CMSC 207- Lecture 15
CHAPTER 5: Sequences, Mathematical Induction, and Recursion (5.6)

Dr. Ahmed Tarek

Defining Sequences Recursively

- A sequence can be defined in different ways.
- One informal way is to write the first few terms with the expectation that the general pattern will be obvious.
- For instance, "consider the sequence 3, 5, 7, ..." Unfortunately, misunderstandings can occur when this approach is used. The next term of the sequence could be 9 if we mean a sequence of odd integers, or it could be 11 if we mean the sequence of odd prime numbers.

Defining Sequences Recursively

- The second way to define a sequence is to give an explicit formula for its *n*th term.
- For example, a sequence $a_0, a_1, a_2 \ldots$ can be specified by writing

$$a_n = \frac{(-1)^n}{n+1} \quad \text{for all integers } n \geq 0.$$

- The advantage of defining a sequence by such an explicit formula is that each term of the sequence is uniquely determined and can be computed in a fixed, finite number of steps, by substitution.

Defining Sequences Recursively

- The third way to define a sequence is to use recursion.

This requires giving both an equation, called a *recurrence relation*, that defines each later term in the sequence by reference to earlier terms and also one or more initial values for the sequence.

• Definition

A **recurrence relation** for a sequence $a_0, a_1, a_2, \ldots$ is a formula that relates each term a_k to certain of its predecessors $a_{k-1}, a_{k-2}, \ldots, a_{k-i}$, where i is an integer with $k - i \geq 0$. The **initial conditions** for such a recurrence relation specify the values of $a_0, a_1, a_2, \ldots, a_{i-1}$, if i is a fixed integer, or $a_0, a_1, \ldots, a_m$, where m is an integer with $m \geq 0$, if i depends on k.

Example 1 – *Computing Terms of a Recursively Defined Sequence*

- Define a sequence $c_0, c_1, c_2, \ldots$ recursively as follows: For all integers $k \geq 2$,

$$\text{(1)} \quad c_k = c_{k-1} + kc_{k-2} + 1 \qquad \text{recurrence relation}$$

$$\text{(2)} \quad c_0 = 1 \quad \text{and} \quad c_1 = 2 \qquad \text{initial conditions.}$$

- Find c_2, c_3, and c_4.
- Solution:
- $$\begin{aligned} c_2 &= c_1 + 2c_0 + 1 && \text{by substituting } k = 2 \text{ into (1)} \\ &= 2 + 2 \cdot 1 + 1 && \text{since } c_1 = 2 \text{ and } c_0 = 1 \text{ by (2)} \end{aligned}$$

Example 1 – *Solution*

(3) • $c_2 = 5$

$c_3 = c_2 + 3c_1 + 1$ by substituting $k = 3$ into (1)

$= 5 + 3 \cdot 2 + 1$ since $c_2 = 5$ by (3) and $c_1 = 2$ by (2)

(4) • $c_3 = 12$

$c_4 = c_3 + 4c_2 + 1$ by substituting $k = 4$ into (1)

$= 12 + 4 \cdot 5 + 1$ since $c_3 = 12$ by (4) and $c_2 = 5$ by (3)

(5) • $c_4 = 33$

Example 4 – *Showing That a Sequence Given by an Explicit Formula Satisfies a Certain Recurrence Relation*

- The sequence of **Catalan numbers**, named after the Belgian mathematician Eugène Catalan (1814–1894), arises in a remarkable variety of different contexts in discrete mathematics. It can be defined as follows: For each integer $n \geq 1$,

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$

- **a.** Find C_1, C_2, and C_3.
- **b.** Show that this sequence satisfies the recurrence relation $C_k = \frac{4k-2}{k+1}C_{k-1}$ for all integers $k \geq 2$

Example 4 – *Solution*

a.

$$C_1 = \frac{1}{2}\binom{2}{1} = \frac{1}{2}\cdot 2 = 1,$$

$$C_2 = \frac{1}{3}\binom{4}{2} = \frac{1}{3}\cdot 6 = 2,$$

$$C_3 = \frac{1}{4}\binom{6}{3} = \frac{1}{4}\cdot 20 = 5$$

Example 4 – *Solution*

b. To obtain the kth and $(k – 1)$st terms of the sequence, just substitute k and $k –1$ in place of n in the explicit formula for $C_1, C_2, C_3, \ldots$.

$$C_k = \frac{1}{k+1}\binom{2k}{k}$$

$$C_{k-1} = \frac{1}{(k-1)+1}\binom{2(k-1)}{k-1} = \frac{1}{k}\binom{2k-2}{k-1}.$$

Example 4 – *Solution*

- Then start with the right-hand side of the recurrence relation and transform it into the left-hand side: For each integer $k \geq 2$,

$$\frac{4k-2}{k+1}C_{k-1} = \frac{4k-2}{k+1}\left[\frac{1}{k}\binom{2k-2}{k-1}\right]$$ by substituting

$$= \frac{2(2k-1)}{k+1} \cdot \frac{1}{k} \cdot \frac{(2k-2)!}{(k-1)!(2k-2-(k-1))!}$$ by the formula for n choose r

$$= \frac{1}{k+1} \cdot (2(2k-1)) \cdot \frac{(2k-2)!}{(k(k-1)!)(k-1)!}$$ by rearranging the factors

Example 4 – *Solution*

$$= \frac{1}{k+1} \cdot (2(2k-1)) \cdot \frac{1}{k!(k-1)!} \cdot (2k-2)! \cdot \frac{1}{2} \cdot \frac{1}{k} \cdot 2k. \quad \text{because } \tfrac{1}{2} \cdot \tfrac{1}{k} \cdot 2k = 1$$

$$= \frac{1}{k+1} \cdot \frac{2}{2} \cdot \frac{1}{k!} \cdot \frac{1}{(k-1)!} \cdot \frac{1}{k} \cdot (2k) \cdot (2k-1) \cdot (2k-2)! \quad \text{by rearranging the factors}$$

$$= \frac{1}{k+1} \cdot \frac{(2k)!}{k!k!}$$

because $k(k-1)! = k!$,
$\frac{2}{2} = 1$, and
$2k \cdot (2k-1) \cdot (2k-2)! = (2k)!$

$$= \frac{1}{k+1} \binom{2k}{k} \quad \text{by the formula for } n \text{ choose } r$$

$$= C_k \quad \text{by defintion of } C_1, C_2, C_3, \ldots.$$

In-class Practice Exercises

- Find the first 5 terms of the following recursively defined sequences
- (a)

  $b_k = b_{k-1} + 3k$, for all integers $k \geq 2$

  $b_1 = 1$
- (b)

  $t_k = t_{k-1} + 2\, t_{k-2}$, for all integers $k \geq 2$

  $t_0 = -1$, $t_1 = 2$

Example 5 – *The Tower of Hanoi*

- In 1883 a French mathematician, Édouard Lucas, invented a puzzle that he called The Tower of Hanoi (La Tour D'Hanoï).
- The puzzle consisted of eight disks of wood with holes in their centers, which were piled in order of decreasing size on one pole in a row of three. Those who played the game were supposed to move all the disks one by one from one pole to another, never placing a larger disk on top of a smaller one.

Example 5 – *The Tower of Hanoi*

- What is the most efficient way to transfer a tower of k disks from one pole to another?

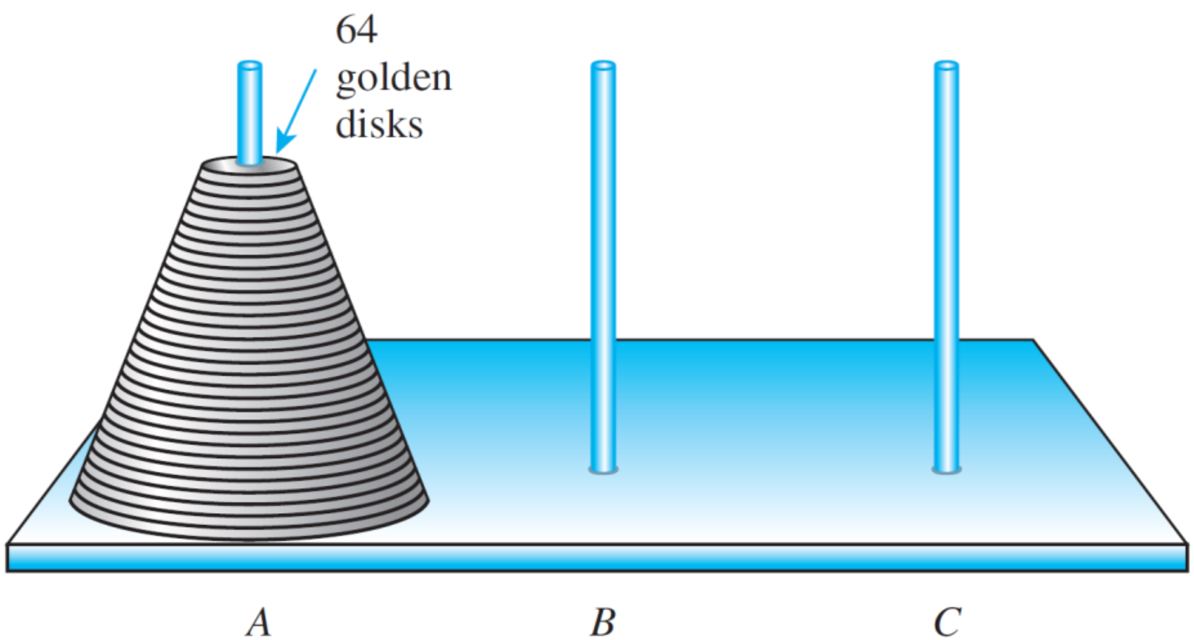
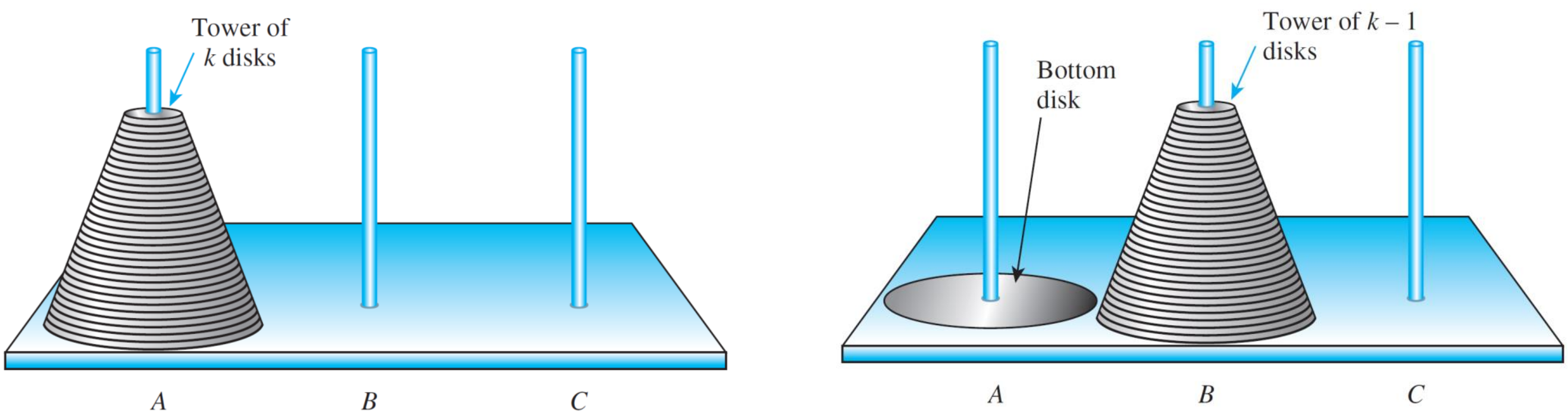


Figure 5.6.2

Example 5 – *Solution*

- The answer is sketched in Figure 5.6.3, where pole *A* is the initial pole and pole *C* is the target pole, and is described as follows:



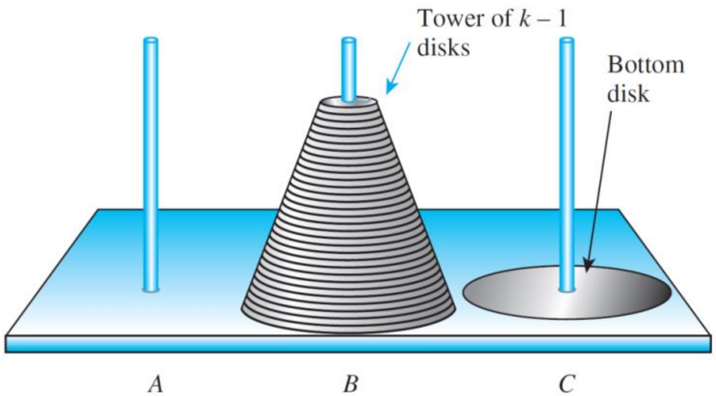
Initial Position
(a)

Position after Transferring $k - 1$ Disks from *A* to *B*
(b)

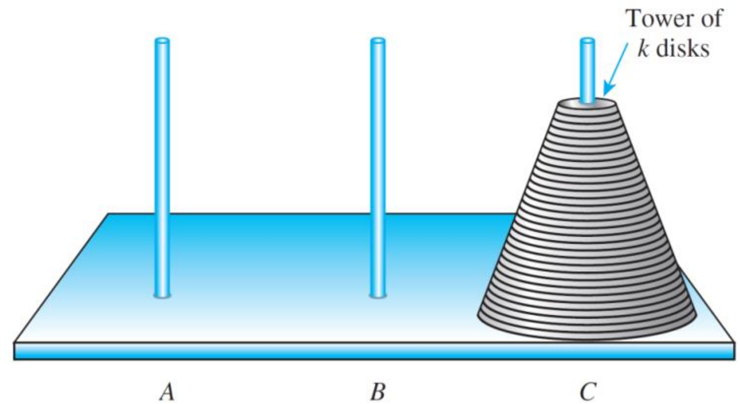
Moves for the Tower of Hanoi

Figure 5.6.3

Example 5 – *Solution*



Position after Moving the Bottom Disk from *A* to *C*
(c)



Position after Transferring k – 1 Disks from *B* to *C*
(d)

Moves for the Tower of Hanoi

Figure 5.6.3

- **Step 1:** Transfer the top $k-1$ disks from pole *A* to pole *B*. If $k > 2$, execution of this step will require a number of moves of individual disks among the three poles.

Example 5 – *Solution*

- **Step 2**: Move the bottom disk from pole *A* to pole *C.*
- **Step 3**: Transfer the top $k - 1$ disks from pole *B* to pole *C*. (Again, if $k > 2$, execution of this step will require more than one move.)
- **To see that this sequence of moves is most efficient**, observe that to move the bottom disk of a stack of k disks from one pole to another, you must first transfer the top $k - 1$ disks to a third pole to get them out of the way.

Example 5 – *Solution*

- Thus transferring the stack of k disks from pole A to pole C requires at least two transfers of the top $k-1$ disks:
- One to transfer them off the bottom disk to free the bottom disk so that it can be moved and another to transfer them back on top of the bottom disk after the bottom disk has been moved to pole C.

Example 5 – *Solution*

- Thus the minimum sequence of moves must include going from the initial position (a) to position (b) to position (c) to position (d).
- **It follows that**

$$\begin{bmatrix}\text{the minimum}\\ \text{number of moves}\\ \text{needed to transfer}\\ \text{a tower of } k \text{ disks}\\ \text{from pole } A \text{ to}\\ \text{pole } C\end{bmatrix} = \begin{bmatrix}\text{the minimum}\\ \text{number of}\\ \text{moves needed}\\ \text{to go from}\\ \text{position (a)}\\ \text{to position (b)}\end{bmatrix} + \begin{bmatrix}\text{The minimum}\\ \text{number of}\\ \text{moves needed}\\ \text{to go from}\\ \text{position (b)}\\ \text{to position (c)}\end{bmatrix} + \begin{bmatrix}\text{the minimum}\\ \text{number of}\\ \text{moves needed}\\ \text{to go from}\\ \text{position (c)}\\ \text{to position (d)}\end{bmatrix} \qquad 5.6.1$$

- **For each integer *n* ≥ 1, let**

$$m_n = \begin{bmatrix}\text{the minimum number of moves needed to transfer}\\ \text{a tower of } n \text{ disks from one pole to another}\end{bmatrix}$$

Example 5 – *Solution*

- Going from position (a) to position (b) requires m_{k-1} moves, going from position (b) to position (c) requires just one move, and going from position (c) to position (d) requires m_{k-1} moves.

Therefore,

$$\begin{aligned} m_k &= m_{k-1} + 1 + m_{k-1} \\ &= 2m_{k-1} + 1 \qquad \text{for all integers } k \geq 2. \end{aligned}$$

$$m_1 = \left[\begin{array}{l}\text{the minimum number of moves needed to move} \\ \text{a tower of one disk from one pole to another}\end{array}\right] = 1.$$

For all integers *k* ≥ 2,

(1) $m_k = 2m_{k-1} + 1$ recurrence relation

(2) $m_1 = 1$ initial conditions

Example 5 – *Solution*

- Here is a computation of the next five terms of the sequence:

$$
\begin{array}{lll}
(3) & m_2 = 2m_1 + 1 = 2\cdot 1 + 1 = 3 & \text{by (1) and (2)} \\
(4) & m_3 = 2m_2 + 1 = 2\cdot 3 + 1 = 7 & \text{by (1) and (3)} \\
(5) & m_4 = 2m_3 + 1 = 2\cdot 7 + 1 = 15 & \text{by (1) and (4)} \\
(6) & m_5 = 2m_4 + 1 = 2\cdot 15 + 1 = 31 & \text{by (1) and (5)} \\
(7) & m_6 = 2m_5 + 1 = 2\cdot 31 + 1 = 63 & \text{by (1) and (6)}
\end{array}
$$

Recursive Definitions of Sum and Product

- Addition and multiplication are called *binary* operations because only two numbers can be added or multiplied at a time. Careful definitions of sums and products of more than two numbers use recursion.

• Definition

Given numbers $a_1, a_2, \ldots, a_n$, where n is a positive integer, the **summation from $i = 1$ to n of the a_i**, denoted $\sum_{i=1}^{n} a_i$, is defined as follows:

$$\sum_{i=1}^{1} a_i = a_1 \quad \text{and} \quad \sum_{i=1}^{n} a_i = \left(\sum_{i=1}^{n-1} a_i\right) + a_n, \quad \text{if } n > 1.$$

The **product from $i = 1$ to n of the a_i**, denoted $\prod_{i=1}^{n} a_i$, is defined by

$$\prod_{i=1}^{1} a_i = a_1 \quad \text{and} \quad \prod_{i=1}^{n} a_i = \left(\prod_{i=1}^{n-1} a_i\right) \cdot a_n, \quad \text{if } n > 1.$$

Recursive Definitions of Sum and Product

•The effect of these definitions is to specify an *order* in which sums and products of more than two numbers are computed. For example,

$$\sum_{i=1}^{4} a_i = \left(\sum_{i=1}^{3} a_i\right) + a_4 = \left(\left(\sum_{i=1}^{2} a_i\right) + a_3\right) + a_4 = ((a_1 + a_2) + a_3) + a_4.$$

•The recursive definitions are used with mathematical induction to establish various properties of general finite sums and products.

Example 9 – *A Sum of Sums*

- Prove that for any positive integer n, if $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ are real numbers, then

$$\sum_{i=1}^{n}(a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i.$$

- Solution:

The proof is by mathematical induction. Let the property $P(n)$ be the equation

$$\sum_{i=1}^{n}(a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i. \qquad \leftarrow P(n)$$

Example 9 – *Solution*

- We must show that *P*(*n*) is true for all integers $n \geq 0$. We do this by mathematical induction on *n*.
- **Show that *P*(1) is true:** To establish *P*(1), we must show that

$$\sum_{i=1}^{1}(a_i + b_i) = \sum_{i=1}^{1} a_i + \sum_{i=1}^{1} b_i. \qquad \leftarrow P(1)$$

- But

$$\begin{aligned}\sum_{i=1}^{1}(a_i + b_i) &= a_1 + b_1 && \text{by definition of } \Sigma \\ &= \sum_{i=1}^{1} a_i + \sum_{i=1}^{1} b_i && \text{also by definition of } \Sigma.\end{aligned}$$

- Hence *P*(1) is true.

Example 9 – *Solution*

- **Show that for all integers $k \geq 1$, if $P(k)$ is true then $P(k + 1)$ is also true:** Suppose $a_1, a_2, \ldots, a_k, a_{k+1}$ and $b_1, b_2, \ldots, b_k, b_{k+1}$ are real numbers and that for some $k \geq 1$

$$\sum_{i=1}^{k}(a_i + b_i) = \sum_{i=1}^{k} a_i + \sum_{i=1}^{k} b_i.$$

$\leftarrow P(k)$
inductive hypothesis

- We must show that

$$\sum_{i=1}^{k+1}(a_i + b_i) = \sum_{i=1}^{k+1} a_i + \sum_{i=1}^{k+1} b_i.$$

$\leftarrow P(k+1)$

Example 9 – *Solution*

- But the left-hand side of the equation is

$$\begin{aligned}
\sum_{i=1}^{k+1}(a_i + b_i) &= \sum_{i=1}^{k}(a_i + b_i) + (a_{k+1} + b_{k+1}) && \text{by definition of } \Sigma \\
&= \left(\sum_{i=1}^{k} a_i + \sum_{i=1}^{k} b_i\right) + (a_{k+1} + b_{k+1}) && \text{by inductive hypothesis} \\
&= \left(\sum_{i=1}^{k} a_i + a_{k+1}\right) + \left(\sum_{i=1}^{k} b_i + b_{k+1}\right) && \text{by the associative and cummutative laws of algebra} \\
&= \sum_{i=1}^{k+1} a_i + \sum_{i=1}^{k+1} b_i && \text{by definition of } \Sigma
\end{aligned}$$

which equals the right-hand side of the equation.

Q.E.D.

In-class Practice Exercise

- Find the first 6 terms of the following recursively defined sequence

  $v_k = v_{k-1} + v_{k-2} + 1$, for all integers $k \geq 3$

  $v_1 = 1$, $v_2 = 3$

Some Applications of Recursive Definitions of Sum & Product

- Recursive definition of the sum of the first n positive integers:

$$F(n) = \sum_{i=1}^{n} i$$

- $F(1) = 1$
- $F(n+1) = F(n) + (n+1), \quad n \geq 1$

Factorials

- $n! = 1$ if $n=1$
- $n! = n.(n-1)!$ if $n \geq 1$

Fibonacci numbers:

- $F(0)=0$, $F(1)=1$ and
- $F(n)=F(n-1)+F(n-2)$ for $n=2,3,\ldots$

- **Assume the alphabet Σ**
  - Example: $\Sigma = \{a,b,c,d\}$
- **A set of all strings containing symbols in Σ: Σ^***
  - Example: $\Sigma^* = \{$"",a,aa,aaa,aaa…, ab, …b,bb, bbb, …$\}$

Recursive definition of Σ^*

- **Basis step:**
  - empty string $\lambda \in \Sigma^*$
- **Recursive step:**
  - If $w \in \Sigma^*$ and $x \in \Sigma$ then $wx \in \Sigma^*$

Recursive Definition for the Length of a String

Length of a String

Example:

Give a recursive definition of $l(w)$, the length of the string w.

Solution:

The length of a string can be recursively defined by:

$l(``) = 0$; the length of the empty string

$l(wx) = l(w) + 1$ if $w \in \Sigma^*$ and $x \in \Sigma$.