Recurrence Relation

Dr. Smita Agrawal
Assistant Professor, CSED
TIET, Patiala
smita.agrawal@thapar.edu

Contents

- Basics of Recurrence Relation
- Linear Recurrence Relation
- Solving Linear Recurrence Relation
- Linear Non-homogeneous Recurrence Relation

Recurrence Relation

$\{a_n\}$

-

- A **recurrence relation** is an equation that recursively defines a sequence where the next term is a function of the previous terms.

- Examples:

1. Fibonacci Series

$$F_n = F_{n-1} + F_{n-2}$$

where, $F_0 = 0$ and $F_1 = 1$

$0, 1, 1, 2, 3, 5, 8, \ldots$

$F_5 = F_4 + F_3 = 3 + 2 = 5$

$F_4 = F_3 + F_2 = 2 + 1 = 3$

$F_3 = F_2 + F_1 = 1 + 1 = 2$

$F_2 = F_1 + F_0 = 1 + 0 = 1$

Examples

- Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have 2 consecutive $0's$. How many bit strings are of there length 5?

Length

$\Rightarrow$ End with a 1 : $\underline{\qquad\qquad\qquad}$ 1 $\quad n-1$

$\Rightarrow$ End with a 0 : $\underline{\qquad\qquad\qquad}$ 1 0 $\quad n-2$

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

$a_5 = a_4 + a_3 = 8 + 5 = 13$ Ans

Let a_n denote the no. of bit strings of length n that do not have two consecutive 0's.

We have to obtain a recurrence relation for $\{a_n\}$.

a_{n-1} = no. of bit strings of length $n-1$ that do not have two consecutive 0's.

a_{n-2} = " " " " " " $n-2$ " " " " "

$$\Rightarrow \boxed{\begin{aligned} a_n &= a_{n-1} + a_{n-2} \\ &\text{when } n \geq 3. \end{aligned}}$$

$\equiv$ Recurrence relation

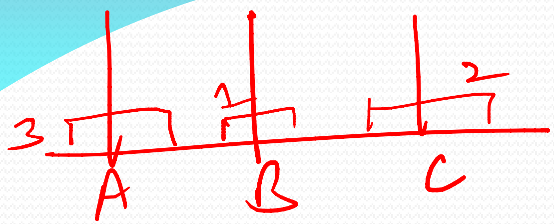
$a_3 = a_2 + a_1 = 5$

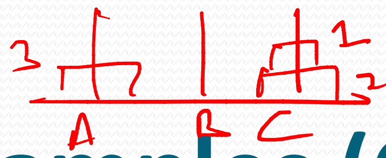
$a_4 = a_3 + a_2 = 5+3 = 8$

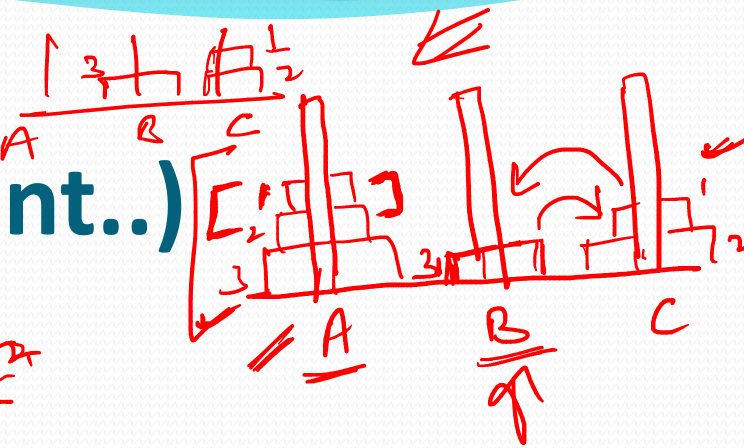
when $n=1$, either 0 or 1, $\therefore a_1 = 2$

when $n=2$, either ~~00~~, 01, 10, 11, $\therefore a_2 = 3$

initial conditions

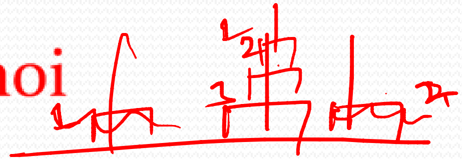






Examples (Cont..)

3. The Tower of Hanoi



- A popular puzzle
- It consists of 3 pegs mounted on a board together with disks of different sizes.
- Initially these disks are placed on the First peg in order of size, with the largest on the bottom.
- The rules of the puzzle allows disks to be moved one at a time from one peg to another as long as a disk is never placed on top of a smaller disk.
- The goal of the puzzle is to have all the disks on the second peg in order of size, with the largest on the bottom.
- Let H_n be the number of moves needed to solve the Tower of Hanoi problem with n disks.
- Set up a recurrence relation for the sequence $\{H_n\}$.

H_n = the no. of moves for ~~solving~~ moving n no. of disks.

H_{n-1} = the no. of moves for moving $n-1$ no. of disks.

n disks $\rightarrow 2^n - 1$ moves

$$\boxed{H_n = H_{n-1} + 1 + H_{n-1}}$$

$$H_n = 2H_{n-1} + 1 \Rightarrow \text{recurrence relation}$$

$H_1 = 1$ ($\because$ if we have one disk, it can be directly moved from peg A to peg B) $\rightarrow$ initial condition.

$$H_2 = 2H_1 + 1 = 2 + 1 = 3$$

$$H_3 = 2H_2 + 1 = 6 + 1 = 7$$

Order and Degree of the Recurrence Relation

- The order of the recurrence relation is defined as the difference between the highest and lowest subscripts of $f(x)$.

- The degree of a recurrence relation is defined to be the highest power of $f(x)$.

Example

- Fibonacci Series

$$\boxed{\underline{F_n} = \underline{F_{n-1}} + \underline{F_{n-2}}}$$

$f(n)$

where, $F_0 = 0$ and $F_1 = 1$

$\underline{n} - \underline{(n-2)}$

Order?

2 ✓

Degree?

1 ✓


Thank you!!!
...