CHAPTER

5 Functions

Prerequisites: Chapter 4

In this chapter we focus our attention on a special type of relation, a function, that plays an important role in mathematics, computer science, and many applications. We also define some functions used in computer science and examine the growth of functions.

LOOKING BACK

The origins of the notion of a function can be traced back to the great Italian philosopher, astronomer, and mathematician Galileo Galilei (1564–1642), who in the 1630s observed the relationship between two variables. The early work on functions in the second half of the seventeenth century concentrated on the study of special functions as curves. These included the power, exponential, logarithmic, and trigonometric functions. Gottfried Wilhelm Leibniz (1646–1716) was the first person to use the word *function* for a quantity whose value varies as a point moves on a curve. Leibniz was an extraordinary person who made brilliant contributions in a number of diverse areas, including logic, philosophy, law, metaphysics, religion, mathematics, diplomacy, and literature. He has often been called a "universal genius." Leibniz was born in Leipzig, and died in Hanover, both in Germany. Early in his career, he developed the foundations for what would later be called symbolic logic (which we discussed in Chapter 2). Leibniz began his study of advanced contemporary mathematics in 1672 at the age of 26. Three years later he discovered the Fundamental Theorem of Calculus independently of Newton, who had also discovered the same result. Indeed, a heated battle raged over a number of years between the supporters of Leibniz and Newton as to who had discovered calculus first. Today, both Newton and Leibniz are considered the fathers of calculus. It is quite surprising to learn what a visionary Leibniz was. In the 1670s he invented a mechanical calculator, known as the *Leibniz wheel*, capable of adding, subtracting, multiplying, and dividing. He almost envisioned the modern age of computing!

LOOKING BACK (Continued)

The commonly used notation for a function value, $f(x)$, is due to Leonhard Euler (1707–1783), who was born in Basel, Switzerland, and died in St. Petersburg, Russia. Euler is one of the greatest and most prolific mathematicians in history. After his death, it took nearly 50 years to publish all his papers and his collected works comprise more than 75 volumes. He was also able to carry out complex calculations in his head. During the last 17 years of his life, Euler was totally blind, but his mathematical output remained undiminished. Euler made significant contributions to many areas of mathematics and used mathematics to solve a wide variety of problems in the sciences.

Gottfried Wilhelm Leibniz

Leonhard Euler

5.1 Functions

In this section we define the notion of a function, a special type of relation. We study its basic properties and then discuss several special types of functions. A number of important applications of functions will occur in later sections of the book, so it is essential to get a good grasp of the material in this section.

Let A and B be nonempty sets. A **function** f from A to B, which is denoted $f: A \to B$, is a relation from A to B such that for all $a \in \text{Dom}(f)$, $f(a)$, the f-relative set of a, contains just one element of B. Naturally, if a is not in $\text{Dom}(f)$, then $f(a) = \emptyset$. If $f(a) = \{b\}$, it is traditional to identify the set $\{b\}$ with the element b and write $f(a) = b$. We will follow this custom, since no confusion results. The relation f can then be described as the set of pairs $\{(a, f(a)) \mid a \in \text{Dom}(f)\}$. Functions are also called **mappings** or **transformations**, since they can be geometrically viewed as rules that assign to each element $a \in A$ the unique element $f(a) \in B$ (see Figure 5.1). The element a is called an **argument** of the function f, and $f(a)$ is called the **value** of the function for the argument a and is also referred to as the **image** of a under f. Figure 5.1 is a schematic or pictorial display

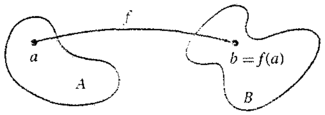


Figure 5.1

of our definition of a function, and we will use several other similar diagrams. They should not be confused with the digraph of the relation f, which we will not generally display.

EXAMPLE 1

Let $A = \{1, 2, 3, 4\}$ and $B = \{a, b, c, d\}$, and let

$$f = \{(1, a), (2, a), (3, d), (4, c)\}.$$

Here we have

$$\begin{aligned} f(1) &= a \\ f(2) &= a \\ f(3) &= d \\ f(4) &= c. \end{aligned}$$

Since each set $f(n)$ is a single value, f is a function.

Note that the element $a \in B$ appears as the second element of two different ordered pairs in f. This does not conflict with the definition of a function. Thus a function may take the same value at two different elements of A. ■

EXAMPLE 2

Let $A = \{1, 2, 3\}$ and $B = \{x, y, z\}$. Consider the relations

$$R = \{(1, x), (2, x)\} \quad \text{and} \quad S = \{(1, x), (1, y), (2, z), (3, y)\}.$$

The relation S is not a function since $S(1) = \{x, y\}$. The relation R is a function with $\text{Dom}(R) = \{1, 2\}$ and $\text{Ran}(R) = \{x\}$. ■

EXAMPLE 3

Let P be a computer program that accepts an integer as input and produces an integer as output. Let $A = B = Z$. Then P determines a relation f_P defined as follows: $(m, n) \in f_P$ means that n is the output produced by program P when the input is m.

It is clear that f_P is a function, since any particular input corresponds to a unique output. (We assume that computer results are reproducible; that is, they are the same each time the program is run.) ■

Example 3 can be generalized to a program with any set A of possible inputs and set B of corresponding outputs. In general, therefore, we may think of functions as **input-output** relations.

EXAMPLE 4

Let $A = \mathbb{R}$ be the set of real numbers, and let $p(x) = a_0 + a_1x + \cdots + a_nx^n$ be a real polynomial. Then p may be viewed as a relation on $\mathbb{R}$. For each r in $\mathbb{R}$ we determine the relative set $p(r)$ by substituting r into the polynomial. Then, since all relative sets $p(r)$ are known, the relation p is determined. Since a unique value is produced by this substitution, the relation p is actually a function. ■

If the formula defining the function does not make sense for all elements of A, then the domain of the function is taken to be the set of elements for A for which the formula does make sense.

In elementary mathematics, the *formula* (in the case of Example 4, the polynomial) is sometimes confused with the *function* it produces. This is not harmful, unless the student comes to expect a formula for every type of function.

Suppose that, in the preceding construction, we used a formula that produced more than one element in $p(x)$, for example, $p(x) = \pm\sqrt{x}$. Then the resulting relation would not be a function. For this reason, in older texts, relations were sometimes called multiple-valued functions.

EXAMPLE 5

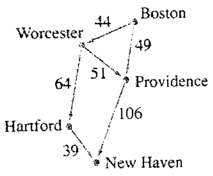


Figure 5.2

A **labeled digraph** is a digraph in which the vertices or the edges (or both) are labeled with information from a set. If V is the set of vertices and L is the set of labels of a labeled digraph, then the labeling of V can be specified to be a function $f: V \to L$, where, for each $v \in V$, $f(v)$ is the label we wish to attach to v. Similarly, we can define a labeling of the edges E as a function $g: E \to L$, where, for each $e \in E$, $g(e)$ is the label we wish to attach to e. An example of a labeled digraph is a map on which the vertices are labeled with the names of cities and the edges are labeled with the distances or travel times between the cities. Figure 5.2 shows an example of a labeled digraph. Another example is a flow chart of a program in which the vertices are labeled with the steps that are to be performed at that point in the program; the edges indicate the flow from one part of the program to another part. ■

EXAMPLE 6

Let $A = B = Z$ and let $f: A \to B$ be defined by

$$f(a) = a + 1, \quad \text{for } a \in A.$$

Here, as in Example 4, f is defined by giving a formula for the values $f(a)$. ■

EXAMPLE 7

Let $A = Z$ and let $B = \{0, 1\}$. Let $f: A \to B$ be found by

$$f(a) = \begin{cases} 0 & \text{if } a \text{ is even} \\ 1 & \text{if } a \text{ is odd.} \end{cases}$$

Then f is a function, since each set $f(a)$ consists of a single element. Unlike the situation in Examples 4 and 6, the elements $f(a)$ are not specified through an algebraic formula. Instead, a verbal description is given. ■

EXAMPLE 8

Let A be an arbitrary nonempty set. The **identity function on A**, denoted by 1_A, is defined by $1_A(a) = a$. ■

The reader may notice that 1_A is the relation we previously called Δ (see Section 4.4), which stands for the diagonal subset of $A \times A$. In the context of functions, the notation 1_A is preferred, since it emphasizes the input-output or functional nature of the relation. Clearly, if $A_1 \subseteq A$, then $1_A(A_1) = A_1$.

Suppose that $f: A \to B$ and $g: B \to C$ are functions. Then the composition of f and g, $g \circ f$ (see Section 4.7), is a relation. Let $a \in \text{Dom}(g \circ f)$. Then, by Theorem 6 of Section 4.7, $(g \circ f)(a) = g(f(a))$. Since f and g are functions, $f(a)$ consists of a single element $b \in B$, so $g(f(a)) = g(b)$. Since g is also a function, $g(b)$ contains just one element of C. Thus each set $(g \circ f)(a)$, for a in $\text{Dom}(g \circ f)$, contains just one element of C, so $g \circ f$ is a function. This is illustrated in Figure 5.3.

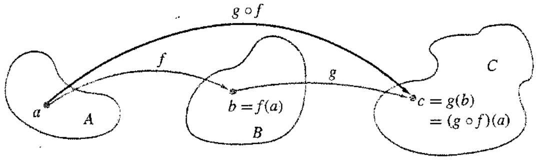


Figure 5.3

EXAMPLE 9

Let $A = B = Z$, and C be the set of even integers. Let $f: A \to B$ and $g: B \to C$ be defined by

$$f(a) = a + 1$$
$$g(b) = 2b.$$

Find $g \circ f$.

Solution

We have

$$(g \circ f)(a) = g(f(a)) = g(a + 1) = 2(a + 1).$$

Thus, if f and g are functions specified by giving formulas, then so is $g \circ f$ and the formula for $g \circ f$ is produced by substituting the formula for f into the formula for g. ■

■ Special Types of Functions

Let f be a function from A to B. Then we say that f is **everywhere defined** if $\text{Dom}(f) = A$. We say that f is **onto** if $\text{Ran}(f) = B$. Finally, we say that f is **one to one** if we cannot have $f(a) = f(a')$ for two distinct elements a and a' of A. The definition of one to one may be restated in the following equivalent form:

$$\text{If } f(a) = f(a'), \quad \text{then } a = a'.$$

The latter form is often easier to verify in particular examples.

EXAMPLE 10

Consider the function f defined in Example 1. Since $\text{Dom}(f) = A$, f is everywhere defined. On the other hand, $\text{Ran}(f) = \{a, c, d\} \neq B$; therefore, f is not onto. Since

$$f(1) = f(2) = a,$$

we can conclude that f is not one to one. ■

EXAMPLE 11

Consider the function f defined in Example 6. Which of the special properties, if any, does f possess?

Solution

Since the formula defining f makes sense for all integers, $\text{Dom}(f) = Z = A$, and so f is everywhere defined.

Suppose that

$$f(a) = f(a')$$

for a and a' in A. Then

$$a + 1 = a' + 1$$

so

$$a = a'.$$

Hence f is one to one.

To see if f is onto, let b be an arbitrary element of B. Can we find an element $a \in A$ such that $f(a) = b$? Since

$$f(a) = a + 1,$$

we need an element a in A such that

$$a + 1 = b.$$

Of course,

$$a = b - 1$$

will satisfy the desired equation since $b - 1$ is in A. Hence $\text{Ran}(f) = B$; therefore, f is onto. ■

EXAMPLE 12

Let $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2, b_3\}$, $C = \{c_1, c_2\}$, and $D = \{d_1, d_2, d_3, d_4\}$. Consider the following four functions, from A to B, A to D, B to C, and D to B, respectively.

(a) $f_1 = \{(a_1, b_2), (a_2, b_3), (a_3, b_1)\}$
(b) $f_2 = \{(a_1, d_2), (a_2, d_1), (a_3, d_4)\}$
(c) $f_3 = \{(b_1, c_2), (b_2, c_2), (b_3, c_1)\}$
(d) $f_4 = \{(d_1, b_1), (d_2, b_2), (d_3, b_1)\}$

Determine whether each function is one to one, whether each function is onto, and whether each function is everywhere defined.

Solution

(a) f_1 is everywhere defined, one to one, and onto.
(b) f_2 is everywhere defined and one to one, but not onto.
(c) f_3 is everywhere defined and onto, but is not one to one.
(d) f_4 is not everywhere defined, not one to one, and not onto. ■

If $f: A \to B$ is a one-to-one function, then f assigns to each element a of $\text{Dom}(f)$ an element $b = f(a)$ of $\text{Ran}(f)$. Every b in $\text{Ran}(f)$ is matched, in this way, with one and only one element of $\text{Dom}(f)$. For this reason, such an f is often called a **bijection** between $\text{Dom}(f)$ and $\text{Ran}(f)$. If f is also everywhere defined and onto, then f is called a **one-to-one correspondence between A and B.**

EXAMPLE 13

Let $\mathcal{R}$ be the set of all equivalence relations on a given set A, and let Π be the set of all partitions on A. Then we can define a function $f: \mathcal{R} \to \Pi$ as follows. For each equivalence relation R on A, let $f(R) = A/R$, the partition of A that corresponds to R. The discussion in Section 4.5 shows that f is a one-to-one correspondence between $\mathcal{R}$ and Π. ■

■ Invertible Functions

A function $f: A \to B$ is said to be **invertible** if its inverse relation, f^{-1}, is also a function. The next example shows that a function is not necessarily invertible.

EXAMPLE 14

Let f be the function of Example 1. Then

$$f^{-1} = \{(a, 1), (a, 2), (d, 3), (c, 4)\}.$$

We see that f^{-1} is not a function, since $f^{-1}(a) = \{1, 2\}$. ■

The following theorem is frequently used.

Theorem 1 Let $f: A \to B$ be a function.

(a) Then f^{-1} is a function from B to A if and only if f is one to one.

If f^{-1} is a function, then

(b) the function f^{-1} is also one to one.

(c) f^{-1} is everywhere defined if and only if f is onto.

(d) f^{-1} is onto if and only if f is everywhere defined.

Proof

(a) We prove the following equivalent statement.

$$f^{-1} \text{ is not a function if and only if } f \text{ is not one to one.}$$

Suppose first that f^{-1} is not a function. Then, for some b in B, $f^{-1}(b)$ must contain at least two distinct elements, a_1 and a_2. Then $f(a_1) = b = f(a_2)$, so f is not one to one.

Conversely, suppose that f is not one to one. Then $f(a_1) = f(a_2) = b$ for two distinct elements a_1 and a_2 of A. Thus $f^{-1}(b)$ contains both a_1 and a_2, so f^{-1} cannot be a function.

(b) Since $(f^{-1})^{-1}$ is the function f, part (a) shows that f^{-1} is one to one.

(c) Recall that $\text{Dom}(f^{-1}) = \text{Ran}(f)$. Thus $B = \text{Dom}(f^{-1})$ if and only if $B = \text{Ran}(f)$. In other words, f^{-1} is everywhere defined if and only if f is onto.

(d) Since $\text{Ran}(f^{-1}) = \text{Dom}(f)$, $A = \text{Dom}(f)$ if and only if $A = \text{Ran}(f^{-1})$. That is, f is everywhere defined if and only if f^{-1} is onto. ∎

As an immediate consequence of Theorem 1, we see that if f is a one-to-one correspondence between A and B, then f^{-1} is a one-to-one correspondence between B and A. Note also that if $f: A \to B$ is a one-to-one function, then the equation $b = f(a)$ is equivalent to $a = f^{-1}(b)$.

EXAMPLE 15

Consider the function f defined in Example 6. Since it is everywhere defined, one to one, and onto, f is a one-to-one correspondence between A and B. Thus f is invertible, and f^{-1} is a one-to-one correspondence between B and A. ■

EXAMPLE 16

Let $\mathbb{R}$ be the set of real numbers, and let $f: \mathbb{R} \to \mathbb{R}$ be defined by $f(x) = x^2$. Is f invertible?

Solution

We must determine whether f is one to one. Since

$$f(2) = f(-2) = 4,$$

we conclude that f is not one to one. Hence f is not invertible. ■

There are some useful results concerning the composition of functions. We summarize these in the following theorem.

Theorem 2 Let $f: A \to B$ be any function. Then

(a) $1_B \circ f = f$.

(b) $f \circ 1_A = f$.

If f is a one-to-one correspondence between A and B, then

(c) $f^{-1} \circ f = 1_A$.

(d) $f \circ f^{-1} = 1_B$.

Proof

(a) $(1_B \circ f)(a) = 1_B(f(a)) = f(a)$, for all a in $\text{Dom}(f)$. Thus, by Theorem 2 of Section 4.2, $1_B \circ f = f$.

(b) $(f \circ 1_A)(a) = f(1_A(a)) = f(a)$, for all a in $\text{Dom}(f)$, so $f \circ 1_A = f$.

Suppose now that f is a one-to-one correspondence between A and B. As we pointed out, the equation $b = f(a)$ is equivalent to the equation $a = f^{-1}(b)$. Since f and f^{-1} are both everywhere defined and onto, this means that, for all a in A and b in B, $f(f^{-1}(b)) = b$ and $f^{-1}(f(a)) = a$. Then

(c) For all a in A, $1_A(a) = a = f^{-1}(f(a)) = (f^{-1} \circ f)(a)$. Thus $1_A = f^{-1} \circ f$.

(d) For all b in B, $1_B(b) = b = f(f^{-1}(b)) = (f \circ f^{-1})(b)$. Thus $1_B = f \circ f^{-1}$. ∎

Theorem 3 **(a)** Let $f: A \to B$ and $g: B \to A$ be functions such that $g \circ f = 1_A$ and $f \circ g = 1_B$. Then f is a one-to-one correspondence between A and B, g is a one-to-one correspondence between B and A, and each is the inverse of the other.

(b) Let $f: A \to B$ and $g: B \to C$ be invertible. Then $g \circ f$ is invertible, and $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

Proof

(a) The assumptions mean that

$$g(f(a)) = a \quad \text{and} \quad f(g(b)) = b, \quad \text{for all } a \text{ in } A \text{ and } b \text{ in } B.$$

This shows in particular that $\text{Ran}(f) = B$ and $\text{Ran}(g) = A$, so each function is onto. If $f(a_1) = f(a_2)$, then $a_1 = g(f(a_1)) = g(f(a_2)) = a_2$. Thus f is one to one. In a similar way, we see that g is one to one, so both f and g are invertible. Note that f^{-1} is everywhere defined since $\text{Dom}(f^{-1}) = \text{Ran}(f) = B$. Now, if b is any element in B,

$$f^{-1}(b) = f^{-1}(f(g(b)) = (f^{-1} \circ f)g(b)) = 1_A(g(b)) = g(b).$$

Thus $g = f^{-1}$, so also $f = (f^{-1})^{-1} = g^{-1}$. Then, since g and f are onto, f^{-1} and g^{-1} are onto, so f and g must be everywhere defined. This proves all parts of part (a).

(b) We know that $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$, since this is true for any two relations. Since g^{-1} and f^{-1} are functions by assumption, so is their composition, and then $(g \circ f)^{-1}$ is a function. Thus $g \circ f$ is invertible. ∎

EXAMPLE 17

Let $A = B = \mathbb{R}$, the set of real numbers. Let $f: A \to B$ be given by the formula $f(x) = 2x^3 - 1$ and let $g: B \to A$ be given by

$$g(y) = \sqrt[3]{\tfrac{1}{2}y + \tfrac{1}{2}}.$$

Show that f is a bijection between A and B and g is a bijection between B and A.

Solution

Let $x \in A$ and $y = f(x) = 2x^3 - 1$. Then $\frac{1}{2}(y + 1) = x^3$; therefore,

$$x = \sqrt[3]{\tfrac{1}{2}y + \tfrac{1}{2}} = g(y) = g(f(x)) = (g \circ f)(x).$$

Thus $g \circ f = 1_A$. Similarly, $f \circ g = 1_B$, so by Theorem 3(a) both f and g are bijections. ■

As Example 17 shows, it is often easier to show that a function, such as f, is one to one and onto by constructing an inverse instead of proceeding directly.

Finally, we discuss briefly some special results that hold when A and B are finite sets. Let $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$, and let f be a function from A to B that is everywhere defined. If f is one to one, then $f(a_1), f(a_2), \dots, f(a_n)$ are n distinct elements of B. Thus we must have all of B, so f is also onto. On the other hand, if f is onto, then $f(a_1), \dots, f(a_n)$ form the entire set B, so they must all be different. Hence f is also one to one. We have therefore shown the following:

Theorem 4 Let A and B be two finite sets with the same number of elements, and let $f: A \to B$ be an everywhere defined function.

(a) If f is one to one, then f is onto.

(b) If f is onto, then f is one to one. ■

Thus for finite sets A and B with the same number of elements, and particularly if $A = B$, we need only prove that a function is one to one *or* onto to show that it is a bijection. This is an application of the pigeonhole principle.

One-to-one functions are a fundamental tool in cryptology, because of the need to both encode and decode. Many secret codes are simple **substitution codes** created as follows. Let $A = \{\text{a, b}, \dots, \text{z}\}$ be the English alphabet, and let $f: A \to A$ be a function agreed on in advance by each party to a correspondence. A message is encoded by replacing each letter with its f image. In order for the message to be decoded, the function f must have an inverse. The recipient decodes the message by applying f^{-1} to each letter. We know by Theorem 3(a) that f must therefore be one to one.

EXAMPLE 18 Suppose that f is defined by the following table:

A	B	C	D	E	F	G	H	I	J	K	L	M
D	E	S	T	I	N	Y	A	B	C	F	G	H
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	K	L	M	O	P	Q	R	U	V	W	X	Z

Thus $f(\text{D}) = \text{T}$, $f(\text{R}) = \text{O}$, and so on. The rearrangement of the alphabet that defines this function is an example of using a keyword to begin, then listing all remaining letters in order.

The phrase THE TRUCK ARRIVES TONIGHT is encoded as

QAIQORSFDOOBUIPQKJBYAQ.

In this case, the inverse function is easily found by using the table from bottom to top. Thus, we can decode the phrase CKAJADPDGKJYEIDOT as

JOHNHASALONGBEARD;

that is, JOHN HAS A LONG BEARD. ■

5.1 Exercises

1. Let $A = \{a, b, c, d\}$ and $B = \{1, 2, 3\}$. Determine whether the relation R from A to B is a function. If it is a function, give its range.
   (a) $R = \{(a, 1), (b, 2), (c, 1), (d, 2)\}$
   (b) $R = \{(a, 1), (b, 2), (a, 2), (c, 1), (d, 2)\}$

2. Let $A = \{a, b, c, d\}$ and $B = \{1, 2, 3\}$. Determine whether the relation R from A to B is a function. If it is a function, give its range.
   (a) $R = \{(a, 3), (b, 2), (c, 1)\}$
   (b) $R = \{(a, 1), (b, 1), (c, 1), (d, 1)\}$

3. Determine whether the relation R from A to B is a function.

   $A =$ the set of all recipients of Medicare in the United States,
   $B = \{x \mid x \text{ is a nine-digit number}\}$,
   $a\ R\ b$ if b is a's Social Security number.

4. Determine whether the relation R from A to B is a function.

   $A =$ a set of people in the United States,
   $B = \{x \mid x \text{ is a nine-digit number}\}$,
   $a\ R\ b$ if b is a's passport number.

In Exercises 5 through 8, verify that the formula yields a function from A to B.

5. $A = B = Z$; $f(a) = a^2$
6. $A = B = \mathbb{R}$; $f(a) = e^a$
7. $A = \mathbb{R}$, $B = \{0, 1\}$; let Z be the set of integers and note that $Z \subseteq \mathbb{R}$. Then for any real number a, let

$$f(a) = \begin{cases} 0 & \text{if } a \notin Z \\ 1 & \text{if } a \in Z. \end{cases}$$

8. $A = \mathbb{R}$, $B = Z$; $f(a) =$ the greatest integer less than or equal to a.
9. Let $A = B = C = \mathbb{R}$, and let $f: A \to B$, $g: B \to C$ be defined by $f(a) = a - 1$ and $g(b) = b^2$. Find
   (a) $(f \circ g)(2)$ **(b)** $(g \circ f)(2)$
   (c) $(g \circ f)(x)$ **(d)** $(f \circ g)(x)$
   (e) $(f \circ f)(y)$ **(f)** $(g \circ g)(y)$
10. Let $A = B = \overline{C} = \mathbb{R}$, and let $f: A \to B$, $g: B \to C$ be defined by $f(a) = a + 1$ and $g(b) = b^2 + 2$. Find
    (a) $(g \circ f)(-2)$ **(b)** $(f \circ g)(-2)$
    (c) $(g \circ f)(x)$ **(d)** $(f \circ g)(x)$
    (e) $(f \circ f)(y)$ **(f)** $(g \circ g)(y)$
11. In each part, sets A and B and a function from A to B are given. Determine whether the function is one to one or onto (or both or neither).
    (a) $A = \{1, 2, 3, 4\} = B$;
    $f = \{(1, 1), (2, 3), (3, 4), (4, 2)\}$
    (b) $A = \{1, 2, 3\}$; $B = \{a, b, c, d\}$;
    $f = \{(1, a), (2, a), (3, c)\}$
12. In each part, sets A and B and a function from A to B are given. Determine whether the function is one to one or onto (or both or neither).
    (a) $A = \left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}\right\}$; $B = \{x, y, z, w\}$;
    $f = \left\{\left(\frac{1}{2}, x\right), \left(\frac{1}{4}, y\right), \left(\frac{1}{3}, w\right)\right\}$
    (b) $A = \{1.1, 7, 0.06\}$; $B = \{p, q\}$;
    $f = \{(1.1, p), (7, q), (0.06, p)\}$
13. In each part, sets A and B and a function from A to B are given. Determine whether the function is one to one or onto (or both or neither).
    (a) $A = B = Z$; $f(a) = a - 1$
    (b) $A = \mathbb{R}$, $B = \{x \mid x \text{ is real and } x \geq 0\}$; $f(a) = |a|$
14. In each part, sets A and B and a function from A to B are given. Determine whether the function is one to one or onto (or both or neither).
    (a) $A = \mathbb{R} \times \mathbb{R}$, $B = \mathbb{R}$; $f((a, b)) = a$
    (b) Let $S = \{1, 2, 3\}$, $T = \{a, b\}$. Let $A = B = S \times T$ and let f be defined by $f(n, a) = (n, b)$, $n = 1, 2, 3$, and $f(n, b) = (1, a)$, $n = 1, 2, 3$.
15. In each part, sets A and B and a function from A to B are given. Determine whether the function is one to one or onto (or both or neither).
    (a) $A = B = \mathbb{R} \times \mathbb{R}$; $f((a, b)) = (a + b, a - b)$
    (b) $A = \mathbb{R}$, $B = \{x \mid x \text{ is real and } x \geq 0\}$; $f(a) = a^2$
16. Let $f(n)$ be the number of divisors of n, $n \in \mathbb{Z}^+$. Determine whether f is one to one or onto (or both or neither).
17. Let $f(n)$ be the maximum of n and 50, $n \in \mathbb{Z}^+$. Determine whether f is one to one or onto (or both or neither).
18. Explain why Theorem 1(a) is equivalent to "f^{-1} is not a function if and only if f is not one to one."
19. Let $f: A \to B$ and $g: B \to A$. Verify that $g = f^{-1}$.
    (a) $A = B = \mathbb{R}$; $f(a) = \frac{a+1}{2}$, $g(b) = 2b - 1$
    (b) $A = \{x \mid x \text{ is real and } x \geq 0\}$; $B = \{y \mid y \text{ is real and } y \geq -1\}$; $f(a) = a^2 - 1$, $g(b) = \sqrt{b + 1}$
20. Let $f: A \to B$ and $g: B \to A$. Verify that $g = f^{-1}$.
    (a) $A = B = P(S)$, where S is a set. If $X \in P(S)$, let $f(X) = \overline{X} = g(X)$.
    (b) $A = B = \{1, 2, 3, 4\}$;
    $f = \{(1, 4), (2, 1), (3, 2), (4, 3)\}$;
    $g = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$
21. Let f be a function from A to B. Find f^{-1}.
    (a) $A = \{x \mid x \text{ is real and } x \geq -1\}$; $B = \{x \mid x \text{ is real and } x \geq 0\}$; $f(a) = \sqrt{a + 1}$
    (b) $A = B = \mathbb{R}$; $f(a) = a^3 + 1$
22. Let f be a function from A to B. Find f^{-1}.
    (a) $A = B = \mathbb{R}$; $f(a) = \frac{2a-1}{3}$
    (b) $A = B = \{1, 2, 3, 4, 5\}$;
    $f = \{(1, 3), (2, 2), (3, 4), (4, 5), (5, 1)\}$
23. Let $f(x, y) = (2x - y, x - 2y)$, $(x, y) \in \mathbb{R} \times \mathbb{R}$.
    (a) Show that f is one to one.
    (b) Find f^{-1}.

In Exercises 24 and 25, let f be a function from $A = \{1, 2, 3, 4\}$ to $B = \{a, b, c, d\}$. Determine whether f^{-1} is a function.

24. $f = \{(1, a), (2, a), (3, c), (4, d)\}$
25. $f = \{(1, a), (2, c), (3, b), (4, d)\}$
26. Let $A = B = C = \mathbb{R}$ and consider the functions $f: A \to B$ and $g: B \to C$ defined by $f(a) = 2a + 1$, $g(b) = b/3$. Verify Theorem 3(b): $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

27. If a set A has n elements, how many functions are there from A to A?

28. If a set A has n elements, how many bijections are there from A to A?

29. If A has m elements and B has n elements, how many functions are there from A to B?

30. Complete the following proof.

    If $f: A \to B$ and $g: B \to C$ are one-to-one functions, then $g \circ f$ is one to one.

    Proof: Let $a_1, a_2 \in A$. Suppose $(g \circ f)(a_1) = (g \circ f)(a_2)$. Then $g(f(a_1)) = g(f(a_2))$ and $f(a_1) = f(a_2)$, because _____. Thus $a_1 = a_2$, because _____. Hence $g \circ f$ is one to one.

31. Complete the following proof.

    If $f: A \to B$ and $g: B \to C$ are onto functions, then $g \circ f$ is onto.

    Proof: Choose $x \in$ _____. Then there exists $y \in$ _____ such that $g(y) = x$. (Why?) Then there exists $z \in$ _____ such that $f(z) = y$ (why?) and $(g \circ f)(z) = x$. Hence, $g \circ f$ is onto.

32. Let $f: A \to B$ and $g: B \to C$ be functions. Show that if $g \circ f$ is one to one, then f is one to one.

33. Let $f: A \to B$ and $g: B \to C$ be functions. Show that if $g \circ f$ is onto, then g is onto.

34. Let A be a set, and let $f: A \to A$ be a bijection. For any integer $k \geq 1$, let $f^k = f \circ f \circ \cdots \circ f$ (k factors), and let $f^{-k} = f^{-1} \circ f^{-1} \circ \cdots \circ f^{-1}$ (k factors). Define f^0 to be 1_A. Then f^n is defined for all $n \in Z$. For any $a \in A$, let $O(a, f) = \{f^n(a) \mid n \in Z\}$. Prove that if $a_1, a_2 \in A$, and $O(a_1, f) \cap O(a_2, f) \neq \varnothing$, then $O(a_1, f) = O(a_2, f)$.

35. Let $f: A \to B$ be a function with finite domain and range. Suppose that $|\mathrm{Dom}(f)| = n$ and $|\mathrm{Ran}(f)| = m$. Prove that

    (a) If f is one to one, then $m = n$.

    (b) If f is not one to one, then $m < n$.

36. Let $|A| = |B| = n$ and let $f: A \to B$ be an everywhere defined function. Prove that the following three statements are equivalent.

    (a) f is one to one. (b) f is onto.

    (c) f is a one-to-one correspondence (that is, f is one to one and onto).

37. Give a one-to-one correspondence between Z^+, the set of positive integers, and $A = \{x \mid x$ is a positive even integer$\}$.

38. Give a one-to-one correspondence between Z^+, the set of positive integers, and $A = \{x \mid x$ is a positive odd integer$\}$.

39. Based on Exercises 37 and 38, does $|Z^+| = |A| = |B|$? Justify your conclusion.

40. (a) Let $A = \mathbb{R}$ and $f: A \to \mathbb{R}$ be defined by $f(a) = a^2$. Prove or disprove that $f(a_1 + a_2) = f(a_1) + f(a_2)$.

    (b) Let $A = \{a, b\}$ and $f: A^* \to \mathbb{Z}$ be defined by $f(s)$ is the length of the string s. Prove or disprove that $f(s_1 \cdot s_2) = f(s_1) + f(s_2)$.

41. Let $A = \{0, 1\}$ and define $a \diamond b = (a + b) \bmod 2$. Let $B = \{true, false\}$. Define $f: A \to B$ by $f(0) = true$ and $f(1) = false$.

    (a) Prove or disprove that $f(a \diamond b) = f(a) \vee f(b)$.

    (b) Prove or disprove that $f(a \diamond b) = f(a) \wedge f(b)$.

42. (a) Use the function in Example 18 to encode the message COME BACK AT ONCE.

    (b) Decode the following message that was encoded using the function of Example 18.
    QODLLITSDJJKQOIQROJ

43. Use the method of Example 18 and the keyword JOURNALISM to encode the message ALL PROJECTS ARE ON TRACK.

44. Substitution codes like the one in Example 18 are not very secure. Describe a commonsense method to break such a code.

5.2 Functions for Computer Science

In previous chapters, we introduced on an informal basis some functions commonly used in computer science applications. In this section we review these and define some others.

EXAMPLE 1

Let A be a subset of the universal set $U = \{u_1, u_2, u_3, \ldots, u_n\}$. The **characteristic function of** A is defined as a function from U to $\{0, 1\}$ by the following:

$$f_A(u_i) = \begin{cases} 1 & \text{if } u_i \in A \\ 0 & \text{if } u_i \notin A. \end{cases}$$

If $A = \{4, 7, 9\}$ and $U = \{1, 2, 3, \ldots, 10\}$, then $f_A(2) = 0$, $f_A(4) = 1$, $f_A(7) = 1$, and $f_A(12)$ is undefined. It is easy to check that f_A is everywhere defined and onto, but is not one to one. ■

EXAMPLE 2

In Section 1.4 we defined a family of mod-n functions, one for each positive integer n. We call these functions f_n; that is, $f_n(m) = m \pmod{n}$. Each f_n is a function from the nonnegative integers to the set $\{0, 1, 2, 3, \ldots, n - 1\}$. For a fixed n, any nonnegative integer z can be written as $z = kn + r$ with $0 \leq r < n$. Then $f_n(z) = r$. We can also express this relation as $z \equiv r \pmod{n}$ (see Section 4.5). Each member of the mod function family is everywhere defined and onto, but not one to one. ■

EXAMPLE 3

Let A be the set of nonnegative integers, $B = Z^+$, and let $f: A \to B$ be defined by $f(n) = n!$. ■

EXAMPLE 4

The general version of the pigeonhole principle (Section 3.3) required the **floor function**, which is defined for rational numbers as $f(q)$ is the largest integer less than or equal to q. Here again is an example of a function that is not defined by a formula. The notation $\lfloor q \rfloor$ is often used for $f(q)$. Thus

$$f(1.5) = \lfloor 1.5 \rfloor = 1, \quad f(-3) = \lfloor -3 \rfloor = -3, \quad f(-2.7) = \lfloor -2.7 \rfloor = -3. \quad ■$$

EXAMPLE 5

A function similar to that in Example 4 is the **ceiling function**, which is defined for rational numbers as $c(q)$ is the smallest integer greater than or equal to q. The notation $\lceil q \rceil$ is often used for $c(q)$. Thus

$$c(1.5) = \lceil 1.5 \rceil = 2, \quad c(-3) = \lceil -3 \rceil = -3, \quad c(-2.7) = \lceil -2.7 \rceil = -2. \quad ■$$

Many common algebraic functions are used in computer science, often with domains restricted to subsets of the integers.

EXAMPLE 6

(a) Any polynomial with integer coefficients, p, can be used to define a function on Z as follows: If $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ and $z \in Z$, then $f(z)$ is the value of p evaluated at z.

(b) Let $A = B = Z^+$ and let $f: A \to B$ be defined by $f(z) = 2^z$. We call f the **base 2 exponential function.** Other bases may be used to define similar functions.

(c) Let $A = B = \mathbb{R}$ and let $f_n: A \to B$ be defined for each positive integer $n > 1$ as $f_n(x) = \log_n(x)$, the logarithm to the base n of x. In computer science applications, the bases 2 and 10 are particularly useful. ■

In general, the unary operations discussed in previous sections can be used to create functions similar to the function in Example 3. The sets A and B in the definition of a function need not be sets of numbers, as seen in the following examples.

EXAMPLE 7

(a) Let A be a finite set and define $l: A^* \to Z$ as $l(w)$ is the length of the string w (see Section 1.3 for the definition of A^* and strings).

(b) Let B be a finite subset of the universal set U and define $pow(B)$ to be the power set of B. Then pow is a function from V, the power set of U, to the power set of V.

(c) Let $A = B =$ the set of all 2×2 matrices with real number entries and let $t(\mathbf{M}) = \mathbf{M}^T$, the transpose of $\mathbf{M}$. Then t is everywhere defined, onto, and one to one. ■

EXAMPLE 8

(a) For elements of $Z^+ \times Z^+$, define $g(z_1, z_2)$ to be $\mathrm{GCD}(z_1, z_2)$. Then g is a function from $Z^+ \times Z^+$ to Z^+. The GCD of two numbers is defined in Section 1.4.

(b) In a similar fashion we can define $m(z_1, z_2)$ to be LCM(z_1, z_2). ■

Another type of function, a Boolean function, plays a key role in nearly all computer programs. Let B = {true, false}. Then a function from a set A to B is called a **Boolean function**. The predicates in Section 2.1 are examples of Boolean functions.

EXAMPLE 9

Let P(x) : x is even and Q(y) : y is odd. Then P and Q are functions from Z to B. We see that P(4) is true and Q(4) is false. The predicate R(x, y) : x is even or y is odd is a Boolean function of two variables from $Z \times Z$ to B. Here R(3, 4) is false and R(6, 4) is true. ■

■ Hashing Functions

In Section 4.6, two methods of storing the data for a relation or digraph in a computer were presented. Here we consider a more general problem of storing data. Suppose that we must store and later examine a large number of data records, customer accounts for example. In general we do not know how many records we may have to store at any given time. This suggests that linked-list storage is appropriate, because storage space is only used when we assign a record to it and we are not holding idle storage space. In order to examine a record we will have to be able to find it, so storing the data in a single linked list may not be practical because looking for an item may take a very long time (relatively speaking). One technique for handling such storage problems is to create a number of linked lists and to provide a method for deciding onto which list a new item should be linked. This method will also determine which list to search for a desired item. A key point is to attempt to assign an item to one of the lists at random. (Remember from Section 3.4 that this means each list has an equal chance of being selected.) This will have the effect of making the lists roughly the same size and thus keep the searching time about the same for any item.

Suppose we must maintain the customer records for a large company and will store the information as computer records. We begin by assigning each customer a unique seven-digit account number. A unique identifier for a record is called its key. For now we will not consider exactly how and what information will be stored for each customer account, but will describe only the storage of a location in the computer's memory where this information will be found. In order to determine to which list a particular record should be assigned, we create a **hashing function** from the set of keys to the set of list numbers. Hashing functions frequently use a mod-n function, as shown in the next example.

EXAMPLE 10

Suppose that (approximately) 10,000 customer account records must be stored and processed. The company's computer is capable of searching a list of 100 items in an acceptable amount of time. We decide to create 101 linked lists for storage, because if the hashing function works well in "randomly" assigning records to lists, we would expect to see roughly 100 records per list. We define a hashing function from the set of seven-digit account numbers to the set {0, 1, 2, 3, . . . , 100} as follows:

$$h(n) = n \text{ (mod 101)}.$$

That is, h is the mod-101 function. Thus,

$$h(2473871) = 2473871 \text{ (mod 101)} = 78.$$

This means that the record with account number 2473871 will be assigned to list 78. Note that the range of h is the set {0, 1, 2, . . . , 100}. ■

Because the function h in Example 10 is not one to one, different account numbers may be assigned to the same list by the hashing function. If the first position on list 78 is already occupied when the record with key 2473871 is to be stored, we say a collision has occurred. There are many methods for resolving collisions. One very simple method that will be sufficient for our work is to insert the new record at the end of the existing list. Using this method, when we wish to find a record, its key will be hashed and the list h(key) will be searched sequentially.

Many other hashing functions are suitable for this situation. For example, we may break the seven-digit account number into a three-digit number and a four-digit number, add these, and then apply the mod-101 function. Chopping the key into pieces to create the function is the origin of the name hashing function. Many factors are considered in addition to the number of records to be stored: the speed with which an average-length list can be searched and the time needed to compute the list number for an account are two possible factors to be taken into account. For reasons that will not be discussed here, the modulus used in the mod function should be a prime. Determining a "good" hashing function for a particular application is a challenging task.

Hashing functions are also employed in other applications such as cryptology where they are used to produce digital fingerprints and other electronic means to verify the authenticity of messages.

5.2 Exercises

1. Let f be the mod-10 function. Compute
   (a) $f(417)$ (b) $f(38)$ (c) $f(253)$
2. Let f be the mod-10 function. Compute
   (a) $f(81)$ (b) $f(316)$ (c) $f(1057)$

In Exercises 3 and 4, use the universal set $U = \{a, b, c, \ldots, y, z\}$ and the characteristic function for the specified subset to compute the requested function values.

3. $A = \{a, e, i, o, u\}$
   (a) $f_A(i)$ (b) $f_A(y)$ (c) $f_A(o)$
4. $B = \{m, n, o, p, q, r, z\}$
   (a) $f_B(a)$ (b) $f_B(m)$ (c) $f_B(s)$
5. Compute each of the following.
   (a) $\lfloor 2.78 \rfloor$ (b) $\lfloor -2.78 \rfloor$ (c) $\lfloor 14 \rfloor$
   (d) $\lfloor -17.3 \rfloor$ (e) $\lfloor 21.5 \rfloor$
6. Compute each of the following.
   (a) $\lceil 2.78 \rceil$ (b) $\lceil -2.78 \rceil$ (c) $\lceil 14 \rceil$
   (d) $\lceil -17.3 \rceil$ (e) $\lceil 21.5 \rceil$
7. Let k, n be positive integers with $k \leq n$. Prove that the number of multiples of k between 1 and n is $\left\lfloor \frac{n}{k} \right\rfloor$.
8. Prove that if n is odd, then $\left\lceil \frac{n^2}{4} \right\rceil = \frac{n^2+3}{4}$.

In Exercises 9 and 10, compute the values indicated. Note that if the domain of these functions is Z^+, then each function is the explicit formula for an infinite sequence. Thus sequences can be viewed as a special type of function.

9. $f(n) = 3n^2 - 1$
   (a) $f(3)$ (b) $f(17)$ (c) $f(5)$ (d) $f(12)$
10. $g(n) = 5 - 2n$
    (a) $g(4)$ (b) $g(14)$ (c) $g(129)$ (d) $g(23)$
11. Let $f_2(n) = 2^n$. Compute each of the following.
    (a) $f_2(1)$ (b) $f_2(3)$ (c) $f_2(5)$ (d) $f_2(10)$
12. Let $f_3(n) = 3^n$. Compute each of the following.
    (a) $f_3(2)$ (b) $f_3(3)$ (c) $f_3(6)$ (d) $f_3(8)$

In Exercises 13 through 16, let $lg(x) = \log_2(x)$.

13. Compute each of the following.
    (a) $lg(16)$ (b) $lg(128)$ (c) $lg(512)$ (d) $lg(1024)$
14. For each of the following find the largest integer less than or equal to the function value and the smallest integer greater than or equal to the function value.
    (a) $lg(10)$ (b) $lg(25)$
15. For each of the following find the largest integer less than or equal to the function value and the smallest integer greater than or equal to the function value.
    (a) $lg(50)$ (b) $lg(100)$
16. For each of the following find the largest integer less than or equal to the function value and the smallest integer greater than or equal to the function value.
    (a) $lg(256)$ (b) $lg(500)$
17. Prove that the function in Example 7(c), r: {2 × 2 matrices with real entries} → {2 × 2 matrices with real entries} is everywhere defined, onto, and one to one.

18. Let $A = \{a, b, c, d\}$. Let l be the function in Example 7(a).
    (a) Prove that l is everywhere defined.
    (b) Prove that l is not one to one.
    (c) Prove or disprove that l is onto.

19. Let A be a set with n elements, S be the set of relations on A, and M the set of $n \times n$ Boolean matrices. Define $f: S \to M$ by $f(R) = \mathbf{M}_R$. Prove that f is a bijection between S and M.

20. Let p be a Boolean variable. How many different Boolean functions of p are there? How many different Boolean functions of two Boolean variables are there?

21. Build a table to represent the Boolean function $f(x, y, z) = (\sim x \wedge y) \vee z$ for all possible values of x, y, and z.

22. Let P be the propositional function defined by $\mathrm{P}(x, y) = (x \vee y) \wedge \sim y$. Evaluate each of the following.
    (a) P(true, true) (b) P(false, true)
    (c) P(true, false)

23. Let Q be the propositional function defined by $\mathrm{Q}(x)\colon \exists(y \in Z^+)(xy = 60)$. Evaluate each of the following.
    (a) Q(3) (b) Q(7) (c) Q(−6) (d) Q(15)

In Exercises 24 through 26, use the hashing function h, which takes the first three digits of the account number as one number and the last four digits as another number, adds them, and then applies the mod-59 function.

24. Assume that there are 7500 customer records to be stored using this hashing function.
    (a) How many linked lists will be required for the storage of these records?
    (b) If an approximately even distribution is achieved, roughly how many records will be stored by each linked list?

25. Determine to which list the given customer account should be attached.
    (a) 3759273 (b) 7149021 (c) 5167249

26. Determine which list to search to find the given customer account.
    (a) 2561384 (b) 6082376 (c) 4984620

27. Refer to Section 3.4, Exercise 37 for the average number of steps needed to search an array of length n for a key. Suppose a hashing function based on mod k is used to store m items. On average, how many steps will be required on average to search for a key?

28. Use the characteristic function of a set to prove that if $|A| = n$, then $|pow(A)| = 2^n$.

29. Let f_A be the characteristic function of A with respect to the universal set U. What does the set $f^{-1}(1)$ represent?

Exercises 30 through 36 use ideas from this section to complete a discussion begun in Section 3.5, Exercises 38 through 40. Pairs of parentheses are often used in mathematical expressions to indicate the order in which operations are to be done. A compiler (or interpreter) for a programming language must check that pairs of parentheses are properly placed. This may involve a number of things, but one simple check is that the number of left and right parentheses are equal and that in reading from left to right the number of left parentheses is always greater than or equal to the number of right parentheses read. An expression that passes this check is called well formed. The task here is to count the number of well-formed strings of n left and n right parentheses. This number is C_n, the nth Catalan number.

30. How many strings of n left and n right parentheses can be made (not just well-formed ones)?

31. List all well-formed strings of n left and n right parentheses for $n = 1, 2, 3$. What are the values of C_1, C_2, and C_3?

32. We will count the strings that are not well formed by making a one-to-one correspondence between them and a set of easier to count strings. Suppose $p_1 p_2 p_3 \dots p_{2n}$ is not well formed; then there is a first p_i that is a right parenthesis and there are fewer left parentheses than right parentheses in $p_1 p_2 \dots p_i$. How many fewer are there? So to the right of p_i the number of left parentheses is ____ than the number of right parentheses. Make a new string $q_1 q_2 \dots q_{2n}$ as follows:

$$q_j = p_j, \quad j = 1, 2, \dots, i$$

and

$$q_j = \begin{cases} (& \text{if } p_j = \,) \\ ) & \text{if } p_j = \,(\end{cases} \quad \text{for } j = i+1, i+2, \dots, 2n.$$

This new string $q_1 q_2 \dots q_{2n}$ has ____ left and ____ right parentheses. Explain your reasoning.

33. To complete the one-to-one correspondence between the p and the q strings of Exercise 32, we must show that any string with $n-1$ left and $n+1$ right parentheses can be paired with exactly one string with n left and n right parentheses that is not well formed. Let $r_1 r_2 r_3 \dots r_{2n}$ consist of $n-1$ left and $n+1$ right parentheses. There must be a first position j where the number of right parentheses is greater than the number of left parentheses. Why? So in $r_1 r_2 r_3 \dots r_j$ there is one more right than left parenthesis. Hence in $r_{j+1} \dots r_{2n}$, the number of left parentheses is ____ than the number of right parentheses. Make a new string $s_1 s_2 \dots s_{2n}$ as follows:

$$s_k = r_k, \quad k = 1, 2, \dots, j$$

and

$$s_k = \begin{cases} (& \text{if } r_k = \,) \\ ) & \text{if } r_k = \,(\end{cases} \quad \text{for } k = j+1, j+2, \dots, 2n.$$

This new string $s_1 s_2 \dots s_{2n}$ has ____ left and ____ right parentheses. Explain how you know $s_1 s_2 \dots s_{2n}$ is not well formed.

34. Using the results of Exercises 32 and 33, the number of strings with n left and n right parentheses that are not well formed is equal to the number of strings with $n-1$ left and $n+1$ right parentheses. By Section 3.2, this number is ____.

35. Use the results of Exercises 30 and 34 to give a formula for C_n. Confirm this result by comparing its values with those found in Exercise 31.

36. Express C_n using the notation for combinations and without this notation.

*Another application of mod functions occurs in assigning an ISBN (International Standard Book Number) to each title published. The 10-digit ISBN encodes information about the language of publication, the publisher, and the book itself. This is an example of coding for error checking rather than for security purposes. For example, the ISBN for the fourth edition of this book is 0-13-083143-3; the 0 indicates the book was published in an English-speaking country and the 13 identifies the publisher. The last digit is a **check digit** chosen to help prevent transcription errors. If $d_1 d_2 d_3 \cdots d_9 c$ is an ISBN, then c is chosen so that*

$$(d_1 + 2d_2 + 3d_3 + \cdots + 9d_9 + 10c) \equiv 0 \pmod{11}.$$

If c is 10, then the Roman numeral X is used.

37. (a) Verify that 3 is the correct check digit for this book's ISBN.
    (b) Compute the check digit c for the following ISBNs.
        (i) 0-471-80075-c (ii) 0-80504826-c
        (iii) 88-8117-275-c (iv) 5-05-001801-c

38. (a) Make one change in the first nine digits of the ISBN 0-183-47381-7 so that the check digit will indicate an error.
    (b) Make two changes in the first nine digits of the ISBN 0-183-47381-7 so that the check digit will not indicate an error.

5.3 Growth of Functions

In the earlier discussion of computer representations of relations (Section 4.6), we saw that one of the factors determining the choice of storage method is the efficiency of handling the data. In the example of testing to see if a relation is transitive, the average number of steps needed was computed for an algorithm with the relation stored as a matrix and for an algorithm with the relation stored using a linked list. The results were that it would take roughly $kn^3 + (1-k)n^2$ steps using matrix storage and $k^3 n^4$ steps using a linked list, where the relation contains kn^2 ordered pairs. Although many details were ignored, these rough comparisons give enough information to make some decisions about appropriate data storage. In this section we apply some concepts from previous sections and lay the groundwork for more sophisticated analysis of algorithms.

The idea of one function growing more rapidly than another arises naturally when working with functions. In this section we formalize this notion.

EXAMPLE 1

Let R be a relation on a set A with $|A| = n$ and $|R| = \frac{1}{2}n^2$. If R is stored as a matrix, then $t(n) = \frac{1}{2}n^3 + \frac{1}{2}n^2$ is a function that describes (roughly) the average number of steps needed to determine if R is transitive using the algorithm TRANS (Section 4.6). Storing R with a linked list and using NEWTRANS, the average number of steps needed is (roughly) given by $s(n) = \frac{1}{8}n^4$. Table 5.1 shows that s grows faster than t.

TABLE 5.1

n	$t(n)$	$s(n)$
10	550	1250
50	63,750	781,250
100	505,000	12,500,000

■

Let f and g be functions whose domains are subsets of Z^+, the positive integers. We say that f is $O(g)$, read **f is big-Oh of g**, if there exist constants c and

k such that $|f(n)| \le c \cdot |g(n)|$ for all $n \ge k$. If f is $O(g)$, then f grows no faster than g does.

Example 2

The function $f(n) = \frac{1}{2}n^3 + \frac{1}{2}n^2$ is $O(g)$ for $g(n) = n^3$. To see this, consider

$$\frac{1}{2}n^3 + \frac{1}{2}n^2 \le \frac{1}{2}n^3 + \frac{1}{2}n^3, \qquad \text{if } n \ge 1.$$

Thus,

$$\frac{1}{2}n^3 + \frac{1}{2}n^2 \le 1 \cdot n^3, \qquad \text{if } n \ge 1.$$

Choosing 1 for c and 1 for k, we have shown that $|f(n)| \le c \cdot |g(n)|$ for all $n \ge 1$ and f is $O(g)$. ■

The reader can see from Example 2 that other choices of c, k, and even g are possible. If $|f(n)| \le c|g(n)|$ for all $n \ge k$, then we have $|f(n)| \le C \cdot |g(n)|$ for all $n \ge k$ for any $C \ge c$, and $|f(n)| \le c \cdot |g(n)|$ for all $n \ge K$ for any $K \ge k$. For the function t in Example 2, t is $O(h)$ for $h(n) = dn^3$, if $d \ge 1$, since $|t(n)| \le 1 \cdot |g(n)| \le |h(n)|$. Observe also that t is $O(r(n))$ for $r(n) = n^4$, because $\frac{1}{2}n^3 + \frac{1}{2}n^2 \le n^3 \le n^4$ for all $n \ge 1$. When analyzing algorithms, we want to know the "slowest growing" simple function g for which f is $O(g)$.

It is common to replace g in $O(g)$ with the formula that defines g. Thus we write that t is $O(n^3)$. This is called big-O notation.

We say that f and g have the **same order** if f is $O(g)$ and g is $O(f)$.

Example 3

Let $f(n) = 3n^4 - 5n^2$ and $g(n) = n^4$ be defined for positive integers n. Then f and g have the same order. First,

$$\begin{aligned} 3n^4 - 5n^2 &\le 3n^4 + 5n^2 \\ &\le 3n^4 + 5n^4, \qquad \text{if } n \ge 1 \\ &= 8n^4. \end{aligned}$$

Let $c = 8$ and $k = 1$, then $|f(n)| \le c \cdot |g(n)|$ for all $n \ge k$. Thus f is $O(g)$. Conversely, $n^4 = 3n^4 - 2n^4 \le 3n^4 - 5n^2$ if $n \ge 2$. This is because if $n \ge 2$, then $n^2 > \frac{5}{2}$, $2n^2 > 5$, and $2n^4 > 5n^2$. Using 1 for c and 2 for k, we conclude that g is $O(f)$. ■

If f is $O(g)$ but g is not $O(f)$, we say that f is **lower order** than g or that f grows more slowly than g.

Example 4

The function $f(n) = n^5$ is lower order than $g(n) = n^7$. Clearly, if $n \ge 1$, then $n^5 \le n^7$. Suppose that there exist c and k such that $n^7 \le cn^5$ for all $n \ge k$. Choose N so that $N > k$ and $N^2 > c$. Then $N^7 \le cN^5 < N^2 \cdot N^5$, but this is a contradiction. Hence f is $O(g)$, but g is not $O(f)$, and f is lower order than g. This agrees with our experience that n^5 grows more slowly than n^7. ■

We define a relation Θ, big-theta, on functions whose domains are subsets of Z^+ as $f \; \Theta \; g$ if and only if f and g have the same order.

Theorem 1 The relation Θ, big-theta, is an equivalence relation.

Proof

Clearly, Θ is reflexive since every function has the same order as itself. Because the definition of same order treats f and g in the same way, this definition is symmetric and the relation Θ is symmetric.

To see that Θ is transitive, suppose f and g have the same order. Then there exist c_1 and k_1 with $|f(n)| \le c_1 \cdot |g(n)|$ for all $n \ge k_1$, and there exist c_2 and k_2 with $|g(n)| \le c_2 \cdot |f(n)|$ for all $n \ge k_2$. Suppose that g and h have the same order; then there exist c_3, k_3 with $|g(n)| \le c_3 \cdot |h(n)|$ for all $n \ge k_3$, and there exist c_4, k_4 with $|h(n)| \le c_4 \cdot |g(n)|$ for all $n \ge k_4$.

Then $|f(n)| \le c_1 \cdot |g(n)| \le c_1(c_3 \cdot |h(n)|)$ if $n \ge k_1$ and $n \ge k_3$. Thus $|f(n)| \le c_1c_3 \cdot |h(n)|$ for all $n \ge$ maximum of k_1 and k_3.

Similarly, $|h(n)| \le c_2c_4 \cdot |f(n)|$ for all $n \ge$ maximum of k_2 and k_4. Thus f and h have the same order and Θ is transitive. ■

The equivalence classes of Θ consist of functions that have the same order. We use any simple function in the equivalence class to represent the order of all functions in that class. One Θ-class is said to be **lower** than another Θ-class if a representative function from the first is of lower order than one from the second class. This means functions in the first class grow more slowly than those in the second. It is the Θ-class of a function that gives the information we need for algorithm analysis.

Example 5

All functions that have the same order as $g(n) = n^3$ are said to have order $\Theta(n^3)$. The most common orders in computer science applications are $\Theta(1)$, $\Theta(n)$, $\Theta(n^2)$, $\Theta(n^3)$, $\Theta(lg(n))$, $\Theta(nlg(n))$, and $\Theta(2^n)$. Here $\Theta(1)$ represents the class of constant functions and lg is the base 2 log function. The continuous versions of some of these functions are shown in Figure 5.4. ■

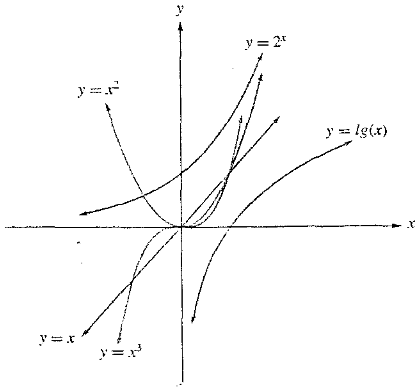


Figure 5.4

Example 6

Every logarithmic function $f(n) = \log_b(n)$ has the same order as $g(n) = lg(n)$. There is a logarithmic change-of-base identity

$$\log_b(x) = \frac{\log_a(x)}{\log_a(b)}$$

in which $\log_a(b)$ is a constant. Thus

$$|\log_b(n)| \leq \frac{1}{lg(b)}|lg(n)|$$

and, conversely,

$$|lg(n)| \leq lg(b) \cdot |\log_b(n)|.$$

Hence g is $O(f)$ and f is $O(g)$. ■

It is sometimes necessary to combine functions that give the number of steps required for pieces of an algorithm as is done in the analysis of TRANS (Section 4.6), where functions are added, and in the analysis of NEWTRANS, where functions are multiplied. There are some general rules regarding the ordering of the Θ-equivalence classes that can be used to determine the class of many functions and the class of the sum or product of previously classified functions.

■ Rules for Determining the Θ-Class of a Function

1. $\Theta(1)$ functions are constant and have zero growth, the slowest growth possible.
2. $\Theta(lg(n))$ is lower than $\Theta(n^k)$ if $k > 0$. This means that any logarithmic function grows more slowly than any power function with positive exponent.
3. $\Theta(n^a)$ is lower than $\Theta(n^b)$ if and only if $0 < a < b$.
4. $\Theta(a^n)$ is lower than $\Theta(b^n)$ if and only if $0 < a < b$.
5. $\Theta(n^k)$ is lower than $\Theta(a^n)$ for any power n^k and any $a > 1$. This means that any exponential function with base greater than 1 grows more rapidly than any power function.
6. If r is not zero, then $\Theta(rf) = \Theta(f)$ for any function f.
7. If h is a nonzero function and $\Theta(f)$ is lower than (or the same as) $\Theta(g)$, then $\Theta(fh)$ is lower than (or the same as) $\Theta(gh)$.
8. If $\Theta(f)$ is lower than $\Theta(g)$, then $\Theta(f + g) = \Theta(g)$.

EXAMPLE 7

Determine the Θ-class of each of the following.

(a) $f(n) = 4n^4 - 6n^7 + 25n^3$
(b) $g(n) = lg(n) - 3n$
(c) $h(n) = 1.1^n + n^{15}$

Solution

(a) By Rules 3, 6, and 8, the degree of the polynomial determines the Θ-class of a polynomial function. $\Theta(f) = \Theta(n^7)$.
(b) Using Rules 2, 6, and 8, we have that $\Theta(g) = \Theta(n)$.
(c) By Rules 5 and 8, $\Theta(h) = \Theta(1.1^n)$. ■

EXAMPLE 8

Using the rules for ordering Θ-classes, arrange the following in order from lowest to highest.

$$\Theta(nlg(n)) \quad \Theta(1000n^2 - n) \quad \Theta(n^{0.2}) \quad \Theta(1{,}000{,}000) \quad \Theta(1.3^n) \quad \Theta(n + 10^7)$$

Solution

$\Theta(1{,}000{,}000)$ is the class of constant functions, so it is the first on the list. By Rules 5 and 8, $\Theta(n + 10^7)$ is lower than $\Theta(1000n^2 - n)$, but higher than $\Theta(n^{0.2})$. To determine the position of $\Theta(nlg(n))$ on the list, we apply Rules 2 and 7. These give that $\Theta(nlg(n))$ is lower than $\Theta(n^2)$ and higher than $\Theta(n)$. Rule 5 says that $\Theta(1.3^n)$ is the highest class on this list. In order, the classes are

$$\Theta(1{,}000{,}000) \quad \Theta(n^{0.2}) \quad \Theta(n + 10^7)$$
$$\Theta(nlg(n)) \quad \Theta(1000n^2 - n) \quad \Theta(1.3^n).$$ ■

The Θ-class of a function that describes the number of steps performed by an algorithm is frequently referred to as the **running time** of the algorithm. For example, the algorithm TRANS has an average running time of n^3. In general, algorithms with exponential running times are impractical for all but very small values of n. In many cases the running time of an algorithm is estimated by examining best, worst, or average cases.

5.3 Exercises

In Exercises 1 and 2, let f be a function that describes the number of steps required to carry out a certain algorithm. The number of items to be processed is represented by n. For each function, describe what happens to the number of steps if the number of items is doubled.

1. (a) $f(n) = 1001$ (b) $f(n) = 3n$
(c) $f(n) = 5n^2$ (d) $f(n) = 2.5n^3$
2. (a) $f(n) = 1.4lg(n)$ (b) $f(n) = 2^n$
(c) $f(n) = nlg(n)$ (d) $f(n) = 100n^4$
3. Show that $g(n) = n!$ is $O(n^n)$.
4. Show that $h(n) = 1 + 2 + 3 + \cdots + n$ is $O(n^2)$.
5. Show that $f(n) = 8n + lg(n)$ is $O(n)$.
6. Show that $g(n) = n^2(7n - 2)$ is $O(n^3)$.
7. Show that $f(n) = nlg(n)$ is $O(g)$ for $g(n) = n^2$, but that g is not $O(f)$.
8. Show that $f(n) = n^{100}$ is $O(g)$ for $g(n) = 2^n$, but that g is not $O(f)$.
9. Show that f and g have the same order for $f(n) = 5n^2 + 4n + 3$ and $g(n) = n^2 + 100n$.
10. Show that f and g have the same order for $f(n) = lg(n^3)$ and $g(n) = \log_5(6n)$.
11. Determine which of the following are in the same Θ-class. A function may be in a class by itself.

$$f_1(n) = 5nlg(n), \quad f_2(n) = 6n^2 - 3n + 7,$$
$$f_3(n) = 1.5^n, \quad f_4(n) = lg(n^4),$$
$$f_5(n) = 13{,}463, \quad f_6(n) = -15n,$$
$$f_7(n) = lg(lg(n)), \quad f_8(n) = 9n^{0.7},$$
$$f_9(n) = n!, \quad f_{10}(n) = n + lg(n),$$
$$f_{11}(n) = \sqrt{n} + 12n, \quad f_{12}(n) = lg(n!)$$

12. Order the Θ-classes in Exercise 11 from lowest to highest.
13. Consider the functions $f_1, f_2, f_4, f_5, f_6, f_{10}, f_{11}$ in Exercise 11. Match each of the functions with its Θ-class from the following list: $\Theta(1)$, $\Theta(n)$, $\Theta(nlg(n))$, $\Theta(lg(n))$, $\Theta(n^2)$, $\Theta(\sqrt{n})$, $\Theta(2^n)$.

In Exercises 14 through 21, analyze the operation performed by the given piece of pseudocode and write a function that describes the number of steps required. Give the Θ-class of the function.

14.
```
1. $A \leftarrow 1$
2. $B \leftarrow 1$
3. UNTIL $(B > 100)$
      a. $B \leftarrow 2A - 2$
      b. $A \leftarrow A + 3$
```

15.
```
1. $X \leftarrow 1$
2. $Y \leftarrow 100$
3. WHILE $(X < Y)$
      a. $X \leftarrow X + 2$
      b. $Y \leftarrow \frac{1}{2}Y$
```

16.
```
1. $I \leftarrow 1$
2. $X \leftarrow 0$
3. WHILE $(I \leq N)$
      a. $X \leftarrow X + 1$
      b. $I \leftarrow I + 1$
```

17.
```
1. $SUM \leftarrow 0$
2. FOR $I = 0$ THRU $2(N - 1)$ BY 2
      a. $SUM \leftarrow SUM + I$
```

18. Assume that N is a power of 2.
```
1. $X \leftarrow 1$
2. $K \leftarrow N$
3. WHILE $(K \geq 1)$
      a. $X \leftarrow 3X$
      b. $K \leftarrow \lfloor K/2 \rfloor$
```

19.
```
1. $I \leftarrow 1$
2. $SUM \leftarrow 0$
3. WHILE $(I \leq N)$
      a. FOR $K = 1$ THRU $I$
          1. $SUM \leftarrow SUM + K$
      b. $I \leftarrow I + 1$
```

20.
```
1. $K \leftarrow 0$
2. FOR $I = 0$ THRU $N$
      a. WHILE $K \leq I$
          1. $K \leftarrow K + 1$
```

21.
```
SUBROUTINE MATMUL(A,B,N,M,P,Q;C)
1. IF (M = P) THEN
     a. FOR I = 1 THRU N
          1. FOR J = 1 THRU Q
               a. C[I,J] ← 0
               b. FOR K = 1 THRU M
                    1. C[I,J] ←
                     C[I,J] +
                     (A[I,K] × B[K,J])
2. ELSE
     a. CALL PRINT ('INCOMPATIBLE')
3. RETURN
END OF SUBROUTINE MATMUL
```

22. Determine the Θ-class of the function defined in Section 1.3, Exercise 38. What is the running time for computing $F(N)$?

23. **(a)** Write a recurrence relation to count the number of ways a 3×3 square can be placed on an $n \times n$ square with the edges of the squares parallel.

    (b) What is the running time of an algorithm that uses the recurrence relation in (a) to count the number of placements?

24. Prove Rule 3.

25. Prove Rule 4.

26. Prove Rule 6.

27. Prove Rule 7.

28. Prove that if $\Theta(f) = \Theta(g) = \Theta(h)$, then $f + g$ is $O(h)$.

29. Prove that if $\Theta(f) = \Theta(g)$ and $c \neq 0$, then $\Theta(cf) = \Theta(g)$.

5.4 Permutation Functions

In this section we discuss bijections from a set A to itself. Of special importance is the case when A is finite. Bijections on a finite set occur in a wide variety of applications in mathematics, computer science, and physics.

A bijection from a set A to itself is called a **permutation** of A.

EXAMPLE 1

Let $A = \mathbb{R}$ and let $f: A \to A$ be defined by $f(a) = 2a + 1$. Since f is one to one and onto (verify), it follows that f is a permutation of A. ■

If $A = \{a_1, a_2, \ldots, a_n\}$ is a finite set and p is a bijection on A, we list the elements of A and the corresponding function values $p(a_1), p(a_2), \ldots, p(a_n)$ in the following form:

$$\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ p(a_1) & p(a_2) & \cdots & p(a_n) \end{pmatrix}. \tag{1}$$

Observe that (1) completely describes p since it gives the value of p for every element of A. We often write

$$p = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ p(a_1) & p(a_2) & \cdots & p(a_n) \end{pmatrix}.$$

Thus, if p is a permutation of a finite set $A = \{a_1, a_2, \ldots, a_n\}$, then the sequence $p(a_1), p(a_2), \ldots, p(a_n)$ is just a rearrangement of the elements of A and so corresponds exactly to a permutation of A in the sense of Section 3.1.

EXAMPLE 2

Let $A = \{1, 2, 3\}$. Then all the permutations of A are

$$1_A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \quad p_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \quad p_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix},$$

$$p_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \quad p_4 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \quad p_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$ ■

[illegible]

Using the permutations of Example 2, compute **(a)** p_4^{-1}; **(b)** $p_3 \circ p_2$.

Solution

(a) Viewing p_4 as a function, we have

$$p_4 = \{(1, 3), (2, 1), (3, 2)\}.$$

Then

$$p_4^{-1} = \{(3, 1), (1, 2), (2, 3)\}$$

or, when written in increasing order of the first component of each ordered pair, we have

$$p_4^{-1} = \{(1, 2), (2, 3), (3, 1)\}.$$

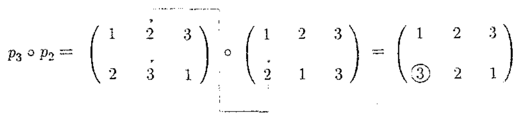
Thus

$$p_4^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = p_3.$$

(b) The function p_2 takes 1 to 2 and p_3 takes 2 to 3, so $p_3 \circ p_2$ takes 1 to 3. Also, p_2 takes 2 to 1 and p_3 takes 1 to 2, so $p_3 \circ p_2$ takes 2 to 2. Finally, p_2 takes 3 to 3 and p_3 takes 3 to 1, so $p_3 \circ p_2$ takes 3 to 1. Thus

$$p_3 \circ p_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

We may view the process of forming $p_3 \circ p_2$ as shown in Figure 5.5. Observe that $p_3 \circ p_2 = p_5$. ■



$p_3 \circ p_2 =$
$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ ③ & 2 & 1 \end{pmatrix}$

Figure 5.5

The composition of two permutations is another permutation, usually referred to as the **product** of these permutations. In the remainder of this chapter, we will follow this convention.

Theorem 1 If $A = \{a_1, a_2, \ldots, a_n\}$ is a set containing n elements, then there are

$$n! = n \cdot (n-1) \cdots 2 \cdot 1 \quad \text{permutations of } A. \tag{2}$$

Proof

This result follows from Theorem 4 of Section 3.1 by letting $r = n$. ■

Let $b_1, b_2, \ldots, b_r$ be r distinct elements of the set $A = \{a_1, a_2, \ldots, a_n\}$. The permutation $p: A \to A$ defined by

$$\begin{aligned} p(b_1) &= b_2 \\ p(b_2) &= b_3 \\ &\vdots \\ p(b_{r-1}) &= b_r \\ p(b_r) &= b_1 \\ p(x) &= x, \quad \text{if } x \in A,\ x \notin \{b_1, b_2, \ldots, b_r\}, \end{aligned}$$

is called a **cyclic permutation** of length r, or simply a **cycle** of length r, and will be denoted by $(b_1, b_2, \ldots, b_r)$. Do not confuse this terminology with that used for cycles in a digraph (Section 4.3). The two concepts are different and we use slightly different notations. If the elements $b_1, b_2, \ldots, b_r$ are arranged uniformly on a circle, as shown in Figure 5.6, then a cycle p of length r moves these elements in a clockwise direction so that b_1 is sent to b_2, b_2 to $b_3, \ldots, b_{r-1}$ to b_r, and b_r to b_1. All the other elements of A are left fixed by p.

EXAMPLE 4

Let $A = \{1, 2, 3, 4, 5\}$. The cycle $(1, 3, 5)$ denotes the permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 5 & 4 & 1 \end{pmatrix}.$$

■

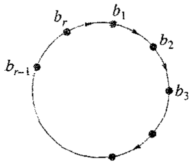


Figure 5.6

Observe that if $p = (b_1, b_2, \ldots, b_r)$ is a cycle of length r, then we can also write p by starting with any b_i, $1 \le i \le r$, and moving in a clockwise direction, as shown in Figure 5.6. Thus, as cycles,

$$(3, 5, 8, 2) = (5, 8, 2, 3) = (8, 2, 3, 5) = (2, 3, 5, 8).$$

Note also that the notation for a cycle does not include the number of elements in the set A. Thus the cycle $(3, 2, 1, 4)$ could be a permutation of the set $\{1, 2, 3, 4\}$ or of $\{1, 2, 3, 4, 5, 6, 7, 8\}$. We need to be told explicitly the set on which a cycle is defined. It follows from the definition that a cycle on a set A is of length 1 if and only if it is the identity permutation, 1_A.

Since cycles are permutations, we can form their product. However, as we show in the following example, the product of two cycles need not be a cycle.

EXAMPLE 5

Let $A = \{1, 2, 3, 4, 5, 6\}$. Compute $(4, 1, 3, 5) \circ (5, 6, 3)$ and $(5, 6, 3) \circ (4, 1, 3, 5)$.

Solution

We have

$$(4, 1, 3, 5) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 5 & 1 & 4 & 6 \end{pmatrix}$$

and

$$(5, 6, 3) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 5 & 4 & 6 & 3 \end{pmatrix}.$$

Then

$$(4, 1, 3, 5) \circ (5, 6, 3) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 5 & 1 & 4 & 6 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 5 & 4 & 6 & 3 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 4 & 1 & 6 & 5 \end{pmatrix}$$

and

$$(5, 6, 3) \circ (4, 1, 3, 5) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 5 & 4 & 6 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 5 & 1 & 4 & 6 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 2 & 6 & 1 & 4 & 3 \end{pmatrix}.$$

Observe that

$$(4, 1, 3, 5) \circ (5, 6, 3) \ne (5, 6, 3) \circ (4, 1, 3, 5)$$

and that neither product is a cycle. ■

Two cycles of a set A are said to be **disjoint** if no element of A appears in both cycles.

EXAMPLE 6

Let $A = \{1, 2, 3, 4, 5, 6\}$. Then the cycles $(1, 2, 5)$ and $(3, 4, 6)$ are disjoint, whereas the cycles $(1, 2, 5)$ and $(2, 4, 6)$ are not. ■

It is not difficult to show that if $p_1 = (a_1, a_2, \ldots, a_r)$ and $p_2 = (b_1, b_2, \ldots, b_s)$ are disjoint cycles of A, then $p_1 \circ p_2 = p_2 \circ p_1$. This can be seen by observing that p_1 affects only the a's, while p_2 affects only the b's.

We shall now present a fundamental theorem and, instead of giving its proof, we shall give an example that imitates the proof.

Theorem 2 A permutation of a finite set that is not the identity or a cycle can be written as a product of disjoint cycles of length ≥ 2. ■

EXAMPLE 7

Write the permutation

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 6 & 5 & 2 & 1 & 8 & 7 \end{pmatrix}$$

of the set $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$ as a product of disjoint cycles.

Solution

We start with 1 and find that $p(1) = 3$, $p(3) = 6$, and $p(6) = 1$, so we have the cycle $(1, 3, 6)$. Next we choose the first element of A that has not appeared in a previous cycle. We choose 2, and we have $p(2) = 4$, $p(4) = 5$, and $p(5) = 2$, so we obtain the cycle $(2, 4, 5)$. We now choose 7, the first element of A that has not appeared in a previous cycle. Since $p(7) = 8$ and $p(8) = 7$, we obtain the cycle $(7, 8)$. We can then write p as a product of disjoint cycles as

$$p = (7, 8) \circ (2, 4, 5) \circ (1, 3, 6).$$

■

It is not difficult to show that in Theorem 2, when a permutation is written as a product of disjoint cycles, the product is unique except for the order of the cycles.

We saw in Section 5.1 how a permutation of the alphabet produces a substitution code. Permutations are also used to produce **transposition codes**. Unlike a substitution code in which each letter is replaced by a substitute, the letters in transposition coded messages are not changed, but are rearranged. Thus if a message TEST THE WATERS is subjected to the permutation

$$(1, 2, 3) \circ (4, 7) \circ (5, 10, 11) \circ (6, 8, 12, 13, 9),$$

where the numbers refer to the positions of the letters in the message, the message becomes STEEEATHSTTWR. If the permutation is known to both parties, then the receiver of the message has only to apply the inverse permutation to decode.

EXAMPLE 8

One commonly used transposition code is the **keyword columnar transposition**. For this it is only necessary to remember a keyword, say JONES. The message to be encoded is written under the keyword in successive rows, padding at the end if necessary. For example, the message THE FIFTH GOBLET CONTAINS THE

GOLD would be arranged as shown:

J	O	N	E	S
T	H	E	F	I
F	T	H	G	O
B	L	E	T	C
O	N	T	A	I
N	S	T	H	E
G	O	L	D	X

Note that the message has been padded with an X to fill out the row. Then the coded message is constructed by writing the columns in succession, beginning with column 4 (since E is the first keyword letter to appear in the alphabet) and following with the letters in columns 1, 3, 2, 5. The encoded message is thus

FGTAHDTFBOGCEHETTLHTLNSOIOCIEX.

The recipient of the message divides the number of letters by 5 to find that there are 6 rows. She writes the coded message, six letters at a time, in columns 4, 1, 3, 2, 5, then reads the original message from the rows. ■

Notice that although the encoded message is a permutation of the original message string, this permutation depends on the length of the message. In Example 8, the permutation of the 30 positions begins

$$\begin{pmatrix} 1 & 2 & 3 & \cdots \\ 7 & 19 & 13 & \cdots \end{pmatrix}.$$

But using the keyword JONES to encode MAKE ME AN OFFER produces a permutation of 15 positions that begin

$$\begin{pmatrix} 1 & 2 & 3 & \cdots \\ 4 & 10 & 7 & \cdots \end{pmatrix}.$$

A common variation of this idea, used in the U.S. Civil War, is to transpose words rather than letters and add some superfluous known words for confusion. These extra words were called arbitraries during the Civil War.

■ Even and Odd Permutations

A cycle of length 2 is called a **transposition**. That is, a transposition is a cycle $p = (a_i, a_j)$, where $p(a_i) = a_j$ and $p(a_j) = a_i$.

Observe that if $p = (a_i, a_j)$ is a transposition of A, then $p \circ p = 1_A$, the identity permutation of A.

Every cycle can be written as a product of transpositions. In fact,

$$(b_1, b_2, \dots, b_r) = (b_1, b_r) \circ (b_1, b_{r-1}) \circ \cdots \circ (b_1, b_3) \circ (b_1, b_2).$$

This case can be verified by induction on r, as follows:

Basis Step

If $r = 2$, then the cycle is just (b_1, b_2), which already has the proper form.

Induction Step

We use P(k) to show P($k+1$). Let $(b_1, b_2, \dots, b_k, b_{k+1})$ be a cycle of length $k+1$. Then $(b_1, b_2, \dots, b_k, b_{k+1}) = (b_1, b_{k+1}) \circ (b_1, b_2, \dots, b_k)$, as may be verified by computing the composition. Using P(k), $(b_1, b_2, \dots, b_k) = (b_1, b_k) \circ (b_1, b_{k-1}) \circ \cdots \circ (b_1, b_2)$. Thus, by substitution,

$$(b_1, b_2, \dots, b_{k+1}) = (b_1, b_{k+1}) \circ (b_1, b_k) \circ \cdots \circ (b_1, b_3)(b_1, b_2).$$

This completes the induction step. Thus, by the principle of mathematical induction, the result holds for every cycle. For example,

$$(1, 2, 3, 4, 5) = (1, 5) \circ (1, 4) \circ (1, 3) \circ (1, 2).$$

We now obtain the following corollary of Theorem 2.

Corollary 1 Every permutation of a finite set with at least two elements can be written as a product of transpositions. ■

Observe that the transpositions in Corollary 1 need not be disjoint.

EXAMPLE 9 Write the permutation p of Example 7 as a product of transpositions.

Solution

We have $p = (7, 8) \circ (2, 4, 5) \circ (1, 3, 6)$. Since we can write

$$(1, 3, 6) = (1, 6) \circ (1, 3)$$
$$(2, 4, 5) = (2, 5) \circ (2, 4),$$

we have $p = (7, 8) \circ (2, 5) \circ (2, 4) \circ (1, 6) \circ (1, 3)$. ■

We have observed that every cycle can be written as a product of transpositions. However, this can be done in many different ways. For example,

$$\begin{aligned}(1, 2, 3) &= (1, 3) \circ (1, 2)\\ &= (2, 1) \circ (2, 3)\\ &= (1, 3) \circ (3, 1) \circ (1, 3) \circ (1, 2) \circ (3, 2) \circ (2, 3).\end{aligned}$$

It then follows that every permutation on a set of two or more elements can be written as a product of transpositions in many ways. However, the following theorem, whose proof we omit, brings some order to the situation.

Theorem 3 If a permutation of a finite set can be written as a product of an even number of transpositions, then it can never be written as a product of an odd number of transpositions, and conversely. ■

A permutation of a finite set is called **even** if it can be written as a product of an even number of transpositions, and it is called **odd** if it can be written as a product of an odd number of transpositions.

EXAMPLE 10 Is the permutation

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 4 & 5 & 7 & 6 & 3 & 1 \end{pmatrix}$$

even or odd?

Solution

We first write p as a product of disjoint cycles, obtaining

$$p = (3, 5, 6) \circ (1, 2, 4, 7). \qquad \text{(Verify this.)}$$

Next we write each of the cycles as a product of transpositions:

$$(1,2,4,7) = (1,7) \circ (1,4) \circ (1,2)$$
$$(3,5,6) = (3,6) \circ (3,5).$$

Then $p = (3,6) \circ (3,5) \circ (1,7) \circ (1,4) \circ (1,2)$. Since p is a product of an odd number of transpositions, it is an odd permutation. ■

From the definition of even and odd permutations, it follows (see Exercises 22 through 24) that

(a) The product of two even permutations is even.

(b) The product of two odd permutations is even.

(c) The product of an even and an odd permutation is odd.

Theorem 4 Let $A = \{a_1, a_2, \ldots, a_n\}$ be a finite set with n elements, $n \geq 2$. There are $n!/2$ even permutations and $n!/2$ odd permutations.

Proof

Let A_n be the set of all even permutations of A, and let B_n be the set of all odd permutations. We shall define a function $f: A_n \to B_n$, which we show is one to one and onto, and this will show that A_n and B_n have the same number of elements.

Since $n \geq 2$, we can choose a particular transposition q_0 of A. Say that $q_0 = (a_{n-1}, a_n)$. We now define the function $f: A_n \to B_n$ by

$$f(p) = q_0 \circ p, \qquad p \in A_n.$$

Observe that if $p \in A_n$, then p is an even permutation, so $q_0 \circ p$ is an odd permutation and thus $f(p) \in B_n$. Suppose now that p_1 and p_2 are in A_n and $f(p_1) = f(p_2)$. Then

$$q_0 \circ p_1 = q_0 \circ p_2. \tag{3}$$

We now compose each side of equation (3) with q_0:

$$q_0 \circ (q_0 \circ p_1) = q_0 \circ (q_0 \circ p_2);$$

so, by the associative property, $(q_0 \circ q_0) \circ p_1 = (q_0 \circ q_0) \circ p_2$ or, since $q_0 \circ q_0 = 1_A$,

$$1_A \circ p_1 = 1_A \circ p_2$$
$$p_1 = p_2.$$

Thus f is one to one.

Now let $q \in B_n$. Then $q_0 \circ q \in A_n$, and

$$f(q_0 \circ q) = q_0 \circ (q_0 \circ q) = (q_0 \circ q_0) \circ q = 1_A \circ q = q,$$

which means that f is an onto function. Since $f: A_n \to B_n$ is one to one and onto, we conclude that A_n and B_n have the same number of elements. Note that $A_n \cap B_n = \varnothing$ since no permutation can be both even and odd. Also, by Theorem 1, $|A_n \cup B_n| = n!$. Thus, by Theorem 2 of Section 1.2,

$$n! = |A_n \cup B_n| = |A_n| + |B_n| - |A_n \cap B_n| = 2|A_n|.$$

We then have

$$|A_n| = |B_n| = \frac{n!}{2}.$$ ■

5.4 Exercises

1. Which of the following functions $f: \mathbb{R} \to \mathbb{R}$ are permutations of $\mathbb{R}$?
   (a) f is defined by $f(a) = a - 1$.
   (b) f is defined by $f(a) = a^2$.

2. Which of the following functions $f: \mathbb{R} \to \mathbb{R}$ are permutations of $\mathbb{R}$?
   (a) f is defined by $f(a) = a^3$.
   (b) f is defined by $f(a) = e^a$.

3. Which of the following functions $f: Z \to Z$ are permutations of Z?
   (a) f is defined by $f(a) = a + 1$.
   (b) f is defined by $f(a) = (a-1)^2$.

4. Which of the following functions $f: Z \to Z$ are permutations of Z?
   (a) f is defined by $f(a) = a^2 + 1$.
   (b) f is defined by $f(a) = a^3 - 3$.

In Exercises 5 through 8, let $A = \{1,2,3,4,5,6\}$ *and*

$$p_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 1 & 2 & 6 & 5 \end{pmatrix},$$

$$p_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 5 & 4 & 6 \end{pmatrix},$$

$$p_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 3 & 2 & 5 & 4 & 1 \end{pmatrix}.$$

5. Compute
   (a) p_1^{-1} (b) $p_3 \circ p_1$

6. Compute
   (a) $(p_2 \circ p_1) \circ p_2$ (b) $p_1 \circ (p_3 \circ p_2^{-1})$

7. Compute
   (a) p_3^{-1} (b) $p_1^{-1} \circ p_2^{-1}$

8. Compute
   (a) $(p_3 \circ p_2) \circ p_1$ (b) $p_3 \circ (p_2 \circ p_1)^{-1}$

In Exercises 9 and 10, let $A = \{1,2,3,4,5,6,7,8\}$. *Compute the products.*

9. (a) $(3,5,7,8) \circ (1,3,2)$
   (b) $(2,6) \circ (3,5,7,8) \circ (2,5,3,4)$

10. (a) $(1,4) \circ (2,4,5,6) \circ (1,4,6,7)$
    (b) $(5,8) \circ (1,2,3,4) \circ (3,5,6,7)$

11. Let $A = \{a,b,c,d,e,f,g\}$. Compute the products.
    (a) $(a,f,g) \circ (b,c,d,e)$
    (b) $(f,g) \circ (b,c,f) \circ (a,b,c)$

In Exercises 12 and 13, let $A = \{1,2,3,4,5,6,7,8\}$. *Write each permutation as the product of disjoint cycles.*

12. (a) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 5 & 1 & 8 & 7 & 6 \end{pmatrix}$
    (b) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 1 & 7 & 5 & 8 & 6 \end{pmatrix}$

13. (a) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 5 & 7 & 8 & 4 & 3 & 2 & 1 \end{pmatrix}$
    (b) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 1 & 4 & 6 & 7 & 8 & 5 \end{pmatrix}$

14. Let $A = \{a,b,c,d,e,f,g\}$. Write each permutation as the product of disjoint cycles.
    (a) $\begin{pmatrix} a & b & c & d & e & f & g \\ g & d & b & a & c & f & e \end{pmatrix}$
    (b) $\begin{pmatrix} a & b & c & d & e & f & g \\ d & e & a & b & g & f & c \end{pmatrix}$

15. Let $A = \{1,2,3,4,5,6,7,8\}$. Write each permutation as a product of transpositions.
    (a) $(2,1,4,5,8,6)$ (b) $(3,1,6) \circ (4,8,2,5)$

16. Code the message WHERE ARE YOU by applying the permutation $(1,7,3,5,11) \circ (2,6,9) \circ (4,8,10)$.

17. Decode the message ATEHAOMOMNTI, which was encoded using the permutation

$$(3,7,1,12) \circ (2,5,8) \circ (4,10,6,11,9).$$

18. (a) Give the complete permutation of the positions for the message in Example 8.
    (b) Write the permutation found in part (a) as the product of disjoint cycles.

19. (a) Encode the message MAKE ME AN OFFER using the keyword JONES and the method of Example 8.
    (b) Write the permutation of the positions for the message in part (a).

In Exercises 20 and 21, let $A = \{1,2,3,4,5,6,7,8\}$. *Determine whether the permutation is even or odd.*

20. (a) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 2 & 1 & 6 & 5 & 8 & 7 & 3 \end{pmatrix}$
    (b) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 3 & 4 & 2 & 1 & 8 & 6 & 5 \end{pmatrix}$

21. (a) $(6,4,2,1,5)$
    (b) $(4,8) \circ (3,5,2,1) \circ (2,4,7,1)$

22. Prove that the product of two even permutations is even.

23. Prove that the product of two odd permutations is even.

24. Prove that the product of an even and an odd permutation is odd.

25. Let $A = \{1,2,3,4,5\}$. Let $f = (5,2,3)$ and $g = (3,4,1)$ be permutations of A. Compute each of the following and write the result as the product of disjoint cycles.
    (a) $f \circ g$ (b) $f^{-1} \circ g^{-1}$

26. Show that if p is a permutation of a finite set A, then $p^2 = p \circ p$ is a permutation of A.

27. Let $A = \{1, 2, 3, 4, 5, 6\}$ and

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 1 & 5 & 6 \end{pmatrix}$$

be a permutation of A.

(a) Write p as a product of disjoint cycles.

(b) Compute p^{-1}. (c) Compute p^2.

(d) Find the period of p, that is, the smallest positive integer k such that $p^k = 1_A$.

28. Let $A = \{1, 2, 3, 4, 5, 6\}$ and

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 5 & 1 & 2 & 6 \end{pmatrix}$$

be a permutation of A.

(a) Write p as a product of disjoint cycles.

(b) Compute p^{-1}. (c) Compute p^2.

(d) Find the period of p, that is, the smallest positive integer k such that $p^k = 1_A$.

29. (a) Use mathematical induction to show that if p is a permutation of a finite set A, then $p^n = p \circ p \circ \cdots \circ p$ (n factors) is a permutation of A for $n \in Z^+$.

(b). If A is a finite set and p is a permutation of A, show that $p^m = 1_A$ for some $m \in Z^+$.

30. Let p be a permutation of a set A. Define the following relation R on A: $a\ R\ b$ if and only if $p^n(a) = b$ for some $n \in Z$. [p^0 is defined as the identity permutation and p^{-n} is defined as $(p^{-1})^n$.] Show that R is an equivalence relation and describe the equivalence classes.

31. Build a composition table for the permutations of $A = \{1, 2, 3\}$ given in Example 2.

32. Describe how to use the composition table in Exercise 31 to identify p^{-1} for any permutation p of A.

33. Find all subsets of $\{1_A, p_1, p_2, p_3, p_4, p_5\}$, the permutations in Example 2, that satisfy the closure property for composition.

34. For each permutation, p, of A in Example 2, determine its period. How does this relate to the subset in Exercise 33 to which p belongs?

35. Let $A = \{1, 2, 3, \ldots, n\}$. How many permutations of A, $p = (a_1, a_2, \ldots, a_n)$, are there for which $a_i < a_{i+1}$, $1 \le i \le n-1$? How many permutations of A, $p = (a_1, a_2, \ldots, a_n)$, are there for which $a_i > a_{i+1}$, $1 \le i \le n-1$?

36. Let $A = \{1, 2, 3, 4, 5\}$. How many different sequences of length 3 can be formed using the elements of A and such that $a_1 < a_2 < a_3$?

37. We call a permutation $p = (a_1, a_2, \ldots, a_n)$ **up-down** if the elements in the odd positions form an increasing sequence and the elements in the even positions form a decreasing sequence.

(a) Let $A = \{1, 2, 3\}$. How many up-down permutations of A are there?

(b) Let $A = \{1, 2, 3, 4\}$. How many up-down permutations of A are there?

38. Let $A = \{1, 2, 3, 4, 5\}$. How many up-down permutations of A are there?

39. Prove that the number of up-down permutations for $A = \{1, 2, 3, \ldots, n\}$ is the same as the number of increasing sequences of length $\lceil \frac{n}{2} \rceil$ that can be formed from elements of A.

Tips for Proofs

Before beginning a proof, you might find it helpful to consider what the statement does *not* say. This can help clarify your thinking about what facts and tools are available for the proof. Consider Theorem 4, Section 5.1. It does not say that if f is one to one, then f is onto. The additional facts that $|A| = |B|$ and that f is everywhere defined will need to be used in the proof.

To show that a function is one to one or onto, we need to use generic elements. See Example 11, Section 5.1. Either the definition of one-to-oneness or its contrapositive may be used to prove this property. We also have the fact that if $f: A \to B$ is everywhere defined and $|A| = |B| = n$, then f is one to one if and only if f is onto. In addition, if we wish to show f is one to one and onto, we may do this by constructing the inverse function f^{-1}. Establishing a one-to-one correspondence is a powerful counting strategy, because it allows us to count a different set than the original one. For example, see Theorem 4, Section 5.4, and Exercises 32 through 34, Section 5.2.

To prove that f and g have the same order or one is of lower order than the other, the principal tools are the rules for Θ-classes or manipulation of inequalities (Section 5.3, Examples 2 and 3).

Key Ideas for Review

- Function: see page 169
- Identity function, 1_A: $1_A(a) = a$
- One-to-one function f from A to B: $a \neq a'$ implies $f(a) \neq f(a')$
- Onto function f from A to B: $\mathrm{Ran}(f) = B$
- Bijection: one-to-one and onto function
- One-to-one correspondence: onto, one-to-one, everywhere defined function
- If f is a function from A to B, $1_B \circ f = f$; $f \circ 1_A = f$
- If f is an invertible function from A to B, $f^{-1} \circ f = 1_A$; $f \circ f^{-1} = 1_B$
- $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$
- Substitution code: see page 176
- Boolean function f: $\mathrm{Ran}(f) \subseteq \{\text{true, false}\}$
- Hashing function: see page 180
- $O(g)$ (big Oh of g): see page 183
- f and g of the same order: f is $O(g)$ and g is $O(f)$
- Theorem: The relation Θ, $f\ \Theta\ g$ if and only if f and g have the same order, is an equivalence relation.
- Lower Θ-class: see page 185
- Rules for determining Θ-class of a function: see page 186
- Running time of an algorithm: Θ-class of a function that describes the number of steps performed by the algorithm
- Permutation function: a bijection from a set A to itself
- Theorem: If A is a set that has n elements, then there are $n!$ permutations of A.
- Cycle of length r: $(b_1, b_2, \ldots, b_r)$: see page 190
- Theorem: A permutation of a finite set that is not the identity or a cycle can be written as a product of disjoint cycles.
- Transposition: a cycle of length 2
- Corollary: Every permutation of a finite set with at least two elements can be written as a product of transpositions.
- Transposition code: see page 191
- Keyword columnar transposition code: see page 191
- Even (odd) permutation: one that can be written as a product of an even (odd) number of transpositions
- Theorem: If a permutation of a finite set can be written as a product of an even number of transpositions, then it can never be written as a product of an odd number of transpositions, and conversely.
- The product of

  (a) Two even permutations is even.

  (b) Two odd permutations is even.

  (c) An even and an odd permutation is odd.
- Theorem: If A is a set that has n elements, then there are $n!/2$ even permutations and $n!/2$ odd permutations of A.

Review Questions

1. How does a function differ from a general relation?

2. What is a common strategy to prove that a function is one-to-one?

3. What is a common strategy to prove that a function is onto?

4. Why are mod functions often used in constructing hashing functions?

5. What does the Θ-class of a function represent?

Chapter 5 Self-Test

1. Let $A = \{a, b, c, d\}$, $B = \{1, 2, 3\}$, and $R = \{(a, 2), (b, 1), (c, 2), (d, 1)\}$. Is R a function? Is R^{-1} a function? Explain your answers.

2. Let $A = B = \mathbb{R}$. Let $f: A \to B$ be the function defined by $f(x) = -5x + 8$. Show that f is one to one and onto.

3. Compute

(a) $\lfloor 16.29 \rfloor$ (b) $\lfloor -1.6 \rfloor$

4. Compute

(a) $\lceil 16.29 \rceil$ (b) $\lceil -1.6 \rceil$

5. Compute

(a) $lg(1)$ (b) $lg(64)$

6. Let Q be the propositional function defined by

$$Q(x): \exists y\ xy = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Evaluate $Q\left(\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}\right)$ and $Q\left(\begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}\right)$.

7. Assume that 9,500 account records need to be stored using the hashing function h, which takes the first two digits of the account number as one number and the last four digits as another number, adds them, and then applies the mod-89 function.

(a) How many linked lists will be needed?

(b) If an approximately even distribution of records is achieved, roughly how many records will be stored in each linked list?

(c) Compute $h(473810)$, $h(125332)$, and $h(308691)$.

8. Show that $f(n) = 2n^2 + 9n + 5$ is $O(n^2)$.

9. Determine the Θ-class of $f(n) = lg(n) + n^2 + 2^n$.

10. Consider the following pseudocode.

```
1. X ← 10
2. I ← 0
3. UNTIL (I > N)
        a. X ← X + I
        b. I ← I + 2
```

Write a function of N that describes the number of steps required and give the Θ-class of the function.

11. Let $A = \{1, 2, 3, 4, 5, 6\}$ and let $p_1 = (3, 6, 2)$ and $p_2 = (5, 1, 4)$ be permutations of A.

    (a) Compute $p_1 \circ p_2$ and write the result as a product of cycles and as the product of transpositions.

    (b) Compute $p_1^{-1} \circ p_2^{-1}$.

12. Let $p_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 3 & 2 & 1 & 4 & 5 & 6 \end{pmatrix}$ and $p_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 3 & 2 & 1 & 5 & 4 & 7 \end{pmatrix}$.

    (a) Compute $p_1 \circ p_2$.

    (b) Compute p_1^{-1}.

    (c) Is p_1 an even or odd permutation? Explain.

Coding Exercises

For each of the following, write the requested program or subroutine in pseudocode (as described in Appendix A) or in a programming language that you know. Test your code either with a paper-and-pencil trace or with a computer run.

1. Let $U = \{u_1, u_2, \ldots, u_n\}$ be the universal set for possible input sets. Write a function CHARFCN that given a set as input returns the characteristic function of the set as a sequence.
2. Write a function TRANSPOSE that, given an $n \times n$ matrix, returns its transpose.
3. Write a program that writes a given permutation as a product of disjoint cycles.
4. Write a program that writes a given permutation as a product of transpositions.
5. Use the program in Exercise 4 as a subroutine in a program that determines whether a given permutation is even or odd.

Experiment 5

The θ-class of a function that describes the number of steps performed by an algorithm is referred to as the **running time** of the algorithm. In this experiment you will analyze several algorithms, presented in pseudocode, to determine their running times.

Part I. The first algorithm is one method for computing the product of two $n \times n$ matrices. Assume that the matrices are each stored in an array of dimension 2 and that $\mathbf{A}[i, j]$ holds the element of $\mathbf{A}$ in row i and column j.

Algorithm MATMUL(**A**, **B**; **C**)

```
1. FOR I = 1 THRU N
     a. FOR J = 1 THRU N
          1. C[I,J] ← 0
          2. FOR K = 1 THRU N
               a. C[I,J] ← C[I,J] + A[I,K] × B[K,J]   ●
```

Assume that each assignment of a value, each addition, and each element multiplication take the same fixed amount of time.

1. How many assignments will be done in the second **FOR** loop?
2. How many element multiplications are done in the third **FOR** loop?
3. What is the running time of MATMUL? Justify your answer.

Part II. The following recursive algorithm will compute $n!$ for any positive integer n.

Algorithm FAC(N)

```
1. IF (N = 1) THEN
     a. A ← 1
2. ELSE
     a. A ← N × FAC(N − 1)
3. RETURN (A)                                  ●
```

1. Let S_n be the number of steps needed to calculate $n!$ using FAC. Write a recurrence relation for S_n in terms of S_{n-1}.
2. Solve the recurrence relation in question 1 and use the result to determine the running time of FAC.

Part III. The function SEEK will give the cell in which a specified value is stored in cells i through $i + n - 1$ (inclusive) of an array A. Assume that $i \geq 1$.

```
FUNCTION SEEK(ITEM, I, I + N − 1)
1. CELL ← 0
2. FOR J = I THRU I + N − 1
     a. IF (A[J] = ITEM) THEN
     b. CELL ← J
3. RETURN (CELL)
END OF FUNCTION SEEK                            ●
```

1. How many cells are there from $A[i]$ to $A[i + n - 1]$ (inclusive)?
2. Give a verbal description of how SEEK operates.
3. What is the running time of SEEK? Justify your answer.

Part IV. The algorithm HUNT will give the cell in which a specified value is stored in cells i through $i + n - 1$ (inclusive) of an array A. Assume that $i \geq 1$. To simplify the analysis of this algorithm, assume that n, the number of cells to be inspected, is a power of 2.

Algorithm HUNT ($ITEM, I, I + N - 1$)

```
1. CELL ← 0
2. IF (N = 1 AND A[I] = ITEM) THEN
     a. CELL ← I
3. ELSE
     a. CELL1 ← HUNT(ITEM, I, I + N/2 − 1)
     b. CELL2 ← HUNT(ITEM, I + N/2, I + N − 1)
4. IF (CELL1 ≠ 0) THEN
     a. CELL ← CELL1
5. ELSE
     a. CELL ← CELL2
6. RETURN (CELL)                                ●
```

1. Give a verbal description of how HUNT operates.
2. What is the running time of HUNT? Justify your answer.
3. Under what circumstances would it be better to use SEEK (Part III) rather than HUNT? When would it be better to use HUNT rather than SEEK?