# Mathematical Foundations of computer science
# BCA-2nd Sem

## Unit-4

## Topics covered: Recurrence Relation, LHRR, LHRRWCC, Recursive procedure


Jyoti Jain

Assistant professor

HIMT, Rohtak

E-mail: jyoti.goyal24@gmail.com

# What is Recurrence Relation

- The procedure for finding the terms of a sequence in a recursive manner is called **recurrence relation.**

Why it is used?

- **It is used for solving counting problems.**

Definition:

- A recurrence relation is an equation that recursively defines a sequence where the next term is a function of the previous terms (Expressing $F_n$ as some combination of $F_i$ with $i<n$).

# Example

1. Fibonacci series

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

2. Tower of Hanoi

$$F_n = 2F(n-1)+1$$

3. Factorial Representation

$$n! = n(n-1)!$$

# Linear recurrences

Linear recurrence:

Each term of a sequence is a linear function of earlier terms in the sequence.

For example:

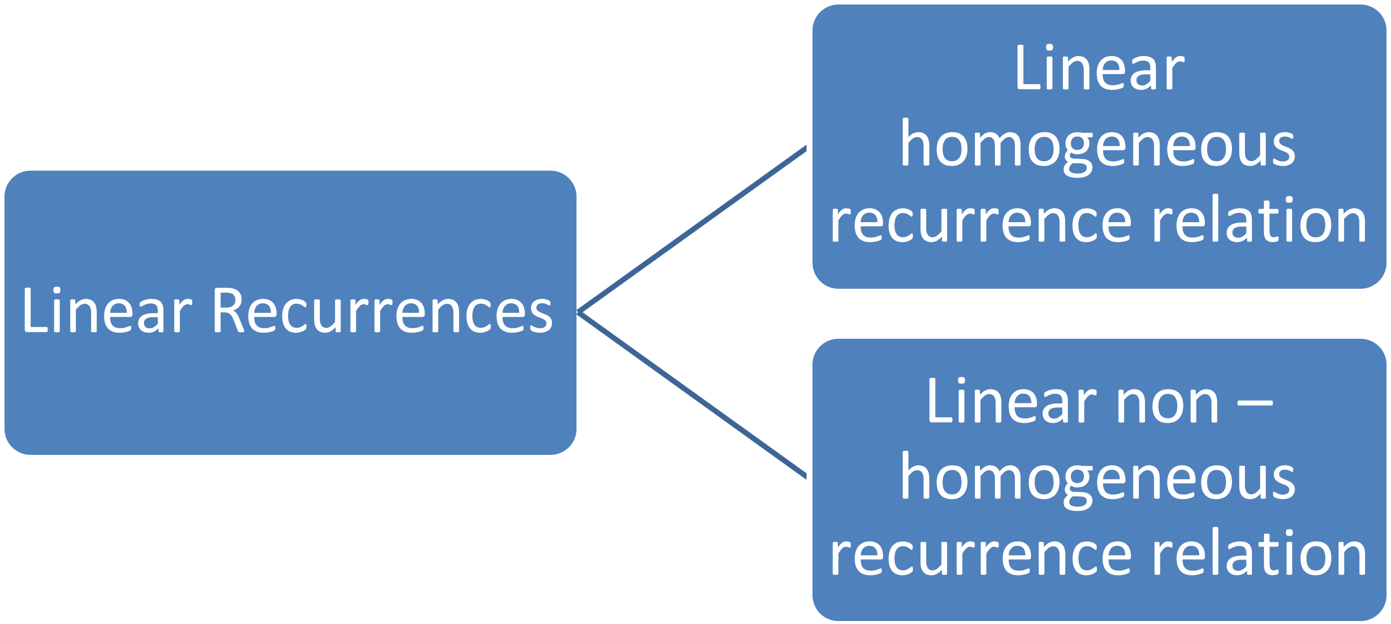
$a_0 = 1$ $a_1 = 6$ $a_2 = 10$

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

$a_3 = a_0 + 2a_1 + 3a_2$

$= 1 + 2(6) + 3(10) = 43$

# Linear recurrences



# Linear homogeneous recurrence relation

A **linear homogenous recurrence relation of degree k** with constant coefficients is a recurrence relation of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k},$$

where $c_1, c_2, \ldots, c_k$ are real numbers, and $c_k / 0$.

$a_n$ is expressed in terms of the previous k terms of the sequence, so its degree is k.

This recurrence includes k initial conditions.

$a_0 = C_0$ $\qquad a_1 = C_1$ $\qquad \ldots$ $\qquad a_k = C_k$

# Example

Determine if the following recurrence relations are linear homogeneous recurrence relations with constant coefficients.

- $P_n = (1.11)P_{n-1}$

  a linear homogeneous recurrence relation of degree one
- $a_n = a_{n-1} + a^2_{n-2}$

  not linear
- $f_n = f_{n-1} + f_{n-2}$

  a linear homogeneous recurrence relation of degree two
- $H_n = 2H_{n-1}+1$

  not homogeneous
- $a_n = a_{n-6}$

  a linear homogeneous recurrence relation of degree six
- $B_n = nB_{n-1}$

  does not have constant coefficient

# Linear non-homogeneous recurrences

A **linear non-homogenous recurrence relation** with constant coefficients is a recurrence relation of the form

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k} + f(n),$$

where $c_1, c_2, \ldots, c_k$ are real numbers, and $f(n)$ is a function depending only on $n$.

The recurrence relation

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k},$$

is called the **associated homogeneous recurrence relation**.

This recurrence includes k initial conditions. $a_0 = C_0$ $a_1 = C_1$ … $a_k = C_k$

# Example

The following recurrence relations are linear non-homogeneous recurrence relations.

- $a_n = a_{n-1} + 2^n$
- $a_n = a_{n-1} + a_{n-2} + n^2 + n + 1$
- $a_n = a_{n-1} + a_{n-4} + n!$
- $a_n = a_{n-6} + n2^n$

# Recursive procedure/Recursion Algorithms

- A *recursive algorithm* is one in which objects are defined in terms of other objects of the same type.

Example

$N! = 1$ if $n = 1$

$\quad n \cdot (n - 1)!$ if $n > 1$

Consider the following recursive algorithm for computing:

- Algorithm (Factorial)
- Input : $n \in N$
- Output : $n!$
- if $n = 1$ then
- return 1
- end
- else
- return Factorial($n - 1$) $\times$ $n$
- end

# Factorial - Analysis?

- How many multiplications $M(n)$ does Factorial perform?
- When $n = 1$ we don't perform any.
- Otherwise we perform 1.
- *Plus* how ever many multiplications we perform in the recursive call, Factorial($n - 1$).
- This can be expressed as a formula (similar to the definition of

$n! =$ $M(0) = 0$

$$M(n) = 1 + M(n - 1)$$

This is known as a *recurrence relation*.

# Recursive procedure

**Advantages**:

- Simplicity of code
- Easy to understand

**Disadvantages**:

- Memory
- Speed
- Possibly redundant work

Thank you