methods are not truly random, they are called **pseudorandom numbers**.

Links

The most commonly used procedure for generating pseudorandom numbers is the **linear congruential method**. We choose four integers: the **modulus** m, **multiplier** a, **increment** c, and **seed** x_0, with $2 \leq a < m$, $0 \leq c < m$, and $0 \leq x_0 < m$. 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

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

(This is an example of a recursive definition, discussed in Section 5.3. In that section we will show that such sequences are well defined.)

Many computer experiments require the generation of pseudorandom numbers between 0 and 1. To generate such numbers, we divide numbers generated with a linear congruential generator by the modulus: that is, we use the numbers x_n/m.

EXAMPLE 3 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: We compute the terms of this sequence by successively using the recursively defined function $x_{n+1} = (7x_n + 4) \bmod 9$, beginning by inserting the seed $x_0 = 3$ to find x_1. We find that

$$\begin{aligned}
x_1 &= 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7,\\
x_2 &= 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8,\\
x_3 &= 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6,\\
x_4 &= 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1,\\
x_5 &= 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2,\\
x_6 &= 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0,\\
x_7 &= 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = \ \ 4 \bmod 9 = 4,\\
x_8 &= 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5,\\
x_9 &= 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3.
\end{aligned}$$

Because $x_9 = x_0$ and because each term depends only on the previous term, we see that the sequence

$$3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \ldots$$

is generated. This sequence contains nine different numbers before repeating. ◀

Most computers do use linear congruential generators to generate pseudorandom numbers. Often, a linear congruential generator with increment $c = 0$ is used. Such a generator is called

a **pure multiplicative generator**. For example, the pure multiplicative generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16{,}807$ is widely used. With these values, it can be shown that $2^{31} - 2$ numbers are generated before repetition begins.

Pseudorandom numbers generated by linear congruential generators have long been used for many tasks. Unfortunately, it has been shown that sequences of pseudorandom numbers generated in this way do not share some important statistical properties that true random numbers have. Because of this, it is not advisable to use them for some tasks, such as large simulations. For such sensitive tasks, other methods are used to produce sequences of pseudorandom numbers, either using some sort of algorithm or sampling numbers arising from a random physical phenomenon. For more details on pseudorandom number, see [Kn97] and [Re10].

4.5.3 Check Digits

Congruences are used to check for errors in digit strings. A common technique for detecting errors in such strings is to add an extra digit at the end of the string. This final digit, or check digit, is calculated using a particular function. Then, to determine whether a digit string is correct, a check is made to see whether this final digit has the correct value. We begin with an application of this idea for checking the correctness of bit strings.

EXAMPLE 4 **Parity Check Bits** Digital information is represented by bit string, split into blocks of a specified size. Before each block is stored or transmitted, an extra bit, called a **parity check bit**, can be appended to each block. The parity check bit x_{n+1} for the bit string $x_1x_2 \dots x_n$ is defined by

$$x_{n+1} = x_1 + x_2 + \cdots + x_n \bmod 2.$$

It follows that x_{n+1} is 0 if there are an even number of 1 bits in the block of n bits and it is 1 if there are an odd number of 1 bits in the block of n bits. When we examine a string that includes a parity check bit, we know that there is an error in it if the parity check bit is wrong. However, when the parity check bit is correct, there still may be an error. A parity check can detect an odd number of errors in the previous bits, but not an even number of errors. (See Exercise 14.)

Suppose we receive in a transmission the bit strings 01100101 and 11010110, each ending with a parity check bit. Should we accept these bit strings as correct?

Solution: Before accepting these strings as correct, we examine their parity check bits. The parity check bit of the first string is 1. Because $0 + 1 + 1 + 0 + 0 + 1 + 0 \equiv 1 \pmod 2$, the parity check bit is correct. The parity check bit of the second string is 0. We find that $1 + 1 + 0 + 1 + 0 + 1 + 1 \equiv 1 \pmod 2$, so the parity check is incorrect. We conclude that the first string may have been transmitted correctly and we know for certain that the second string was transmitted incorrectly. We accept the first string as correct (even though it still may contain an even number of errors), but we reject the second string. ◀

Check bits computed using congruences are used extensively to verify the correctness of various kinds of identification numbers. Examples 5 and 6 show how check bits are computed for codes that identify products (Universal Product Codes) and books (International Standard Book Numbers). The preambles to Exercises 18, 28, and 32 introduce the use of congruences to find and use check digits in money order numbers, airline ticket numbers, and identification numbers for periodicals, respectively. Note that congruences are also used to compute check digits for bank account numbers, drivers license numbers, credit card numbers, and many other types of identification numbers.

EXAMPLE 5 **UPCs** Retail products are identified by their **Universal Product Codes (UPCs)**. The most common form of a UPC has 12 decimal digits: the first digit identifies the product category, the

next five digits identify the manufacturer, the following five identify the particular product, and the last digit is a check digit. The check digit is determined by the congruence

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

Answer these questions:

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

Solution: (a) We insert the digits of 79357343104 into the congruence for UPC check digits. This gives $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$. Simplifying, we have $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$. Hence, $98 + x_{12} \equiv 0 \pmod{10}$. It follows that $x_{12} \equiv 2 \pmod{10}$, so the check digit is 2.

(b) To check whether 041331021641 is valid, we insert the digits into the congruence these digits must satisfy. This gives $3 \cdot 0 + 4 + 3 \cdot 1 + 3 + 3 \cdot 3 + 1 + 3 \cdot 0 + 2 + 3 \cdot 1 + 6 + 3 \cdot 4 + 1 \equiv 0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 \equiv 4 \not\equiv 0 \pmod{10}$. Hence, 041331021641 is not a valid UPC. ◀

EXAMPLE 6

Remember that the check digit of an ISBN-10 can be an X!

ISBNs All books are identified by an **International Standard Book Number (ISBN-10)**, a 10-digit code $x_1x_2 \ldots x_{10}$, assigned by the publisher. (Recently, a 13-digit code known as ISBN-13 was introduced to identify a larger number of published works; see the preamble to Exercise 42 in the Supplementary Exercises.) An ISBN-10 consists of blocks identifying the language, the publisher, the number assigned to the book by its publishing company, and finally, a check digit that is either a digit or the letter X (used to represent 10). This check digit is selected so that

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

or equivalently, so that

$$\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}.$$

Answer these questions about ISBN-10s:

(a) The first nine digits of the ISBN-10 of the sixth edition of this book are 007288008. What is the check digit?

(b) Is 084930149X a valid ISBN-10?

Solution: (a) The check digit is determined by the congruence $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$. Inserting the digits 007288008 gives $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \pmod{11}$. This means that $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$, so $x_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $x_{10} = 2$.

(b) To see whether 084930149X is a valid ISBN-10, we see if $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$. We see that $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10 = 0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$. Hence, 084930149X is not a valid ISBN-10. ◀

Several kinds of errors often arise in identification numbers. A **single error**, an error in one digit of an identification number, is perhaps the most common type of error. Another common kind of error is a **transposition error**, which occurs when two digits are accidentally interchanged. For each type of identification number, including a check digit, we would like to be able to detect these common types of errors, as well as other types of errors. We will investigate whether the check digit for ISBNs can detect single errors and transposition errors. Whether check digits for UPCs can detect these kinds of errors is left as Exercises 26 and 27.

Publishers sometimes do not calculate ISBNs correctly for their books, as was done for an earlier edition of this text.

Suppose that $x_1x_2 \dots x_{10}$ is a valid ISBN (so that $\sum_{i=1}^{10} x_i \equiv 0 \pmod{10}$). We will show that we can detect a single error and a transposition of two digits (where we include the possibility that one of the two digits is the check digit X, representing 10). Suppose that this ISBN has been printed with a single error as $y_1y_2 \dots y_{10}$. If there is a single error, then, for some integer j, $y_i = x_i$ for $i \neq j$ and $y_j = x_j + a$, where $-10 \leq a \leq 10$ and $a \neq 0$. Note that $a = y_j - x_j$ is the error in the jth place. It then follows that

$$\sum_{i=1}^{10} iy_i = \Big(\sum_{i=1}^{10} ix_i\Big) + ja \equiv ja \not\equiv 0 \pmod{11}.$$

These last two congruences hold because $\sum_{i=1}^{10} x_i \equiv 0 \pmod{10}$ and $11 \nmid ja$, because $11 \nmid j$ and $11 \nmid a$. We conclude that $y_1y_2 \dots y_{10}$ is not a valid ISBN. So, we have detected the single error.

Now suppose that two unequal digits have been transposed. It follows that there are distinct integers j and k such that $y_j = x_k$ and $y_k = x_j$, and $y_i = x_i$ for $i \neq j$ and $i \neq k$. Hence,

$$\sum_{i=1}^{10} iy_i = \Big(\sum_{i=1}^{10} ix_i\Big) + (jx_k - jx_j) + (kx_j - kx_k) \equiv (j-k)(x_k - x_j) \not\equiv 0 \pmod{11},$$

because $\sum_{i=1}^{10} x_i \equiv 0 \pmod{10}$ and $11 \nmid (j-k)$ and $11 \nmid (x_k - x_j)$. We see that $y_1y_2 \dots y_{10}$ is not a valid ISBN. Thus, we can detect the interchange of two unequal digits.

Exercises

1. Which memory locations are assigned by the hashing function $h(k) = k$ **mod** 97 to the records of insurance company customers with these Social Security numbers?
a) 034567981 **b)** 183211232
c) 220195744 **d)** 987255335

2. Which memory locations are assigned by the hashing function $h(k) = k$ **mod** 101 to the records of insurance company customers with these Social Security numbers?
a) 104578690 **b)** 432222187
c) 372201919 **d)** 501338753

3. A parking lot has 31 visitor spaces, numbered from 0 to 30. Visitors are assigned parking spaces using the hashing function $h(k) = k$ **mod** 31, where k is the number formed from the first three digits on a visitor's license plate.
a) Which spaces are assigned by the hashing function to cars that have these first three digits on their license plates: 317, 918, 007, 100, 111, 310?
b) Describe a procedure visitors should follow to find a free parking space, when the space they are assigned is occupied.

Another way to resolve collisions in hashing is to use *double hashing*. We use an initial hashing function $h(k) = k$ **mod** p, where p is prime. We also use a second hashing function $g(k) = (k + 1)$ **mod** $(p - 2)$. When a collision occurs, we use a *probing sequence* $h(k, i) = (h(k) + i \cdot g(k))$ **mod** p.

4. Use the double hashing procedure we have described with $p = 4969$ to assign memory locations to files for employees with social security numbers $k_1 = 132489971$, $k_2 = 509496993$, $k_3 = 546332190$, $k_4 = 034367980$, $k_5 = 047900151$, $k_6 = 329938157$, $k_7 = 212228844$, $k_8 = 325510778$, $k_9 = 353354519$, $k_{10} = 053708912$.

5. What sequence of pseudorandom numbers is generated using the linear congruential generator $x_{n+1} = (3x_n + 2)$ **mod** 13 with seed $x_0 = 1$?

6. What sequence of pseudorandom numbers is generated using the linear congruential generator $x_{n+1} = (4x_n + 1)$ **mod** 7 with seed $x_0 = 3$?

7. What sequence of pseudorandom numbers is generated using the pure multiplicative generator $x_{n+1} = 3x_n$ **mod** 11 with seed $x_0 = 2$?

8. Write an algorithm in pseudocode for generating a sequence of pseudorandom numbers using a linear congruential generator.

The **middle-square method** for generating pseudorandom numbers begins with an n-digit integer. This number is squared, initial zeros are appended to ensure that the result has $2n$ digits, and its middle n digits are used to form the next number in the sequence. This process is repeated to generate additional terms.

9. Find the first eight terms of the sequence of four-digit pseudorandom numbers generated by the middle square method starting with 2357.

10. Explain why both 3792 and 2916 would be bad choices for the initial term of a sequence of four-digit pseudorandom numbers generated by the middle square method.

The **power generator** is a method for generating pseudorandom numbers. To use the power generator, parameters p and d are specified, where p is a prime, d is a positive integer such that $p \nmid d$, and a seed x_0 is specified. The pseudorandom numbers $x_1, x_2, \ldots$ are generated using the recursive definition $x_{n+1} = x_n^d \bmod p$.

11. Find the sequence of pseudorandom numbers generated by the power generator with $p = 7$, $d = 3$, and seed $x_0 = 2$.

12. Find the sequence of pseudorandom numbers generated by the power generator with $p = 11$, $d = 2$, and seed $x_0 = 3$.

13. Suppose you received these bit strings over a communications link, where the last bit is a parity check bit. In which string are you sure there is an error?

a) 00000111111 **b)** 10101010101
c) 11111100000 **d)** 10111101111

14. Prove that a parity check bit can detect an error in a string if and only if the string contains an odd number of errors.

15. The first nine digits of the ISBN-10 of the European version of the fifth edition of this book are 0-07-119881. What is the check digit for that book?

16. The ISBN-10 of the sixth edition of *Elementary Number Theory and Its Applications* is 0-321-500Q1-8, where Q is a digit. Find the value of Q.

17. Determine whether the check digit of the ISBN-10 for this textbook (the eighth edition of *Discrete Mathematics and Its Applications*) was computed correctly by the publisher.

The United States Postal Service (USPS) sells money orders identified by an 11-digit number $x_1x_2 \ldots x_{11}$. The first ten digits identify the money order; x_{11} is a check digit that satisfies $x_{11} = x_1 + x_2 + \cdots + x_{10} \bmod 9$.

18. Find the check digit for the USPS money orders that have identification number that start with these ten digits.

a) 7555618873 **b)** 6966133421
c) 8018927435 **d)** 3289744134

19. Determine whether each of these numbers is a valid USPS money order identification number.

a) 74051489623 **b)** 88382013445
c) 56152240784 **d)** 66606631178

20. One digit in each of these identification numbers of a postal money order is smudged. Can you recover the smudged digit, indicated by a Q, in each of these numbers?

a) Q1223139784 **b)** 6702120Q988
c) 27Q41007734 **d)** 213279032Q1

21. One digit in each of these identification numbers of a postal money order is smudged. Can you recover the smudged digit, indicated by a Q, in each of these numbers?

a) 493212Q0688 **b)** 850Q9103858
c) 2Q941007734 **d)** 66687Q03201

22. Determine which single digit errors are detected by the USPS money order code.

23. Determine which transposition errors are detected by the USPS money order code.

24. Determine the check digit for the UPCs that have these initial 11 digits.

a) 73232184434 **b)** 63623991346
c) 04587320720 **d)** 93764323341

25. Determine whether each of the strings of 12 digits is a valid UPC code.

a) 036000291452 **b)** 012345678903
c) 782421843014 **d)** 726412175425

26. Does the check digit of a UPC code detect all single errors? Prove your answer or find a counterexample.

27. Determine which transposition errors the check digit of a UPC code finds.

Some airline tickets have a 15-digit identification number $a_1a_2 \ldots a_{15}$, where a_{15} is a check digit that equals $a_1a_2 \ldots a_{14} \bmod 7$.

28. Find the check digit a_{15} that follows each of these initial 14 digits of an airline ticket identification number.

a) 10237424413392 **b)** 00032781811234
c) 00611232134231 **d)** 00193222543435

29. Determine whether each of these 15-digit numbers is a valid airline ticket identification number.

a) 101333341789013 **b)** 007862342770445
c) 113273438882531 **d)** 000122347322871

30. Which errors in a single digit of a 15-digit airline ticket identification number can be detected?

***31.** Can the accidental transposition of two consecutive digits in an airline ticket identification number be detected using the check digit?

Periodicals are identified using an **International Standard Serial Number (ISSN)**. An ISSN consists of two blocks of four digits. The last digit in the second block is a check digit. This check digit is determined by the congruence $d_8 \equiv 3d_1 + 4d_2 + 5d_3 + 6d_4 + 7d_5 + 8d_6 + 9d_7 \pmod{11}$. When $d_8 \equiv 10 \pmod{11}$, we use the letter X to represent d_8 in the code.

32. For each of these initial seven digits of an ISSN, determine the check digit (which may be the letter X).

a) 1570-868 **b)** 1553-734
c) 1089-708 **d)** 1383-811

33. Are each of these eight-digit codes possible ISSNs? That is, do they end with a correct check digit?

a) 1059-1027 **b)** 0002-9890
c) 1530-8669 **d)** 1007-120X

34. Does the check digit of an ISSN detect every single error in an ISSN? Justify your answer with either a proof or a counterexample.

35. Does the check digit of an ISSN detect every error where two consecutive digits are accidentally interchanged? Justify your answer with either a proof or a counterexample.

4.6 Cryptography

4.6.1 Introduction

Number theory plays a key role in cryptography, the subject of transforming information so that it cannot be easily recovered without special knowledge. Number theory is the basis of many classical ciphers, first used thousands of years ago, and used extensively until the 20th century. These ciphers encrypt messages by changing each letter to a different letter, or each block of letters to a different block of letters. We will discuss some classical ciphers, including shift ciphers, which replace each letter by the letter a fixed number of positions later in the alphabet, wrapping around to the beginning of the alphabet when necessary. The classical ciphers we will discuss are examples of private key ciphers where knowing how to encrypt allows someone to also decrypt messages. With a private key cipher, two parties who wish to communicate in secret must share a secret key. The classical ciphers we will discuss are also vulnerable to cryptanalysis, which seeks to recover encrypted information without access to the secret information used to encrypt the message. We will show how to cryptanalyze messages sent using shift ciphers.

Number theory is also important in public key cryptography, a type of cryptography invented in the 1970s. In public key cryptography, knowing how to encrypt does not also tell someone how to decrypt. The most widely used public key system, called the RSA cryptosystem, encrypts messages using modular exponentiation, where the modulus is the product of two large primes. Knowing how to encrypt requires that someone know the modulus and an exponent. (It does not require that the two prime factors of the modulus be known.) As far as it is known, knowing how to decrypt requires someone to know how to invert the encryption function, which can only be done in a practical amount of time when someone knows these two large prime factors. In this chapter we will explain how the RSA cryptosystem works, including how to encrypt and decrypt messages.

The subject of cryptography also includes the subject of cryptographic protocols, which are exchanges of messages carried out by two or more parties to achieve a specific security goal. We will discuss two important protocols in this chapter. One allows two people to share a common secret key. The other can be used to send signed messages so that a recipient can be sure that they were sent by the purported sender. Finally, we will introduce the notion of a homomorphic cryptosystem, now playing an important role in cloud computing. Data can become vulnerable if they must be decrypted for use as input to programs. Homomorphic encryption eliminates this vulnerability by allowing programs to be run on encrypted data. The output of these programs is then the encryption of the desired output.

4.6.2 Classical Cryptography

One of the earliest known uses of cryptography was by Julius Caesar. He made messages secret by shifting each letter three letters forward in the alphabet (sending the last three letters of the alphabet to the first three). For instance, using this scheme the letter B is sent to E and the letter X is sent to A. This is an example of **encryption**, that is, the process of making a message secret.

To express Caesar's encryption process mathematically, first replace each letter by an element of $\mathbf{Z}_{26}$, that is, an integer from 0 to 25 equal to one less than its position in the alphabet. For example, replace A by 0, K by 10, and Z by 25. Caesar's encryption method can be represented by the function f that assigns to the nonnegative integer p, $p \leq 25$, the integer $f(p)$ in the set $\{0, 1, 2, \ldots, 25\}$ with

$$f(p) = (p + 3) \bmod 26.$$

In the encrypted version of the message, the letter represented by p is replaced with the letter represented by $(p + 3) \bmod 26$.

EXAMPLE 1 What is the secret message produced from the message "MEET YOU IN THE PARK" using the Caesar cipher?

Solution: First replace the letters in the message with numbers. This produces

12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3) \bmod 26$. This gives

15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Translating this back to letters produces the encrypted message "PHHW BRX LQ WKH SDUN." ◀

To recover the original message from a secret message encrypted by the Caesar cipher, the function f^{-1}, the inverse of f, is used. Note that the function f^{-1} sends an integer p from $\mathbf{Z}_{26}$, to $f^{-1}(p) = (p - 3) \bmod 26$. In other words, to find the original message, each letter is shifted back three letters in the alphabet, with the first three letters sent to the last three letters of the alphabet. The process of determining the original message from the encrypted message is called **decryption**.

There are various ways to generalize the Caesar cipher. For example, instead of shifting the numerical equivalent of each letter by 3, we can shift the numerical equivalent of each letter by k, so that

$$f(p) = (p + k) \bmod 26.$$

Such a cipher is called a *shift cipher*. Note that decryption can be carried out using

$$f^{-1}(p) = (p - k) \bmod 26.$$

Here the integer k is called a **key**. We illustrate the use of a shift cipher in Examples 2 and 3.

EXAMPLE 2 Encrypt the plaintext message "STOP GLOBAL WARMING" using the shift cipher with shift $k = 11$.

Solution: To encrypt the message "STOP GLOBAL WARMING" we first translate each letter to the corresponding element of $\mathbf{Z}_{26}$. This produces the string

18 19 14 15 6 11 14 1 0 11 22 0 17 12 8 13 6.

We now apply the shift $f(p) = (p + 11)$ **mod** 26 to each number in this string. We obtain

3 4 25 0 17 22 25 12 11 22 7 11 2 23 19 24 17.

Translating this last string back to letters, we obtain the ciphertext "DEZA RWZMLW HLCXTYR." ◀

EXAMPLE 3 Decrypt the ciphertext message "LEWLYPLUJL PZ H NYLHA ALHJOLY" that was encrypted with the shift cipher with shift $k = 7$.

Solution: To decrypt the ciphertext "LEWLYPLUJL PZ H NYLHA ALHJOLY" we first translate the letters back to elements of $\mathbf{Z}_{26}$. We obtain

11 4 22 11 24 15 11 20 9 11 15 25 7 13 24 11 7 0 0 11 7 9 14 11 24.

Next, we shift each of these numbers by $-k = -7$ modulo 26 to obtain

4 23 15 4 17 8 4 13 2 4 8 18 0 6 17 4 0 19 19 4 0 2 7 4 17.

Finally, we translate these numbers back to letters to obtain the plaintext. We obtain "EXPERIENCE IS A GREAT TEACHER." ◀

We can generalize shift ciphers further to slightly enhance security by using a function of the form

$$f(p) = (ap + b) \textbf{ mod } 26,$$

where a and b are integers, chosen so that f is a bijection. (The function $f(p) = (ap + b)$ **mod** 26 is a bijection if and only if $\gcd(a, 26) = 1$.) Such a mapping is called an *affine transformation*, and the resulting cipher is called an *affine cipher*.

EXAMPLE 4 What letter replaces the letter K when the function $f(p) = (7p + 3)$ **mod** 26 is used for encryption?

Solution: First, note that 10 represents K. Then, using the encryption function specified, it follows that $f(10) = (7 \cdot 10 + 3)$ **mod** $26 = 21$. Because 21 represents V, K is replaced by V in the encrypted message. ◀

We will now show how to decrypt messages encrypted using an affine cipher. Suppose that $c = (ap + b)$ **mod** 26 with $\gcd(a, 26) = 1$. To decrypt we need to show how to express p in terms of c. To do this, we apply the encrypting congruence $c \equiv ap + b \pmod{26}$, and solve it for p. To do this, we first subtract b from both sides, to obtain $c - b \equiv ap \pmod{26}$. Because $\gcd(a, 26) = 1$, we know that there is an inverse $\overline{a}$ of a modulo 26. Multiplying both sides of the last equation by $\overline{a}$ gives us $\overline{a}(c - b) \equiv \overline{a}ap \pmod{26}$. Because $\overline{a}a \equiv 1 \pmod{26}$, this tells us that $p \equiv \overline{a}(c - b) \pmod{26}$. This determines p because p belongs to $\mathbf{Z}_{26}$.

CRYPTANALYSIS The process of recovering plaintext from ciphertext without knowledge of both the encryption method and the key is known as **cryptanalysis** or **breaking codes**. In general, cryptanalysis is a difficult process, especially when the encryption method is unknown.

We will not discuss cryptanalysis in general, but we will explain how to break messages that were encrypted using a shift cipher.

If we know that a ciphertext message was produced by enciphering a message using a shift cipher, we can try to recover the message by shifting all characters of the ciphertext by each of the 26 possible shifts (including a shift of zero characters). One of these is guaranteed to be the plaintext. However, we can use a more intelligent approach, which we can build upon to cryptanalyze ciphertext resulting from other ciphers. The main tool for cryptanalyzing ciphertext encrypted using a shift cipher is the count of the frequency of letters in the ciphertext. The nine most common letters in English text and their approximate relative frequencies are E 13%, T 9%, A 8%, O 8%, I 7%, N 7%, S 7%, H 6%, and R 6%. To cryptanalyze ciphertext that we know was produced using a shift cipher, we first find the relative frequencies of letters in the ciphertext. We list the most common letters in the ciphertext in frequency order; we hypothesize that the most common letter in the ciphertext is produced by encrypting E. Then, we determine the value of the shift under this hypothesis, say k. If the message produced by shifting the ciphertext by $-k$ makes sense, we presume that our hypothesis is correct and that we have the correct value of k. If it does not make sense, we next consider the hypothesis that the most common letter in the ciphertext is produced by encrypting T, the second most common letter in English; we find k under this hypothesis, shift the letters of the message by $-k$, and see whether the resulting message makes sense. If it does not, we continue the process working our way through the letters from most common to least common.

Mathematicians make the best code breakers. Their work in World War II changed the course of the war.

EXAMPLE 5 Suppose that we intercepted the ciphertext message ZNK KGXRE HOXJ MKZY ZNK CUXS that we know was produced by a shift cipher. What was the original plaintext message?

Solution: Because we know that the intercepted ciphertext message was encrypted using a shift cipher, we begin by calculating the frequency of letters in the ciphertext. We find that the most common letter in the ciphertext is K. So, we hypothesize that the shift cipher sent the plaintext letter E to the ciphertext letter K. If this hypothesis is correct, we know that $10 = 4 + k$ **mod** 26, so $k = 6$. Next, we shift the letters of the message by -6, obtaining THE EARLY BIRD GETS THE WORM. Because this message makes sense, we assume that the hypothesis that $k = 6$ is correct. ◀

Links

BLOCK CIPHERS Shift ciphers and affine ciphers proceed by replacing each letter of the alphabet by another letter in the alphabet. Because of this, these ciphers are called **character** or **monoalphabetic ciphers**. Encryption methods of this kind are vulnerable to attacks based on the analysis of letter frequency in the ciphertext, as we just illustrated. We can make it harder to successfully attack ciphertext by replacing blocks of letters with other blocks of letters instead of replacing individual characters with individual characters; such ciphers are called **block ciphers**.

We will now introduce a simple type of block cipher, called the **transposition cipher**. As a key we use a permutation σ of the set $\{1, 2, \ldots, m\}$ for some positive integer m, that is, a one-to-one function from $\{1, 2, \ldots, m\}$ to itself. To encrypt a message we first split its letters into blocks of size m. (If the number of letters in the message is not divisible by m we add some random letters at the end to fill out the final block.) We encrypt the block $p_1 p_2 \ldots p_m$ as $c_1 c_2 \ldots c_m = p_{\sigma(1)} p_{\sigma(2)} \cdots, p_{\sigma(m)}$. To decrypt a ciphertext block $c_1 c_2 \ldots c_m$, we transpose its letters using the permutation σ^{-1}, the inverse of σ. Example 6 illustrates encryption and decryption for a transposition cipher.

EXAMPLE 6 Using the transposition cipher based on the permutation σ of the set $\{1, 2, 3, 4\}$ with $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 4$, and $\sigma(4) = 2$,

(a) Encrypt the plaintext message PIRATE ATTACK.

(b) Decrypt the ciphertext message SWUE TRAE OEHS, which was encrypted using this cipher.

Solution: (a) We first split the letters of the plaintext into blocks of four letters. We obtain PIRA TEAT TACK. To encrypt each block, we send the first letter to the third position, the second letter to the first position, the third letter to the fourth position, and the fourth letter to the second position. We obtain IAPR ETTA AKTC.

(b) We note that σ^{-1}, the inverse of σ, sends 1 to 2, sends 2 to 4, sends 3 to 1, and sends 4 to 3. Applying $\sigma^{-1}(m)$ to each block gives us the plaintext: USEW ATER HOSE. (Grouping together these letters to form common words, we surmise that the plaintext is USE WATER HOSE.) ◀

CRYPTOSYSTEMS We have defined two families of ciphers: shift ciphers and affine ciphers. We now introduce the notion of a cryptosystem, which provides a general structure for defining new families of ciphers.

Definition 1

A *cryptosystem* is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{P}$ is the set of plaintext strings, $\mathcal{C}$ is the set of ciphertext strings, $\mathcal{K}$ is the *keyspace* (the set of all possible keys), $\mathcal{E}$ is the set of encryption functions, and $\mathcal{D}$ is the set of decryption functions. We denote by E_k the encryption function in $\mathcal{E}$ corresponding to the key k and D_k the decryption function in $\mathcal{D}$ that decrypts ciphertext that was encrypted using E_k, that is, $D_k(E_k(p)) = p$, for all plaintext strings p.

We now illustrate the use of the definition of a cryptosystem.

EXAMPLE 7 Describe the family of shift ciphers as a cryptosytem.

Solution: To encrypt a string of English letters with a shift cipher, we first translate each letter to an integer between 0 and 25, that is, to an element of $\mathbf{Z}_{26}$. We then shift each of these integers by a fixed integer modulo 26, and finally, we translate the integers back to letters. To apply the definition of a cryptosystem to shift ciphers, we assume that our messages are already integers, that is, elements of $\mathbf{Z}_{26}$. That is, we assume that the translation between letters and integers is outside of the cryptosystem. Consequently, both the set of plaintext strings $\mathcal{P}$ and the set of ciphertext strings $\mathcal{C}$ are the set of strings of elements of $\mathbf{Z}_{26}$. The set of keys $\mathcal{K}$ is the set of possible shifts, so $\mathcal{K} = \mathbf{Z}_{26}$. The set $\mathcal{E}$ consists of functions of the form $E_k(p) = (p + k) \bmod 26$, and the set $\mathcal{D}$ of decryption functions is the same as the set of encrypting functions where $D_k(p) = (p - k) \bmod 26$. ◀

The concept of a cryptosystem is useful in the discussion of additional families of ciphers and is used extensively in cryptography.

4.6.3 Public Key Cryptography

All classical ciphers, including shift ciphers and affine ciphers, are examples of **private key cryptosystems**. In a private key cryptosystem, once you know an encryption key, you can quickly find the decryption key. So, knowing how to encrypt messages using a particular key allows you to decrypt messages that were encrypted using this key. For example, when a shift cipher is used with encryption key k, the plaintext integer p is sent to

$$c = (p + k) \bmod 26.$$

Decryption is carried out by shifting by $-k$; that is,

$$p = (c - k) \bmod 26.$$

So knowing how to encrypt with a shift cipher also tells you how to decrypt.

When a private key cryptosystem is used, two parties who wish to communicate in secret must share a secret key. Because anyone who knows this key can both encrypt and decrypt messages, two people who want to communicate securely need to securely exchange this key. (We will introduce a method for doing this later in this section.) The shift cipher and affine cipher cryptosystems are private key cryptosystems. They are quite simple and are extremely vulnerable to cryptanalysis. However, the same is not true of many modern private key cryptosystems. In particular, the current US government standard for private key cryptography, the Advanced Encryption Standard (AES), is extremely complex and is considered to be highly resistant to cryptanalysis. (See [St06] for details on AES and other modern private key cryptosystems.) AES is widely used in government and commercial communications. However, it still shares the property that for secure communications keys be shared. Furthermore, for extra security, a new key is used for each communication session between two parties, which requires a method for generating keys and securely sharing them.

To avoid the need for keys to be shared by every pair of parties that wish to communicate securely, in the 1970s cryptologists introduced the concept of **public key cryptosystems**. When such cryptosystems are used, knowing how to send an encrypted message does not help decrypt messages. In such a system, everyone can have a publicly known encryption key. Only the decryption keys are kept secret, and only the intended recipient of a message can decrypt it, because, as far as it is currently known, knowledge of the encryption key does not let someone recover the plaintext message without an extraordinary amount of work (such as billions of years of computer time).

The first public key cryptosystems were invented in the mid-1970s. Many additional public key cryptosystems have been developed in the ensuing decades. We will introduce the most commonly used public key cryptosystem, known as the RSA system, in this book. Besides RSA, there are several other commonly used public key cryptosystems that are now used for many applications. These other public key cryptosystems will play an important role in the future when advances in commuting may make the RSA cryptosystem obsolete, as often happens in cryptography. We will explain why this may be so shortly.

Although public key cryptography has the advantage that two parties who wish to communicate securely do not need to exchange keys, it has the disadvantage that encryption and decryption can be extremely time-consuming. For many applications, this makes public key cryptography impractical. In such situations, private key cryptography is used instead. However, public key cryptography may still be used in the key exchange process.

4.6.4 The RSA Cryptosystem

M.I.T. is also known as the 'Tute.

In 1976, three researchers at the Massachusetts Institute of Technology—Ronald Rivest, Adi Shamir, and Leonard Adleman—introduced to the world a public key cryptosystem, known as the **RSA system**, from the initials of its inventors. As often happens with cryptographic discoveries, the RSA system had been discovered several years earlier in secret government research in the United Kingdom. Clifford Cocks, working in secrecy at the United Kingdom's Government Communications Headquarters (GCHQ), had discovered this cryptosystem in 1973. However, his invention was unknown to the outside world until the late 1990s, when he was allowed to share classified GCHQ documents from the early 1970s. (An excellent account of this earlier discovery, as well as the work of Rivest, Shamir, and Adleman, can be found in [Si99].)

Unfortunately, no one calls this the Cocks cryptosystem.

In the RSA cryptosystem, each individual has an encryption key (n, e), where $n = pq$, the modulus is the product of two large primes p and q, say with 300 digits each, and an exponent e that is relatively prime to $(p - 1)(q - 1)$. To produce a usable key, two large primes must be found. This can be done quickly on a computer using probabilistic primality tests, referred to earlier in this section. However, the product of these primes $n = pq$, with approximately 600 digits, cannot, as far as is currently known, be factored in a reasonable length of time. As we will see, this is an important reason why decryption cannot, as far as is currently known, be done quickly without a separate decryption key.

Remark: With the steady increase of the speed of computers, the recommended size of the primes p and q used to produce a RSA public key has increased. But the larger n is, the slower RSA encryption and decryption become. When considering this tradeoff, the number of years a message needs to be remain secret is important. A more important consideration is that the development of quantum computing threatens the security of the RSA cryptosystem, because factorization algorithms have been developed for quantum computers that could then be used to quickly factor large primes. So, once quantum computing becomes practical, perhaps in the next 20 to 30 years, other public key cryptosystems that cannot be broken using quantum computing will need to be used.

4.6.5 RSA Encryption

To encrypt messages using a particular key (n, e), we first translate a plaintext message M into sequences of integers. To do this, we first translate each plaintext letter into a two-digit number, using the same translation we employed for shift ciphers, with one key difference. That is, we include an initial zero for the letters A through J, so that A is translated into 00, B into 01, … , and J into 09. Then, we concatenate these two-digit numbers into strings of digits. Next, we divide this string into equally sized blocks of $2N$ digits, where $2N$ is the largest even number such that the number $2525 \ldots 25$ with $2N$ digits does not exceed n. (When necessary, we pad the plaintext message with dummy Xs to make the last block the same size as all other blocks.)

After these steps, we have translated the plaintext message M into a sequence of integers $m_1, m_2, \ldots, m_k$ for some integer k. Encryption proceeds by transforming each block m_i to a ciphertext block c_i. This is done using the function

$$c = m^e \bmod n.$$

(To perform the encryption, we use an algorithm for fast modular exponentiation, such as Algorithm 5 in Section 4.2.) We leave the encrypted message as blocks of numbers and send these to the intended recipient. Because the RSA cryptosystem encrypts blocks of characters into blocks of characters, it is a block cipher.

Example 8 illustrates how RSA encryption is performed. For practical reasons we use small primes p and q in this example, rather than primes with 300 or more digits. Although the cipher described in this example is not secure, it does illustrate the techniques used in the RSA cipher.

EXAMPLE 8 Encrypt the message STOP using the RSA cryptosystem with key (2537, 13). Note that $2537 = 43 \cdot 59$, $p = 43$ and $q = 59$ are primes, and

$$\gcd(e, (p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1.$$

Links

Attribution: The Royal Society

CLIFFORD COCKS (BORN 1950) Clifford Cocks, born in Cheshire, England, was a talented mathematics student. He attended the Manchester Grammar School. In 1968 he won a silver medal at the International Mathematical Olympiad. Cocks attended King's College, Cambridge, studying mathematics. He also spent a short time at Oxford University working in number theory. In 1973 he decided not to complete his graduate work, instead taking a mathematical job at the Government Communications Headquarters (GCHQ) of British intelligence. Two months after joining GCHQ, Cocks learned about public key cryptography from an internal GCHQ report written by James Ellis. Cocks used his number theory knowledge to invent what is now called the RSA cryptosystem. He quickly realized that a public key cryptosystem could be based on the difficulty of reversing the process of multiplying two large primes. In 1997 he was allowed to reveal declassified GCHQ internal documents describing his discovery. Cocks is also known for his invention of a secure identity-based encryption scheme, which uses information about a user's identity as a public key. In 2001, Cocks became the Chief Mathematician at GCHQ. He has also set up the Heilbronn Institute for Mathematical Research, a partnership between GCHQ and the University of Bristol.

Solution: To encrypt, we first translate the letters in STOP into their numerical equivalents. We then group these numbers into blocks of four digits (because $2525 < 2537 < 252525$), to obtain

$$1819 \quad 1415.$$

We encrypt each block using the mapping

$$c = m^{13} \bmod 2537.$$

Computations using fast modular multiplication show that $1819^{13} \bmod 2537 = 2081$ and $1415^{13} \bmod 2537 = 2182$. The encrypted message is 2081 2182. ◀

4.6.6 RSA Decryption

The plaintext message can be quickly recovered from a ciphertext message when the decryption key d, an inverse of e modulo $(p-1)(q-1)$, is known. [Such an inverse exists because $\gcd(e, (p-1)(q-1)) = 1$.] To see this, note that if $de \equiv 1 \pmod{(p-1)(q-1)}$, there is an integer k such that $de = 1 + k(p-1)(q-1)$. It follows that

$$c^d \equiv (m^e)^d = m^{de} = m^{1+k(p-1)(q-1)} \pmod{n}.$$

By Fermat's little theorem [assuming that $\gcd(m, p) = \gcd(m, q) = 1$, which holds except in rare cases, which we cover in Exercise 28], it follows that $m^{p-1} \equiv 1 \pmod{p}$ and $m^{q-1} \equiv 1 \pmod{q}$. Consequently,

$$c^d \equiv m \cdot (m^{p-1})^{k(q-1)} \equiv m \cdot 1 = m \pmod{p}$$

Links

Courtesy of Ronald L. Rivest

RONALD RIVEST (BORN 1948) Ronald Rivest received a B.A. from Yale in 1969 and his Ph.D. in computer science from Stanford in 1974. Rivest is a computer science professor at M.I.T. and was a cofounder of RSA Data Security, which held the patent on the RSA cryptosystem that he invented together with Adi Shamir and Leonard Adleman. Areas that Rivest has worked in besides cryptography include machine learning, VLSI design, and computer algorithms. He is a coauthor of a popular text on algorithms ([CoLeRiSt09]).

©The Asahi Shimbun via Getty Images

ADI SHAMIR (BORN 1952) Adi Shamir was born in Tel Aviv, Israel. His undergraduate degree is from Tel Aviv University (1972) and his Ph.D. is from the Weizmann Institute of Science (1977). Shamir was a research assistant at the University of Warwick and an assistant professor at M.I.T. He is currently a professor in the Applied Mathematics Department at the Weizmann Institute and leads a group studying computer security. Shamir's contributions to cryptography, besides the RSA cryptosystem, include cracking knapsack cryptosystems, cryptanalysis of the Data Encryption Standard (DES), and the design of many cryptographic protocols.

Courtesy of Leonard Adleman

LEONARD ADLEMAN (BORN 1945) Leonard Adleman was born in San Francisco, California. He received a B.S. in mathematics (1968) and his Ph.D. in computer science (1976) from the University of California, Berkeley. Adleman was a member of the mathematics faculty at M.I.T. from 1976 until 1980, where he was a coinventor of the RSA cryptosystem, and in 1980 he took a position in the computer science department at the University of Southern California (USC). He was appointed to a chaired position at USC in 1985. Adleman has worked on computer security, computational complexity, immunology, and molecular biology. He invented the term "computer virus." Adleman's recent work on DNA computing has sparked great interest. He was a technical adviser for the movie *Sneakers*, in which computer security played an important role.

and

$$c^d \equiv m \cdot (m^{q-1})^{k(p-1)} \equiv m \cdot 1 = m \pmod{q}.$$

Because $\gcd(p, q) = 1$, it follows by the Chinese remainder theorem that

$$c^d \equiv m \pmod{pq}.$$

Example 9 illustrates how to decrypt messages sent using the RSA cryptosystem.

EXAMPLE 9 We receive the encrypted message 0981 0461. What is the decrypted message if it was encrypted using the RSA cipher from Example 8?

Solution: The message was encrypted using the RSA cryptosystem with $n = 43 \cdot 59$ and exponent 13. As Exercise 2 in Section 4.4 shows, $d = 937$ is an inverse of 13 modulo $42 \cdot 58 = 2436$. We use 937 as our decryption exponent. Consequently, to decrypt a block c, we compute

$$m = c^{937} \textbf{ mod } 2537.$$

To decrypt the message, we use the fast modular exponentiation algorithm to compute 0981^{937} **mod** $2537 = 0704$ and 0461^{937} **mod** $2537 = 1115$. Consequently, the numerical version of the original message is 0704 1115. Translating this back to English letters, we see that the message is HELP. ◀

4.6.7 RSA as a Public Key System

Links

Why is the RSA cryptosystem suitable for public key cryptography? First, it is possible to rapidly construct a public key by finding two large primes p and q, each with more than 300 digits, and to find an integer e relatively prime to $(p - 1)(q - 1)$. When we know the factorization of the modulus n, that is, when we know p and q, we can quickly find an inverse d of e modulo $(p - 1)(q - 1)$. [This is done by using the Euclidean algorithm to find Bézout coefficients s and t for d and $(p - 1)(q - 1)$, which shows that the inverse of d modulo $(p - 1)(q - 1)$ is s **mod** $(p - 1)(q - 1)$.] Knowing d lets us decrypt messages sent using our key. However, no method is known to decrypt messages that is not based on finding a factorization of n, or that does not also lead to the factorization of n.

Factorization is believed to be a difficult problem, as opposed to finding large primes p and q, which can be done quickly. The most efficient factorization methods known (as of 2017) require billions of years to factor 600-digit integers. Consequently, when p and q are 300-digit primes, it is believed that messages encrypted using $n = pq$ as the modulus cannot be decrypted in a reasonable time unless the primes p and q are known.

Although no polynomial-time algorithm is known for factoring large integers, active research is under way to find new ways to efficiently factor integers. Integers that were thought, as recently as several years ago, to be far too large to be factored in a reasonable amount of time can now be factored routinely. Integers with more than 230 digits have been factored using team efforts. When new factorization techniques are found, it will be necessary to use larger primes to ensure the secrecy of messages. Unfortunately, messages that were considered secure earlier can be saved and subsequently decrypted by unintended recipients when it becomes feasible to factor the $n = pq$ in the key used for RSA encryption. (Note that the RSA system will be insecure once quantum computing is available.)

The RSA method is now widely used. However, the most commonly used cryptosystems are private key cryptosystems. The use of public key cryptography, via the RSA system, is growing. Nevertheless, there are applications that use both private key and public key systems. For example, a public key cryptosystem, such as RSA, can be used to distribute private keys

to pairs of individuals when they wish to communicate. These people then use a private key system for the encryption and decryption of messages.

4.6.8 Cryptographic Protocols

So far we have shown how cryptography can be used to make messages secure. However, there are many other important applications of cryptography. Among these applications are **cryptographic protocols**, which are exchanges of messages carried out by two or more parties to achieve a particular security goal. In particular, we will show how cryptography can be used to allow two people to exchange a secret key over an insecure communication channel. We will also show how cryptography can be used to send signed secret messages so that the recipient can be sure that the message came from the purported sender. We refer the reader to [St05] for thorough discussions of a variety of cryptographic protocols.

KEY EXCHANGE We now discuss a protocol that two parties can use to exchange a secret key over an insecure communications channel without having shared any information in the past. Generating a key that two parties can share is important for many applications of cryptography. For example, for two people to send secure messages to each other using a private key cryptosystem they need to share a common key. The protocol we will describe is known as the **Diffie-Hellman key agreement protocol**, after Whitfield Diffie and Martin Hellman, who described it in 1976. However, this protocol was invented in 1974 by Malcolm Williamson in secret work at the British GCHQ. It was not until 1997 that his discovery was made public.

Suppose that Alice and Bob want to share a common key. The protocol follows these steps, where the computations are done in $\mathbf{Z}_p$.

(1) Alice and Bob agree to use a prime p and a primitive root a of p.

(2) Alice chooses a secret integer k_1 and sends $a^{k_1} \bmod p$ to Bob.

(3) Bob chooses a secret integer k_2 and sends $a^{k_2} \bmod p$ to Alice.

(4) Alice computes $(a^{k_2})^{k_1} \bmod p$.

(5) Bob computes $(a^{k_1})^{k_2} \bmod p$.

At the end of this protocol, Alice and Bob have computed their shared key, namely,

$$(a^{k_2})^{k_1} \bmod p = (a^{k_1})^{k_2} \bmod p.$$

To analyze the security of this protocol, note that the messages sent in steps (1), (2), and (3) are not assumed to be sent securely. We can even assume that these communications were in the clear and that their contents are public information. So, p, a, $a^{k_1} \bmod p$, and $a^{k_2} \bmod p$ are assumed to be public information. The protocol ensures that k_1, k_2, and the common key $(a^{k_2})^{k_1} \bmod p = (a^{k_1})^{k_2} \bmod p$ are kept secret. To find the secret information from this public information requires that an adversary solves instances of the discrete logarithm problem, because the adversary would need to find k_1 and k_2 from $a^{k_1} \bmod p$ and $a^{k_2} \bmod p$, respectively. Furthermore, no other method is known for finding the shared key using just the public information. We have remarked that this is thought to be computationally infeasible when p and a are sufficiently large. With the computing power available now, this system is considered unbreakable when p has more than 300 decimal digits and k_1 and k_2 have more than 100 decimal digits each.

DIGITAL SIGNATURES Not only can cryptography be used to secure the confidentiality of a message, but it also can be used so that the recipient of the message knows that it came from the person they think it came from. We first show how a message can be sent so that a recipient of the message will be sure that the message came from the purported sender of the message. In particular, we can show how this can be accomplished using the RSA cryptosystem to apply a **digital signature** to a message.

Suppose that Alice's RSA public key is (n, e) and her private key is d. Alice encrypts a plaintext message x using the encryption function $E_{(n,e)}(x) = x^e \bmod n$. She decrypts a ciphertext message y using the decryption function $D_{(n,e)} = x^d \bmod n$. Alice wants to send the message M so that everyone who receives the message knows that it came from her. Just as in RSA encryption, she translates the letters into their numerical equivalents and splits the resulting string into blocks $m_1, m_2, \ldots, m_k$ such that each block is the same size, which is as large as possible so that $0 \leq m_i \leq n$ for $i = 1, 2, ..., k$. She then applies her *decryption function* $D_{(n,e)}$ to each block, obtaining $D_{n,e}(m_i)$, $i = 1, 2, \ldots, k$. She sends the result to all intended recipients of the message.

When a recipient receives her message, they apply Alice's encryption function $E_{(n,e)}$ to each block, which everyone has available because Alice's key (n, e) is public information. The result is the original plaintext block because $E_{(n,e)}(D_{(n,e)}(x)) = x$. So, Alice can send her message to as many people as she wants and by signing it in this way, every recipient can be sure it came from Alice. Example 10 illustrates this protocol.

EXAMPLE 10 Suppose Alice's public RSA cryptosystem key is the same as in Example 8. That is, $n = 43 \cdot 59 = 2537$ and $e = 13$. Her decryption key is $d = 937$, as described in Example 9. She wants to send the message "MEET AT NOON" to her friends so that they are sure it came from her. What should she send?

Solution: Alice first translates the message into blocks of digits, obtaining 1204 0419 0019 1314 1413 (as the reader should verify). She then applies her decryption transformation $D_{(2537,13)}(x) = x^{937} \bmod 2537$ to each block. Using fast modular exponentiation (with the help of a computational aid), she finds that $1204^{937} \bmod 2537 = 817$, $419^{937} \bmod 2537 = 555$, $19^{937} \bmod 2537 = 1310$, $1314^{937} \bmod 2537 = 2173$, and $1413^{937} \bmod 2537 = 1026$.

So, the message she sends, split into blocks, is 0817 0555 1310 2173 1026. When one of her friends gets this message, they apply her encryption transformation $E_{(2537,13)}$ to each block. When they do this, they obtain the blocks of digits of the original message, which they translate back to English letters. ◀

We have shown that signed messages can be sent using the RSA cryptosystem. We can extend this by sending signed secret messages. To do this, the sender applies RSA encryption using the publicly known encryption key of an intended recipient to each block that was encrypted using the sender's decryption transformation. The recipient then first applies his private decryption transformation and then the sender's public encryption transformation. (Exercise 32 asks for this protocol to be carried out.)

4.6.9 Homomorphic Encryption

A cryptosystem, such as RSA, can be used to encrypt files to keep them secret. Today, many users store encrypted files in the **cloud**, where they reside on remote computers. It is often necessary to run programs using data from files stored on the cloud. These data are vulnerable to users with access to the remote computer where our data are stored if we run programs on the cloud without downloading these data. These data are also vulnerable to eavesdroppers when

we download files, run programs on our computer, and upload the output to the cloud. Could we avoid these vulnerabilities by just running programs on encrypted data? Although this seems farfetched at first blush, in 1979, soon after RSA was introduced, the question was proposed whether there was a cryptosystem that allowed arbitrary computations to be run on encrypted data that would produce the encryption of the unencrypted output produced by this unencrypted input. For such a cryptosystem, it would not be necessary to decrypt input data because the program could be run on a remote system without disclosing either the input or the output. So, the search began for a **fully homomorphic cryptosystem** that allows arbitrary computations to be run remotely on encrypted data.

Before we discuss progress towards fully homomorphic encryption, we will show that the RSA cryptosytem is not fully homomorphic, although it allows some computations to be done on encrypted data.

EXAMPLE 11 **RSA is Partially Homomorphic** Let (n, e) be a public key for the RSA cryptosystem and suppose that M_1 and M_2 are plaintext messages, so that $0 \leq M_1 < n$ and $0 \leq M_2 < n$. Then

$$\begin{aligned} E_{(n,e)}(M_1)E_{(n,e)}(M_2) \bmod m &= (M_1^e \bmod m \cdot M_2^e \bmod m) \bmod m \\ &= (M_1M_2)^e \bmod m = E_{(n,e)}(M_1M_2). \end{aligned}$$

From this equation we see that $E_{(n,e)}(M_1) \cdot_n E_{(n,e)}(M_2) = E_{(n,e)}(M_1M_2)$ in $\mathbf{Z}_n$. Because of this, we say that RSA is **multiplicatively homomorphic**. When we have encrypted using RSA, we can carry out multiplications without first decrypting because the encryption of the product of two plaintexts equals the product of their encryptions.

However, it is not true that $E_{(n,e)}(M_1) +_n E_{(n,e)}(M_2) = E_{(n,e)}(M_1 + M_2)$ for all M_1 and M_2 in $\mathbf{Z}_n$. (For instance, this is easy to see when $M_2 = 1$.) That is, when we have encrypted with RSA, we cannot add the encryptions of two numbers to obtain the encryption of their sum. Moreover, there is no method known for determining $E_{(n,e)}(M1 + M2)$ from $E_{(n,e)}(M1)$ and $E_{(n,e)}(M2)$ without decrypting these two ciphertexts. We say that RSA is not **additively homomorphic**. Because it is multiplicatively homomorphic, but not additively homomorphic, RSA is **partially homomorphic**. ◀

In 2009, Craig Gentry described the first fully homomorphic cryptosystem, based on what is known as lattice-based cryptography. Unfortunately, no practical fully homomorphic cryptosystems have yet been developed, because all require extremely large amounts of processing and memory. However, there is hope that new advances will lead to practical fully homomorphic cryptosystems in the not so distant future.

Links

Source: John D. & Catherine T. MacArthur Foundation

CRAIG B. GENTRY (BORN 1972) Craig B. Gentry received his B.S. from Duke University in 1993 and his J.D. from Harvard University Law School in 1998. For two years he worked as an intellectual property lawyer. From 2000 to 2005 he was a senior research engineer at NTT DoCoMo USA Labs. He decided to return to school, and in 2009 he received a Ph.D. in computer science from Stanford University. Gentry joined the Cryptography Research Group of the IBM Watson Research Center in 2009 where he currently works.

Gentry's invention of a fully homomorphic scheme, which resolved an open problem proposed in 1978, earned him the 2010 ACM Grace Murray Hopper Award. In 2013, Gentry with others constructed the first cryptographic multilinear maps, using them to construct the first cryptographic program obfuscation schemes, which many had thought might not exist. Gentry's work on fully homomorphic cryptography and on cryptographic multilinear maps is based on lattice-based cryptography, which cannot be broken by quantum computing, unlike the RSA cryptosystem. Gentry, along with his colleagues, advanced the area of verifiable computation, which allows a computer to offload the computation of functions to other computers while maintaining verifiable results. In 2014, Gentry was awarded a MacArthur Fellowship (also known as a Genuis Grant).

Exercises

1. Encrypt the message DO NOT PASS GO by translating the letters into numbers, applying the given encryption function, and then translating the numbers back into letters.

a) $f(p) = (p + 3)$ **mod** 26 (the Caesar cipher)
b) $f(p) = (p + 13)$ **mod** 26
c) $f(p) = (3p + 7)$ **mod** 26

2. Encrypt the message STOP POLLUTION by translating the letters into numbers, applying the given encryption function, and then translating the numbers back into letters.

a) $f(p) = (p + 4)$ **mod** 26
b) $f(p) = (p + 21)$ **mod** 26
c) $f(p) = (17p + 22)$ **mod** 26

3. Encrypt the message WATCH YOUR STEP by translating the letters into numbers, applying the given encryption function, and then translating the numbers back into letters.

a) $f(p) = (p + 14)$ **mod** 26
b) $f(p) = (14p + 21)$ **mod** 26
c) $f(p) = (-7p + 1)$ **mod** 26

4. Decrypt these messages that were encrypted using the Caesar cipher.

a) EOXH MHDQV
b) WHVW WRGDB
c) HDW GLP VXP

5. Decrypt these messages encrypted using the shift cipher $f(p) = (p + 10)$ **mod** 26.

a) CEBBOXNOB XYG
b) LO WI PBSOXN
c) DSWO PYB PEX

6. Suppose that when a long string of text is encrypted using a shift cipher $f(p) = (p + k)$ **mod** 26, the most common letter in the ciphertext is X. What is the most likely value for k, assuming that the distribution of letters in the text is typical of English text?

7. Suppose that when a string of English text is encrypted using a shift cipher $f(p) = (p + k)$ **mod** 26, the resulting ciphertext is DY CVOOZ ZOBMRKXMO DY NBOKW. What was the original plaintext string?

8. Suppose that the ciphertext DVE CFMV KF NFEUVI, REU KYRK ZJ KYV JVVU FW JTZVETV was produced by encrypting a plaintext message using a shift cipher. What is the original plaintext?

9. Suppose that the ciphertext ERC WYJJMGMIRXPC EHZERGIH XIGLRSPSKC MW MRHMWXMRKYMWLEFPI JVSQ QEKMG was produced by encrypting a plaintext message using a shift cipher. What is the original plaintext?

10. Determine whether there is a key for which the enciphering function for the shift cipher is the same as the deciphering function.

11. What is the decryption function for an affine cipher if the encryption function is $c = (15p + 13)$ **mod** 26?

***12.** Find all pairs of integers keys (a, b) for affine ciphers for which the encryption function $c = (ap + b)$ **mod** 26 is the same as the corresponding decryption function.

13. Suppose that the most common letter and the second most common letter in a long ciphertext produced by encrypting a plaintext using an affine cipher $f(p) = (ap + b)$ **mod** 26 are Z and J, respectively. What are the most likely values of a and b?

14. Encrypt the message GRIZZLY BEARS using blocks of five letters and the transposition cipher based on the permutation of $\{1, 2, 3, 4, 5\}$ with $\sigma(1) = 3$, $\sigma(2) = 5$, $\sigma(3) = 1$, $\sigma(4) = 2$, and $\sigma(5) = 4$. For this exercise, use the letter X as many times as necessary to fill out the final block of fewer then five letters.

15. Decrypt the message EABW EFRO ATMR ASIN, which is the ciphertext produced by encrypting a plaintext message using the transposition cipher with blocks of four letters and the permutation σ of $\{1, 2, 3, 4\}$ defined by $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 4$, and $\sigma(4) = 2$.

***16.** Suppose that you know that a ciphertext was produced by encrypting a plaintext message with a transposition cipher. How might you go about breaking it?

17. Suppose you have intercepted a ciphertext message and when you determine the frequencies of letters in this message, you find the frequencies are similar to the frequency of letters in English text. Which type of cipher do you suspect was used?

The **Vigenère cipher** is a block cipher, with a key that is a string of letters with numerical equivalents $k_1 k_2 \dots k_m$, where $k_i \in \mathbf{Z}_{26}$ for $i = 1, 2, \dots, m$. Suppose that the numerical equivalents of the letters of a plaintext block are $p_1 p_2 \dots p_m$. The corresponding numerical ciphertext block is $(p_1 + k_1)$ **mod** 26 $(p_2 + k_2)$ **mod** 26 $\dots$ $(p_m + k_m)$ **mod** 26. Finally, we translate back to letters. For example, suppose that the key string is RED, with numerical equivalents 17 4 3. Then, the plaintext ORANGE, with numerical equivalents 14 17 00 13 06 04, is encrypted by first splitting it into two blocks 14 17 00 and 13 06 04. Then, in each block we shift the first letter by 17, the second by 4, and the third by 3. We obtain 5 21 03 and 04 10 07. The cipherext is FVDEKH.

18. Use the Vigenère cipher with key BLUE to encrypt the message SNOWFALL.

19. The ciphertext OIKYWVHBX was produced by encrypting a plaintext message using the Vigenère cipher with key HOT. What is the plaintext message?

20. Express the Vigenère cipher as a cryptosystem.

To break a Vigenère cipher by recovering a plaintext message from the ciphertext message without having the key, the first step is to figure out the length of the key string. The second

step is to figure out each character of the key string by determining the corresponding shift. Exercises 21 and 22 deal with these two aspects.

21. Suppose that when a long string of text is encrypted using a Vigenère cipher, the same string is found in the ciphertext starting at several different positions. Explain how this information can be used to help determine the length of the key.

22. Once the length of the key string of a Vigènere cipher is known, explain how to determine each of its characters. Assume that the plaintext is long enough so that the frequency of its letters is reasonably close to the frequency of letters in typical English text.

***23.** Show that we can easily factor n when we know that n is the product of two primes, p and q, and we know the value of $(p-1)(q-1)$.

In Exercises 24–27 first express your answers without computing modular exponentiations. Then use a computational aid to complete these computations.

24. Encrypt the message ATTACK using the RSA system with $n = 43 \cdot 59$ and $e = 13$, translating each letter into integers and grouping together pairs of integers, as done in Example 8.

25. Encrypt the message UPLOAD using the RSA system with $n = 53 \cdot 61$ and $e = 17$, translating each letter into integers and grouping together pairs of integers, as done in Example 8.

26. What is the original message encrypted using the RSA system with $n = 53 \cdot 61$ and $e = 17$ if the encrypted message is 3185 2038 2460 2550? (To decrypt, first find the decryption exponent d, which is the inverse of $e = 17$ modulo $52 \cdot 60$.)

27. What is the original message encrypted using the RSA system with $n = 43 \cdot 59$ and $e = 13$ if the encrypted message is 0667 1947 0671? (To decrypt, first find the decryption exponent d which is the inverse of $e = 13$ modulo $42 \cdot 58$.)

***28.** Suppose that (n, e) is an RSA encryption key, with $n = pq$, where p and q are large primes and $\gcd(e, (p-1)(q-1)) = 1$. Furthermore, suppose that d is an inverse of e modulo $(p-1)(q-1)$. Suppose that $C \equiv M^e \pmod{pq}$. In the text we showed that RSA decryption, that is, the congruence $C^d \equiv M \pmod{pq}$ holds when $\gcd(M, pq) = 1$. Show that this decryption congruence also holds when $\gcd(M, pq) > 1$. [*Hint:* Use congruences modulo p and modulo q and apply the Chinese remainder theorem.]

29. Describe the steps that Alice and Bob follow when they use the Diffie-Hellman key exchange protocol to generate a shared key. Assume that they use the prime $p = 23$ and take $a = 5$, which is a primitive root of 23, and that Alice selects $k_1 = 8$ and Bob selects $k_2 = 5$. (You may want to use some computational aid.)

30. Describe the steps that Alice and Bob follow when they use the Diffie-Hellman key exchange protocol to generate a shared key. Assume that they use the prime $p = 101$ and take $a = 2$, which is a primitive root of 101, and that Alice selects $k_1 = 7$ and Bob selects $k_2 = 9$. (You may want to use some computational aid.)

In Exercises 31–32 suppose that Alice and Bob have these public keys and corresponding private keys: $(n_{\text{Alice}}, e_{\text{Alice}}) = (2867, 7) = (61 \cdot 47, 7)$, $d_{\text{Alice}} = 1183$ and $(n_{\text{Bob}}, e_{\text{Bob}}) = (3127, 21) = (59 \cdot 53, 21)$, $d_{\text{Bob}} = 1149$. First express your answers without carrying out the calculations. Then, using a computational aid, if available, perform the calculation to get the numerical answers.

31. Alice wants to send to all her friends, including Bob, the message "SELL EVERYTHING" so that he knows that she sent it. What should she send to her friends, assuming she signs the message using the RSA cryptosystem.

32. Alice wants to send to Bob the message "BUY NOW" so that he knows that she sent it and so that only Bob can read it. What should she send to Bob, assuming she signs the message and then encrypts it using Bob's public key?

33. We describe a basic key exchange protocol using private key cryptography upon which more sophisticated protocols for key exchange are based. Encryption within the protocol is done using a private key cryptosystem (such as AES) that is considered secure. The protocol involves three parties, Alice and Bob, who wish to exchange a key, and a trusted third party Cathy. Assume that Alice has a secret key k_{Alice} that only she and Cathy know, and Bob has a secret key k_{Bob} which only he and Cathy know. The protocol has three steps:

(i) Alice sends the trusted third party Cathy the message "request a shared key with Bob" encrypted using Alice's key k_{Alice}.

(ii) Cathy sends back to Alice a key $k_{\text{Alice,Bob}}$, which she generates, encrypted using the key k_{Alice}, followed by this same key $k_{\text{Alice,Bob}}$, encrypted using Bob's key, k_{Bob}.

(iii) Alice sends to Bob the key $k_{\text{Alice,Bob}}$ encrypted using k_{Bob}, known only to Bob and to Cathy.

Explain why this protocol allows Alice and Bob to share the secret key $k_{\text{Alice,Bob}}$, known only to them and to Cathy.

The **Paillier cryptosystem** is a public key cryptosystem described in 1999 by P. Paillier, used in some electronic voting systems. A public key (n, g) and a corresponding private key (p, q) are created by randomly selecting primes p and q so that $\gcd(pq, \lambda) = 1$, where $\lambda = (p-1)(q-1)$, and a nonzero element g of $\mathbf{Z}_{n^2}$ with $\gcd((g^{\lambda} \bmod n^2 - 1)/n, n) = 1$. To encrypt a message $m \in \mathbf{Z}_n$ a random nonzero element r of $\mathbf{Z}_n$ is first selected, and then used to find $c = g^m r^n \bmod n^2$.

34. a) Show that we can take $p = 149$, $q = 179$, and $g = 5$ to generate a public key for the Paillier cryptosystem by checking that all the conditions required for these parameters hold, and find the public and private keys that are generated.

b) Find the ciphertext corresponding to the plaintext $m = 67$ where we choose $r = 81$.

35. Show that the Paillier cryptosystem is additively homomorphic.

Key Terms and Results

TERMS

$a \mid b$ (a divides b): there is an integer c such that $b = ac$

a and b are congruent modulo m: m divides $a - b$

modular arithmetic: arithmetic done modulo an integer $m \geq 2$

prime: an integer greater than 1 with exactly two positive integer divisors

composite: an integer greater than 1 that is not prime

Mersenne prime: a prime of the form $2^p - 1$, where p is prime

gcd(a, b) (greatest common divisor of a and b): the largest integer that divides both a and b

relatively prime integers: integers a and b such that $\gcd(a, b) = 1$

pairwise relatively prime integers: a set of integers with the property that every pair of these integers is relatively prime

lcm(a, b) (least common multiple of a and b): the smallest positive integer that is divisible by both a and b

a mod b: the remainder when the integer a is divided by the positive integer b

$a \equiv b \pmod{m}$ (a is congruent to b modulo m): $a - b$ is divisible by m

$n = (a_k a_{k-1} \ldots a_1 a_0)_b$: the base b representation of n

binary representation: the base 2 representation of an integer

octal representation: the base 8 representation of an integer

hexadecimal representation: the base 16 representation of an integer

linear combination of a and b with integer coefficients: an expression of the form $sa + tb$, where s and t are integers

Bézout coefficients of a and b: integers s and t such that the **Bézout identity** $sa + tb = \gcd(a, b)$ holds

inverse of a modulo m: an integer $\overline{a}$ such that $\overline{a}a \equiv 1 \pmod{m}$

linear congruence: a congruence of the form $ax \equiv b \pmod{m}$, where x is an integer variable

pseudoprime to the base b: a composite integer n such that $b^{n-1} \equiv 1 \pmod{n}$

Carmichael number: a composite integer n such that n is a pseudoprime to the base b for all positive integers b with $\gcd(b, n) = 1$

primitive root of a prime p: an integer r in $\mathbf{Z}_p$ such that every integer not divisible by p is congruent modulo p to a power of r

discrete logarithm of a to the base r modulo p: the integer e with $0 \leq e \leq p - 1$ such that $r^e \equiv a \pmod{p}$

encryption: the process of making a message secret

decryption: the process of returning a secret message to its original form

encryption key: a value that determines which of a family of encryption functions is to be used

shift cipher: a cipher that encrypts the plaintext letter p as $(p + k)$ **mod** m for an integer k

affine cipher: a cipher that encrypts the plaintext letter p as $(ap + b)$ **mod** m for integers a and b with $\gcd(a, 26) = 1$

character cipher: a cipher that encrypts characters one by one

block cipher: a cipher that encrypts blocks of characters of a fixed size

cryptanalysis: the process of recovering the plaintext from ciphertext without knowledge of the encryption method, or with knowledge of the encryption method, but not the key

cryptosystem: a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ where $\mathcal{P}$ is the set of plaintext messages, $\mathcal{C}$ is the set of ciphertext messages, $\mathcal{K}$ is the set of keys, $\mathcal{E}$ is the set of encryption functions, and $\mathcal{D}$ is the set of decryption functions

private key encryption: encryption where both encryption keys and decryption keys must be kept secret

public key encryption: encryption where encryption keys are public knowledge, but decryption keys are kept secret

RSA cryptosystem: the cryptosystem where $\mathcal{P}$ and $\mathcal{C}$ are both $\mathbf{Z}_{26}$, $\mathcal{K}$ is the set of pairs $k = (n, e)$ where $n = pq$ where p and q are large primes and e is a positive integer, $E_k(p) = p^e$ **mod** n, and $D_k(c) = c^d$ **mod** n where d is the inverse of e modulo $(p - 1)(q - 1)$

key exchange protocol: a protocol used for two parties to generate a shared key

digital signature: a method that a recipient can use to determine that the purported sender of a message actually sent the message

fully homomorphic cryptosystem: a cryptosystem that allows arbitrary computations to be run on encrypted data so that the ouptut is the encryption of the unencrypted output of the unencrypted input

RESULTS

division algorithm: Let a and d be integers with d positive. Then there are unique integers q and r with $0 \leq r < d$ such that $a = dq + r$.

Let b be an integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$.

The algorithm for finding the base b expansion of an integer (see Algorithm 1 in Section 4.2)

The conventional algorithms for addition and multiplication of integers (given in Section 4.2)

The fast modular exponentiation algorithm (see Algorithm 5 in Section 4.2)

Euclidean algorithm: for finding greatest common divisors by successively using the division algorithm (see Algorithm 1 in Section 4.3)

Bézout's theorem: If a and b are positive integers, then $\gcd(a, b)$ is a linear combination of a and b.

sieve of Eratosthenes: a procedure for finding all primes not exceeding a specified number n, described in Section 4.3

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.

If a and b are positive integers, then $ab = \gcd(a, b) \cdot \operatorname{lcm}(a, b)$.

If m is a positive integer and $\gcd(a, m) = 1$, then a has a unique inverse modulo m.

Chinese remainder theorem: A system of linear congruences modulo pairwise relatively prime integers has a unique solution modulo the product of these moduli.

Fermat's little theorem: If p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p}$.

Review Questions

1. Find 210 **div** 17 and 210 **mod** 17.
2. **a)** Define what it means for a and b to be congruent modulo 7.
   b) Which pairs of the integers $-11, -8, -7, -1, 0, 3$, and 17 are congruent modulo 7?
   c) Show that if a and b are congruent modulo 7, then $10a + 13$ and $-4b + 20$ are also congruent modulo 7.
3. Show that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$.
4. Describe a procedure for converting decimal (base 10) expansions of integers into hexadecimal expansions.
5. Convert $(1101\ 1001\ 0101\ 1011)_2$ to octal and hexadecimal representations.
6. Convert $(7206)_8$ and $(\text{A0EB})_{16}$ to a binary representation.
7. State the fundamental theorem of arithmetic.
8. **a)** Describe a procedure for finding the prime factorization of an integer.
   b) Use this procedure to find the prime factorization of 80,707.
9. **a)** Define the greatest common divisor of two integers.
   b) Describe at least three different ways to find the greatest common divisor of two integers. When does each method work best?
   c) Find the greatest common divisor of 1,234,567 and 7,654,321.
   d) Find the greatest common divisor of $2^3 3^5 5^7 7^9 11$ and $2^9 3^7 5^5 7^3 13$.
10. **a)** How can you find a linear combination (with integer coefficients) of two integers that equals their greatest common divisor?
    b) Express $\gcd(84, 119)$ as a linear combination of 84 and 119.
11. **a)** What does it mean for $\bar{a}$ to be an inverse of a modulo m?
    b) How can you find an inverse of a modulo m when m is a positive integer and $\gcd(a, m) = 1$?
    c) Find an inverse of 7 modulo 19.
12. **a)** How can an inverse of a modulo m be used to solve the congruence $ax \equiv b \pmod{m}$ when $\gcd(a, m) = 1$?
    b) Solve the linear congruence $7x \equiv 13 \pmod{19}$.
13. **a)** State the Chinese remainder theorem.
    b) Find the solutions to the system $x \equiv 1 \pmod{4}$, $x \equiv 2 \pmod{5}$, and $x \equiv 3 \pmod{7}$.
14. Suppose that $2^{n-1} \equiv 1 \pmod{n}$. Is n necessarily prime?
15. Use Fermat's little theorem to evaluate 9^{200} **mod** 19.
16. Explain how the check digit is found for a 10-digit ISBN.
17. Encrypt the message APPLES AND ORANGES using a shift cipher with key $k = 13$.
18. **a)** What is the difference between a public key and a private key cryptosystem?
    b) Explain why using shift ciphers is a private key system.
    c) Explain why the RSA cryptosystem is a public key system.
19. Explain how encryption and decryption are done in the RSA cryptosystem.
20. Describe how two parties can share a secret key using the Diffie-Hellman key exchange protocol.

Supplementary Exercises

1. The odometer on a car goes to up 100,000 miles. The present owner of a car bought it when the odometer read 43,179 miles. He now wants to sell it; when you examine the car for possible purchase, you notice that the odometer reads 89,697 miles. What can you conclude about how many miles he drove the car, assuming that the odometer always worked correctly?
2. **a)** Explain why n **div** 7 equals the number of complete weeks in n days.
   b) Explain why n **div** 24 equals the number of complete days in n hours.
3. Find four numbers congruent to 5 modulo 17.
4. Show that if a and d are positive integers, then there are integers q and r such that $a = dq + r$, where $-d/2 < r \leq d/2$.
5. * Show that if $ac \equiv bc \pmod{m}$, where a, b, c, and m are integers with $m > 2$, and $d = \gcd(m, c)$, then $a \equiv b \pmod{m/d}$.
6. Show that the sum of the squares of two odd integers cannot be the square of an integer.
7. Show that if $n^2 + 1$ is a perfect square, where n is an integer, then n is even.

8. Prove that there are no solutions in integers x and y to the equation $x^2 - 5y^2 = 2$. [*Hint:* Consider this equation modulo 5.]

9. Develop a test for divisibility of a positive integer n by 8 based on the binary expansion of n.

10. Develop a test for divisibility of a positive integer n by 3 based on the binary expansion of n.

11. Devise an algorithm for guessing a number between 1 and $2^n - 1$ by successively guessing each bit in its binary expansion.

12. Determine the complexity, in terms of the number of guesses, needed to determine a number between 1 and $2^n - 1$ by successively guessing the bits in its binary expansion.

13. Show that an integer is divisible by 9 if and only if the sum of its decimal digits is divisible by 9.

****14.** Show that if a and b are positive irrational numbers such that $1/a + 1/b = 1$, then every positive integer can be uniquely expressed as either $\lfloor ka \rfloor$ or $\lfloor kb \rfloor$ for some positive integer k.

15. Prove there are infinitely many primes by showing that $Q_n = n! + 1$ must have a prime factor greater than n whenever n is a positive integer.

16. Find a positive integer n for which $Q_n = n! + 1$ is not prime.

17. Use Dirichlet's theorem, which states there are infinitely many primes in every arithmetic progression $ak + b$ where $\gcd(a, b) = 1$, to show that there are infinitely many primes that have a decimal expansion ending with a 1.

18. Prove that if n is a positive integer such that the sum of the divisors of n is $n + 1$, then n is prime.

***19.** Show that every integer greater than 11 is the sum of two composite integers.

20. Find the five smallest consecutive composite integers.

21. Show that Goldbach's conjecture, which states that every even integer greater than 2 is the sum of two primes, is equivalent to the statement that every integer greater than 5 is the sum of three primes.

22. Find an arithmetic progression of length six beginning with 7 that contains only primes.

***23.** Prove that if $f(x)$ is a nonconstant polynomial with integer coefficients, then there is an integer y such that $f(y)$ is composite. [*Hint:* Assume that $f(x_0) = p$ is prime. Show that p divides $f(x_0 + kp)$ for all integers k. Obtain a contradiction of the fact that a polynomial of degree n, where $n > 1$, takes on each value at most n times.]

***24.** How many zeros are at the end of the binary expansion of $(100_{10})!$?

25. Use the Euclidean algorithm to find the greatest common divisor of 10,223 and 33,341.

26. How many divisions are required to find gcd(144, 233) using the Euclidean algorithm?

27. Find $\gcd(2n + 1, 3n + 2)$, where n is a positive integer. [*Hint:* Use the Euclidean algorithm.]

28. a) Show that if a and b are positive integers with $a \geq b$, then $\gcd(a, b) = a$ if $a = b$, $\gcd(a, b) = 2\gcd(a/2, b/2)$ if a and b are even, $\gcd(a, b) = \gcd(a/2, b)$ if a is even and b is odd, and $\gcd(a, b) = \gcd(a - b, b)$ if both a and b are odd.

b) Explain how to use part (a) to construct an algorithm for computing the greatest common divisor of two positive integers that uses only comparisons, subtractions, and shifts of binary expansions, without using any divisions.

c) Find gcd(1202, 4848) using this algorithm.

29. Adapt the proof that there are infinitely many primes (Theorem 3 in Section 4.3) to show that there are infinitely many primes in the arithmetic progression $6k + 5$, $k = 1, 2, \ldots$.

30. Explain why you cannot directly adapt the proof that there are infinitely many primes (Theorem 3 in Section 4.3) to show that there are infinitely many primes in the arithmetic progression $3k + 1$, $k = 1, 2, \ldots$.

31. Explain why you cannot directly adapt the proof that there are infinitely many primes (Theorem 3 in Section 4.3) to show that there are infinitely many primes in the arithmetic progression $4k + 1$, $k = 1, 2, \ldots$.

32. Show that if the smallest prime factor p of the positive integer n is larger than $\sqrt[3]{n}$, then n/p is prime or equal to 1.

A set of integers is called **mutually relatively prime** if the greatest common divisor of these integers is 1.

33. Determine whether the integers in each of these sets are mutually relatively prime.

a) 8, 10, 12 **b)** 12, 15, 25

c) 15, 21, 28 **d)** 21, 24, 28, 32

34. Find a set of four mutually relatively prime integers such that no two of them are relatively prime.

***35.** For which positive integers n is $n^4 + 4^n$ prime?

36. Show that the system of congruences $x \equiv 2 \pmod 6$ and $x \equiv 3 \pmod 9$ has no solutions.

37. Find all solutions of the system of congruences $x \equiv 4 \pmod 6$ and $x \equiv 13 \pmod{15}$.

***38. a)** Show that the system of congruences $x \equiv a_1 \pmod{m_1}$ and $x \equiv a_2 \pmod{m_2}$, where a_1, a_2, m_1, and m_2 are integers with $m_1 > 0$ and $m_2 > 0$, has a solution if and only if $\gcd(m_1, m_2) \mid a_1 - a_2$.

b) Show that if the system in part (a) has a solution, then it is unique modulo $\operatorname{lcm}(m_1, m_2)$.

39. Prove that 30 divides $n^9 - n$ for every nonnegative integer n.

40. Prove that $n^{12} - 1$ is divisible by 35 for every integer n for which $\gcd(n, 35) = 1$.

41. Show that if p and q are distinct prime numbers, then $p^{q-1} + q^{p-1} \equiv 1 \pmod{pq}$.

The check digit a_{13} for an ISBN-13 with initial digits $a_1 a_2 \ldots a_{12}$ is determined by the congruence $(a_1 + a_3 + \cdots + a_{13}) + 3(a_2 + a_4 + \cdots + a_{12}) \equiv 0 \pmod{10}$.

42. Determine whether each of these 13-digit numbers is a valid ISBN-13.
   a) 978-0-073-20679-1
   b) 978-0-45424-521-1
   c) 978-3-16-148410-0
   d) 978-0-201-10179-9

43. Show that the check digit of an ISBN-13 can always detect a single error.

44. Show that there are transpositions of two digits that are not detected by an ISBN-13.

A **routing transit number (RTN)** is a bank code used in the United States which appears on the bottom of checks. The most common form of an RTN has nine digits, where the last digit is a check digit. If $d_1 d_2 \ldots d_9$ is a valid RTN, the congruence $3(d_1 + d_4 + d_7) + 7(d_2 + d_5 + d_8) + (d_3 + d_6 + d_9) \equiv 0 \pmod{10}$ must hold.

45. Show that if $d_1 d_2 \ldots d_9$ is a valid RTN, then $d_9 = 7(d_1 + d_4 + d_7) + 3(d_2 + d_5 + d_8) + 9(d_3 + d_6) \bmod 10$. Furthermore, use this formula to find the check digit that follows the eight digits 11100002 in a valid RTN.

46. Show that the check digit of an RTN can detect all single errors and determine which transposition errors an RTN check digit can catch and which ones it cannot catch.

47. The encrypted version of a message is LJMKG MGMXF QEXMW. If it was encrypted using the affine cipher $f(p) = (7p + 10) \bmod 26$, what was the original message?

Autokey ciphers are ciphers where the nth letter of the plaintext is shifted by the numerical equivalent of the nth letter of a keystream. The keystream begins with a seed letter; its subsequent letters are constructed using either the plaintext or the ciphertext. When the plaintext is used, each character of the keystream, after the first, is the previous letter of the plaintext. When the ciphertext is used, each subsequent character of the keystream, after the first, is the previous letter of the ciphertext computed so far. In both cases, plaintext letters are encrypted by shifting each character by the numerical equivalent of the corresponding keystream letter.

48. Use the autokey cipher to encrypt the message NOW IS THE TIME TO DECIDE (ignoring spaces) using
   a) the keystream with seed X followed by letters of the plaintext.
   b) the keystream with seed X followed by letters of the ciphertext.

49. Use the autokey cipher to encrypt the message THE DREAM OF REASON (ignoring spaces) using
   a) the keystream with seed X followed by letters of the plaintext.
   b) the keystream with seed X followed by letters of the ciphertext.

Computer Projects

Write programs with these inputs and outputs.

1. Given integers n and b, each greater than 1, find the base b expansion of this integer.

2. Given the positive integers a, b, and m with $m > 1$, find $a^b \bmod m$.

3. Given a positive integer, find the Cantor expansion of this integer (see the preamble to Exercise 54 of Section 4.2).

4. Given a positive integer, determine whether it is prime using trial division.

5. Given a positive integer, find the prime factorization of this integer.

6. Given two positive integers, find their greatest common divisor using the Euclidean algorithm.

7. Given two positive integers, find their least common multiple.

8. Given positive integers a and b, find Bézout coefficients s and t of a and b.

9. Given relatively prime positive integers a and b, find an inverse of a modulo b.

10. Given n linear congruences modulo pairwise relatively prime moduli, find the simultaneous solution of these congruences modulo the product of these moduli.

11. Given a positive integer N, a modulus m, a multiplier a, an increment c, and a seed x_0, where $0 \le a < m$, $0 \le c < m$, and $0 \le x_0 < m$, generate the sequence of N pseudorandom numbers using the linear congruential generator $x_{n+1} = (ax_n + c) \bmod m$.

12. Given a set of identification numbers, use a hash function to assign them to memory locations where there are k memory locations.

13. Compute the check digit when given the first nine digits of an ISBN-10.

14. Given a message and a positive integer k less than 26, encrypt this message using the shift cipher with key k; and given a message encrypted using a shift cipher with key k, decrypt this message.

15. Given a message and positive integers a and b less than 26 with $\gcd(a, 26)$, encrypt this message using an affine cipher with key (a, b); and given a message encrypted using the affine cipher with key (a, b), decrypt this message, by first finding the decryption key and then applying the appropriate decryption transformation.

16. Find the original plaintext message from the ciphertext message produced by encrypting the plaintext message

using a shift cipher. Do this using a frequency count of letters in the ciphertext.

*17. Construct a valid RSA encryption key by finding two primes p and q with 200 digits each and an integer $e > 1$ relatively prime to $(p-1)(q-1)$.

18. Given a message and an integer $n = pq$ where p and q are odd primes and an integer $e > 1$ relatively prime to $(p-1)(q-1)$, encrypt the message using the RSA cryptosystem with key (n, e).

19. Given a valid RSA key (n, e), and the primes p and q with $n = pq$, find the associated decryption key d.

20. Given a message encrypted using the RSA cryptosystem with key (n, e) and the associated decryption key d, decrypt this message.

21. Generate a shared key using the Diffie-Hellman key exchange protocol.

22. Given the RSA public and private keys of two parties, send a signed secret message from one of the parties to the other.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Determine whether $2^p - 1$ is prime for each of the primes not exceeding 100.

2. Test a range of large Mersenne numbers $2^p - 1$ to determine whether they are prime. (You may want to use software from the GIMPS project.)

3. Determine whether $Q_n = p_1 p_2 \cdots p_n + 1$ is prime where $p_1, p_2, \ldots, p_n$ are the n smallest primes, for as many positive integers n as possible.

4. Look for polynomials in one variables whose values at long runs of consecutive integers are all primes.

5. Find as many primes of the form $n^2 + 1$ where n is a positive integer as you can. It is not known whether there are infinitely many such primes.

6. Find 10 different primes each with 100 digits.

7. How many primes are there less than 1,000,000, less than 10,000,000, and less than 100,000,000? Can you propose an estimate for the number of primes less than x where x is a positive integer?

8. Find a prime factor of each of 10 different 20-digit odd integers, selected at random. Keep track of how long it takes to find a factor of each of these integers. Do the same thing for 10 different 30-digit odd integers, 10 different 40-digit odd integers, and so on, continuing as long as possible.

9. Find all pseudoprimes to the base 2 that do not exceed 10,000.

Writing Projects

Respond to these with essays using outside sources.

1. Describe the Lucas–Lehmer test for determining whether a Mersenne number is prime. Discuss the progress of the GIMPS project in finding Mersenne primes using this test.

2. Explain how probabilistic primality tests are used in practice to produce extremely large numbers that are almost certainly prime. Do such tests have any potential drawbacks?

3. The question of whether there are infinitely many Carmichael numbers was solved recently after being open for more than 75 years. Describe the ingredients that went into the proof that there are infinitely many such numbers.

4. Summarize the current status of factoring algorithms in terms of their complexity and the size of numbers that can currently be factored. When do you think that it will be feasible to factor 200-digit numbers?

5. Describe the algorithms that are actually used by modern computers to add, subtract, multiply, and divide positive integers.

6. Describe the history of the Chinese remainder theorem. Describe some of the relevant problems posed in Chinese and Hindu writings and how the Chinese remainder theorem applies to them.

7. When are the numbers of a sequence truly random numbers, and not pseudorandom? What shortcomings have been observed in simulations and experiments in which pseudorandom numbers have been used? What are the properties that pseudorandom numbers can have that random numbers should not have?

8. Explain how a check digit is found for an International Bank Account Number (IBAN) and discuss the types of errors that can be found using this check digit.

9. Describe the Luhn algorithm for finding the check digit of a credit card number and discuss the types of errors that can be found using this check digit.

10. Show how a congruence can be used to tell the day of the week for any given date.

11. Describe how public key cryptography is being applied. Are the ways it is applied secure given the status of factoring algorithms? Will information kept secure using public key cryptography become insecure in the future?

12. Describe how public key cryptography can be used to produce signed secret messages so that the recipient is relatively sure the message was sent by the person expected to have sent it.

13. Describe the Rabin public key cryptosystem, explaining how to encrypt and how to decrypt messages and why it is suitable for use as a public key cryptosystem.

***14.** Explain why it would be unsuitable to use p, where p is a large prime, as the modulus for encryption in the RSA cryptosystem. That is, explain how someone could, without excessive computation, find a private key from the corresponding public key if the modulus were a large prime, rather than the product of two large primes.

15. Explain what is meant by a cryptographic hash function. What are the important properties such a function must have?

***16.** Explain the steps that Gentry used to construct a fully homomorphic cryptosystem.