Data Science & Artificial Intelligence

Data Structures Through Python

TREES

Lecture No.- 01

By- Satya sir

Open addressing Techniques $\qquad h_1(k) = k \% N$

- Linear Probing $= h(k,i) = (h_1(k) + i) \% N$

- Quadratic Probing $= h(k,i) = (h_1(k) + i^2) \% N$

- Double Hashing $= h(k,i) = [h_1(k) + i\, h_2(k)] \% N$

$$h_2(k) = k \% N_1 \qquad i = 0,1,2,\ldots \text{ for Each key}$$

$$N = \text{Table Size}$$

Topics to be Covered

TREES Data structure

- Terminology

- Types of Binary Trees

TREES

- Non-linear DS

- Elements (Nodes) are represented in multiple levels
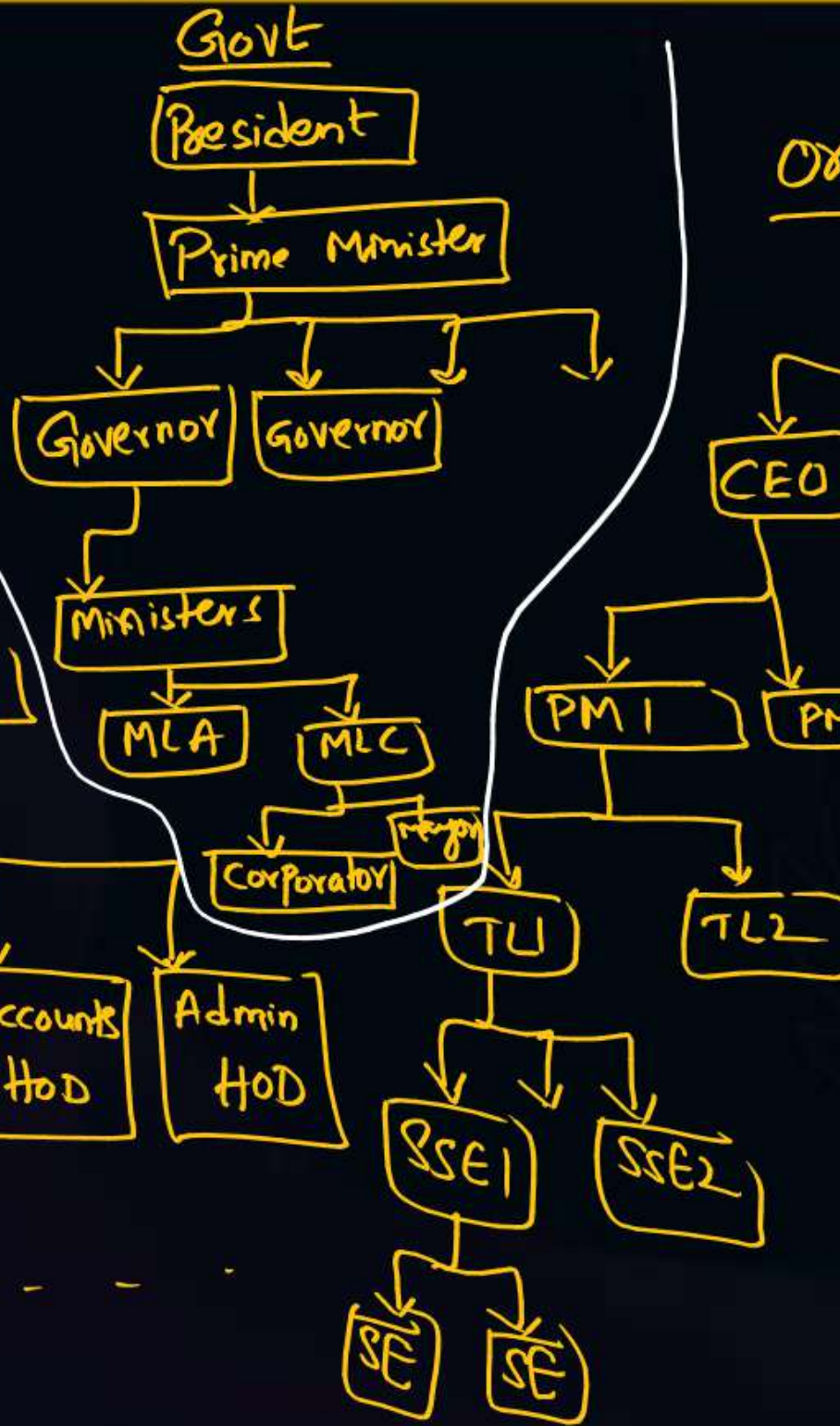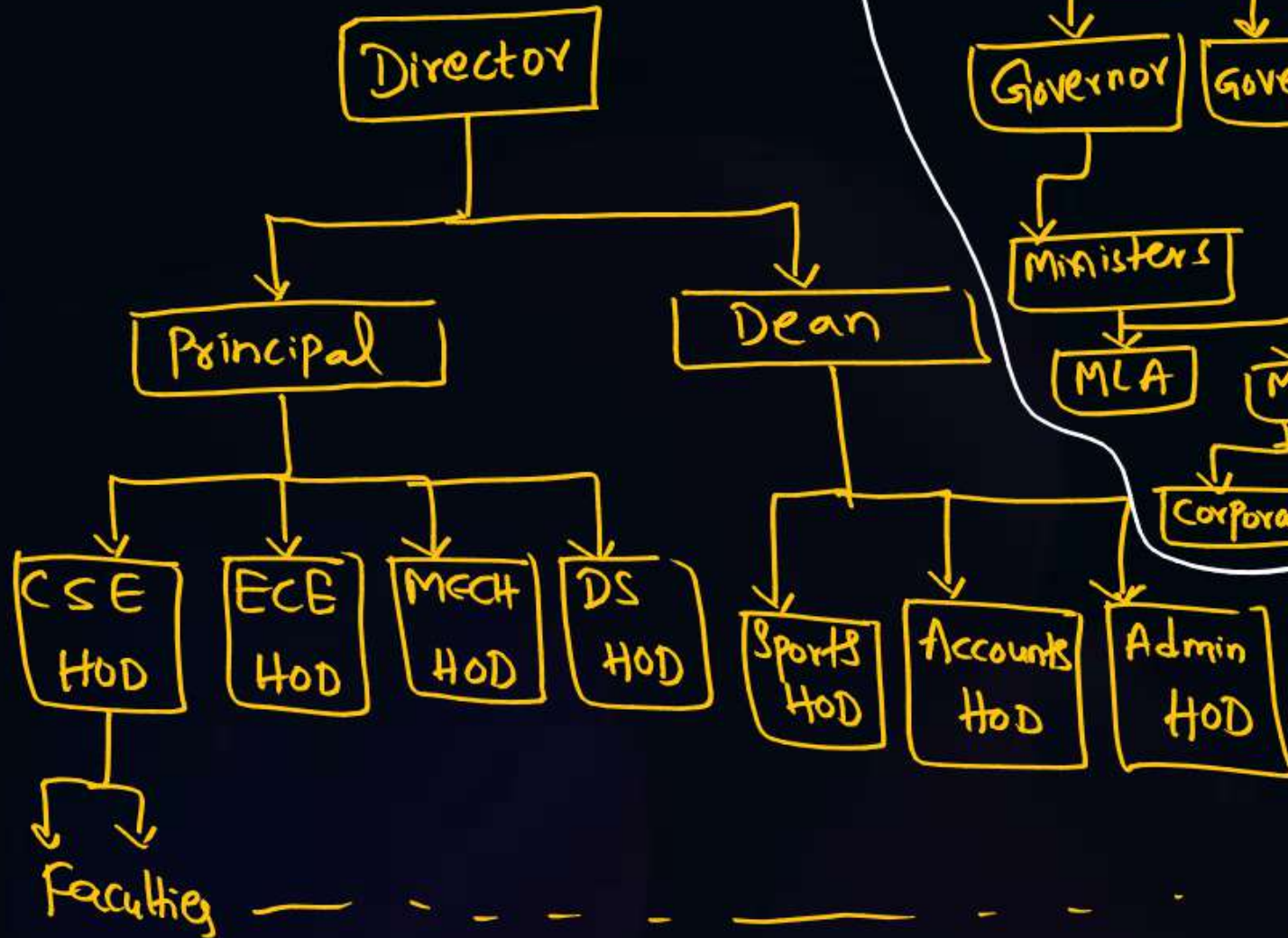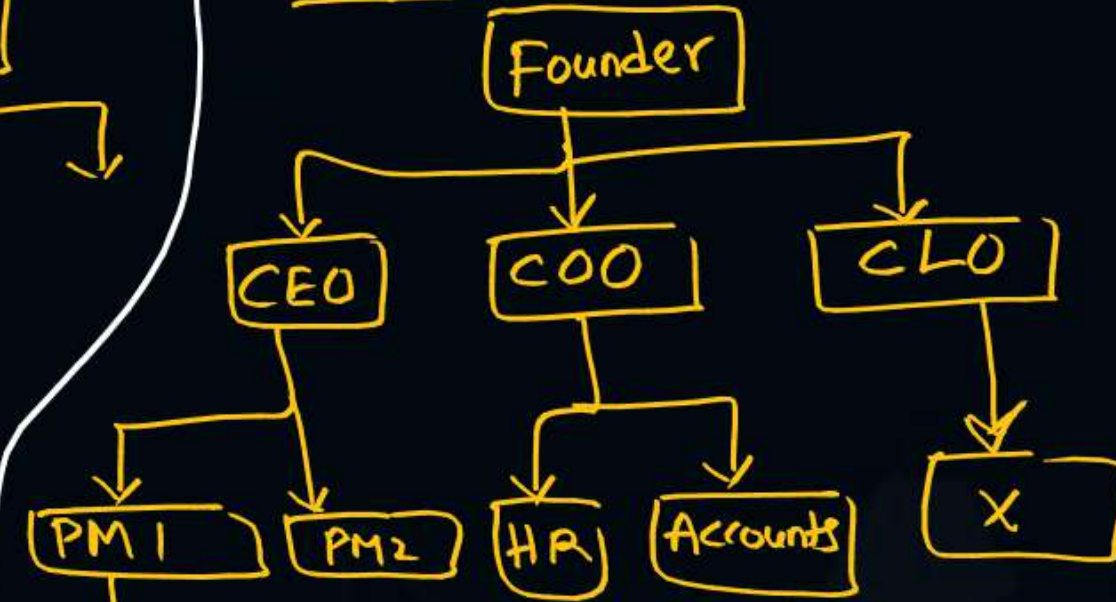- So, It is also called as Hierarchical Data Structure

DS

Linear DS | Non-Linear DS

Linear DS:
- Arrays (Lists)
- Stack
- Queue
- Linked List
- Hash Tables

Non-Linear DS:
- Trees ✓
- Graphs (Algorithms)
- Sets
- Tries
- Maps

Out of Syllabus

Examples

College-Management

Director
- Principal
  - CSE HOD
    - Faculties
  - ECE HOD
  - MECH HOD
  - DS HOD
- Dean
  - Sports HOD
  - Accounts HOD
  - Admin HOD

Govt

President
- Prime Minister
  - Governor
    - Ministers
      - MLA
      - MLC
        - Mayor
        - Corporator
          - TL1
            - SSE1
              - SE
              - SE
            - SSE2
  - Governor
- PM1
- PM2
- TL2

Organization

Founder
- CEO
- COO
  - HR
  - Accounts
- CLO
  - X

Facebook Account

User
- Post
- Tag
- Message
- Upload

Trees Terminology

Root Node (Internal Node without Parent)

Parent — vertices / Nodes

A

The Number of children Per Node is Called as Degree of a Node

Siblings — Children

Links / Edges

B   C   D

E   F   G   H   I   J   K   L

Path (Connection of one Node to other Node)

A — C — J — T

Successors to A

Predecessor / Ancestor

M   N   O   P   Q   R   S   T   U

Leaf Nodes (or) External Nodes
(or) Terminal Node

Types of Nodes

① Internal / Non-leaf / Non-Terminal
  - Nodes with degree > 0

② External / Leaf Nodes / Terminal
  - Nodes with degree == 0

Degree of a Tree

Max (degrees of Nodes)

== 4

Node	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Degree	3	2	4	2	2	2	0	2	0	2	0	1	0	0	0	0	0	0	0	0	0

Level, Depth, Height (Numbering starts from Zero)

Depth : The Number of Edges from Root Node to respective Node (Root depth == 0)

Height : The Maximum Number of Edges in a path from Leaf Node to respective Node (Leaf Height == 0)

Example :

$h(A) == 4$ $d(A) = 0$
$Max\{4, 2, 4, 3, 4, 4, 2, 2, 4\} == 4$ Level 0
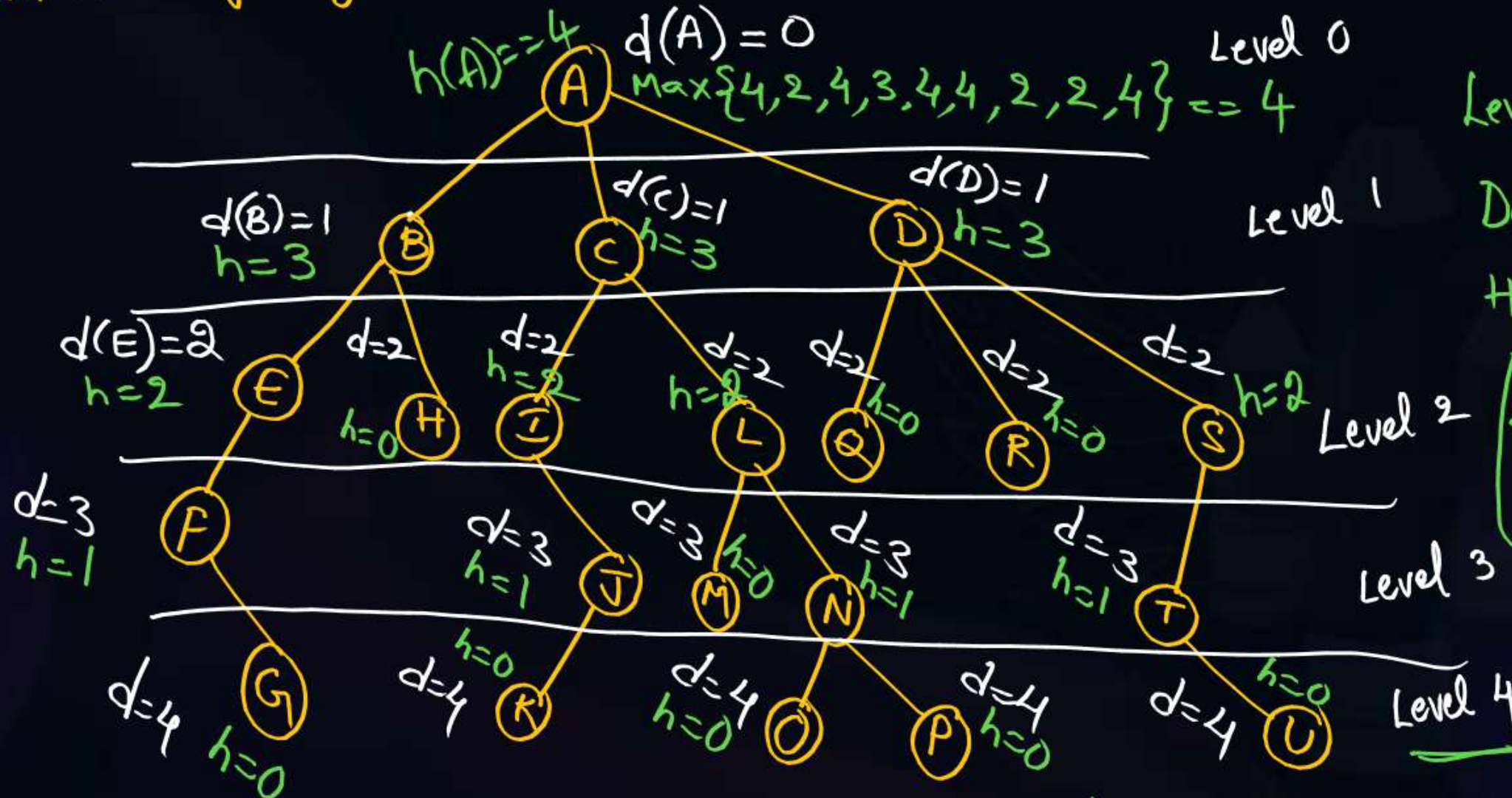
Level (Tree) = Maximum Level

$d(B) = 1$ $h = 3$
$d(C) = 1$ $h = 3$
$d(D) = 1$ $h = 3$ Level 1

Depth (Tree) = Max $\{$depths$\}$

Height (Tree) = Max $\{$heights$\}$

| Level of a Node == depth of a Node |

$d(E) = 2$ $h = 2$
$d = 2$
$d = 2$ $h = 2$
$d = 2$ $h = 0$
$d = 2$
$d = 2$ $h = 0$
$d = 2$ $h = 0$
$d = 2$ $h = 2$ Level 2

| Level (T) == depth (T) == Height (T) |

$d = 3$ $h = 1$
$d = 3$ $h = 1$
$d = 3$ $h = 0$
$d = 3$ $h = 1$
$d = 3$ $h = 1$ Level 3

$d = 4$ $h = 0$
$d = 4$
$d = 4$ $h = 0$
$d = 4$ $h = 0$
$d = 4$
$d = 4$ $h = 0$ Level 4

Binary Tree : A Tree whose degree $<= 2$

- A Binary Tree is a Tree in which Maximum Number of Children Per Node $== 2$

i.e A Node can have either No child, (or) 1 child (or) 2 children.

Ex: 1

Empty Tree

- Binary Tree

Ex: 2

(A)

Binary Tree

Ex: 3

(A)
  \
  (B)

Binary Tree

Ex: 4

Left (A)
     |
    (B)
     |
    (C)
    / \
  (D) (E)

Binary Tree

Ex: 5

Left (A) Right
    /   \
  (B)   (C)
  /   \
(D)   (E)

Binary Tree

Ex: 6

(A)
/   \
(B)   (C)
/ \   \
(D) (G)   (J)
/ \ / \   \
(E)(F)(H)(I) (K)
      /|\   \
    (L)(M)(N) (O)

$deg(H) == 3 == deg(Tree) == 3$
Not a Binary Tree

Types of Binary Trees

- Full Binary Tree ✓
- Complete Binary Tree ✓
- Perfect Binary Tree ✓
- Skewed Binary Tree ✓
- Degenerated Binary Tree ✓
- Binary Search Tree ✓
- Binary Heap ✓
- AVL Tree
- Red Black Tree
  - Splay Tree, Segment Tree, B-Tree, B^t-Tree ....

Full Binary Tree

- A Binary Tree Whose degree = 0 or 2.
- Each node can have either 0 child (or) 2 children

Ex:1

Not a FBT.
bcoz, deg (B) == 1

Ex:2

deg(A)=2
deg(B)=0
deg(C)=2
deg(D)=0
deg(E)=0
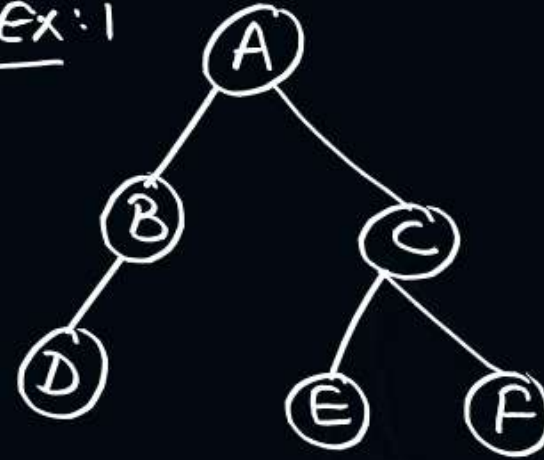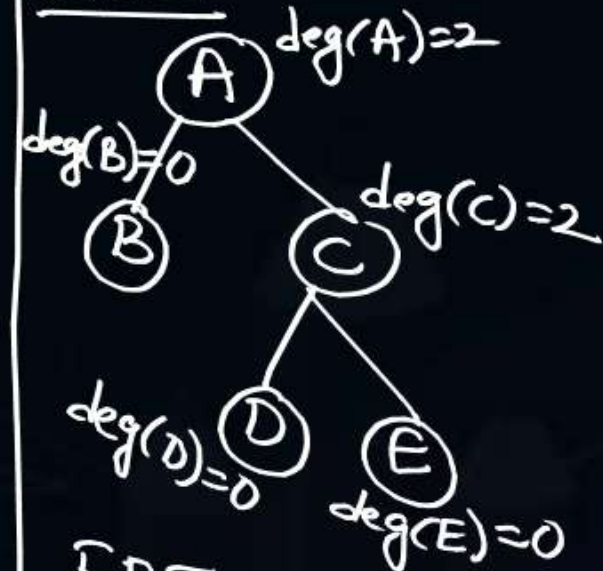
FBT

Ex:3

FBT

Ex:4

FBT

Ex:5

FBT
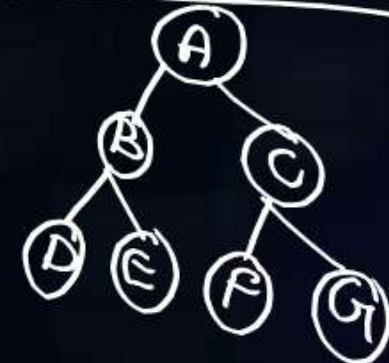
Perfect Binary Tree

− A Binary Tree which is FBT and all leaf Nodes at the Same Level.

Ex:1
(A)

FBT, PBT

Ex:2
(A)
(B)   (C)

FBT, PBT

Ex:3
(A) .... Level 0
(B)   (C) .... Level 1
(D)   (E) .... Level 2

FBT, Not PBT

Ex:4
(A)
(B)   (I)
(C)   (H)
(D) (G)
(E) (F)

FBT, Not PBT

Ex:5
(A)
(B)   (C)
(D) (E)   (P)   (G)
(H)(I)(J)(K)(L)(M)(N)
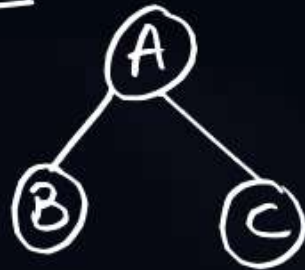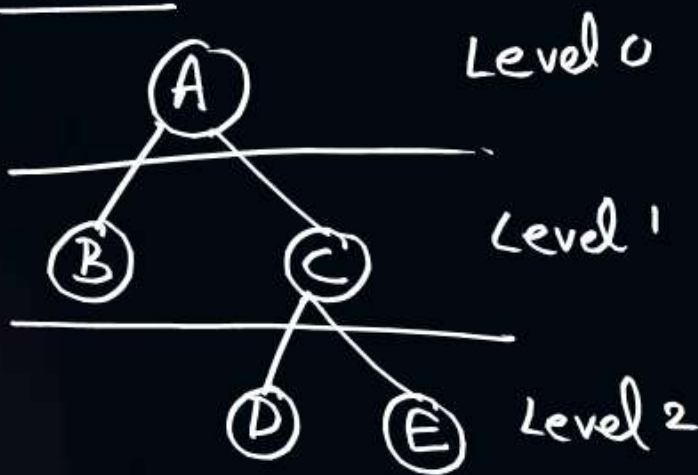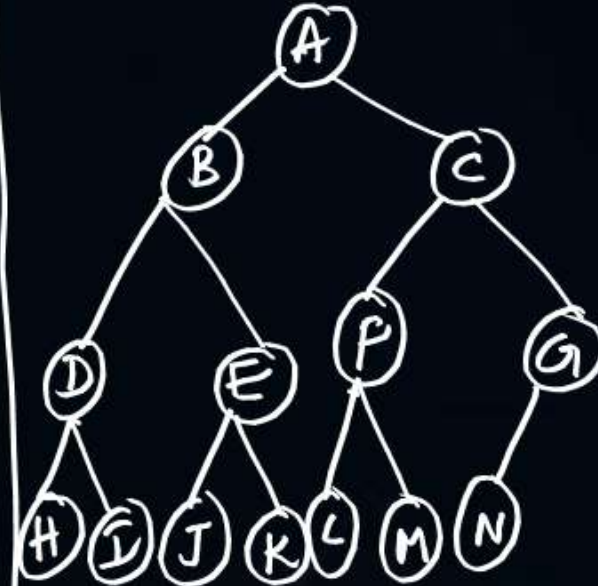
Not a FBT, $\deg(G)=1$
Not a PBT

NOTE: Every PBT is FBT, but Every FBT is Not PBT

Complete Binary Tree

- A Binary Tree in which,

  - Filling of Nodes must be from Top to bottom, Left to right.

  - There can be only left child (or) both left and right children, but not only right child for any Node.

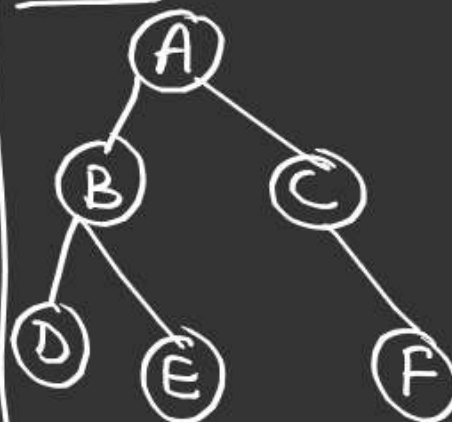Ex:1

(A)

FBT, PBT, CBT

Ex:2

(A)
 /
(B)

CBT, Not FBT
Not PBT

Ex:3

(A)
 / \
(B) (C)
     / \
   (D) (E)

- Not co CBT
- FBT
- Not PBT

Ex:4

(A)
 / \
(B) (C)
 /   / \
(D) (E) (F)
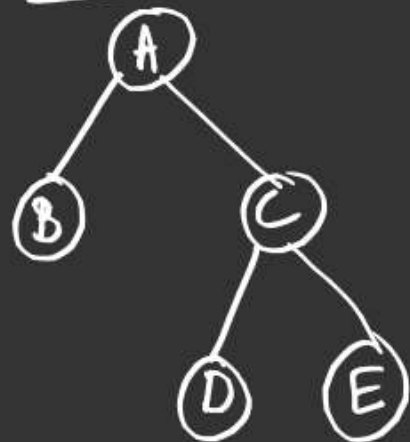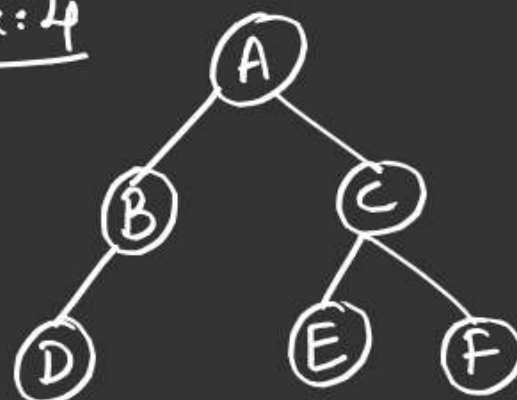
- Not CBT
- Not FBT
- Not PBT

Ex:5

(A)
 / \
(B) (C)
 / \   \
(D)(E) (F)

- Not CBT
- Not FBT
- Not PBT

Ex:6

(A)
 / \
(B) (C)
 / \
(D)(E)

- CBT
- FBT
- Not PBT

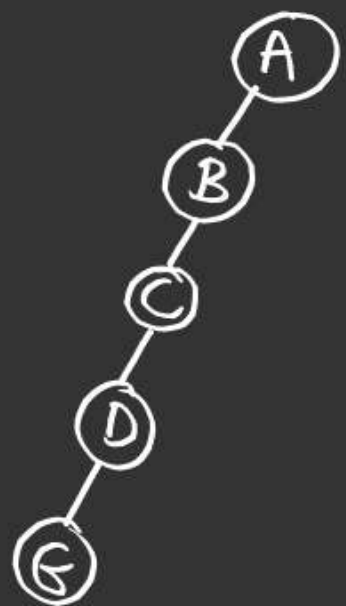Ex:7

(A)
 / \
(B) (C)
 / \ / \
(D)(E)(F)(G)
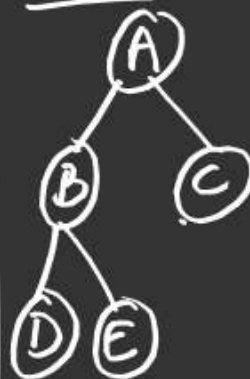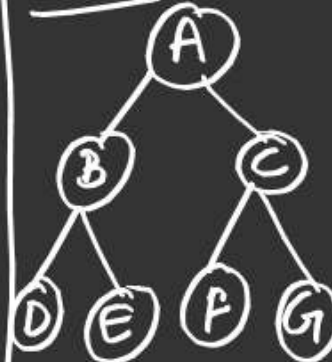
- CBT
- FBT
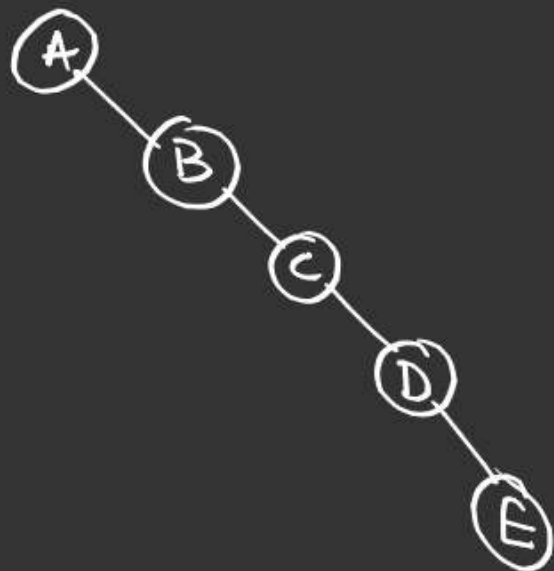- PBT

Skewed Binary Tree

A Binary Tree in which
- Each Non-leaf Node degree == 1
- All Non-leaf Nodes have
  Either only left child (or)
  only right child

Left Skewed

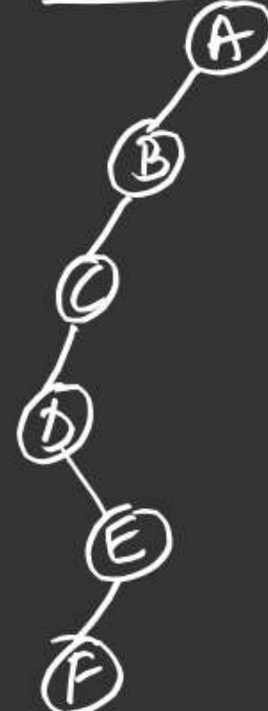
Right Skewed

Degenerated Binary Tree

- Each Node degree == 1 (Except leaf Node)
- It can be either left (or) Right child
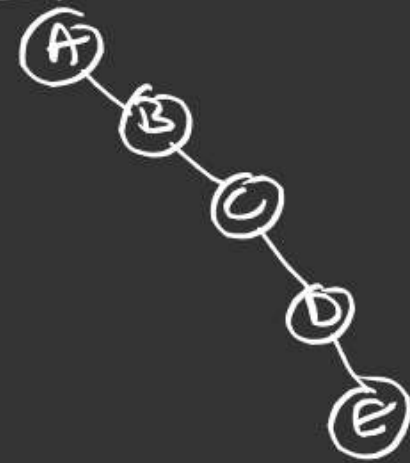
Ex:1

- DBT
- Not SBT

Ex:2

- DBT
- Not SBT

Ex:3

- SBT
- DBT

Ex:4

- SBT
- DBT

Every SBT is DBT, but Not Every DBT is SBT

- Tree Terminology

    - degree, depth, height, level

    - Root, siblings, leaf, ancestor, Predecessor

- Types of Binary Trees

    - FBT, PBT, CBT, SBT, DBT

To be Contd ...

THANK - YOU