Lecture 9. Recurrence Relations

Ref.: Rosen Section 6.1 & 6.2

Main Topics:

- Recurrence Relations
- Solving Recurrence Relations

Recurrence Relations

- Recursively Defined Sequences
- Modeling with Recurrence Relations
- Examples
- References:
  - Rosen 5.1

Recursively Defined Sequences

Sequences can be defined in a variety of ways:

- Write the first terms with the expectation that the general pattern will be obvious, for instance: 3, 5, 7, 9, …

- Explicit formula for the n^{th} term, for instance a sequence a_0, a_1, a_2,… can be specified by:

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

Recursively Defined Sequences

- Use recursion: this requires

  recurrence relation: relates later terms in the sequence to earlier terms

  initial conditions: specifies the first few terms of the sequence

Recursively Defined Sequences

Example:

Define a sequence b_0, b_1, b_2, … recursively as follows:

(1) $b_k = b_{k-1} + b_{k-2}$ for all $k \geq 2$ (recurrence relation)

(2) $b_0 = 1$, $b_1 = 3$ (initial conditions)

b_2 , b_3, b_4, b_5, … can be computed using the recurrence relation:

$b_2 = b_1 + b_0 = 3 + 1 = 4$

$b_3 = b_2 + b_1 = 4 + 3 = 7$

$b_4 = b_3 + b_2 = 7 + 4 = 11$

$b_5 = b_4 + b_3 = 11 + 7 = 18$

Recursively Defined Sequences

Why using recursion to define a sequence?

Sometimes it is very difficult or impossible to find an explicit formula for a sequence, but it is possible to define the sequence using recursion.

Remark

Defining sequences recursively is similar to a prove by mathematical induction:

- initial conditions similar to the basis step
- recurrence relation similar to the inductive step

Recursively Defined Sequences

Definition 1

A **recurrence relation** for a sequence a_0, a_1, a_2, … is a formula that relates each term a_k to certain of its predecessors a_{k-1}, a_{k-2}, … , a_{k-i}, where i is a fixed integer and $k \geq i$.

The **initial conditions** for such a recurrence relation specify the values of a_0, a_1, a_2, … ,a_{i-1}.

Remark: A sequence need not always start with a subscript of 0.

Recursively Defined Sequences

Example: Computing terms of a recursively defined sequence.

Define a sequence c_0, c_1, c_2, … recursively as follows:

(1) $c_k = c_{k-1} + k \cdot c_{k-2} + 1$ for all $k \geq 2$ (recurrence relation)

(2) $c_0 = 1$, $c_1 = 2$ (initial conditions)

Find c_2 , c_3, and c_4:

$c_2 = c_1 + 2 \cdot c_0 + 1 = 2 + 2 \cdot 1 + 1 = 5$

$c_3 = c_2 + 3 \cdot c_1 + 1 = 5 + 3 \cdot 2 + 1 = 12$

$c_4 = c_3 + 4 \cdot c_2 + 1 = 12 + 4 \cdot 5 + 1 = 33$

Recursively Defined Sequences

Example: Writing a recursion in more than one way.

Let s_0, s_1, s_2, … be a sequence that satisfies the following recurrence relation:

$$s_k = 3s_{k-1} - 1 \text{ for all } k \geq 1$$

Then this recurrence relation defines the same sequence:

$$s_{k+1} = 3s_k - 1 \text{ for all } k \geq 0$$

Recursively Defined Sequences

Example: Actual values of the sequence are determined by the initial conditions.

Let $a_0, a_1, a_2, \ldots$ and $b_0, b_1, b_2, \ldots$ be sequences that satisfy the same recurrence relation:

$$a_k = 3a_{k-1} \text{ and } b_k = 3b_{k-1} \text{ for all } k \geq 2$$

But suppose that the initial conditions are different:

$$a_1 = 2 \text{ and } b_1 = 1$$

Find a_2, a_3, a_4 and b_2, b_3, b_4,:

$a_2 = 3a_1 = 3 \cdot 2 = 6$ $\quad$ $b_2 = 3b_1 = 3 \cdot 1 = 3$

$a_3 = 3a_2 = 3 \cdot 6 = 18$ $\quad$ $b_3 = 3b_2 = 3 \cdot 3 = 9$

$a_4 = 3a_3 = 3 \cdot 18 = 54$ $\quad$ $b_4 = 3b_3 = 3 \cdot 9 = 27$

Recursively Defined Sequences

<u>Example:</u> Show that an explicit formula satisfies a recurrence relation.
Show that the sequence $\mathbf{(-1)^{\mathit{n}}\,\mathit{n}!}$ for $n \geq 0$, satisfies the recurrence relation $\boldsymbol{s_k = -k \cdot s_{k-1}}$ for all $k \geq 1$

Substitute k and $k-1$ for n to get:
$\boldsymbol{s_k = (-1)^k \cdot k!}$ and $\boldsymbol{s_{k-1} = (-1)^{k-1} \cdot (k-1)!}$

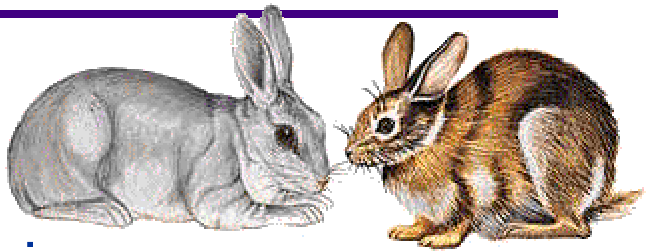
It follows that: $-k \cdot s_{k-1} = -k \cdot [(-1)^{k-1} \cdot (k-1)!] = -1 \cdot k \cdot (-1)^{k-1} \cdot (k-1)!$
$= -1 \cdot (-1)^{k-1} \cdot k \cdot (k-1)! = (-1)^k \cdot k! = s_k$

Modeling with Recurrence Relations

<u>Compound Interest</u>:
Suppose that a person deposits \$10,000 in a savings account at a bank yielding 11% per year with interest compounded annually.
Find a recurrence relation for the amount of money in the account after *n* years. What are the initial conditions?

<u>Solution:</u> Let P_n denote the amount of money after *n* years.
Then $P_n = P_{n-1} + 0.11\ P_{n-1} = 1.11\ P_{n-1}$.
Initial Condition: $P_0 = 10{,}000$.
$P_1 = 1.11 * P_0 = 1.11 * 10{,}000 = 11{,}100$
$P_2 = 1.11 * P_1 = 1.11 * 11{,}100 = 12{,}321$
$P_3 = 1.11 * P_2 = 1.11 * 12{,}321 = 13{,}676.31$

Modeling with Recurrence Relations



Rabbits and Fibonacci Numbers:

A young pair of rabbits (male and female) is placed on an island. Assume the following conditions:

1. A pair of rabbits does not breed until they are 2 month old.
2. After they are 2 month old, each pair of rabbits produces another pair each month.
3. No rabbits die.

Find a recurrence relation for the number of pairs of rabbits on the island after *n* month. What are the initial conditions?

Modeling with Recurrence Relations

Solution: Denote f_n the number of pairs of rabbits after n months.

Month	newborn pairs in month n	pairs after month $n-1$	f_n= sum
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8

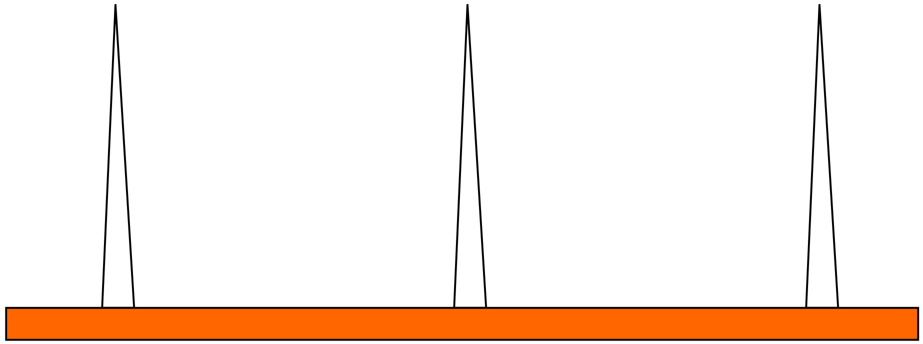
Number of pairs after month $n-1 = f_{n-1}$

Number of newborn pairs in month n = Number of pairs after month $n-2 = f_{n-2}$

Thus, $f_n = f_{n-1} + f_{n-2}$ for all $n \geq 3$. Initial conditions: f_1=1 and f_2=1.

Modeling with Recurrence Relations

Towers of Hanoi:

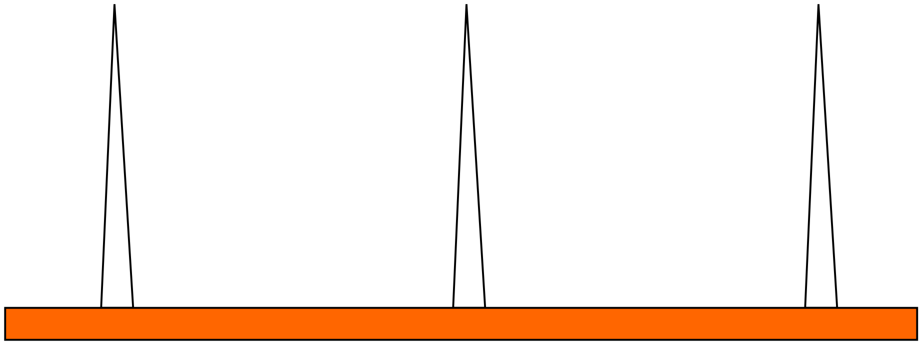


Legend has it that there were three diamond needles set into the floor of the temple of Brahma in Hanoi.
God placed 64 golden disks that decrease in size as they rise from the base.

Priests were to transfer all the disks one by one from the first pole to one of the others, but they must never place a larger disk on top of a smaller one. When completed, the world would end!

Modeling with Recurrence Relations

Towers of Hanoi:

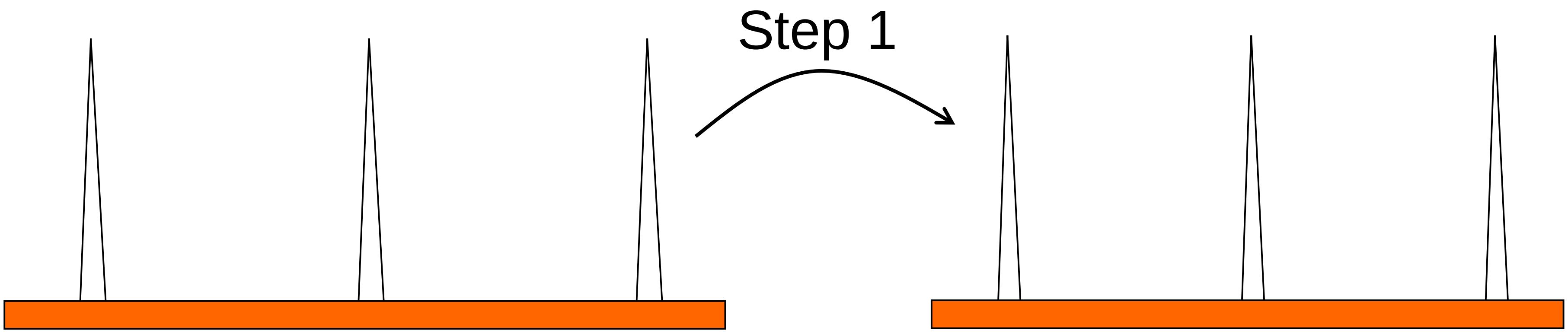


Assuming the priests need 1 second to move a disk, how long after the priests started will the world end?

Let H_n denote the number of moves needed to solve the Tower of Hanoi problem with n disks.

Set up a recurrence relation for H_n.

Modeling with Recurrence Relations

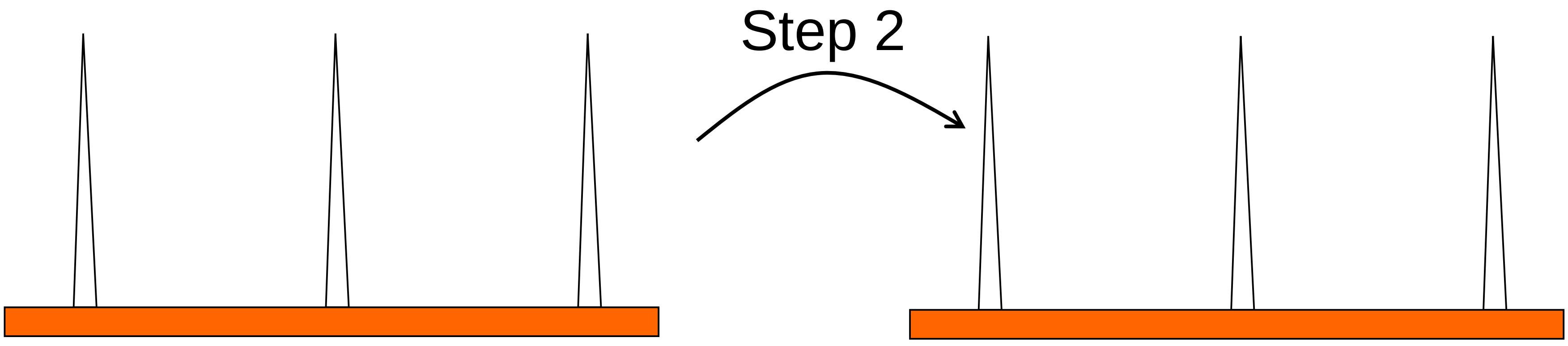


Step 1: Transfer the top n–1 disks one by one to needle 2, obeying the restriction that you never place a larger disk on top of another.

We keep the largest disk fixed during these moves

This requires H_{n-1} moves.

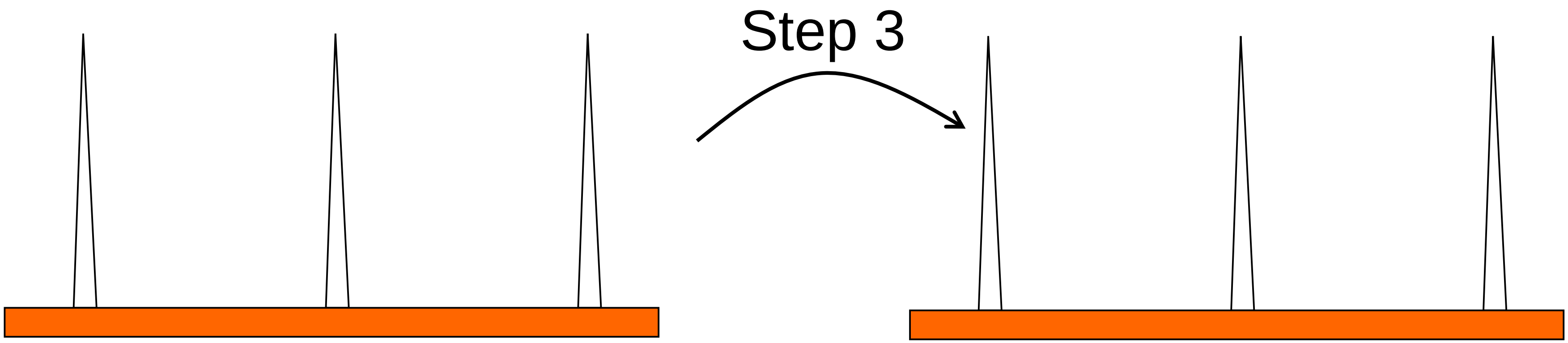
Modeling with Recurrence Relations



Step 2: Move the largest disk from needle 1 to needle 3.

This requires 1 move.

Modeling with Recurrence Relations



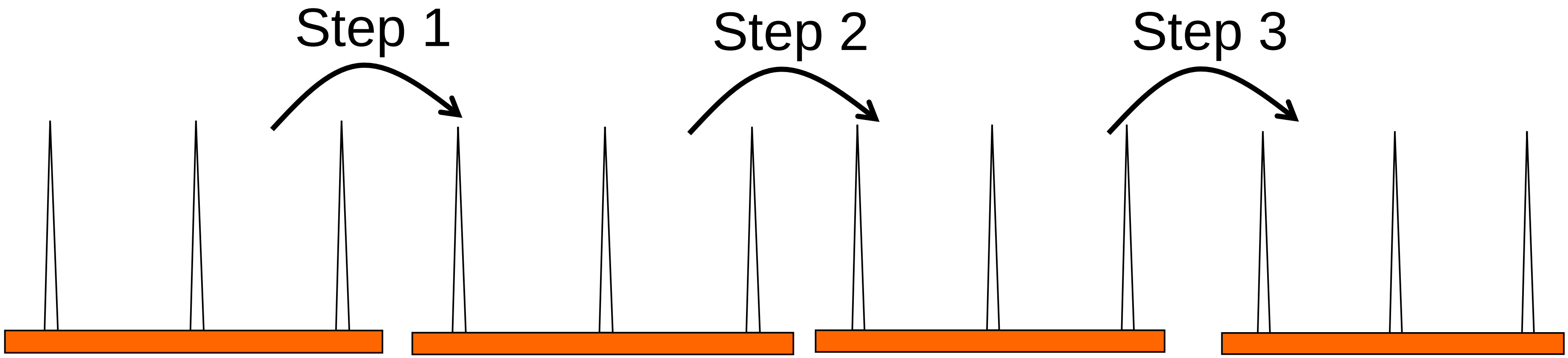
Step 3: Move all n–1 disks from needle 2 to needle 3.

We keep the largest disk fixed during these moves.

This requires H_{n-1} moves.

$$\boldsymbol{H_n = \text{Step 1} + \text{Step 2} + \text{Step 3} = H_{n-1} + 1 + H_{n-1} = 2H_{n-1} + 1}$$

Modeling with Recurrence Relations

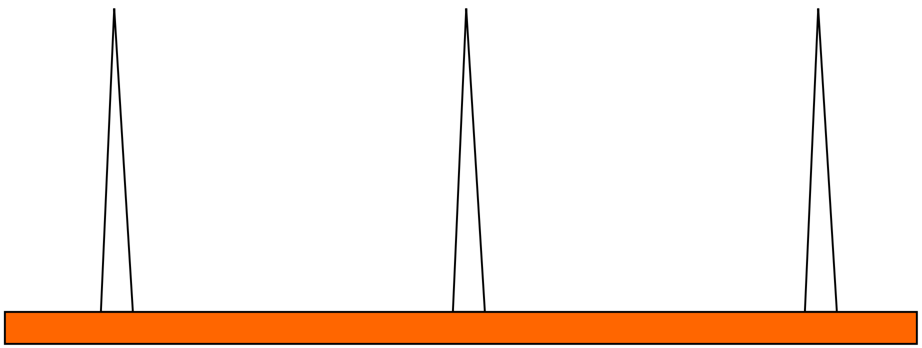


Can the puzzle be solved in fewer steps?

No, since before moving the bottom disk to needle 3, you must move the top n–1 disks to needle 2 to get them out of the way. Thus moving n disks from needle 1 to needle 3 requires at least 2 transfers of the top n–1 disks, one to move them off the bottom disk to free the disk (Step 1) so it can be moved (Step 2) and another to move them back on top of the bottom disk (Step 3).

Modeling with Recurrence Relations

Towers of Hanoi:



Recurrence relation:

$H_n = 2\ H_{n-1} + 1$, for all $n > 1$.

Initial condition: $H_1 = 1$

Find H_2, H_3, H_4 , H_5 , H_6:

$H_2 = 2\ H_1 + 1 = 3$
$H_3 = 2\ H_2 + 1 = 7$
$H_4 = 2\ H_3 + 1 = 15$
$H_5 = 2\ H_4 + 1 = 31$
$H_6 = 2\ H_5 + 1 = 63$

Compute H_{64}:

$H_{64} \approx 1.84 \times 10^{19}$ seconds $\approx$ 584 billion years

Modeling with Recurrence Relations

Number of bit strings with a certain property:

Find a recurrence relations and give initial conditions for the number of bit strings of length n that do **not have two consecutive 0s**. How many such bit strings are there of length 7?

Modeling with Recurrence Relations

Solution: Let a_n denote the number of bit strings of length n that do not have two consecutive 0s.

Assume $n \geq 3$. Partition all the bit strings in into two disjoint sets:

- All bit strings of length n not containing two consecutive 0s and start with a 1
- All bit strings of length n not containing two consecutive 0s and start with a 0

Modeling with Recurrence Relations

Bit strings of length n not containing two consecutive 0s and starting with a 1:

$\mathbf{1}\underbrace{\ldots\ldots\ldots\ldots}$

any bit string of length $n-1$ with no two consecutive 0s

There are a_{n-1} of these

Bit strings of length n not containing two consecutive 0s and starting with a 0:

$\mathbf{01}\underbrace{\ldots\ldots\ldots\ldots}$

any bit string of length $n-2$ with no two consecutive 0s

There are a_{n-2} of these

Modeling with Recurrence Relations

Therefore, $a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$.
Initial conditions $a_1 = 2$, $a_2 = 3$.

Then $a_3 = a_2 + a_1 = 5$, $a_4 = a_3 + a_2 = 8$, $a_5 = a_4 + a_3 = 13$, $a_6 = a_5 + a_4 = 21$, $a_7 = a_6 + a_5 = 34$.

Modeling with Recurrence Relations

Find a recurrence relations and give initial conditions for the number of bit strings of length *n* that **contain two consecutive 0s**. How many such bit strings are there of length 8?

Modeling with Recurrence Relations

Solution: Let a_n denote the number of bit strings of length n that contain two consecutive 0s.

Assume $n \geq 3$. Partition all the bit strings in into three disjoint sets:

- All bit strings of length n containing two consecutive 0s and start with a 1
- All bit strings of length n containing two consecutive 0s and start with a 01
- All bit strings of length n containing two consecutive 0s and start with a 00

Modeling with Recurrence Relations

Bit strings of length n with two consecutive 0s and starting with a 1:

$\underbrace{\textbf{1............}}$

any bit string of length n–1 with two consecutive 0s

There are a_{n-1} of these

Bit strings of length n with two consecutive 0s and starting with a 01:

$\underbrace{\textbf{01............}}$

any bit string of length n–2 with two consecutive 0s

There are a_{n-2} of these

Bit strings of length n with two consecutive 0s and starting with a 00:

$\underbrace{\textbf{00............}}$

any bit string of length n–2

There are 2^{n-2} of these

Modeling with Recurrence Relations

Recurrence Relation: $a_n = a_{n-1} + a_{n-2} + 2^{n-2}$ for $n \geq 3$

Initial Conditions: $a_1 = 0$ and $a_2 = 1$.

$a_3 = a_2 + a_1 + 2^1 = 1 + 0 + 2 = 3$
$a_4 = a_3 + a_2 + 2^2 = 3 + 1 + 4 = 8$
$a_5 = a_4 + a_3 + 2^3 = 8 + 3 + 8 = 19$
$a_6 = a_5 + a_4 + 2^4 = 19 + 8 + 16 = 43$
$a_7 = a_6 + a_5 + 2^5 = 43 + 19 + 32 = 94$
$a_8 = a_7 + a_6 + 2^6 = 94 + 43 + 64 = 201$

Solving Recurrence Relations

- Method of Iteration
- References:

  Rosen 6.2

Method of Iteration

- Given a sequence a_0, a_1, a_2, … defined by a recurrence relation and initial conditions
- start from the initial conditions and calculate successive terms until you see the pattern developing
- guess an explicit formula

Method of Iteration

Example: Sequence b_0, b_1, b_2, … recursively defined as follows:

(1) $a_k = a_{k-1} + 2$ for all $k \geq 1$ (recurrence relation)

(2) $a_0 = 1$ (initial condition)

Calculate successive terms until you see the pattern developing:

$a_0 = 1$

$a_1 = a_0 + 2 = 1 + 2$

$a_2 = a_1 + 2 = (1 + 2) + 2$

$a_3 = a_2 + 2 = (1 + 2 + 2) + 2$

$a_4 = a_3 + 2 = (1 + 2 + 2 + 2) + 2$

$a_0 = 1 + 0 \cdot 2$

$a_1 = 1 + 1 \cdot 2$

$a_2 = 1 + 2 \cdot 2$

$a_3 = 1 + 3 \cdot 2$

$a_4 = 1 + 4 \cdot 2$

Guess for an explicit formula: $a_n = 1 + n \cdot 2 = 1 + 2n$, for all $n \geq 0$

Method of Iteration

A sequence a_0, a_1, a_2, … is called an **arithmetic sequence** if and only if, there is a constant d such that

$$a_k = a_{k-1} + d \text{ for all integers and } k \geq 1.$$

Then, it holds

$$a_n = a_0 + d \cdot n \quad \text{for all integers and } n \geq 0.$$

Method of Iteration

<u>Example:</u> Let $r \neq 0$. Sequence $b_0, b_1, b_2, \ldots$ defined as follows:

(1) $a_k = r \cdot a_{k-1}$ for all $k \geq 1$ (recurrence relation)

(2) $a_0 = a$ (initial condition)

Use iteration to guess an explicit formula:

$a_0 = a$

$a_1 = r \cdot a_0 = r \cdot a$

$a_2 = r \cdot a_1 = r \cdot (r \cdot a) = r^2 \cdot a$

$a_3 = r \cdot a_2 = r \cdot (r^2 \cdot a) = r^3 \cdot a$

$a_4 = r \cdot a_3 = r \cdot (r^3 \cdot a) = r^4 \cdot a$

Guess for an explicit formula: $a_n = r^n \cdot a$, for all $n \geq 0$

Method of Iteration

A sequence a_0, a_1, a_2, … is called an **geometric sequence** if and only if, there is a constant r such that

$$a_k = r \cdot a_{k-1} \text{ for all integers and } k \geq 1.$$

Then, it holds

$$a_n = a_0 + r^n \text{ for all integers and } n \geq 0.$$

Method of Iteration

Example: Explicit formula for the Tower of Hanoi sequence:

(1) $m_k = 2 \cdot m_{k-1} + 1$ for all $k \geq 2$ (recurrence relation)

(2) $m_1 = 1$ (initial condition)

$m_1 = 1$

$m_2 = 2 \cdot m_1 + 1 = 2 \cdot 1 + 1 = 2 + 1$

$m_3 = 2 \cdot m_2 + 1 = 2 \cdot (2 + 1) + 1 = 2^2 + 2 + 1$

$m_4 = 2 \cdot m_3 + 1 = 2 \cdot (2^2 + 2 + 1) + 1 = 2^3 + 2^2 + 2 + 1$

$m_5 = 2 \cdot m_4 + 1 = 2 \cdot (2^3 + 2^2 + 2 + 1) + 1 = 2^4 + 2^3 + 2^2 + 2 + 1$

Guess for an explicit formula:

$m_n = 2^{n-1} + 2^{n-2} + \ldots + 2^2 + 2^1 + 2^0 = 2^n - 1$, for all $n \geq 1$

Method of Iteration

Using mathematical induction to verify the correctness of a solution to a recurrence relation.

<u>Example:</u> Explicit formula for the Tower of Hanoi sequence:
if m_1, m_2, m_3, … is the sequence defined by
$m_k = 2 \cdot m_{k-1} + 1$ for all $k \geq 2$, and
$m_1 = 1$,
then $m_n = 2^n - 1$, for all $n \geq 1$.

<u>Poof by induction:</u>
<u>Basis step (n = 1):</u> $m_1 = 1 = 2^1 - 1$, .

Method of Iteration

Poof by induction:

Inductive Hypothesis ($n = k$): Suppose that $m_k = 2^k - 1$.

To show ($n = k+1$): $m_{k+1} = 2^{k+1} - 1$.

It holds

$$\begin{aligned} m_{k+1} &= 2 \cdot m_k + 1 \\ &= 2 \cdot (2^k - 1) + 1 \\ &= 2^{k+1} - 2 + 1 \\ &= 2^{k+1} - 1. \end{aligned}$$

Method of Iteration

<u>Example:</u> A sequence is recursively defined as follows:
$s_k = 2 \cdot s_{k-2}$ for all $k \geq 2$
$s_0 = 1$, $s_1 = 2$.
(a) Use iteration to guess an explicit formula for the sequence.
(b) Use induction to check the correctness of the formula.

$s_0 = 1$
$s_1 = 2$
$s_2 = 2 \cdot s_0 = 2$
$s_3 = 2 \cdot s_1 = 2 \cdot 2 = 2^2$
$s_4 = 2 \cdot s_2 = 2 \cdot 2 = 2^2$
$s_5 = 2 \cdot s_3 = 2 \cdot 2^2 = 2^3$
$s_6 = 2 \cdot s_4 = 2 \cdot 2^2 = 2^3$

Guess an explicit formula:

$$s_n = \begin{cases} 2^{(n+1)/2} & \text{if } n \text{ is odd} \\ 2^{n/2} & \text{if } n \text{ is even} \end{cases}$$

Method of Iteration

Let s_0, s_1, s_2, … be a sequence that satisfies the recurrence relation $s_k = 2 \cdot s_{k-2}$ for all $k \geq 2$ and the initial conditions $s_0 = 1$ and $s_1 = 2$.

Show by induction that for all $n \geq 0$:

$$s_n = \begin{cases} 2^{(n+1)/2} & \text{if } n \text{ is odd} \\ 2^{n/2} & \text{if } n \text{ is even} \end{cases}$$

<u>Basis steps (n=0, n=1):</u> $s_0 = 2^{0/2} = 2^0 = 1$, $s_1 = 2^{(1+1)/2} = 2^1 = 2$.

<u>Inductive hypothesis (n=k):</u> Let $k \geq 1$ and suppose

$$s_i = \begin{cases} 2^{(i+1)/2} & \text{if } i \text{ is odd} \\ 2^{i/2} & \text{if } i \text{ is even} \end{cases}$$

for all integers i with $1 \leq i \leq k$

Method of Iteration

To show ($n=k+1$):

$$s_{k+1} = \begin{cases} 2^{(k+2)/2} & \text{if } k+1 \text{ is odd} \\ 2^{(k+1)/2} & \text{if } k+1 \text{ is even} \end{cases}$$

$$s_{k+1} = 2 \cdot s_{k-1} = \begin{cases} 2 \cdot 2^{k/2} & \text{if } k-1 \text{ is odd} \\ 2 \cdot 2^{(k-1)/2} & \text{if } k-1 \text{ is even} \end{cases}$$

$$= \begin{cases} 2^{(k/2)+1} & \text{if } k+1 \text{ is odd} \\ 2^{((k-1)/2)+1} & \text{if } k+1 \text{ is even} \end{cases}$$

$$= \begin{cases} 2^{(k+2)/2} & \text{if } k+1 \text{ is odd} \\ 2^{(k+1)/2} & \text{if } k+1 \text{ is even} \end{cases}$$

Summary

- Recursively Defined Sequences
- Modeling with Recurrence Relations
- Method of Iteration