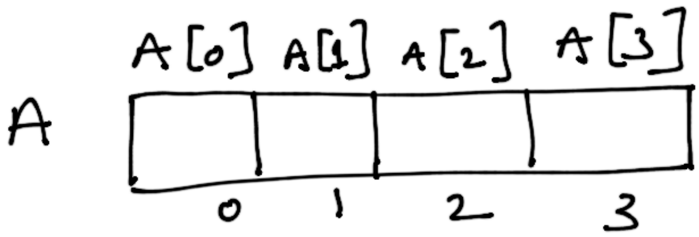
Data Structures

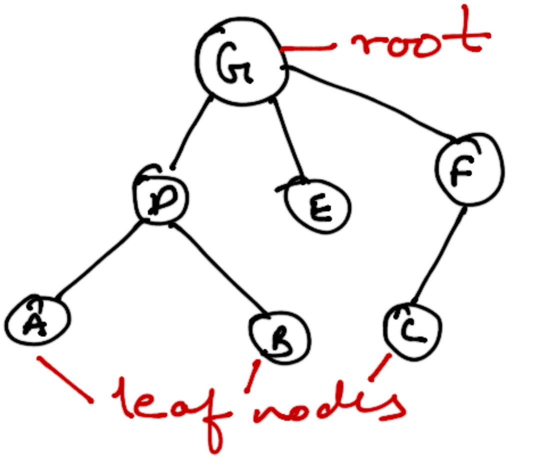
Linear

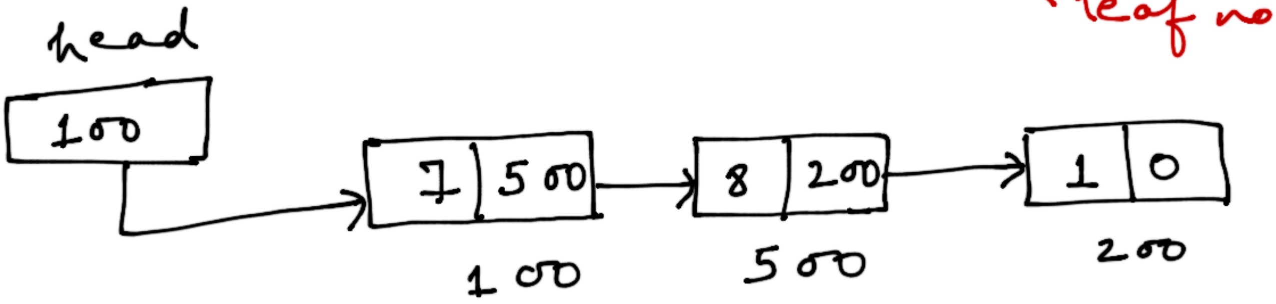
{arrays, linked list, stacks, Queues}

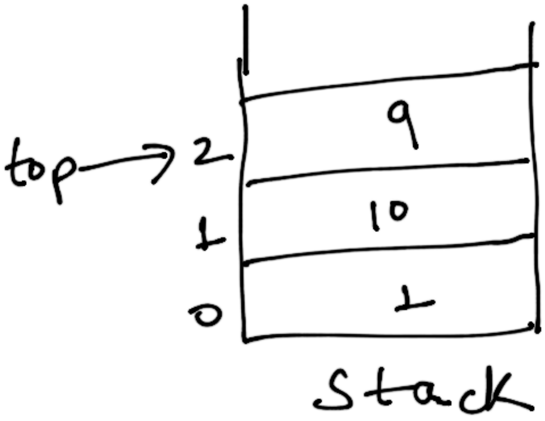
Non-linear

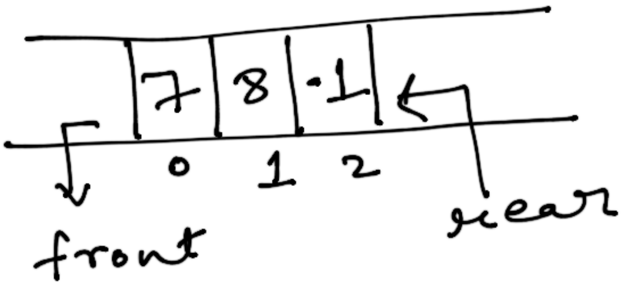
{trees, Graphs}





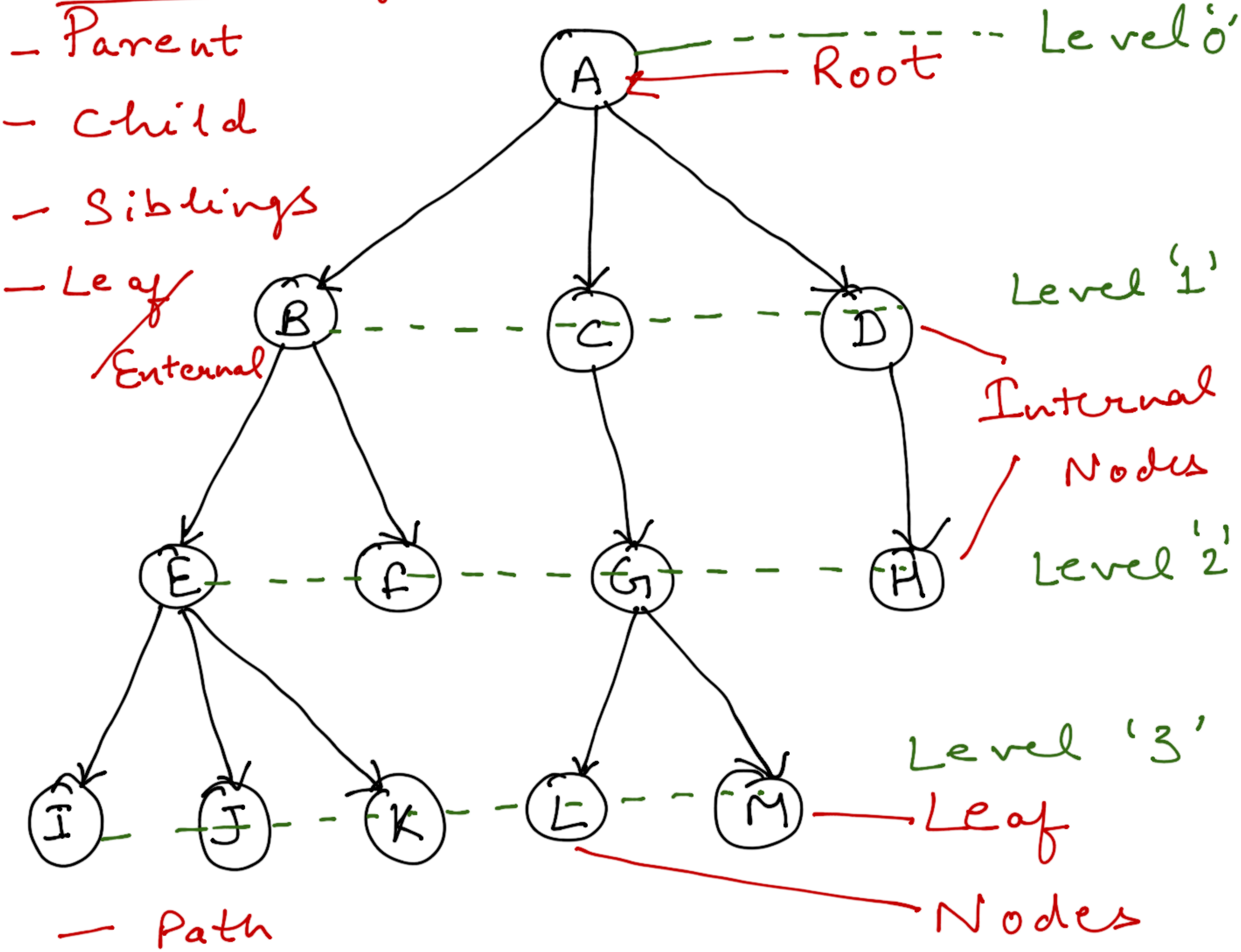






Terminology

- Parent
- Child
- Siblings
- Leaf / External



- Path
- Predecessor / Successor
- Ancestor / Descendant
- Sub-Tree
- Height of node (No. of edges in longest path from node to leaf)

- Degree (No. of children
- Degree of tree (Max. degree among nodes)
- Depth of node (Edges from Root to node

Root =

Leaf Nodes =

Internal Nodes =

Child of G =

Parent of G =

Predecessor of G =

Successor of G =

Ancestor of G =

Descendant of G =

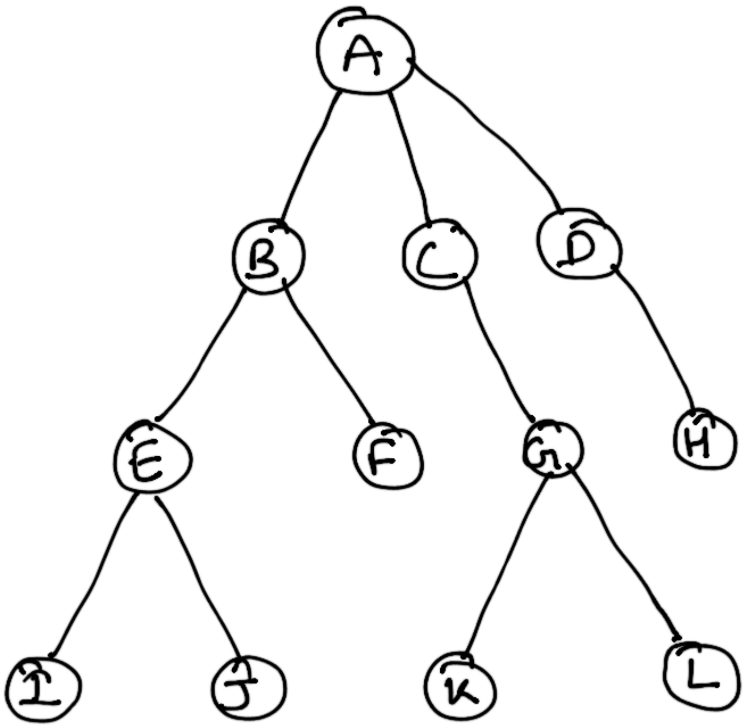
Height of G =

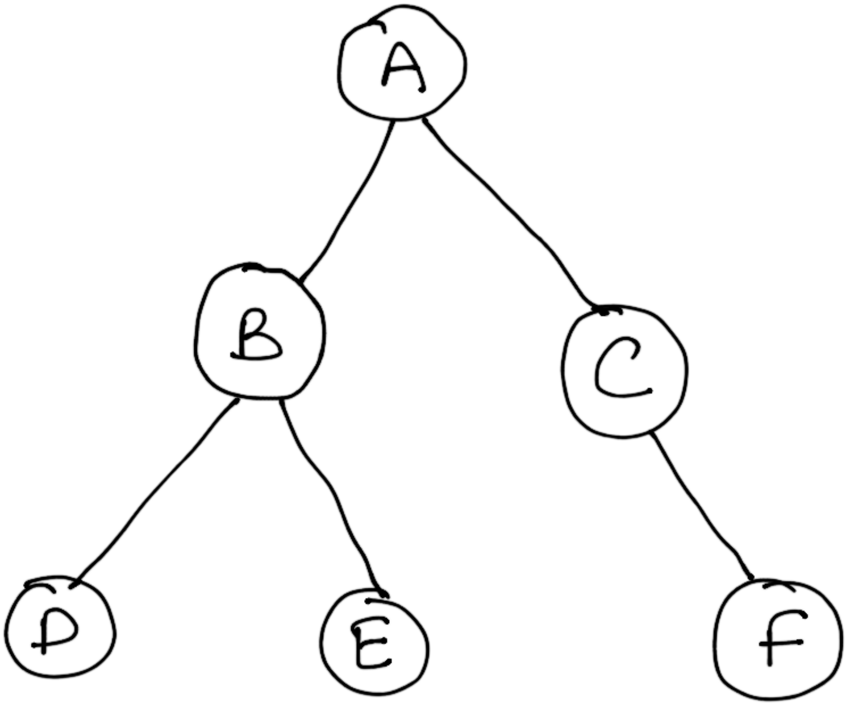
Depth of G =

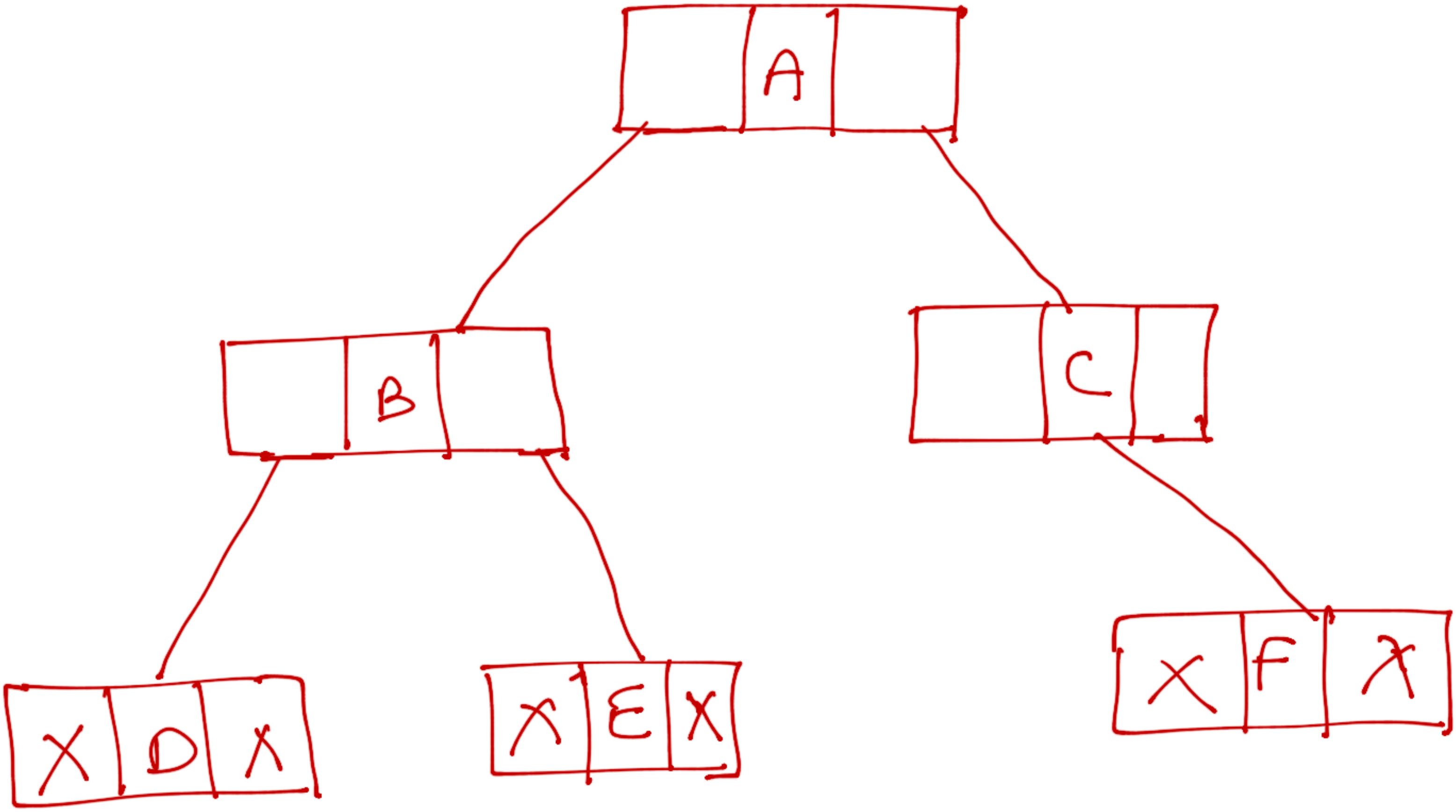
Level of G =

Degree of Tree =

Height of Tree =





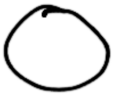


Hierarchial Model

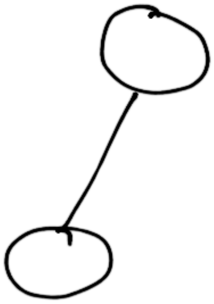
Binary Tree & its types

- Each node can have 'atmost' two children

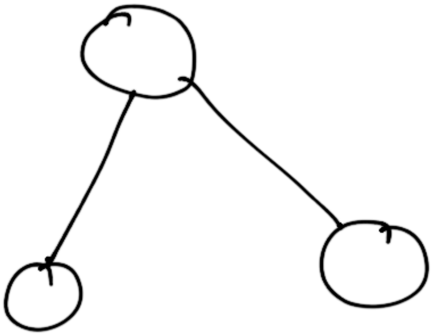
Children – 0, 1, 2



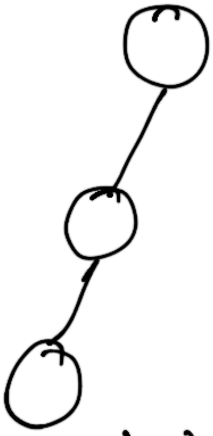
(i) ___



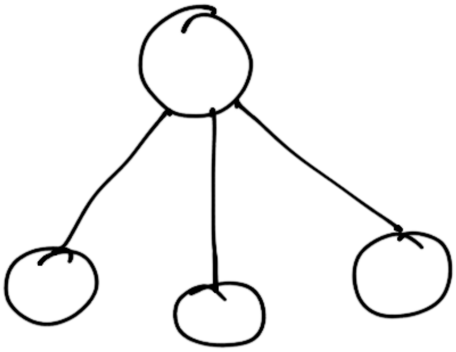
(ii) ___



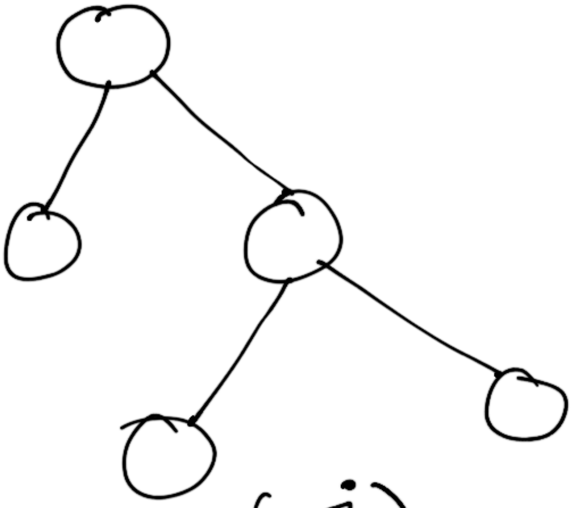
(iii) ___



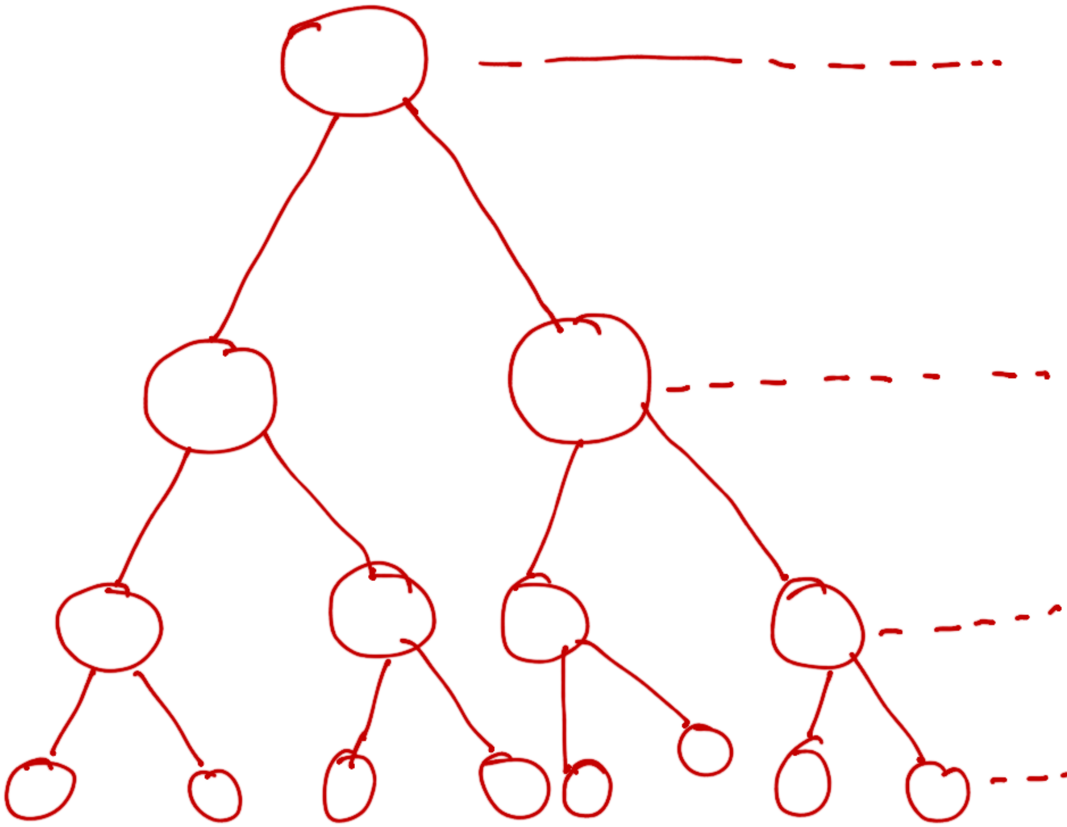
(iv) ___



(v) ___



(vi) ___



Level	Max. No. of Nodes
0	$2^0 = 1$
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$

Max. No. of nodes at any level 'i' $= 2^i$

Height of tree = Height of Root Nod

= No. of edges in longest path from Root to the leaf node

For this tree, height is 3

Max. no. of nodes in tree of height '3'

$= 2^0 + 2^1 + 2^2 + 2^3$

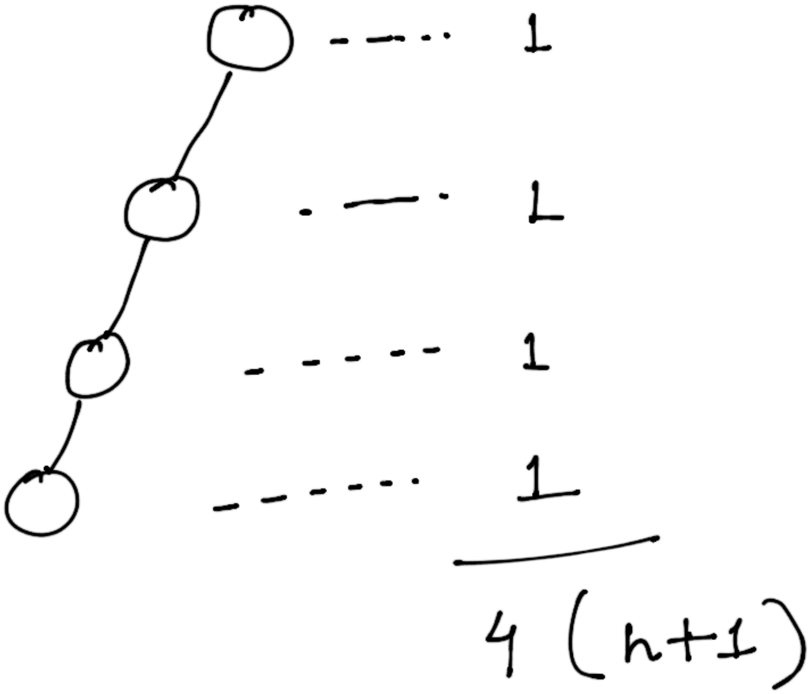
$= 1 + 2 + 4 + 8 = 15 \left(2^{h+1} - 1\right)$

Max. no. of nodes in tree of height 'h'

$$= 2^0 + 2^1 + 2^2 + \ldots\ldots\ldots\ldots + 2^h$$

$$= 2^{h+1} - 1$$

Min. no. of nodes in tree of ht. 'h' $= h+1$



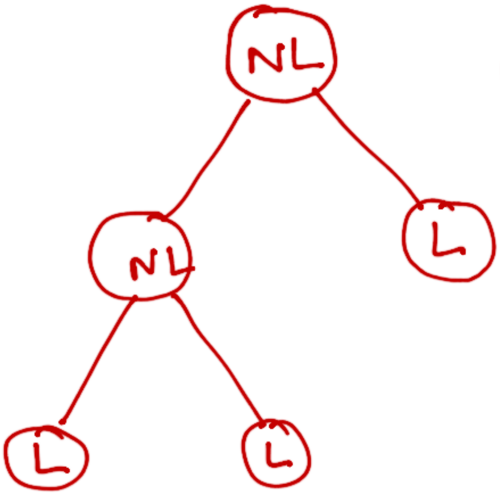
Max. ht. = when min. no. of nodes $= n-1$

Min. ht = when max. no. of nodes $= \lceil \log_2(n+1) - 1 \rceil$

Types of Binary Tree

→ Full / Proper / Strict

- All nodes have 2 children, except the leaf nodes
- No. of leaf nodes = No. of non-leaf nodes + 1



No. of non-leaf nodes = 2

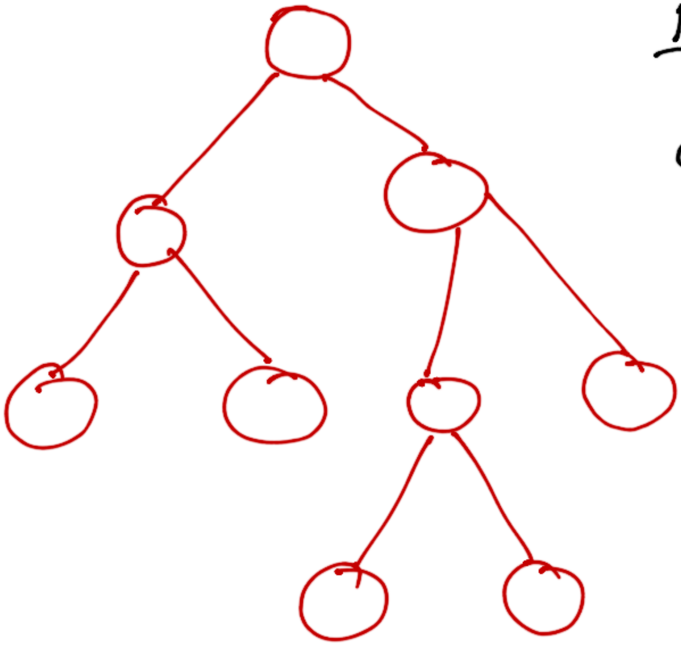
Leaf nodes = 2 + 1 = 3

Max. no. of nodes = $2^{h+1} - 1$

Min no of nodes = $2h + 1$

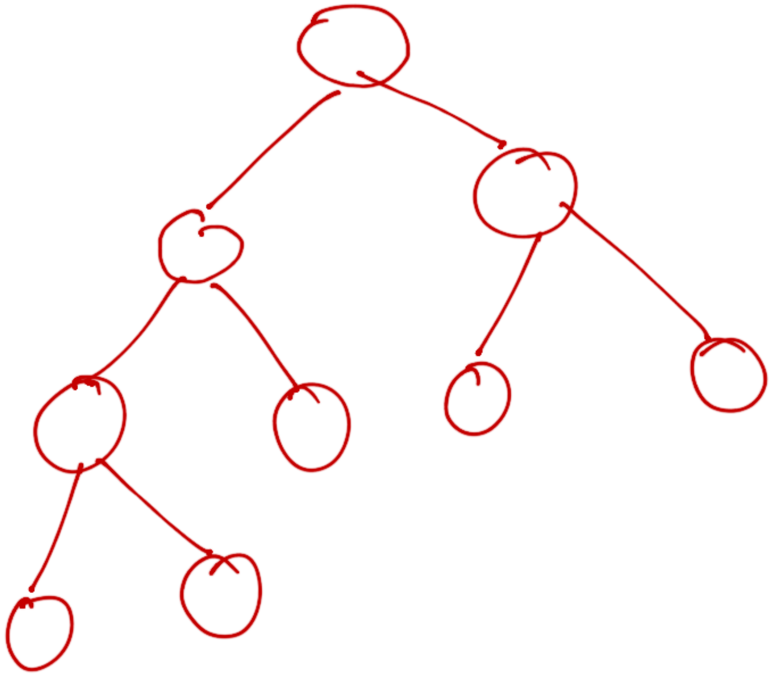
→ Min ht = $\lceil \log_2(n+1) - 1 \rceil$ Max ht = $(n-1)/2$

Complete Binary Tree



1) All levels completely filled except the last

2) In the last level, node must be as left as possible



Max nodes $= 2^{h+1}-1$

Min nodes $= 2^h$

Max ht $= \lceil \log_2(n+1)-1 \rceil$

Min ht $= \log_2 n$