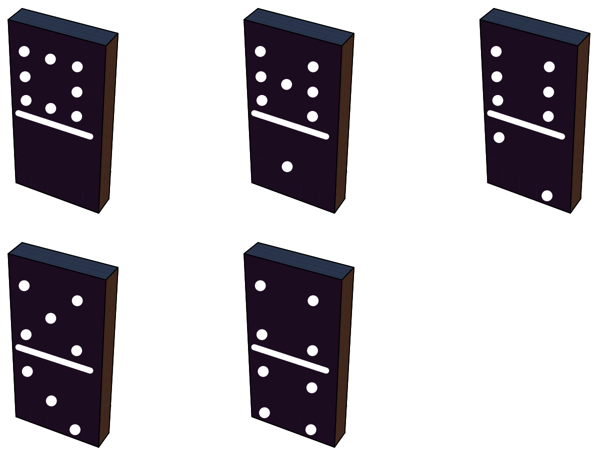
chapter 8



RECURSION AND RECURRENCE RELATIONS

GOALS

An essential tool that anyone interested in computer science must master is how to think recursively. The ability to understand definitions, concepts, algorithms, etc., that are presented recursively and the ability to put thoughts into a recursive framework are essential in computer science. One of our goals in this chapter is to help the reader become more comfortable with recursion in its commonly encountered forms.

A second goal is to discuss recurrence relations. We will concentrate on methods of solving recurrence relations, including an introduction to generating functions.

8.1 The Many Faces of Recursion

Consider the following definitions, all of which should be somewhat familiar to you. When reading them, concentrate on how they are similar.

Example 8.1.1. A very common alternate notation for the binomial coefficient $\binom{n}{k}$ is $C(n;\ k)$. We will use the latter notation in this chapter.

Here is a recursive definition of binomial coefficients.

> ***Definition: Binomial Coefficients.*** *Assume* $n \geq 0$ *and* $n \geq k \geq 0$.
> $C(n; 0) = 1$
> $C(n,\ n) = 1$
> *and* $C(n;\ k) = C(n-1;\ k) + C(n-1;\ k-1)$ *if* $n > k > 0$.

POLYNOMIALS AND THEIR EVALUATION

Definition: Polynomial Expression in x over S (Non-Recursive). *Let n be an integer,* $n \geq 0$. *An* n^{th} *degree polynomial in x is an expression of the form* $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$, *where* $a_n, a_{n-1}, \ldots, a_1, a_0$ *are elements of some designated set of numbers, S, called the set of coefficients and* $a_n \neq 0$.

We refer to x as a variable here, although the more precise term for x is an *indeterminate*. There is a distinction between the terms indeterminate and variable, but that distinction will not come into play in our discussions.

Zeroth degree polynomials are called constant polynomials and are simply elements of the set of coefficients.

This definition is often introduced in algebra courses to describe expressions such as $f(n) = 4n^3 + 2n^2 - 8n + 9$, a third-degree, or cubic, polynomial in n. This definitions has has drawbacks when the variable is given a value and the expression must be evaluated, For example, suppose that n = 7. Your first impulse is likely to do this:

$$\begin{aligned} f(7) &= 4\times 7^3 + 2\times 7^2 - 8\times 7 + 9 \\ &= 4\times 343 + 2\times 49 - 8\times 7 + 9 = 1423 \end{aligned}$$

A count of the number of operations performed shows that five multiplications and three additions/subtractions were performed. The first two multiplications compute 7^2 and 7^3, and the last three mutiply the powers of 7 times the coefficients. This gives you the four terms; and adding/-subtacting a list of k numbers requires $k-1$ addition/subtractions. The following definition of a polynomial expression suggests another more efficient method of evaluation.

Definition: Polynomial Expression in x over S (Recursive)*. Let S be a set of coefficients and x any variable.*

(a) A zeroth degree polynomial expression in x over S is a nonzero element of S.

(b) For $n \geq 1$, an n^{th} degree polynomial expression in x over S is an expression of the form $p(x)\,x + a$ where $p(x)$ is an $(n-1)^{st}$ degree polynomial expression in x and $a \in S$.

We can easily verify that f(n) is a third-degree polynomial expression in n over the $\mathbb{Z}$ based on this definition:

$$f(n) = (4n^2 + 2n - 8)\,n + 9 = ((4n + 2)\,n - 8)\,n + 9$$

Notice that 4 is a zeroth degree polynomial since it is an integer. Therefore $4n + 2$ is a first-degree polynomial; therefore, $(4n + 2)\,n - 8$ is a second-degree polynomial in n over $\mathbb{Z}$; therefore, $f(n)$ is a third-degree polynomial in n over $\mathbb{Z}$. The final expression for $f(n)$ is called its *telescoping form*. If we use it to calculate $f(7)$, we need only three multiplications and three additions /subtractions. This is called Horner's method for evaluating a polynomial expression.

Example 8.12. (a) The telescoping form of $p(x) = 5x^4 + 12x^3 - 6x^2 + x + 6$ is $(((5x + 12)\,x - 6)\,x + 1)\,x + 6$. Using Horner's method, computing the value of $p(c)$ requires four multiplications and four additions/subtractions for any real number c.

(b) $g(x) = -x^5 + 3x^4 + 2x^2 + x$ has the telescoping form $((((-x + 3)\,x)\,x + 2)\,x + 1)\,x$.

Many computer languages represent polynomials as lists of coefficients, usually starting with the constant term. For example, $g(x) = -x^5 + 3x^4 + 2x^2 + x$ would be represented with the list $\{0, 1, 2, 0, 3, -1\}$. In both *Mathematica* and Sage, polynomial expressions can be entered and manipulated, so the list representation is only internal. Some lower-leveled languages do require users to program polynomial operations with lists. We will leave these programming issues to another source.

Example 8.1.3. A recursive algorithm for a binary search of a sorted list of items: $r = \{r(1), r(2) \ldots, r(n)\}$ represent a list of n items sorted by a numeric key in descending order. The j^{th} item is denoted $r(j)$ and its key value by $r(j)$.key. For example, each item might contain data on the buildings in a city and the key value might be the height of the building. Then $r(1)$ would be the item for the tallest building. The algorithm BinarySearch(j, k) can be applied to search for an item in r with key value C. This would be accomplished by the execution of BinarySearch$(1, n)$. When the algorithm is completed, the variable `Found` will have a value of true if an item with the desired key value was found, and the value of `location` will be the index of an item whose key is C. If `Found` stays false, no such item exists in the list. The general idea behind the algorithm is illustrated in Figure 8.1.2.

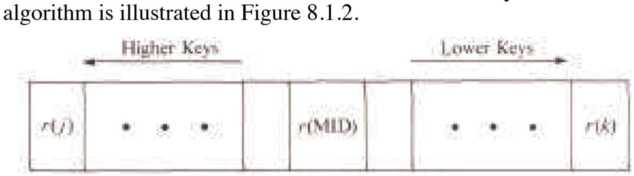


FIGURE 8.1.2 Illustration of BinarySearch

In this algorithm, `Found` and `location` are "global" variables to execution of the algorithm.

```
BinarySearch (j, k) :
 Found = False
 If J < K
   Then
      Mid = ⌊( j + k ) / 2⌋
      If r(Mid).key == C
        Then
          location = Mid
          Found = TRUE
        Else
          If r(Mid).key < C
             Then execute BinarySearch(j, Mid - 1)
             Else execute BinarySearch(Mid + 1 , k)
```

For the next two examples, consider a sequence of numbers to be a list of numbers consisting of a zeroth number, first number, second number, … . If a sequence is given the name S, the k^{th} number of S, is usually written S_k or $S(k)$.

Example 8.1.4 Define the sequence of numbers B by

$$B_0 = 100 \text{ and}$$

$$B_k = 1.08\ B_{k-1} \text{ for } k \geq 1$$

These rules stipulate that each number in the list is 1.08 times the previous number, with the starting number equal to 100. For example

$$\begin{aligned} B_3 &= 1.08\, B_2 \\ &= 1.08\,(1.08\, B_1) \\ &= 1.08\,(1.08\,(1.08\, B_0)) \\ &= 1.08\,(1.08\,(1.08\ 100)) \\ &= 1.08^3\ 100 \\ &= 125.971 \end{aligned}$$

Example 8.1.5. The Fibonacci sequence is the sequence F defined by

$$F_0 = 1,\ \ F_1 = 1 \text{ and}$$

$$F_k = F_{k-2} + F_{k-1} \text{ for } k \geq 2.$$

RECURSION

All of the previous examples were presented recursively. That is, every "object" is described in one of two forms. One form is by a simple definition, which is usually called the basis for the recursion. The second form is by a recursive description in which objects are described in terms of themselves, with the following qualification. What is essential for a proper use of recursion is that the objects can be expressed in terms of simpler objects, where "simpler" means closer to the basis of the recursion. To avoid what might be considered a circular definition, the basis must be reached after a finite number of applications of the recursion.

To determine, for example, the fourth item in the Fibonacci sequence we repeatedly apply the recursive rule for F until we are left with an expression involving F_0 and F_1:

$$\begin{aligned} F_4 &= F_2 + F_3 \\ &= (F_0 + F_1) + (F_1 + F_2) \\ &= (F_0 + F_1) + (F_1 + (F_0 + F_1)) \\ &= (1 + 1) + (1 + (1 + 1)) \\ &= 5 \end{aligned}$$

ITERATION

On the other hand, we could compute a term in the Fibonacci sequence, say F_5 by starting with the basis terms and working forward as follows:

$$\begin{aligned} F_2 &= F_0 + F_1 = 1 + 1 = 2 \\ F_3 &= F_1 + F_2 = 1 + 2 = 3 \\ F_4 &= F_2 + F_3 = 2 + 3 = 5 \\ F_5 &= F_3 + F_4 = 3 + 5 = 8 \end{aligned}$$

This is called an iterative computation of the Fibonacci sequence. Here we start with the basis and work our way forward to a less simple number, such as. Try to compute F_5 using the recursive definition for F as we did for F_4 . It will take much more time than it would have taken to do the computations above. Iterative computations usually tend to be faster than computations that apply recursion. Therefore, one useful skill is being able to convert a recursive formula into a nonrecursive formula, such as one that requires only iteration or a faster method, if possible.

An iterative formula for $C(n;\ k)$ is also much more efficient than an application of the recursive definition. The recursive definition is not without its merits, however. First, the recursive equation is often useful in manipulating algebraic expressions involving binomial coefficients. Second, it gives us an insight into the combinatoric interpretation of $C(n;\ k)$. In choosing k elements from $\{1,\ 2,\ \ldots,\ n\}$, there are $C(n-1;\ k)$ ways of choosing all k from $\{1, 2, \ldots, n-1\}$, and there are $C(n-1; k-1)$ ways of choosing the k elements if n is to be selected and the remaining $k-1$ elements come from $\{1,\ 2,\ \ldots,\ n-1\}$. Note how we used the Law of Addition from Chapter 2 in our reasoning.

BinarySearch Revisited. In the binary search algorithm, the place where recursion is used is easy to pick out. When an item is examined and the key is not the one you want, the search is cut down to a sublist of no more than half the number of items that you were searching in before. Obviously, this is a simpler search. The basis is hidden in the algorithm. The two cases that complete the search can be thought of as the basis. Either you find an item that you want, or the sublist that you have been left to search in is empty (j > k).

BinarySearch can be translated without much difficulty into any language that allows recursive calls to its subprograms. The advantage to such a program is that its coding would be much shorter than a nonrecursive program that does a binary search. However, in most cases the recursive version will be slower and require more memory at execution time.

INDUCTION AND RECURSION

The definition of the positive integers in terms of Peano's Postulates (Section 3.7) is a recursive definition. The basis element is the number 1 and the recursion is that if n is a positive integer, then so is its successor. In this case, n is the simple object and the recursion is of a forward type. Of course, the validity of an induction proof is based on our acceptance of this definition. Therefore, the appearance of induction proofs when recursion is used is no coincidence.

Example 8.1.6. A formula for the sequence B in Example 8.1.4 is $B = 100\,(1.08)^k$ for $k \geq 0$. A proof by induction follows: If $k = 0$, then $B = 100\,(1.08)^0 = 100$, as defined. Now assume that for some $k \geq 1$, the formula for B_k is true.

$$\begin{aligned} B_{k+1} &= 1.08\,B_k && \text{by the recursive definition} \\ &= 1.08\left(100\,(1.08)^k\right) && \text{by the induction hypothesis} \\ &= 100\;(1.08)^{k+1} && \text{hence the formula is true for } k+1 \end{aligned}$$

The formula that we have just proven for B is called a closed form expression. It involves no recursion or summation signs.

Definition: Closed Form Expression. *Let $E = E(x_1, x_2, \ldots, x_n)$ he an algebraic expression involving variables $x_1, x_2, \ldots, x_n$ which are allowed to take on values from some predetermined set. E is a closed form expression if there exists a number B such that the evaluation of E with any allowed values of the variables will take no more than B operations (alternatively, B time units).*

Example 8.1.7. The sum $E(n) = \sum_{k=1}^{n} k$ is not a closed form expression because the number of additions needed evaluate $E(n)$ grows indefinitely with n. A closed form expression that computes the value of $E(n)$ is $\frac{n(n+1)}{2}$, which only requires $B = 3$ operations.

EXERCISES FOR SECTION 8.1

A Exercises

1. By the recursive definition of binomial coefficients, $C(5;\,2) = C(4;\,2) + C(4;\,1)$. Continue expanding $C(5;\,2)$ to express it in terms of quantities defined by the basis. Check your result by applying the factorial definition of $C(n;\,k)$.

2. Define the sequence L by $L_0 = 5$ and for $k \geq 1$, $L_k = 2\,L_{k-1} - 7$. Determine L_4 and prove by induction that $L_k = 7 - 2^{k+1}$.

3. Let $p(x) = x^5 + 3\,x^4 - 15\,x^3 + x - 10$.

(a) Write $p(x)$ in telescoping form.

(b) Use a calculator to compute $p(3)$ using the original form of $p(x)$.

(c) Use a calculator to compute $p(3)$ using the telescoping form of $p(x)$.

(d) Compare your speed in parts b and c.

B Exercises

4. Suppose that a list of nine items, (r(l), r(2), . . . , r(9)), is sorted by key in decending order so that $r(3)$.key $= 12$ and $r(4)$.key $= 10$. List the executions of BinarySearch that would be needed to complete BinarySearch(1,9) for:

(a) C = 12
(b) C = 11

Assume that distinct items have distinct keys.

5. What is wrong with the following definition of $f : \mathbb{R} \to \mathbb{R}$?

$f(0) = 1$ and $f(x) = f(x/2)/2$ if $x \neq 0$.

8.2 Sequences

Definition: Sequence. *A sequence is a function from the natural numbers into some predetermined set. The image of any natural number k can be written interchangeably as $S(k)$ or S_k and is called the k^{th} term of S. The variable k is called the index or argument of the sequence.*

For example, a sequence of integers would be a function $S : \mathbb{N} \to \mathbb{Z}$.

Example 8.2.1.

(a) The sequence A defined by $A(k) = k^2 - k, \;\; k \geq 0$, is a sequence of integers.

(b) The sequence B defined recursively by $B(0) = 2$ and $B(k) = B(k-1) + 3$ for $k \geq 1$ is a sequence of integers. The terms of B can be computed either by applying the recursion formula or by iteration. For example;

$$\begin{aligned} B(3) &= B(2) + 3 \\ &= (B(1) + 3) + 3 \\ &= ((B(0) + 3) + 3) + 3) \\ &= ((2 + 3) + 3) + 3 \\ &= 11 \end{aligned}$$

or

$$B(1) = B(0) + 3 = 2 + 3 = 5$$

$$B(2) = B(1) + 3 = 5 + 8 = 8$$

$$B(3) = B(2) + 3 = 8 + 3 = 11.$$

(c) Let C_r be the number of strings of 0's and 1's of length r having no consecutive zeros. These terms define a sequence C of integers.

Remarks;

(1) A sequence is often called a *discrete function*.

(2) Although it is important to keep in mind that a sequence is a function, another useful way of visualizing a sequence is as a list. For example, the sequence A could be written as $(0, 0, 2, 6, 12, 20, \ldots)$. Finite sequences can appear much the same way when they are the input to or output from a computer. The index of a sequence can be thought of as a time variable. Imagine the terms of a sequence flashing on a screen every second. The s_k would be what you see in the k^{th} second. It is convenient to use terminology like this in describing sequences. For example, the terms that precede the k^{th} term of A would be $A(0), \; A(1), \; \ldots, \; A(k-1)$. They might be called the earlier terms.

A FUNDAMENTAL PROBLEM

Given the definition of any sequence, a fundamental problem that we will concern ourselves with is to devise a method for determining any specific term in a minimum amount of time. Generally, time can be equated with the number of operations needed. In counting operations, the application of a recursive formula would be considered an operation.

Example 8.2.2.

(a) The terms of A in Example 8.2.1 are very easy to compute because of the closed form expression. No matter what term you decide to compute, only three operations need to be performed.

(b) How to compute the terms of B is not so clear. Suppose that you wanted to know $B(100)$. One approach would be to apply the definition recursively:

$$B(100) = B(99) + 3 = (B(98) + 3) + 3 = \ldots$$

The recursion equation for B would be applied 100 times and 100 additions would then follow. To compute $B(k)$ by this method, $2k$ operations are needed. An iterative computation of $B(k)$ is an improvement:

$$\begin{aligned} B(1) &= B(0) + 3 = 2 + 3 = 5 \\ B(2) &= B(1) + 3 = 5 + 3 = 8 \end{aligned}$$

etc.

Only k additions are needed. This still isn't a good situation. As k gets large, we take more and more time to compute $B(k)$. The formula $B(k) = B(k-1) + 3$ is called a recurrence relation on B. The process of finding a closed form expression for $B(k)$, one that requires no more than some fixed number of operations, is called solving the recurrence relation.

(c) The determination of C_k is a standard kind of problem in combinatorics. One solution is by way of a recurrence relation. In fact, many problems in combinatorics are most easily solved by first searching for a recurrence relation and then solving it. The following observation will suggest the recurrence relation that we need to determine C_k : If $k \geq 2$, then every string of 0's and 1's with length k and no two consecutive 0's is either $1\,s_{k-1}$ or $01\,s_{k-2}$, where s_{k-1} and s_{k-2} are strings with no two consecutive 0's of length $k-1$ and $k-2$ respectively. From this observation we can see that $C_k = C_{k-2} + C_{k-1}$ for $k \geq 2$. The terms $C_0 = 1$ and $C_1 = 2$ are easy to determine by enumeration. Now, by iteration, any C_k can be easily determined. For example, $C_5 = 21$ can be computed with five additions. A closed form expression for C_k would be an improvement. Note that the recurrence relation for C_k is identical to the one for the Fibonacci sequence (Example 8.1.4). Only the basis is different.

EXERCISES FOR SECTION 8.2

A Exercises

1. Prove by induction that $B(k) = 3k + 2, k \geq 0$, is a closed form expression for the sequence B in Example 8.2.1.

2. (a) Consider sequence Q defined by $Q(k) = 2k + 9, k \geq 1$. Complete the table below and determine a recurrence relation that describes Q.

k	$Q(k)$	$Q(k) - Q(k-1)$
2		
3		
4		
5		
6		
7		

(b) Let $A(k) = k^2 - k, \quad k \geq 0$. Complete the table below and determine a recurrence relation for A. Notice that $(A(k) - A(k-1) - (A(k-1) - A(k-2)) = A(k) - 2A(k-1) + A(k-2)$

k	$A(k)$	$A(k) - A(k-1)$	$A(k) - 2A(k-1) + A(k-2)$
2			
3			
4			
5			

3. Given k lines $(k \geq 0)$ on a plane such that no two lines are parallel and no three lines meet at the same point, let $P(k)$ be the number of regions into which the lines divide the plane (including the infinite ones (see Figure 8.2.1). Describe geometrically how the recurrence relation $P(k) = P(k-1) + k$ can be obtained. Given that $P(0) = 1$, determine $P(5)$.

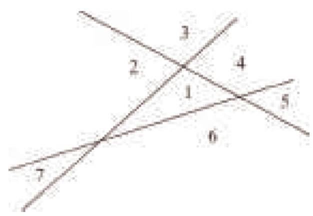


FIGURE 8.2.1 Exercise 3

4. A sample of a radioactive substance is expected to decay by 0.15 percent each hour. If $w_t, t \geq 0$, is the weight of the sample t hours into an experiment, write a recurrence relation for w.

B Exercise

5. Let $M(n)$ be the number of multiplications needed to evaluate an n^{th} degree polynomial. Use the recursive definition of a polynomial expression to define M recursively.

8.3 Recurrence Relations

In this section we will begin our study of recurrence relations and their solutions. Our primary focus will be on the class of finite order linear recurrence relations with constant coefficients (shortened to finite order linear relations). First, we will examine closed form expressions from which these relations arise. Second, we will present an algorithm for solving them. In later sections we will consider some other common relations (8.4) and introduce two additional tools for studying recurrence relations: generating functions (8.5) and matrix methods (Chapter 12).

Definition: Recurrence Relation. *Let S be a sequence of numbers, A recurrence relation on S is a formula that relates all but a finite number of terms of S to previous terms of S. That is, there is a k_0 in the domain of S such that if $k \geq k_0$, then S(k) is expressed in terms of some (and possibly all) of the terms that precede S(k). If the domain of S is $\{0, 1, 2, \ldots\}$, the terms $S(0), S(1), \ldots, S(k_0 - 1)$ are not defined by the recurrence formula. Their values are the initial conditions (or boundary conditions, or basis) that complete the definition of S.*

Example 8.3.1.

(a) The Fibonacci sequence is defined by the recurrence relation $F_k = F_{k-2} + F_{k-1},\ k \geq 2$, with the initial conditions $F_0 = 1$ and $F_1 = 1$. The recurrence relation is called a second-order relation because F_k depends on the two previous terms of F. Recall that the sequence C in Section 8.2 can be defined with the same recurrence relation, but with different initial conditions.

(b) The relation $T(k) = 2\,T(k-1)^2 - k\,T(k-3)$ is a third-order recurrence relation. If values of $T(0)$, $T(1)$, and $T(2)$ are specified, then T is completely defined.

(c) The recurrence relation $S(n) = S(\lfloor n/2 \rfloor) + 5$, $n > 0$, with $S(0) = 0$ has infinite order. To determine $S(n)$ when n is even, you must go back $n/2$ terms. Since $n/2$ grows unbounded with n, no finite order can be given to S.

SOLVING RECURRENCE RELATIONS

Sequences are often most easily defined with a recurrence relation; however, the calculation of terms by directly applying a recurrence relation can be time consuming. The process of determining a closed form expression for the terms of a sequence from its recurrence relation is called solving the relation. There is no single technique or algorithm that can be used to solve all recurrence relations. In fact, some recurrence relations cannot be solved. The relation that defines T above is one such example. Most of the recurrence relations that you are likely to encounter in the future as classified as finite order linear recurrence relations with constant coefficients. This class is the one that we will spend most of our time with in this chapter.

Definition: n^{th} Order Linear Recurrence Relation. *Let S be a sequence of numbers with domain $k \geq 0$. An n^{th} order linear recurrence relation on S with constant coefficients is a recurrence relation that can be written in the form*

$$S(k) + C_1\,S(k-1) + \ldots + C_n\,S(k-n) = f(k) \quad \text{for } k \geq n$$

where $C_1, C_2, \ldots, C_n$ are constants and f is a numeric function that is defined for $k \geq n$.

Note: We will shorten the name of this class of relations to n^{th} order linear relations. Therefore, in further discussions, $S(k) + 2\,k\,S(k-1) = 0$ would not be considered a first-order linear relation.

Example 8.3.2.

(a) The Fibonacci sequence is defined by the second-order linear relation because $F_k - F_{k-1} - F_{k-2} = 0$

(b) The relation $P(j) + 2\,P(j-3) = j^2$ is a third-order linear relation. In this case, $C_1 = C_2 = 0$.

(c) The relation $A(k) = 2\,(A(k-1) + k)$ can be written as $A(k) - 2\,A(k-1) = 2\,k$. Therefore, it is a first-order linear relation.

RECURRENCE RELATIONS OBTAINED FROM "SOLUTIONS"

Before giving an algorithm for solving finite order linear relations, we will examine recurrence relations that arise from certain closed form expressions. The closed form expressions are selected so that we will obtain finite order linear relations from them. This approach may seem a bit contrived, but if you were to write down a few simple algebraic expressions, chances are that most of them would be similar to the ones we are about to examine.

Example 8.3.3.

(a) Consider D, defined by $D(k) = 5 \cdot 2^k$, $k \geq 0$. If $k \geq 1$,

$$D(k) = 5 \cdot 2^k = 2 \cdot 5 \cdot 2^{k-1} = 2\,D(k-1).$$

Therefore, D satisfies the first order linear relation $D(k) - 2\,D(k-1) = 0$ and the initial condition $D(0) = 5$ serves as an initial condition for D.

(b) If $C(k) = 3^{k-1} + 2^{k+1} + k$, $k \geq 0$, quite a bit more algebraic manipulation is required to get our result:

$C(k) = 3^{k-1} + 2^{k+1} + k$	Original equation
$3\,C(k-1) = 3^{k-1} + 3 \cdot 2^k + 3\,(k-1)$	Substitute $k-1$ for k and multipy by 3
	Subtract the second equation from the first
$C(k) - 3\,C(k-1) = -2^k - 2\,k + 3$	3^{k-1} term is eliminated, this is a first order relation
$2\,C(k-1) - 6\,C(k-2) = -2^k - 2\,(2\,(k-1) + 3)$	Substitute $k-1$ for k in the 3rd equation, mult. by 2
	Subtract the fourth equation from the third equation
$C(k) - 5\,C(k-1) - 6\,C(k-2) = 2\,k - 7$	2^{k+1} term eliminated, this is a 2nd order relation

The recurrence relation that we have just obtained, defined for k ≥ 2, together with the initial conditions $C(0) = 7/3$ and $C(1) = 5$, define C. We could do more algebra to obtain a third-order linear relation in this case.

Table 8.3.1 summarizes our results together with a few other examples that we will let the reader derive. Based on these results, we might conjecture that any closed form expression for a sequence that combines exponential expressions and polynomial expressions will be solutions of finite order linear relations. Not only is this true, but the converse is true: a finite order linear relation defines a closed form expression that is similar to the ones that were just examined. The only additional information that is needed is a set of initial conditions.

Closed Form Expression	Recurrence Relation
$D(k) = 5 \cdot 2^k$	$D(k) - 2\,D(k-1) = 0$
$C(k) = 3^{k-1} + 2^{k+1} + k$	$C(k) - 2\,C(k-1) - 6\,C(k-2) = 2\,k - 7$
$Q(k) = 2\,k + 9$	$Q(k) - Q(k-1) = 2$
$A(k) = k^2 - k$	$A(k) - 2\,A(k-1) + A(k-2) = 2$
$B(k) = 2\,k^2 + 1$	$B(k) - 2\,B(k-1) + B(k-2) = 4$
$G(k) = 2 \cdot 4^k - 5\,(-3)^k$	$G(k) - G(k-1) + 12\,G(k-2) = 0$
$J(k) = (3 + k)\,2^k$	$J(k) - 4\,J(k-1) + 4\,J(k-2) = 0$

Table 8.3.1

Recurrence Relation Obtained from Certain Sequences

Definition: Homogeneous Recurrence Relation. *An n^{th} order linear relation is homogeneous if $f(k) = 0$ for all k. For each recurrence relation $S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = f(k)$, the associated homogeneous relation is $S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = 0$*

Example 8.3.4. $D(k) - 2\,D(k-1) = 0$ is a first-order homogeneous relation. Since it can also be written as $D(k) = 2\,D(k-1)$, it should be no surprise that it arose from an expression that involves powers of 2 (see Example 8.3.3a). More generally, you would expect that the solution of $L(k) - a\,L(k-1)$ would involve a^k . Actually, the solution is $L(k) = L(0)\,a^k$, where the value of $L(0)$ is given by the the initial condition.

Example 8.3.5. Consider the second-order homogeneous relation $S(k) - 7\,S(k-1) + 12\,S(k-2) = 0$ together with the initial conditions $S(0) = 4$ and $S(1) = 4$. From our discussion above, we can predict that the solution to this relation involves terms of the form $b\,a^k$, where b and a are nonzero constants that must be determined. If the solution were to equal this quantity exactly, then

$$S(k) = b\,a^k$$
$$S(k-1) = b\,a^{k-1}$$
$$S(k-2) = b\,a^{k-2}$$

Substitute these expressons into the recurrence relation to get

$$b\,a^k - 7\,b\,a^{k-1} + 12\,b\,a^{k-1} = 0 \qquad \text{(Eq 8.3 } a\text{)}$$

Each term on the left-hand side of the equation has a factor of $b\,a^{k-2}$, which is nonzero. Dividing through by this common factor yields

$$a^2 - 7\,a + 12 = (a-3)\,(a-4) = 0. \quad \text{(Eq 8.3b)}$$

Therefore, the only possible values of a are 3 and 4. Equation (8.3b) is called the characteristic equation of the recurrence relation. The fact is that our original recurrence relation is true for any sequence of the form $S(k) = b_1\,3^k + b_2\,4^k$, where b_1 and b_2 are real numbers. This set of sequences is called the general solution of the recurrence relation. If we didn't have initial conditions for S, we would stop here. The initial conditions make it possible for us to obtain definite values for b_1 and b_2.

$$\left\{\begin{array}{c} S(0) = 4 \\ S(1) = 4 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1\,3^0 + b_2\,4^0 = 4 \\ b_1\,3^1 + b_2\,4^1 = 4 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1 + b_2 = 4 \\ 3\,b_1 + 4\,b_2 = 4 \end{array}\right\}$$

The solution of this set of simultaneous equations is $b_1 = 12$ and $b_2 = -8$ and so the solution is $S(k) = 12\ 3^k - 8\ 4^k$.

Definition: Characteristic Equation. *The characteristic equation of the homogeneous n^{th} order linear relation $S(k) + C_1\,S(k-1) + \ldots + C_n\,S(k-n) = 0$ is the nth degree polynomial equation*

$$a^n + \sum_{j=1}^{n} C_j\, a^{n-j} = a^n + C_1\, a^{n-1} + \cdots + C_{n-1}\, x + C_n = 0$$

The left-hand side of this equation is called the characteristic polynomial.

Example 8.3.6.

(a) The characteristic equation of $F(k) - F(k-1) - F(k-2) = 0$ is $a^2 - a - 1 = 0$.

(b) The characteristic equation of $Q(k) + 2\,Q(k-1) - 3\,Q(k-2) - 6\,Q(k-4) = 0$ is $a^4 + 2\,a^3 - 3\,a^2 - 6 = 0$. Note that the absence of a $Q(k-3)$ term means that there is not an $x^{4-3} = x$ term appearing in the characteristic equation.

Algorithm 8.3.1: Algorithm for Solving Homogeneous n^{th} Order Linear Relations.

(a) Write out the characteristic equation of the relation $S(k) + C_1\, S(k-1) + \ldots + C_n\, S(k-n) = 0$, which is $a^n + C_1\, a^{n-1} + \cdots + C_{n-1}\, x + C_n = 0$.

(b) Find all roots of the characteristic equation, called characteristic roots.

(c) If there are n distinct characteristic roots, $a_1, a_2, \ldots, a_n$, then the general solution of the recurrence relation is $S(k) = b_1\, a_1^{\,k} + b_2\, a_2^{\,k} + \cdots + b_n\, a_n^{\,k}$. If there are fewer than n characteristic roots, then at least one root is a multiple root. If a_j is a double root, then the $b_j\, a_j^{\,k}$ term is replaced with $\left(b_{j0} + b_{j1}\, k\right) a_j^{\,k}$. In general, if a_j is a root of multiplicity p, then the $b_j\, a_j^{\,k}$ term is replaced with $\left(b_{j0} + b_{j1}\, k + \cdots + b_{j(p-1)}\, k^{p-1}\right) a_j^{\,k}$.

(d) If n initial conditions are given, obtain n linear equations in n unknowns (the b_j's from Step (c)) by substitution. If possible, solve these equations to determine a final form for $S(k)$.

Although this algorithm is valid for all values of n, there are limits to the size of n for which the algorithm is feasible. Using just a pencil and paper, we can always solve second-order equations. The quadratic formula for the roots of $a\,x^2 + b\,x + c = 0$ is

$$x = \frac{-b \pm \sqrt{b^2 - 4\,a\,c}}{2\,a}$$

The solutions of $a^2 + C_1\, a + C_2 = 0$ are then

$$\frac{1}{2}\left(-C_1 + \sqrt{C_1^{\,2} - 4\,C_2}\right) \quad \text{and} \quad \frac{1}{2}\left(-C_1 - \sqrt{C_1^{\,2} - 4\,C_2}\right)$$

Although cubic and quartic formulas exist, they are too lengthy to introduce here. For this reason, the only higher-order relations ($n \geq 3$) that you could be expected to solve by hand are ones for which there is an easy factorization of the characteristic polynomial.

Example 8.3.7. Suppose that T is defined by $T(k) = 7\,T(k-1) - 10\,T(k-2)$, with , $T(0) = 4$ and $T(1) = 17$. We can solve this recurrence relation with Algorithm 8.3.1:

(a) Note that we had written the recurrence relation in "nonstandard" form. To avoid errors in this easy step, you might consider a rearrangement of the equation to, in this case, $T(k) - 7\,T(k-1) + 10\,T(k-2) = 0$. Therefore, the characteristic equation is $a^2 - 7\,a + 10 = 0$.

(b) The characteristic roots are $\frac{1}{2}\left(7 + \sqrt{49 - 40}\right) = 5$ and $\frac{1}{2}\left(7 - \sqrt{49 - 40}\right) = 2$. These roots can be just as easily obtained by factoring the characteristic polynomial into $(a-5)(a-2)$.

(c) The general solution of the recurrence relation is $T(k) = b_1\, 2^k + b_2\, 5^k$,

(d) $$\left\{\begin{array}{c} T(0) = 4 \\ T(1) = 17 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1\, 2^0 + b_2\, 5^0 = 4 \\ b_1\, 2^1 + b_2\, 5^1 = 4 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1 + b_2 = 4 \\ 2\,b_1 + 5\,b_2 = 17 \end{array}\right\}$$

The simulations equations have the solution $b_1 = 1$ and $b_2 = 3$, Therefore, $T(k) = 2^k + 3 \cdot 5^k$.

Here is one rule that might come in handy: If the coefficients of the characteristic polynomial are all integers, with the constant term equal to m, then the only possible rational characteristic roots are divisors of m (both positive and negative).

With the aid of a computer (or possibly only a calculator), we can increase n. Approximations of the characteristic roots can be obtained by any of several well-known methods, some of which are part of standard software packages. There is no general rule that specifies the values of n for which numerical approximations will be feasible. The accuracy that you get will depend on the relation that you try to solve. (See Exercise 17 of this section.)

Example 8.3.8. Solve $S(k) - 7\,S(k-2) + 6\,S(k-3) = 0$, where $S(0) = 8$, $S(1) = 6$, and $S(2) = 22$.

(a) The characteristic equation is $a^3 - 7\,a + 6 = 0$.

(b) The only rational roots that we can attempt are ± 1, ± 2, ± 3, and ± 6. By checking these, we obtain the three roots 1, 2, and -3.

(c) The general solution is $S(k) = b_1\, 1^k + b_2\, 2^k + b_3 (-3)^k$. The first term can simply be written b_1 .

(d) $\left\{\begin{array}{c} S(0)=8 \\ S(1)=6 \\ S(20=22 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1+b_2+b_3=8 \\ b_1+2b_2-3b_3=6 \\ b_1+4b_2+9b_3=22 \end{array}\right\}$

You can solve this system by elimination to obtain $b_1=5$, $b_2=2$, and $b_3=1$. Therefore,

$$S(k) = 5 + 2\cdot 2^k + (-3)^k = 5 + 2^{k+1} + (-3)^k$$

Example 8.3.9. Solve $D(k) - 8D(k-1) + 16D(k-2) = 0$, where $D(2) = 16$ and $D(3) = 80$.

(a) Characteristic equation; $a^2 - 8a + 16 = 0$.

(b) $a^2 - 8a + 16 = (a-4)^2$. Therefore, there is a double characteristic root, 4.

(c) General solution: $D(k) = (b_{10} + b_{11}k)4^k$.

(d) $\left\{\begin{array}{c} D(2)=16 \\ D(3)=80 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} (b_{10}+b_{11}2)4^2=16 \\ (b_{10}+b_{11}3)4^3=80 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 16b_{10}+32b_{11}=16 \\ 64b_{10}+192b_{11}=80 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_{10}=\frac{1}{2} \\ b_{11}=\frac{1}{4} \end{array}\right\}$

Therefore $D(k) = (1/2 + (1/4)k)4^k = (2 + k)4^{k-1}$.

SOLUTION OF NONHOMOGENEOUS FINITE ORDER LINEAR RELATIONS

Our algorithm for nonhomogeneous relations will not be as complete as for the homogeneous case. This is due to the fact that different right-hand sides (f(k)'s) call for different procedures in obtaining a particular solution in Steps (b) and (c).

Algorithm 8.3.2: Algorithm for Solving Nonhomogeneous Finite Order Linear Relations.

To solve the recurrence relation $S(k) + C_1 S(k-1) + \ldots + C_n S(k-n) = f(k)$:

(a) Write the associated homogeneous relation and find its general solution (Steps (a) through (c) of Algorithm 8.3.1). Call this the homogeneous solution, $S^{(h)}(k)$.

(b) Start to obtain what is called a particular solution, $S^{(p)}(k)$ *of the recurrence relation by taking an educated guess at the form of a particular solu tion. For a large class of right-hand sides, this is not really a guess, since the particular solution is often the same type of function as* $f(k)$ *(see Table 8.3.2).*

Right Hand Side, $f(k)$	*Form of* a *particular Solution,* $S^{(p)}(k)$
constant, q	*constant,* d
linear function $q_0 + q_1 k$	*linear function* $d_0 + d_1 k$
m^{th} *degree polynomial,* $q_0 + q_1 k + \cdots + q_m k^m$	m^{th} *degree polynomial,* $d_0 + d_1 k + \cdots + d_m k^m$
exponential function $q\,a^k$	*exponential function* $d\,a^k$

Table 8.3.2
Particular Solutions for Given Right-hand Sides

(c) Substitute your guess from Step (b) into the recurrence relation. If you made a good guess, you should be able to determine the unknown coefficients of your guess. If you made a wrong guess, it should be apparent from the result of this substitution, so go back to Step (b).

(d) The general solution of the recurrence relation is the sum of the homogeneous and particular solutions. If no conditions are given, then you are finished. If n initial conditions are given, they will translate to n linear equations in n unknowns and solve the system, if possible, to get a complete solution.

Example 8.3.10. Solve $S(k) + 5S(k-1) = 9$, with $S(0) = 6$.

(a) The associated homogeneous relation, $S(k) + 5S(k-1) = 0$ has the characteristic equation $a + 5 = 0$; therefore, $a = -5$. The homogeneous solution is $S^{(h)}(k) = b(-5)^k$.

(b) Since the right-hand side is a constant, we guess that the particular solution will be a constant, d.

(c) If we substitute $S^{(p)}(k) = d$ into the recurrence relation, we get $d + 5d = 9$, or $6d = 9$. Therefore, $S^{(p)}(k) = 1.5$

(d) The general solution of the recurrence relation is

$$S(k) = S^{(h)}(k) + S^{(p)}(k) = b(-5)^k + 1.5$$

The initial condition will give us one equation to solve in order to determine b.

$$S(0) = 6 \Rightarrow b(-5)^0 + 1.5 = 6 \Rightarrow b + 1.5 = 6$$

Therefore, $b = 4.5$ and $S(k) = 4.5(-5)^k + 1.5$.

Example 8.3.11. Consider $T(k) - 7T(k-1) + 10T(k-2) = 6 + 8k$ with $T(0) = 1$ and $T(1) = 2$.

(a) From Example 8.3.7, we know that $T^{(h)}(k) = b_1 2^k + b_2 5^k$. Caution: Don't apply the initial conditions to $T^{(h)}$ until you add $T^{(p)}$!

(b) Since the right-hand side is a linear polynomial, $T^{(p)}$ is linear; that is, $T^{(p)}(k) = d_0 + d_1 k$.

(c) Substitution into the recurrence relation yields:

$$(d_0 + d_1 k) - 7(d_0 + d_1(k-1)) + 10(d_0 + d_1(k-2)) = 6 + 8k$$

$$\Rightarrow (4d_0 - 13d_1) + (4d_1)k = 6 + 8k$$

Two polynomials are equal only if their coefficients are equal. Therefore,

$$\left\{\begin{array}{c} 4d_0 - 13d_1 = 6 \\ 4d_1 = 8 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} d_0 = 8 \\ d_1 = 2 \end{array}\right\}$$

(d) Use the general solution $T(k) = b_1 2^k + b_2 5^k + 8 + 2k$ and the initial conditions to get a final solution:

$$\left\{\begin{array}{c} T(0) = 1 \\ T(1) = 2 \end{array}\right\} \Rightarrow \left\{\begin{array}{c} b_1 + b_2 + 8 = 1 \\ 2b_1 + 5b_2 + 10 = 2 \end{array}\right\}$$

$$\Rightarrow \left\{\begin{array}{c} b_1 + b_2 = -7 \\ 2b_1 + 5b_2 = -8 \end{array}\right\}$$

$$\Rightarrow \left\{\begin{array}{c} b_1 = -9 \\ b_2 = 2 \end{array}\right\}$$

Therefore, $T(k) = -9 \cdot 2^k + 2 \cdot 5^k + 8 + 2k$

A quick note on interest rates: When a quantity, such as a savings account balance, is increased by some fixed percent, it is most easily computed with a multipier. In the case of an 8% increase, the multier is 1.08 because any original amount A, has $0.08\,A$ added to it, so that the new balance is

$$A + 0.08\,A = (1 + 0.08)\,A = 1.08\,A\,.$$

Another example is that if the interest rate is 3.5%, the multiplier would be 1.035. This presumes that the interest is applied a the end of year for 3.5% annual interest, often called *simple interest*. If the interest is applied monthly, and we assume a simplifed case where each month has the same length, the multiplier after every month would be $\left(1 + \frac{0.35}{12}\right) \approx 1.0292$. After a year passes, this multiplier would be applied 12 times, which is the same as multiplying by $1.0292^{12} \approx 1.3556$. That increase from 1.035 to 1.3556 is the effect of *compound interest*.

Example 8.3.12. Suppose you open a savings account that pays an annual interest rate of 8%. In addition, suppose you decide to deposit one dollar when you open the account, and you intend to double your deposit each year. Let $B(k)$ be your balance after k years. B can be described by the relation $B(k) = 1.08\,B(k-1) + 2^k$, with $S(0) = 1$. If, instead of doubling the deposit each year, you deposited a constant amount, q, the 2^k term would be replaced with q, A sequence of regular deposits such as this is called an annuity.

Returning to the original situation, we can obtain a closed form expression for $B^{(h)}$:

(a) $B^{(h)}(k) = b_1(1.08)^k$

(b) $B^{(p)}(k)$ should be of the form $d\,2^k$.

(c) $d\,2^k = 1.08\,d\,2^{k-1} + 2^k$

$$\Rightarrow (2d)\,2^{k-1} = 1.08\,d\,2^{k-1} + 2 \cdot 2^{k-1}$$
$$\Rightarrow 2d = 1.08\,d + 2$$
$$\Rightarrow .92\,d = 2$$
$$\Rightarrow d = 2.174 \text{ (to the nearest thousandth)}$$

Therefore $B^{(p)}(k) = 2.174 \cdot 2^k$

(d) $B(0) = 1 \Rightarrow b_1 + 2.174 = 1$

$$\Rightarrow b_1 = -1.174$$

$$B(k) = -1.174 \cdot 1.08^k + 2.174 \cdot 2^k.$$

Example 8.3.13. Find the general solution to $S(k) - 3S(k-1) - 4S(k-2) = 4^k$.

(a) The characteristic roots of the associated homogeneous relation are -1 and 4. Therefore, $S^{(h)}(k) = b_1(-1)^k + b_2 4^k$.

(b) A function of the form $d\,4^k$ will not be a particular solution of the nonhomogeneous relation since it solves the associated homogeneous relation. When the right-hand side involves an exponential function with a base that equals a characteristic root, you should multiply your guess at a particular solution by k. Our guess at $S^{(p)}(k)$ would then be $d\,k\,4^k$. See below for a more complete description of this procedure.

(c) Substitute $d\,k\,4^k$ into the recurrence relation for $S(k)$:

$$d\,k\,4^k - 3\,d\,(k-1)\,4^{k-1} - 4\,d\,(k-2)\,4^{k-2} = 4^k$$

$$16\,d\,k\,4^{k-2} - 12\,d\,(k-1)\,4^{k-2} - 4\,d\,(k-2)\,4^{k-2} = 4^k$$

Each term on the left-hand side has a factor of 4^{k-2}

$$I6\,d\,k - 12\,d\,(k-1) - 4\,d(k-2) = 4^2$$
$$20\,d = 16 \;\Rightarrow\; d = 0.8$$

Therefore, $S^{(p)}(k) = 0.8\,k\;4^4$

(d) The general solution to the recurrence relation is

$$S(k) = b_1(-1)^k + b_2\,4^k + 0.8\,k\,4^k$$

BASE OF RIGHT-HAND SIDE EQUAL TO CHARACTERISTIC ROOT

If the right-hand side of a nonhomogeneous relation involves an exponential with base a, and a is also a characteristic root of multiplicity p, then multiply your guess at a particular solution as prescribed in Table 8,3.2 by k^p , where k is the index of the sequence.

Example 8.3.14.

(a) If $S(k) - 9\,S(k-1) + 20\,S(k-2) = 2\cdot 5^k$, the characteristic roots are 4 and 5. $S^{(p)}(k)$ will take the form $d\,k\,5^k$.

(b) If $S(n) - 6\,S(n-1) + 9\,S(n-2) = 3^{n+1}$ the only characteristic root is 3, but it is a double root (multiplicity 2). Therefore, the form of the particular solution is $d\,n^2\,3^n$.

(c) If $Q(j) - Q(j-1) - 12\,Q(j-2) = (-3)^j + 6\cdot 4^j$, the characteristic roots are -3 and 4. The form of the particular solution will be $d_1\,j\,(-3)^j + d_2\,j\cdot 4^j$.

(d) If $S(k) - 9\,S(k-1) + 8\,S(k-2) = 9\,k + 1 = (9\,k + 1)\,1^k$, the characteristic roots are 1 and 8. If the right-hand side is a polynomial, as it is in this case, then the exponential factor 1^k can be introduced. The particular solution will take the form $k(d_0 + d_1\,k)$.

We conclude this section with a comment on the situation in which the characteristic equation gives rise to complex roots. If we restrict the coefficients of our finite order linear relations to real numbers, or even to integers, we can still encounter characteristic equations whose roots are complex. Here, we will simply take the time to point out that our algorithms are still valid with complex characteristic roots, but the customary method for expressing the solutions of these relations is different. Since an understanding of these representations requires some background in complex numbers, we will simply suggest that an interested reader can refer to a more advanced treatment of recurrence relations (see also difference equations).

EXERCISES FOR SECTION 8.3

A Exercises

Solve the following sets of recurrence relations and initial conditions:

1. $S(k) - 10\,S(k-1) + 9\,S(k-2) = 0,\; S(0) = 3,\; S(1) = 11$

2. $S(k) - 9\,S(k-1) + 18\,S(k-2) = 0\;\; S(0) = 0,\; S(1) = 3$

3. $S(k) - 0.25\,S(k-1) = 0\,,\; S(0) = 6$

4. $S(k) - 20\,S(k-1) + 100\,S(k-2) = 0,\; S(0) = 2,\; S(1) = 50$

5. $S(k) - 2\,S(k-1) + S(k-2) = 2\;\; S(0) = 25,\; S(1) = 16$

6. $S(k) - S(k-1) - 6\,S(k-2) = -30\;\; S(0) = 7,\; S(1) = 10$

7. $S(k) - 5\,S(k-1) = 5^k,\;\; S(0) = 3$

8. $S(k) - 5\,S(k-1) + 6\,S(k-2) = 2,\;\; S(0) = -1,\; S(1) = 0$

9. $S(k) - 4\,S(k-1) + 4\,S(k-2) = 3\,k + 2^k.\; S(0) = 1,\; S(1) = 1$

10. $S(k) = r\,S(k-1) + a\,,\;\; S(0) = 0,\;\; r,\; a \geq 0,\; r \neq 1$

11. $S(k) - 4\,S(k-1) - 11\,S(k-2) + 30\,S(k-3) = 0,$

    $S(0) = 0,\;\; S(1) = -35,\;\; S(2) = -85$

12. Find a closed form expression for $P(k)$ in Exercise 3 of Section 8.2.

13. (a) Find a closed form expression for the terms of the Fibonacci sequence (see Example 8.1.4).

(b) The sequence C was defined by C_r = the number of strings of zeros and ones with length r having no consecutive zeros (Example 8.2.1(c)). Its recurrence, relation is the same as that of the Fibonacci sequence. Determine a closed form expression for C_r, $r \geq 1$,

14. If $S(n) = \sum_{j=1}^{n} g(j)$, $n \geq 1$, then S can be described with the recurrence relation $S(n) = S(n-1) + g(n)$. For each of the following sequences that are defined using a summation, find a closed form expression:

(a) $S(n) = \sum_{j=1}^{n} j$, $n \geq 1$

(b) $Q(n) = \sum_{j=1}^{n} j^2$, $n \geq 1$

(c) $P(n) = \sum_{j=1}^{n} \left(\frac{1}{2}\right)^j$, $n \geq 0$

(d) $T(n) = \sum_{j=1}^{n} j^3$, $n \geq 1$

B Exercises

15. Let $D(n)$ be the number of ways that the set $\{1, 2, \ldots, n\}$, $n \geq 1$, can be partitioned into two nonempty subsets.

(a) Find a recurrence relation for D. (Hint: It will be a first-order linear relation.)

(b) Solve the recurrence relation.

16. If you were to deposit a certain amount of money at the end of each year for a number of years, this sequence of payment would be called an annuity (see Example 8.3.12,).

(a) Find a closed form expression for the balance or value of an annuity that consists of payments of q dollars at a rate of interest of i. Note that for a normal annuity, the first payment is made after one year.

(b) With an interest rate of 12.5%, how much would you need to deposit into an annuity to have a value of one million dollars after 18 years?

(c) The payment of a loan is a form of annuity in which the initial value is some negative amount (the amount of the loan) and the annuity ends when the value is raised to zero. How much could you borrow if you can afford to pay $5,000 per year for 25 years at 14% interest?

C Exercises

17. Suppose that C is a small positive number. Consider the recurrence relation $B(k) - 2B(k-1) + \left(1 - C^2\right)B(k-2) = C^2$, with initial conditions $B(0) = 1$ and $B(1) = 1$. If C is small enough, we might consider approximating the relation by replacing $1 - C^2$ with 1 and C^2 with 0. Solve the original relation and its approximation. Let B_a a be the solution of the approximation. Compare closed form expressions for $B(k)$ and $B_a(k)$. Their forms are very different because the characteristic roots of the original relation were close together and the approximation resulted in one double characteristic root. If characteristic roots of a relation are relatively far apart, this problem will not occur. For example, compare the general solutions of

$S(k) + 1.001\, S(k-1) - 2.004002\, S(k-2) = 0.0001$ and

$S_a(k) + S_a(k-1) - 2\, S_a(k-2) = 0$.