UMN
UNIVERSITAS MULTIMEDIA NUSANTARA

IF120 Discrete Mathematics

09 Recurrence Relation

Angga Aditya Permana, Januar Wahjudi, Yaya Suryana, Meriske, Muhammad Fahrury Romdendine

REVIEW

- Discrete Probability
- Binomial Coefficient
- Combinatorial Identity

OUTLINE

- Recursive Algorithm
- Recurrence Relations
- Solving Recurrence Relations

Recursive Algorithm

- A **recursive function** is a function that invokes itself.
- A **recursive algorithm** is an algorithm that contains a recursive function.
- Recursion is a powerful, elegant, and natural way to solve a large class of problems.
- A problem in this class can be solved using a **divide-and-conquer** technique in which the problem is decomposed into problems of the same type as the original problem.

Computing n Factorial

This recursive algorithm computes $n!$.

Input: n, an integer greater than or equal to 0

Output: $n!$

```
factorial(n) {
    if (n == 0)
        return 1
    return n * factorial(n − 1)
}
```

Recursive Algorithm

- A robot can take steps of 1 meter or 2 meters. We write an algorithm to calculate the number of ways the robot can walk *n* meters. As examples:

Distance	*Sequence of Steps*	*Number of Ways to Walk*
1	1	1
2	1, 1 or 2	2
3	1, 1, 1 or 1, 2 or 2, 1	3
4	1, 1, 1, 1 or 1, 1, 2 or 1, 2, 1 or 2, 1, 1 or 2, 2	5

Let *walk*(*n*) denote the number of ways the robot can walk *n* meters. We have observed that

$$walk(1) = 1, walk(2) = 2.$$

Now suppose that $n > 2$. The robot can begin by taking a step of 1 meter or a step of 2 meters. If the robot begins by taking a 1-meter step, a distance of $n-1$ meters remains; but, by definition, the remainder of the walk can be completed in $walk(n-1)$ ways.

Similarly, if the robot begins by taking a 2-meter step, a distance of $n-2$ meters remains and, in this case, the remainder of the walk can be completed in *walk*($n-2$) ways. Since the walk must begin with either a 1-meter or a 2-meter step, all of the ways to walk n meters are accounted for. We obtain the formula

Robot Walking

This algorithm computes the function defined by

$$walk(n) = \begin{cases} 1, & n = 1 \\ 2, & n = 2 \\ walk(n-1) + walk(n-2) & n > 2. \end{cases}$$

Input: n

Output: $walk(n)$

```
walk(n) {
   if (n == 1 ∨ n == 2)
      return n
   return walk(n − 1) + walk(n − 2)
}
```

$$walk(n) = walk(n-1) + walk(n-2).$$

For example,

$$walk(4) = walk(3) + walk(2) = 3 + 2 = 5.$$

We can write a recursive algorithm to compute *walk*(n) by translating the equation

$$walk(n) = walk(n-1) + walk(n-2)$$

directly into an algorithm.

The base cases are $n = 1$ and $n = 2$.

Recurrence Relations

- Definition 9.1: A **recurrence relation** for the sequence $a_0, a_1, \ldots$ is an equation that relates a_n to certain of its predecessors $a_0, a_1, \ldots, a_{n-1}$.
- **Initial conditions** for the sequence $a_0, a_1, \ldots$ are explicitly given values for a finite number of the terms of the sequence.

❑ The Fibonacci sequence is defined by the recurrence relation

$$f_n = f_{n-1} + f_{n-2}, \qquad n \geq 3,$$

and initial conditions

$$f_1 = 1, \qquad f_2 = 1.$$

❑ Let S_n denote the number of subsets of an n-element set. Since going from an $(n - 1)$-element set to an n-element set doubles the number of subsets (remember the Set Theory), we obtain the recurrence relation

$$S_n = 2S_{n-1}.$$

The initial condition is

$$S_0 = 1.$$

Recurrence Relations

- A person invests \$1000 at 12 percent interest compounded annually. If A_n represents the amount at the end of *n* years, find a recurrence relation and initial conditions that define the sequence $\{A_n\}$.

  At the end of *n* − 1 years, the amount is A_{n-1}. After one more year, we will have the amount A_{n-1} plus the interest. Thus

$$A_n = A_{n-1} + (0.12)A_{n-1} = (1.12)A_{n-1}, \qquad n \geq 1.$$

Computing Compound Interest

This recursive algorithm computes the amount of money at the end of *n* years assuming an initial amount of \$1000 and an interest rate of 12 percent compounded annually.

Input: *n*, the number of years

Output: The amount of money at the end of *n* years

```
compound_interest(n) {
    if (n == 0)
        return 1000
    return 1.12 * compound_interest(n − 1)
}
```

To apply this recurrence relation for *n* = 1, we need to know the value of A_0.
Since A_0 is the beginning amount, we have the initial condition

$$A_0 = 1000.$$

Solving Recurrence Relations

- To solve a recurrence relation involving the sequence $a_0, a_1, \ldots$ is to find an explicit formula for the general term a_n.
- In this section we discuss two methods of solving recurrence relations:

1. **iteration** and
2. **linear homogeneous recurrence relations with constant coefficients.**

- To solve a recurrence relation involving the sequence $a_0, a_1, \ldots$ by **iteration**, we use the recurrence relation to write the *n*th term a_n in terms of certain of its predecessors $a_{n-1}, \ldots, a_0$.
- We then successively use the recurrence relation to replace each of $a_{n-1}, \ldots$ by certain of their predecessors.
- We continue until an explicit formula is obtained.

Iteration Method

- We can solve the recurrence relation

$$a_n = a_{n-1} + 3, \quad (9.1)$$

subject to the initial condition

$$a_1 = 2,$$

by iteration. Replacing *n* by *n* − 1 in (9.1), we obtain

$$a_{n-1} = a_{n-2} + 3.$$

If we substitute this expression for a_{n-1} into (9.1), we obtain

$$\begin{aligned} a_n &= a_{n-2} + 3 + 3 \\ &= a_{n-2} + 2 \cdot 3. \quad (9.2) \end{aligned}$$

Replacing *n* by *n* − 2 in (9.1), we obtain

$$a_{n-2} = a_{n-3} + 3.$$

If we substitute this expression for a_{n-2} into (9.2), we obtain

$$\begin{aligned} a_n &= a_{n-3} + 3 + 2 \cdot 3 \\ &= a_{n-3} + 3 \cdot 3. \end{aligned}$$

Iteration Method

In general, we have

$$a_n = a_{n-k} + k \cdot 3.$$

If we set $k = n - 1$ in this last expression, we have

$$a_n = a_1 + (n - 1) \cdot 3.$$

Since $a1 = 2$, we obtain the explicit formula

$$a_n = 2 + 3(n - 1)$$

for the sequence a.

- We can solve the recurrence relation

$$S_n = 2S_{n-1}$$

subject to the initial condition

$$S_0 = 1,$$

by iteration:

$$S_n = 2S_{n-1} = 2(2S_{n-2}) = \cdots = 2^n S_0 = 2^n.$$

Iteration Method

- Assume that the deer population of Rustic County is 1000 at time $n = 0$ and that the increase from time $n - 1$ to time n is 10 percent of the size at time $n - 1$. Write a recurrence relation and an initial condition that define the deer population at time n and then solve the recurrence relation.

  Let d_n denote the deer population at time n. We have the initial condition

  $$d_0 = 1000.$$

  The increase from time $n - 1$ to time n is $d_n - d_{n-1}$. Since this increase is 10 percent of the size at time $n - 1$, we obtain the recurrence relation

  $$d_n - d_{n-1} = 0.1\, d_{n-1},$$

  which may be rewritten

  $$d_n = 1.1 d_{n-1}.$$

  The recurrence relation may be solved by iteration:

  $$\begin{aligned} d_n &= 1.1 d_{n-1} = 1.1(1.1 d_{n-2}) = (1.1)^2 d_{n-2} \\ &= \cdots = (1.1)^n d_0 = (1.1)^n 1000. \end{aligned}$$

  The assumptions imply exponential population growth.

Linear Homogeneous Recurrence Relation

- Definition 9.2: A **linear homogeneous recurrence relation of order k with constant coefficients** is a recurrence relation of the form
$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}, \qquad c_k \neq 0.$$
- Notice that a linear homogeneous recurrence relation of order k with constant coefficients, together with the k initial conditions
$$a_0 = C_0, a_1 = C_1, \ldots, a_{k-1} = C_{k-1},$$
uniquely defines a sequence $a_0, a_1, \ldots$.

❑ The recurrence relations
$$S_n = 2S_{n-1}$$
and
$$f_n = f_{n-1} + f_{n-2},$$
which defines the Fibonacci sequence, are both linear homogeneous recurrence relations with constant coefficients. The first recurrence relation is of **order** 1 and the second is of **order** 2.

- The recurrence relation

$$a_n = 3a_{n-1}a_{n-2}$$

is **not** a linear homogeneous recurrence relation with constant coefficients. In a linear homogeneous recurrence relation with constant coefficients, each term is of the form ca_k. Terms such as $a_{n-1}a_{n-2}$ are not permitted. Recurrence relations such as the example above are said to be **nonlinear**.

- The recurrence relation

$$a_n - a_{n-1} = 2n$$

is **not** a linear homogeneous recurrence relation with constant coefficients because the expression on the right side of the equation is not zero. (Such an equation is said to be **inhomogeneous**.)

- The recurrence relation

$$a_n = 3na_{n-1}$$

is **not** a linear homogeneous recurrence relation with constant coefficients because the coefficient $3n$ is not constant. It is a linear homogeneous recurrence relation with nonconstant coefficients.

Linear Homogeneous Recurrence Relation

- Theorem 9.1: Let

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} \quad (9.3)$$

be a second-order, linear homogeneous recurrence relation with constant coefficients.
If S and T are solutions of (9.3), then $U = bS + dT$ is also a solution of (9.3).
If r is a root of

$$t^2 - c_1 t - c_2 = 0, \qquad (9.4)$$

then the sequence $r^n, n = 0, 1, \ldots,$ is a solution of (9.3).
If a is the sequence defined by (9.3),

$$a_0 = C_0, \ a_1 = C_1, \qquad (9.5)$$

and r_1 and r_2 are roots of (9.4) with $r_1 \neq r_2$, then there exist constants b and d such that

$$a_n = br_1^n + dr_2^n, \qquad n = 0,1, \ldots.$$

Linear Homogeneous Recurrence Relation

- Assume that the deer population of Rustic County is 200 at time *n* = 0 and 220 at time *n* = 1 and that the increase from time *n* − 1 to time *n* is twice the increase from time *n* − 2 to time *n* − 1. Write a recurrence relation and an initial condition that define the deer population at time *n* and then solve the recurrence relation.

  Let d_n denote the deer population at time *n*. We have the initial conditions

$$d_0 = 200, d_1 = 220.$$

  The increase from time *n* − 1 to time *n* is $d_n - d_{n-1}$, and the increase from time *n* − 2 to time *n* − 1 is $d_{n-1} - d_{n-2}$. Thus we obtain the recurrence relation

$$d_n - d_{n-1} = 2\,(d_{n-1} - d_{n-2}),$$

  which may be rewritten

$$d_n = 3d_{n-1} - 2d_{n-2}.$$

Linear Homogeneous Recurrence Relation

To solve this recurrence relation, we first solve the quadratic equation

$$t^2 - 3t + 2 = 0$$

to obtain roots 1 and 2. The sequence *d* is of the form

$$d_n = b \cdot 1^n + c \cdot 2^n = b + c2^n.$$

To meet the initial conditions, we must have

$$200 = d_0 = b + c, \qquad 220 = d_1 = b + 2c.$$

Solving for *b* and *c*, we find that *b* = 180 and *c* = 20. Thus d_n is given by

$$d_n = 180 + 20 \cdot 2^n.$$

The growth is exponential.

Linear Homogeneous Recurrence Relation

- Find an explicit formula for the Fibonacci sequence.

The Fibonacci sequence is defined by the linear homogeneous, second-order recurrence relation

$$f_n - f_{n-1} - f_{n-2} = 0, \qquad n \geq 3,$$

and initial conditions

$$f_1 = 1, \qquad f_2 = 1.$$

We begin by using the quadratic formula to solve

$$t^2 - t - 1 = 0.$$

The solutions are

$$t = \frac{1 \pm \sqrt{5}}{2}.$$

Thus the solution is of the form

$$f_n = b\left(\frac{1+\sqrt{5}}{2}\right)^n + d\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Linear Homogeneous Recurrence Relation

To satisfy the initial conditions, we must have

$$b\left(\frac{1+\sqrt{5}}{2}\right) + d\left(\frac{1-\sqrt{5}}{2}\right) = 1$$

$$b\left(\frac{1+\sqrt{5}}{2}\right)^2 + d\left(\frac{1-\sqrt{5}}{2}\right)^2 = 1.$$

Solving these equations for *b* and *d*, we obtain

$$b = \frac{1}{\sqrt{5}}, \qquad d = -\frac{1}{\sqrt{5}}.$$

Therefore, an explicit formula for the Fibonacci sequence is

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Surprisingly, even though f_n is an integer, the preceding formula involves the irrational number $\sqrt{5}$.

Linear Homogeneous Recurrence Relation

- Theorem 9.2: Let

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} \qquad (9.6)$$

be a second-order linear homogeneous recurrence relation with constant coefficients. Let a be the sequence satisfying (9.6) and

$$a_0 = C_0, \qquad a_1 = C_1$$

If both roots of

$$t^2 - c_1 t - c_2 = 0, \qquad (9.7)$$

are equal to r, then there exist constants b and d such that

$$a_n = br^n + dnr^n, \qquad n = 0, 1, \ldots.$$

Linear Homogeneous Recurrence Relation

- Solve the recurrence relation

$$d_n = 4(d_{n-1} - d_{n-2}) \quad (9.8)$$

subject to the initial conditions

$$d_0 = 1 = d_1.$$

According to Theorem 9.1, $S_n = r^n$ is a solution of (9.8), where r is a solution of

$$t^2 - 4t + 4 = 0. \quad (9.9)$$

Thus we obtain the solution

$$S_n = 2^n$$

of (9.8). Since 2 is the only solution of (9.9), by Theorem 9.2,

$$T_n = n2^n.$$

is also a solution of (9.8). Thus the general solution of (9.8) is of the form

$$U = aS + bT.$$

Linear Homogeneous Recurrence Relation

We must have

$$U_0 = 1 = U_1.$$

These last equations become

$$aS_0 + bT_0 = a + 0b = 1, \qquad aS_1 + bT_1 = 2a + 2b = 1.$$

Solving for a and b, we obtain

$$a = 1, \qquad b = -\frac{1}{2}.$$

Therefore, the solution of (9.8) is

$$d_n = 2^n - n2^{n-1}.$$

PRACTICE

PRACTICE 1

1. Use the formulas

$$s_1 = 1, s_n = s_{n-1} + n \qquad\qquad for\ all\ n \geq 2,$$

to write a recursive algorithm that computes

$$s_n = 1 + 2 + 3 + \cdots + n.$$

2. Give a proof using mathematical induction that your algorithm for part (1) is correct.

PRACTICE 2

- *In Exercises below, find a recurrence relation and initial conditions that generate a sequence that begins with the given terms.*

1. $3, 7, 11, 15, \ldots$
2. $3, 6, 9, 15, 24, 39, \ldots$

PRACTICE 3

- *In Exercises below, solve the given recurrence relation for the initial conditions given.*

1. $a_n = 2na_{n-1}; \quad a_0 = 1$
2. $n = 6a_{n-1} - 8a_{n-2}, a_0 = 1 = a_1$
3. $a_n = 7a_{n-1} - 10a_{n-2}, a_0 = 5, a_1 = 16$

NEXT WEEK'S OUTLINE

- Graph Terminologies
- Path and Cycle
- Euler Cycle and Hamiltonian Cycle

REFERENCES

- Johnsonbaugh, R., 2005, *Discrete Mathematics*, New Jersey: Pearson Education, Inc.
- Rosen, Kenneth H., 2005, *Discrete Mathematics and Its Applications*, 6th edition, McGraw-Hill.
- Hansun, S., 2021, *Matematika Diskret Teknik*, Deepublish.
- Lipschutz, Seymour, Lipson, Marc Lars, *Schaum's Outline of Theory and Problems of Discrete Mathematics*, McGraw-Hill.
- Liu, C.L., 1995, *Dasar-Dasar Matematika Diskret*, Jakarta: Gramedia Pustaka Utama.
- Other offline and online resources.

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

INFORMATIKA
UMN
UNIVERSITAS MULTIMEDIA NUSANTARA

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.