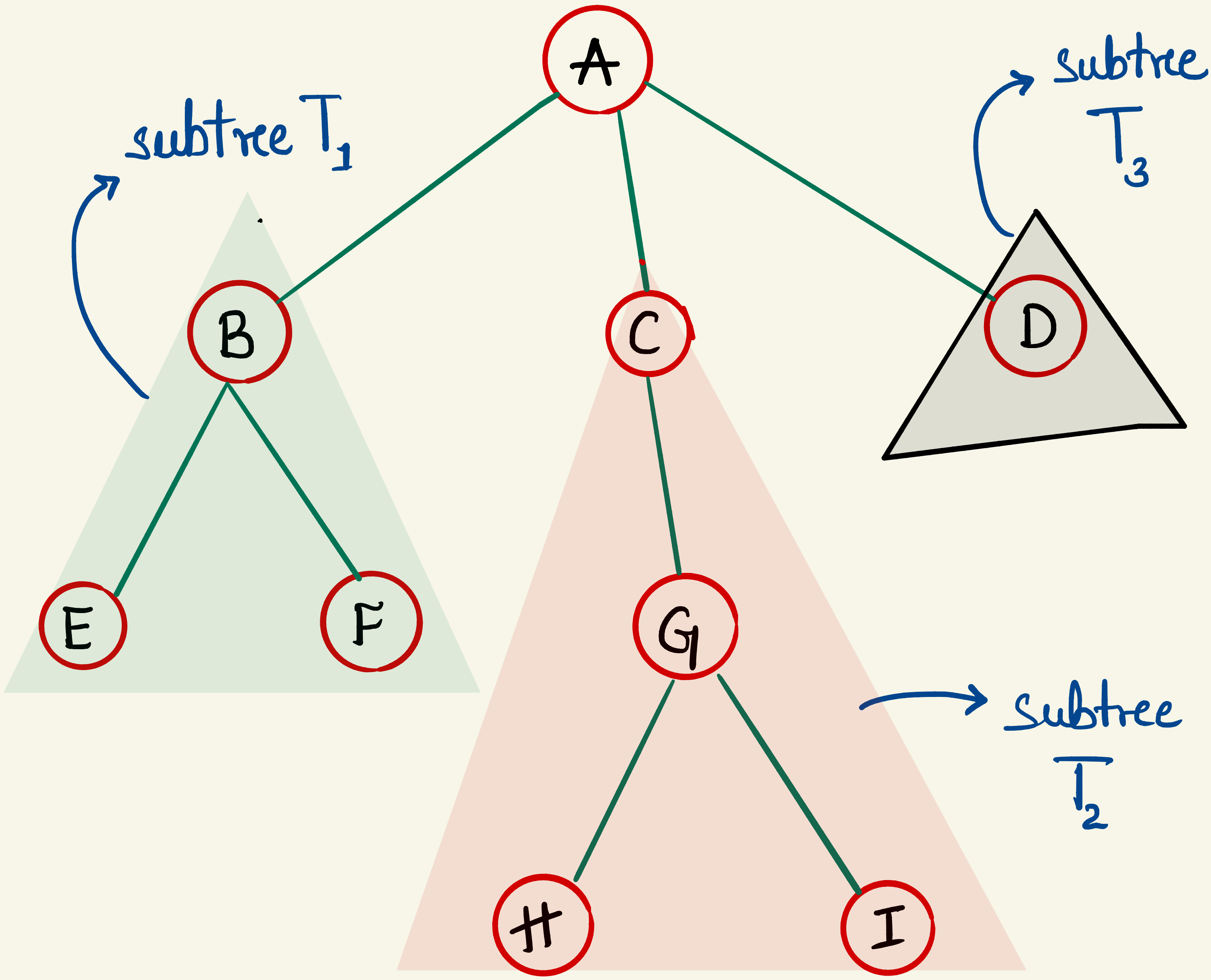
# Trees

## Definitions and Terminology

**Definition:**

A tree is a finite set of one or more objects (nodes) such that:

1. There is a specially designated node called root.
2. Remaining nodes are partitioned into $n (\geq 0)$ disjoint sets $T_1, T_2, \ldots, T_n$, where each of these sets ($T_i$, $\forall i = 1 \ldots n$) are called subtrees of the root.
   (subtrees are trees)



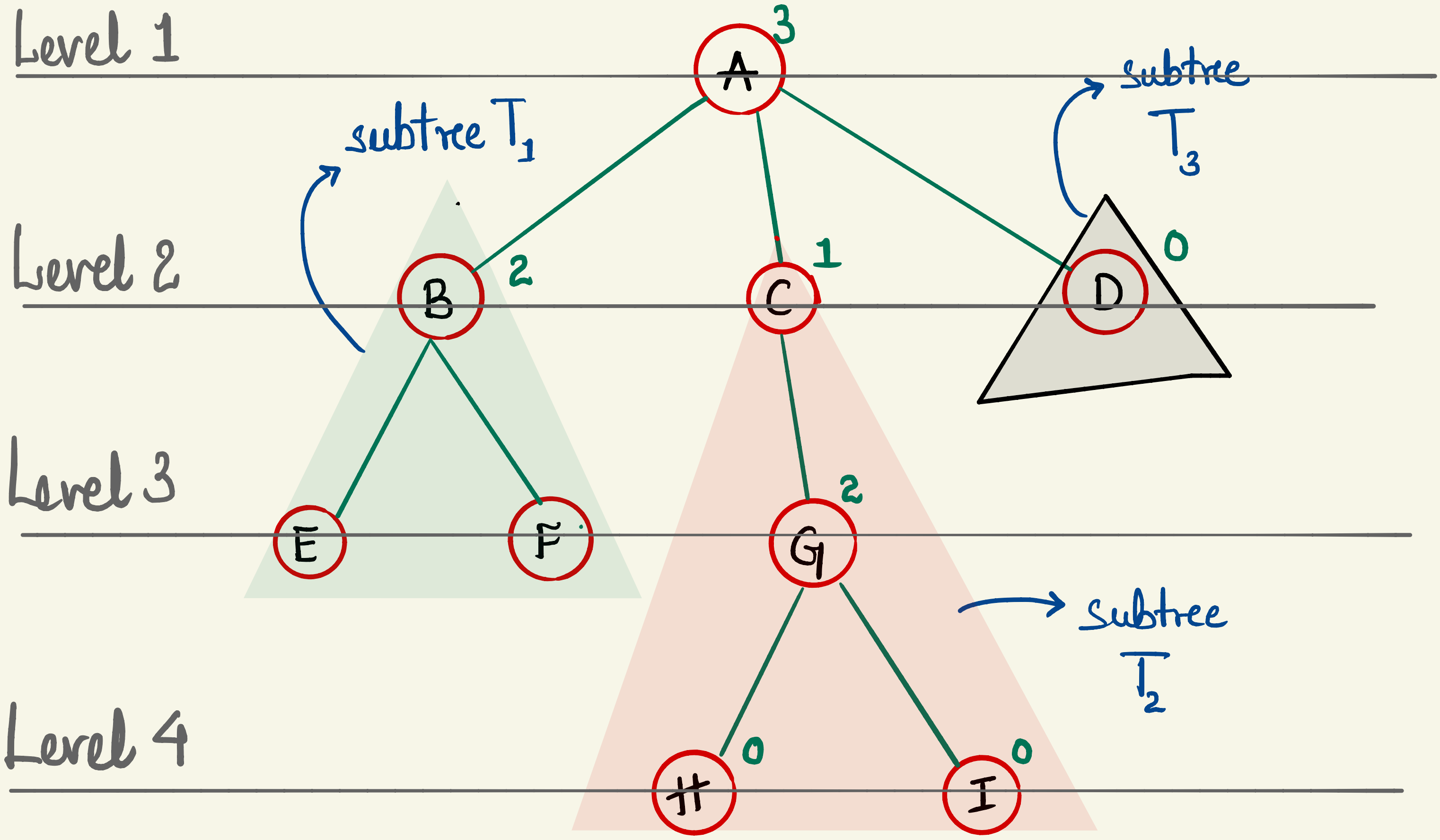

Notice $T_1, T_2, T_3$ are disjoint sets and $T_1$, $T_2$ and $T_3$ are trees.

Let's call the tree as "T"

A is root node of tree T

Nodes E, F, H, I, D are leaf nodes of the tree

(terminal nodes)

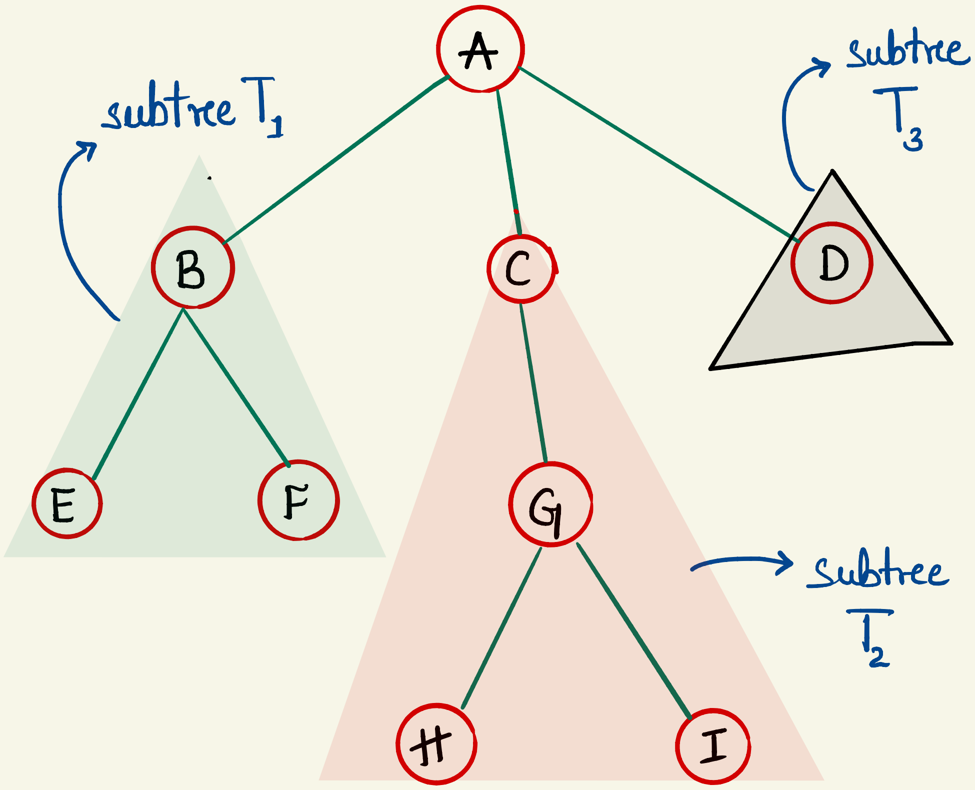



No. of subtrees of a node is called its **degree**.

eg :- A's degree is 3, G's degree is 2.
degree of E, F, H, I and D is 0.

**Degree of a tree** is the max. of degree of the nodes.

**Level** = 1 + (No. of edges between root and a node)




Nodes B, C, D are children of node A and therefore A is the parent node of B, C, D.

||ly, children of G are H and I. and parent of H and I is G.

Children of same parent are siblings.

eg:- E and F are siblings

Ancestors of a node are all the nodes along the path from root to that node eg. Ancestors of node H are G, C, and A

**Descentants** of a node are all the nodes from that node to the leaf node (including leaf node:

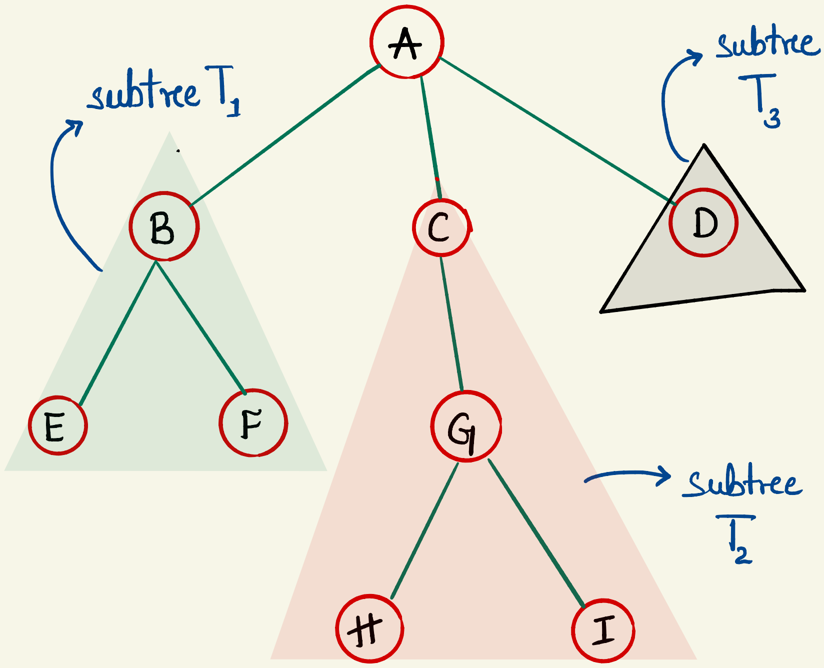
eg :- E and F are desentants of B.

Note :- If a node is at level 'l' then its children are at level l+1.
(In some books levels are defined from 0)

**Height** or **depth** of a tree is defined as the maximum level of any node in the tree.

"Representions"

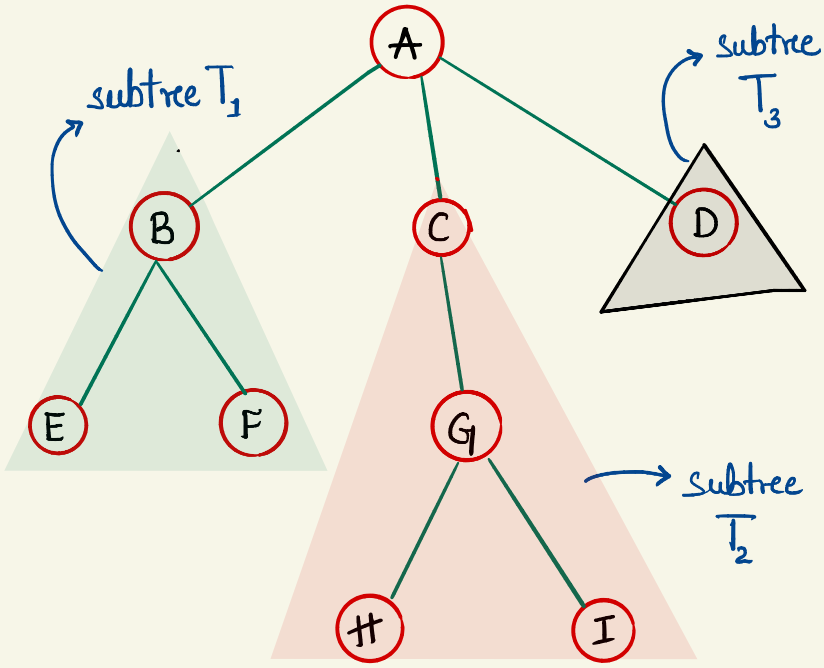
(i) List Representation



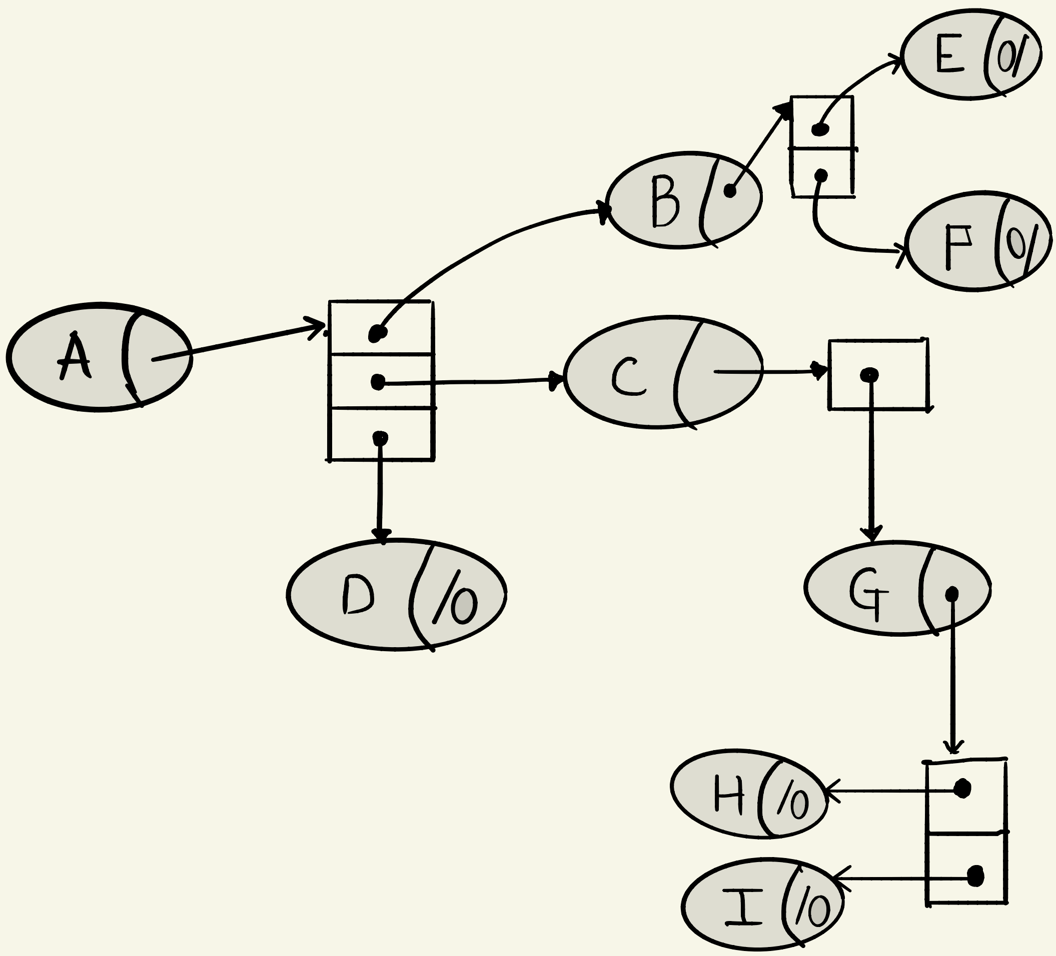

## List Representation :-

$$(\text{root}\ (T_1, T_2, T_3, \ldots, T_n))$$

$$(A(B(E,F), C(G(H,I)), D)$$

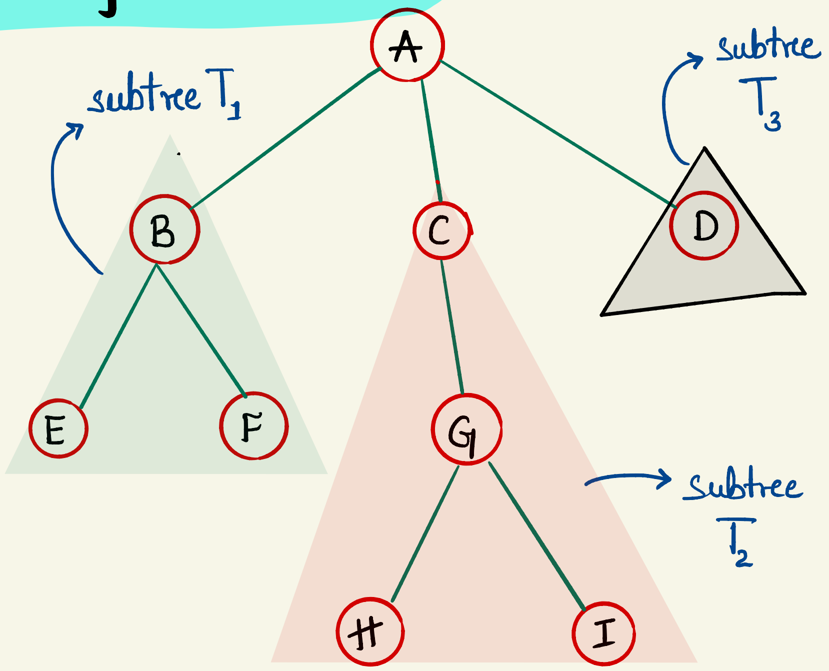


A
B
C
D
E
F
G
H
I
subtree $T_1$
subtree $T_2$
subtree $T_3$

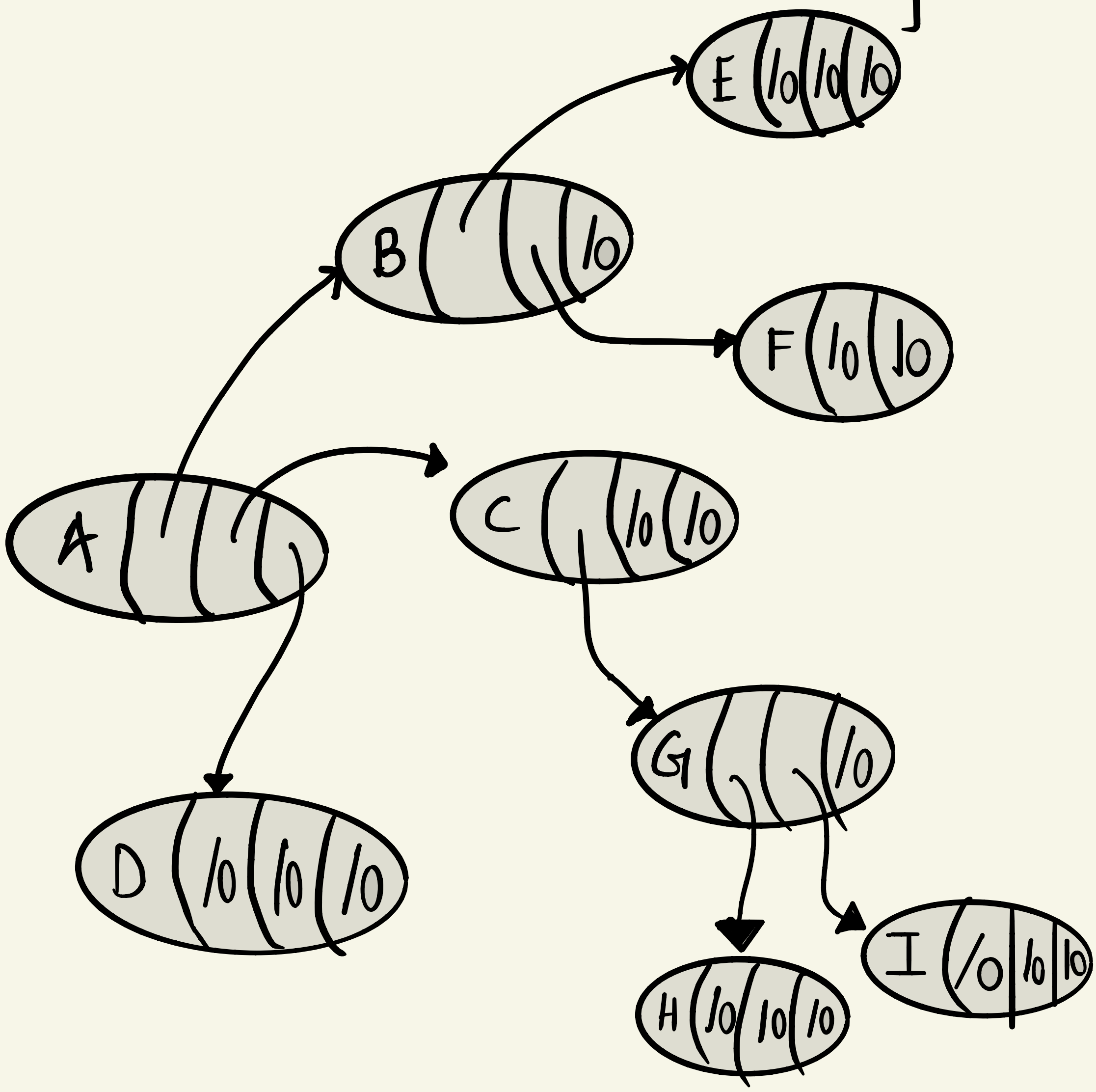
A
B
C
D
E
F
G
H
I
0
0
0
0
0

# K-ary tree Representation:




→ 3-ary tree

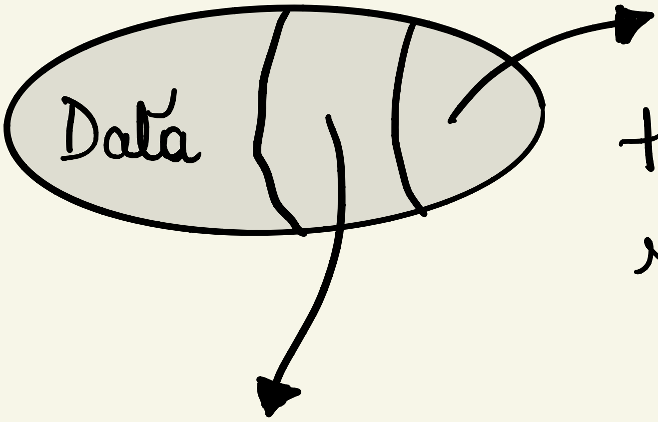
K → degree of tree (Maximum of degree of nodes in a tree)



# Left Child - Right Sibling Representation.

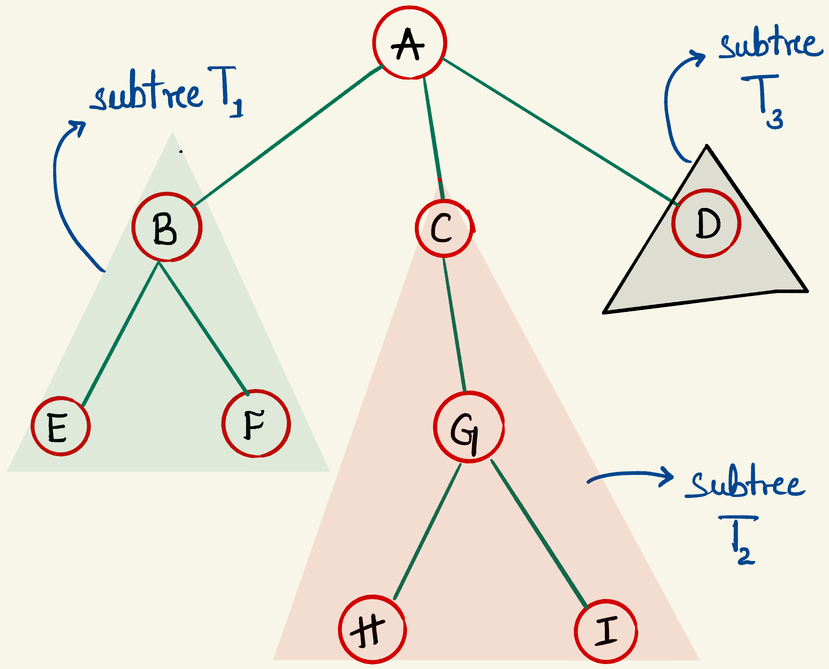
A node of a tree is represented as.

node 'n'

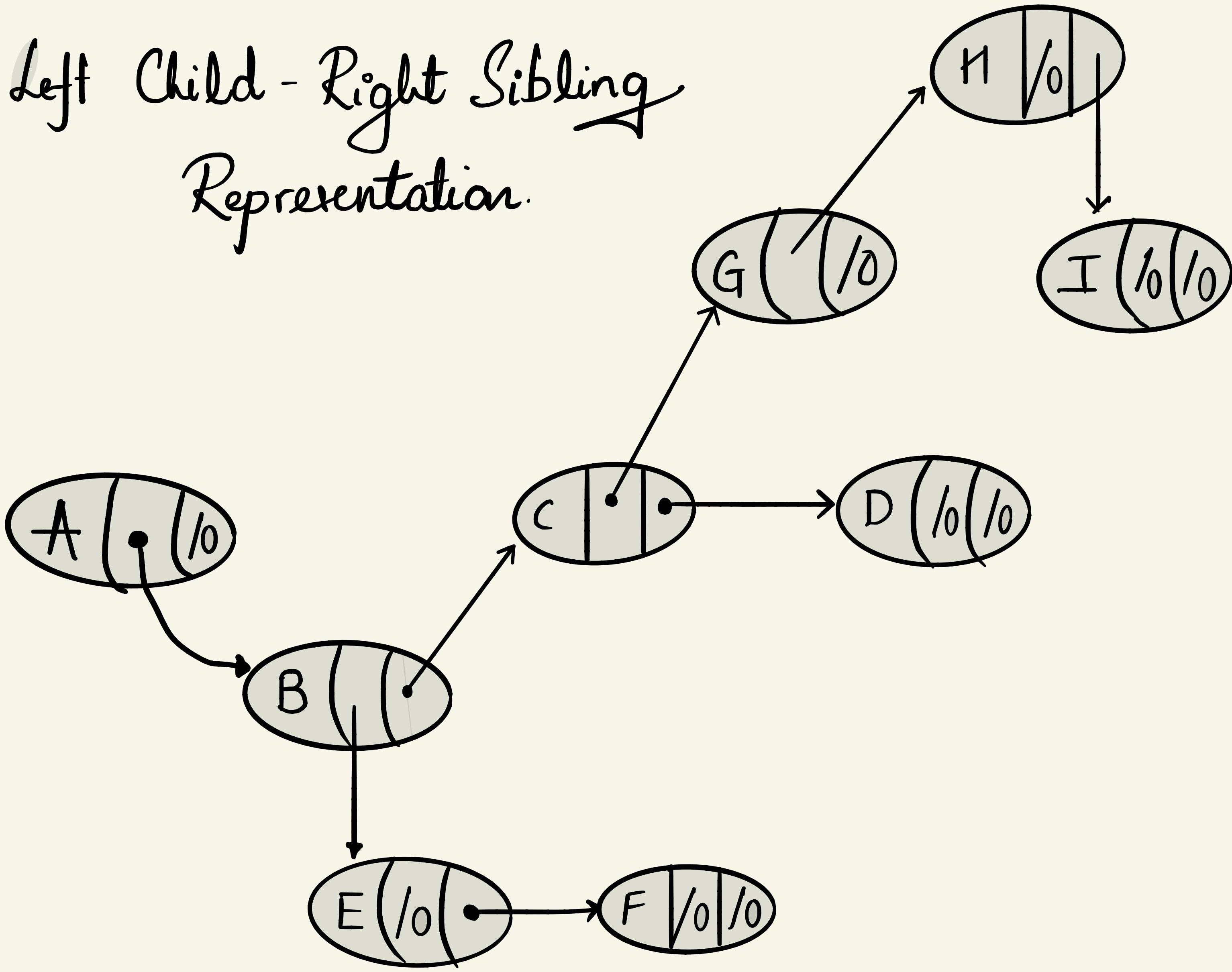



Points to the closest right sibling of "node 'n'"

Points to leftmost child of the node 'n'




# Left Child - Right Sibling Representation.



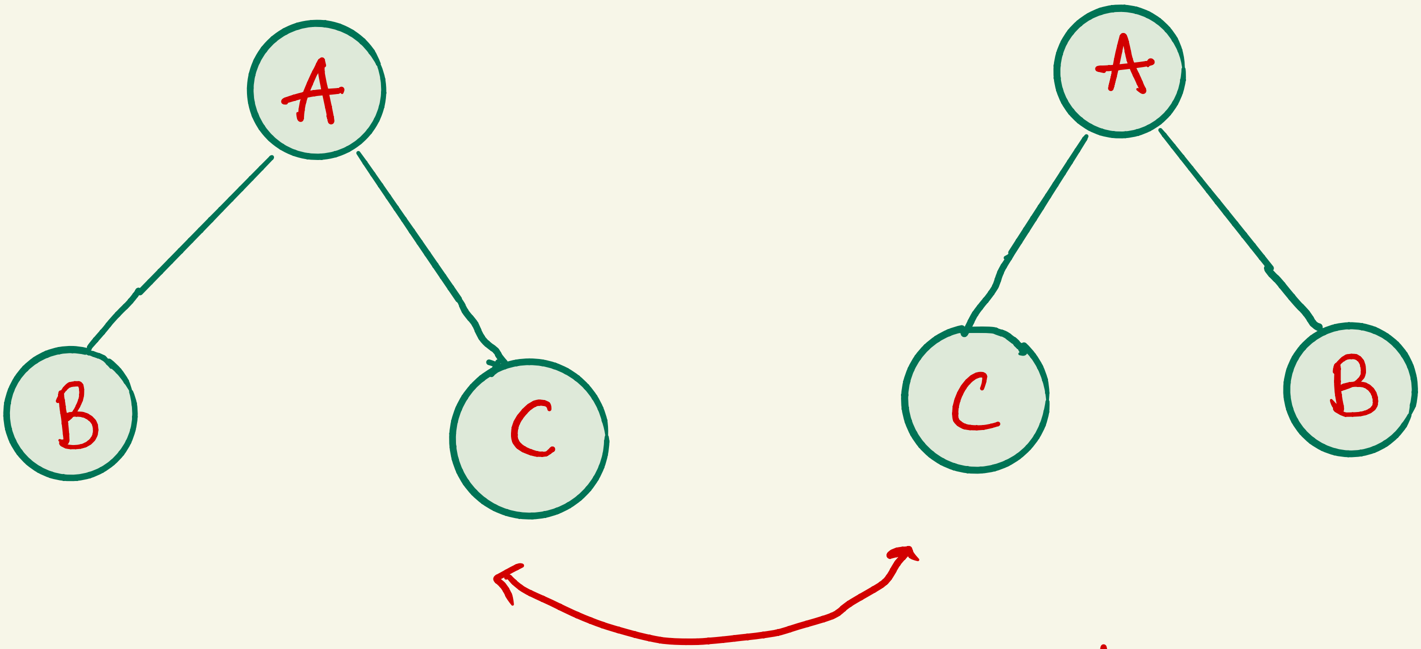
# Binary Trees

A binary tree is a finite set of objects that is either empty, or consists of root and two disjoint binary trees called the left subtree and the right subtree.

Notice the differences between the definition of a tree and a Binary tree

1. A tree must have atleast one node (root node) whereas a binary tree can be empty (with no nodes)

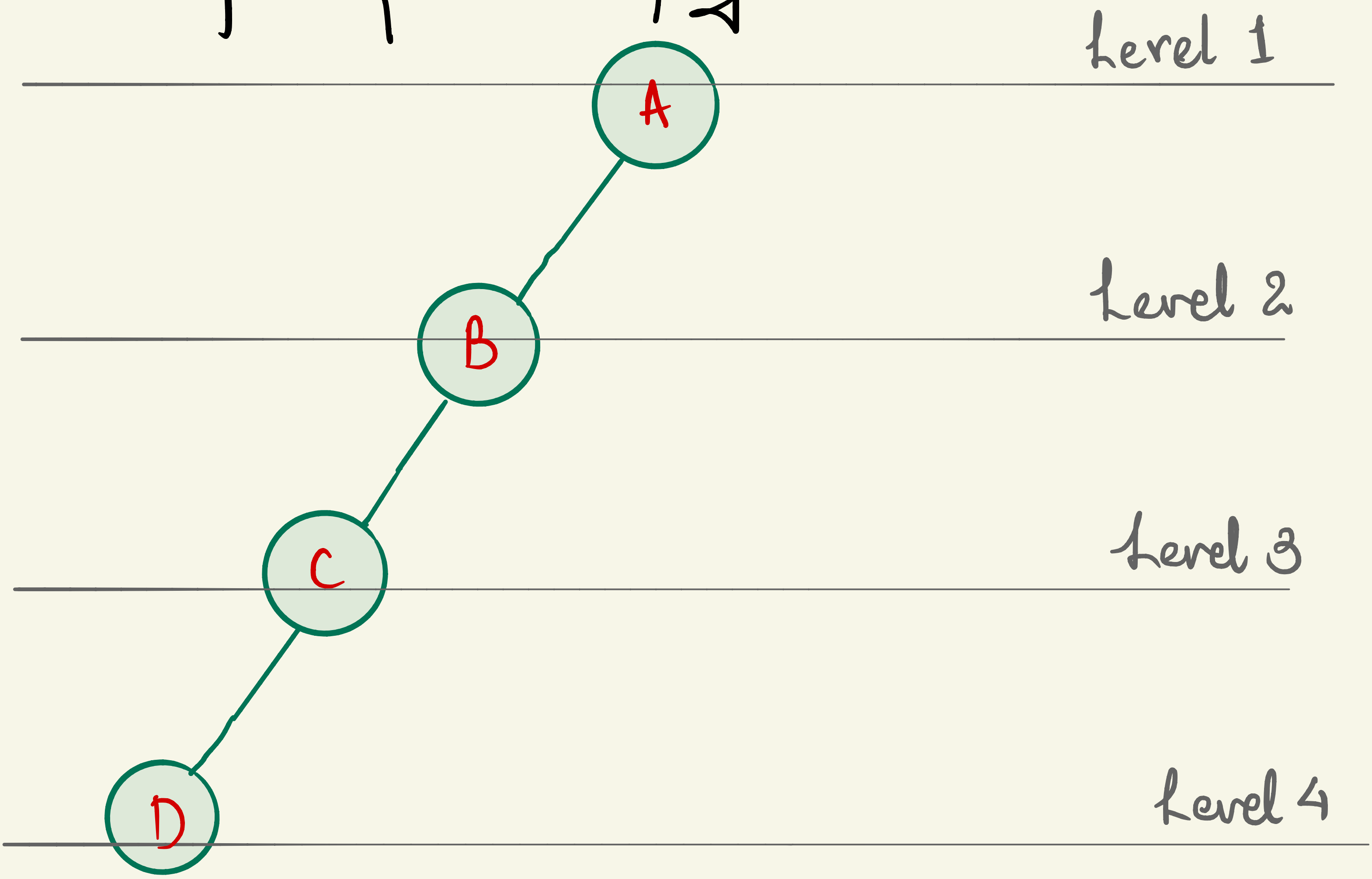
2. In binary trees, we distingush between the order of the children (left child and right child). In a tree we donot make this distinction.



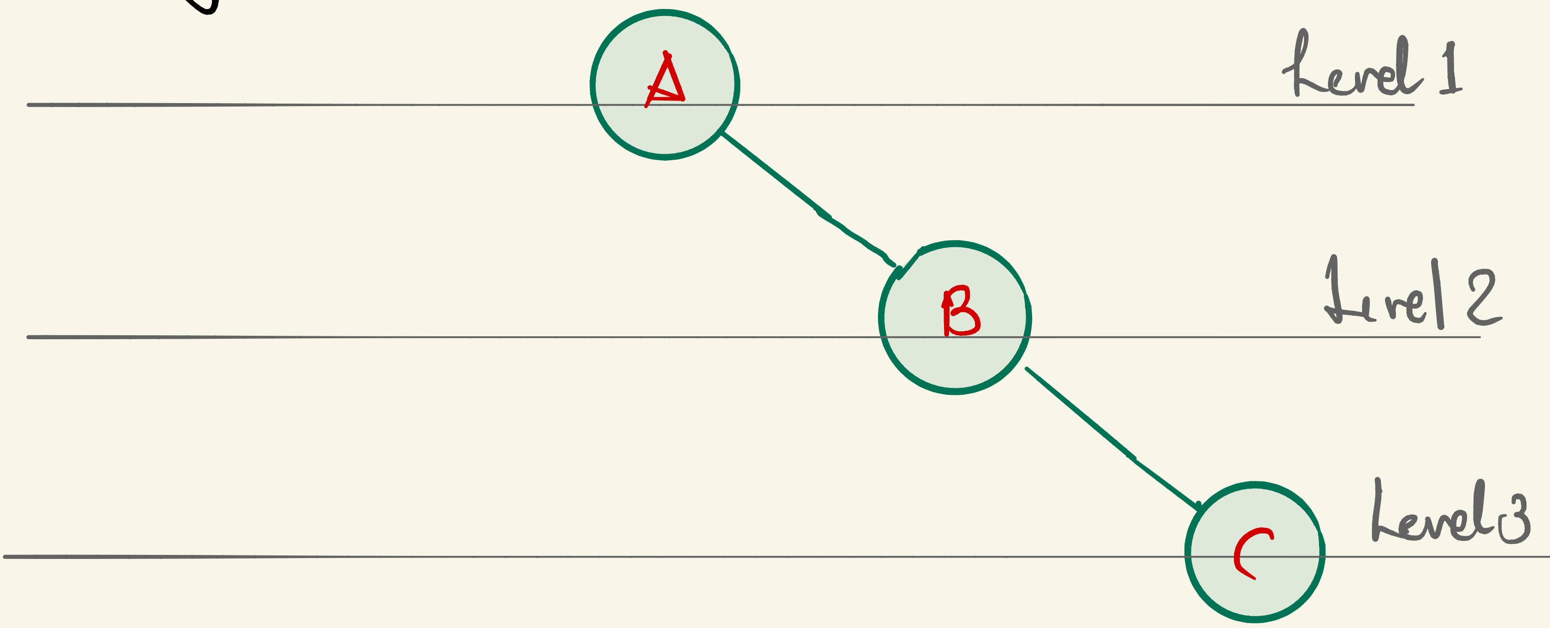

different Binary trees.

## Types of Binary trees.

1. Left-Skewed Binary tree



## 2. Right - Skewed Binary Tree.




**Observations:**

1) Maximum no. of nodes on level '$l$' of a binary tree is $\underline{\quad 2^{l-1} \quad}$.

$l = 1$ max. #nodes $= 1$

$l = 2$ max. #nodes $= 2^{2-1} = 2$

$l = 3$ max. #nodes $= 2^{3-1} = 4$

$\vdots \qquad \vdots \qquad \vdots$

(Proof By induction)

**Base:** $l = 1$

max # of nodes on level 1 $= 2^{l-1} = 2^0$
$= 1$

This one node is nothing but the root node.

**Hypothesis:** Let at any level $l = (k-1)$
the max # of nodes on level $l$ is $2^{k-2}$.

To prove:
the max # of nodes on level $l = k$ is
$2^{k-1}$.

Since, each node in a binary tree has a maximum degree 2 (By definition of binary tree). So, if the max no. of nodes at level $k-1$ is $2^{k-2}$ (induction hypothesis) then, maximum no. of nodes at level k of the binary must be

$$= 2 * (\text{max. no. of nodes at level } k-1)$$

$$= 2 * 2^{k-2}$$

$$= 2^{k-1}$$

(2) Maximum no. of nodes in a binary tree of depth k is ________.

depth = maximum level of any node in binary tree.

ie., if depth (d) = k then maximum no. of nodes in a binary tree = sum of maximum no. of nodes at each level of binary tree.

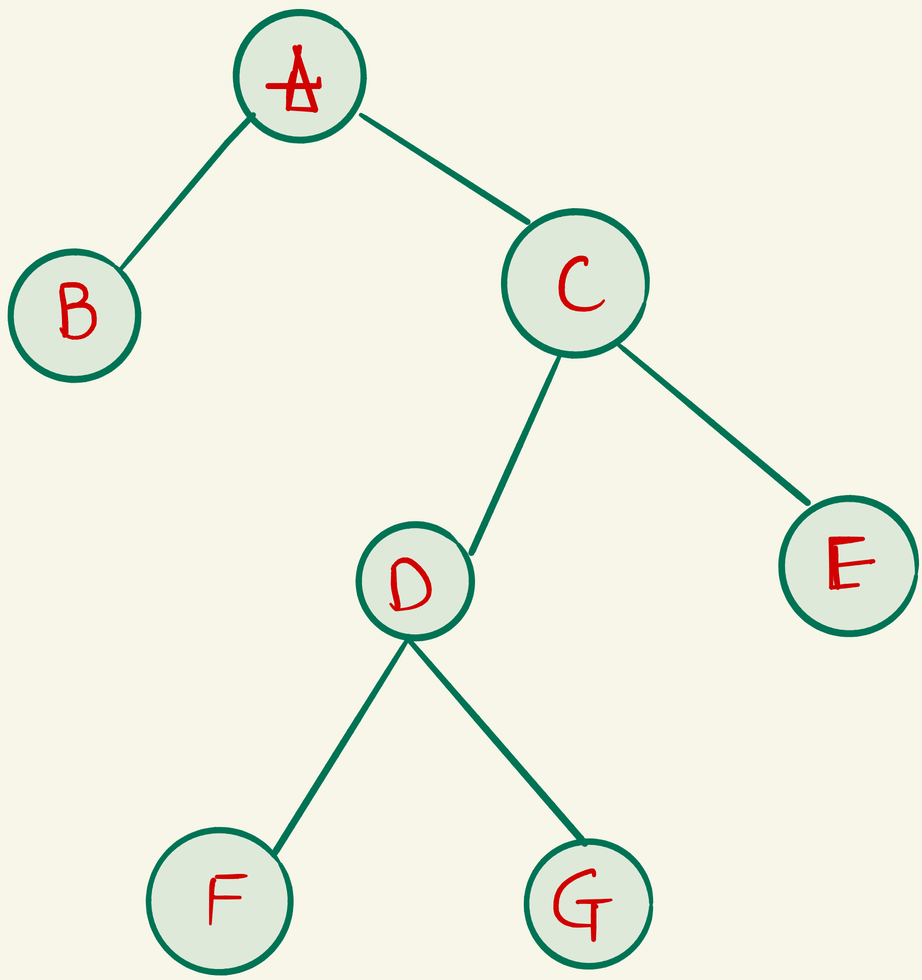
$$\text{Max \# of nodes in a binary tree} = \sum_{l=1}^{k} (\text{max \# of nodes at level } l)$$

$$= 2^0 + 2^1 + 2^2 + \cdots + 2^{k-1}$$

$$= \frac{1(2^k - 1)}{2-1} = 2^k - 1$$

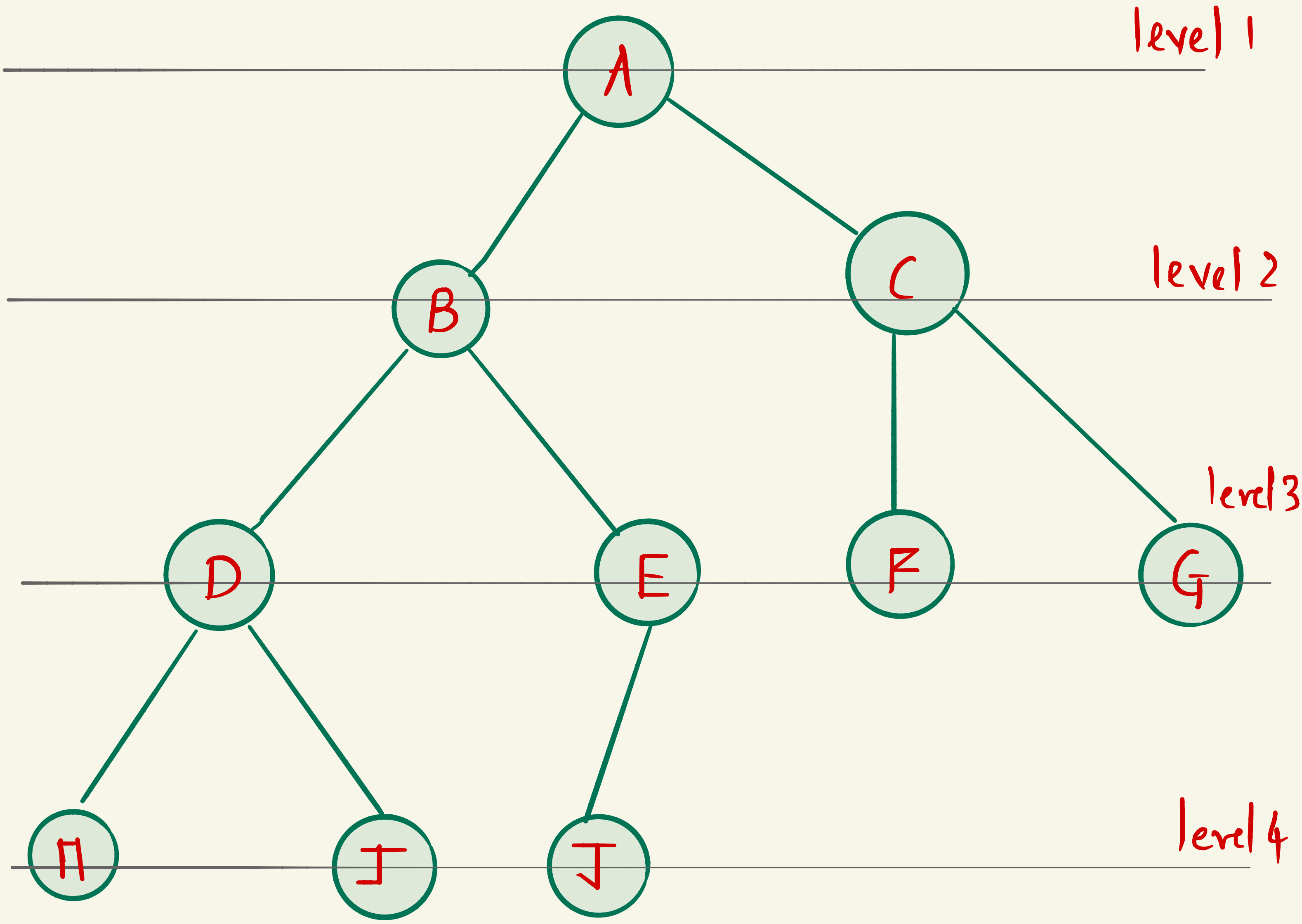
(3) Full Binary Tree.

Definition :- A full Binary tree is a binary tree in which every node has either 0 or 2 children.



# (4) Complete Binary Tree

A Complete Binary tree is a binary tree in which every level, except possibly last is completely filled and all nodes are as far left as possible.



# Binary Tree Representations.

## 1. Array Representations.

$\lfloor x \rfloor$ : greatest integer $\leq x$

If a complete binary tree with 'n' nodes is represented sequentially, then for any node with index $i$, $1 \leq i \leq n$, we have

(i) parent$(i)$ is at $\lfloor i/2 \rfloor$ if $i \neq 1$

if $i = 1$, then it is root node and has no parent.

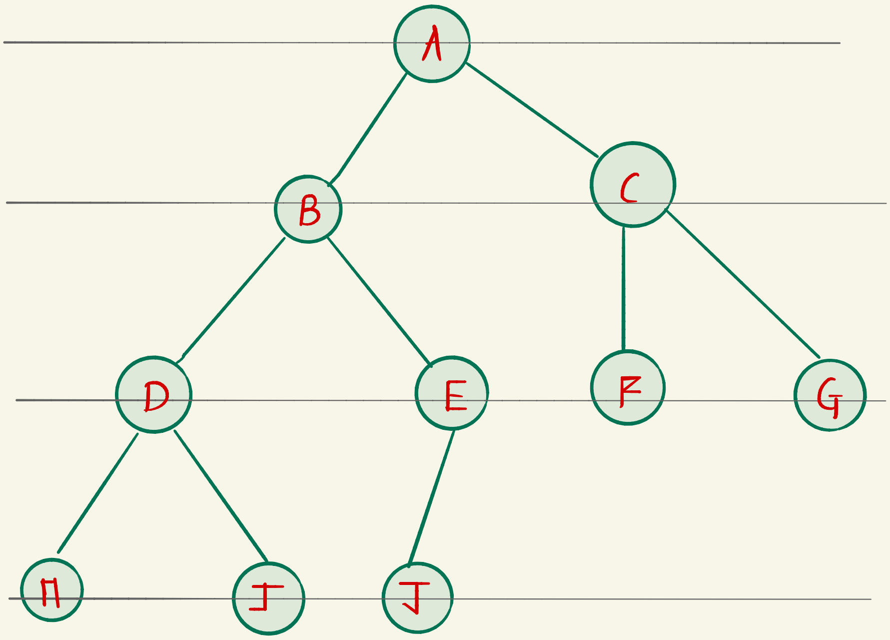
(ii) leftChild$(i)$ is at $2i$ if $2i \leq n$

If $2i > n$, then $i$ has no left child.

(iii) rightChild$(i)$ is at $2i+1$ if $(2i+1) \leq n$

if $(2i+1) > n$ then $i$ has no right child.

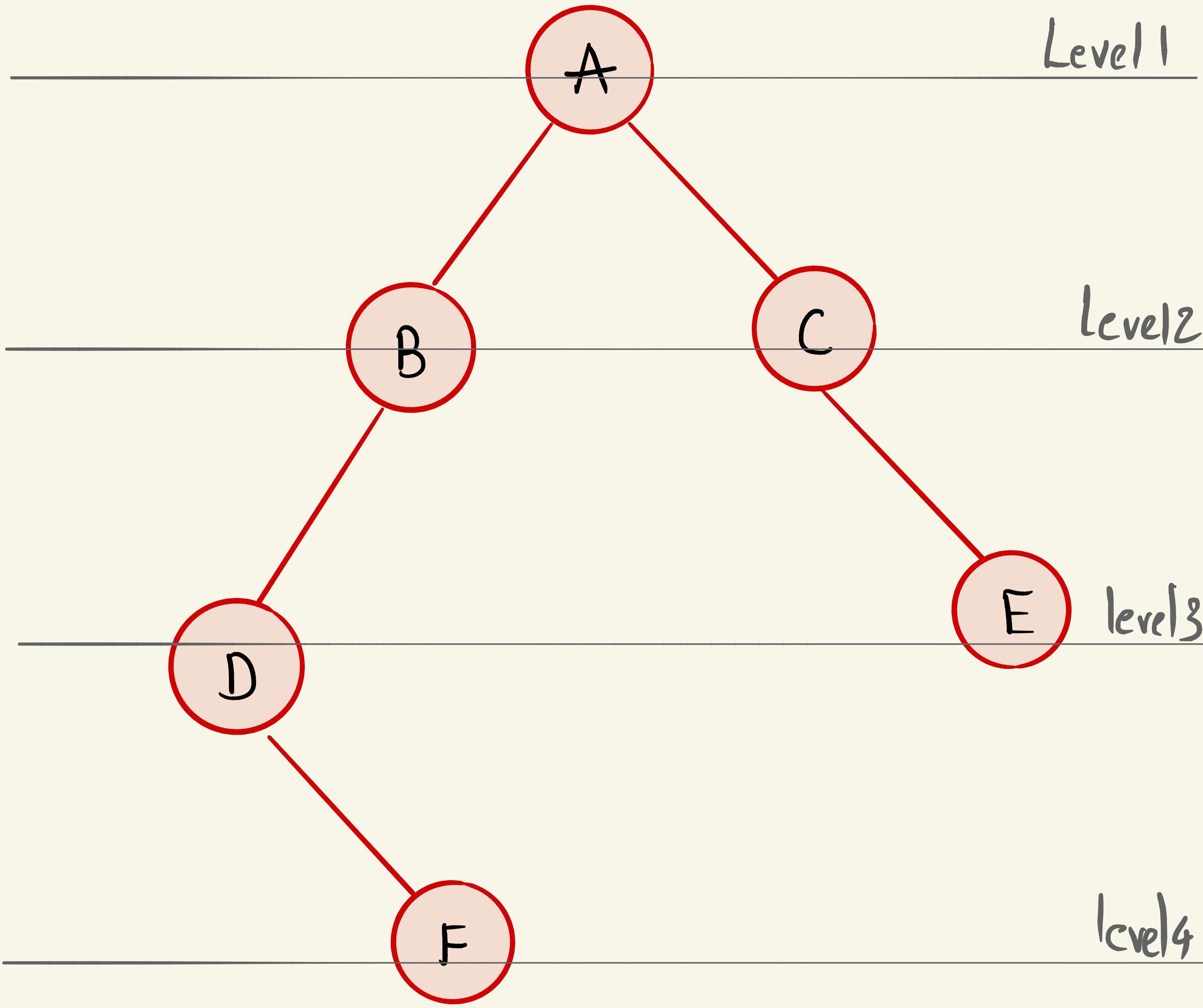
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| - | A | B | C | D | E | F | G | H | I | J |



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | A | B | C | D | - | - | E | - | F | - | - | - | - | - | - |

Max no. of nodes

$= 2^4 - 1 = 15$



# Linked Representation

```
template<class T>
class TreeNode
{
    private:
        T data;
        TreeNode<T>* leftChild;
        TreeNode<T>* rightChild;

}
template <class T>
class Tree {
    public:
        // Tree Operation.
    private:
        TreeNode<T>* root;
}
```