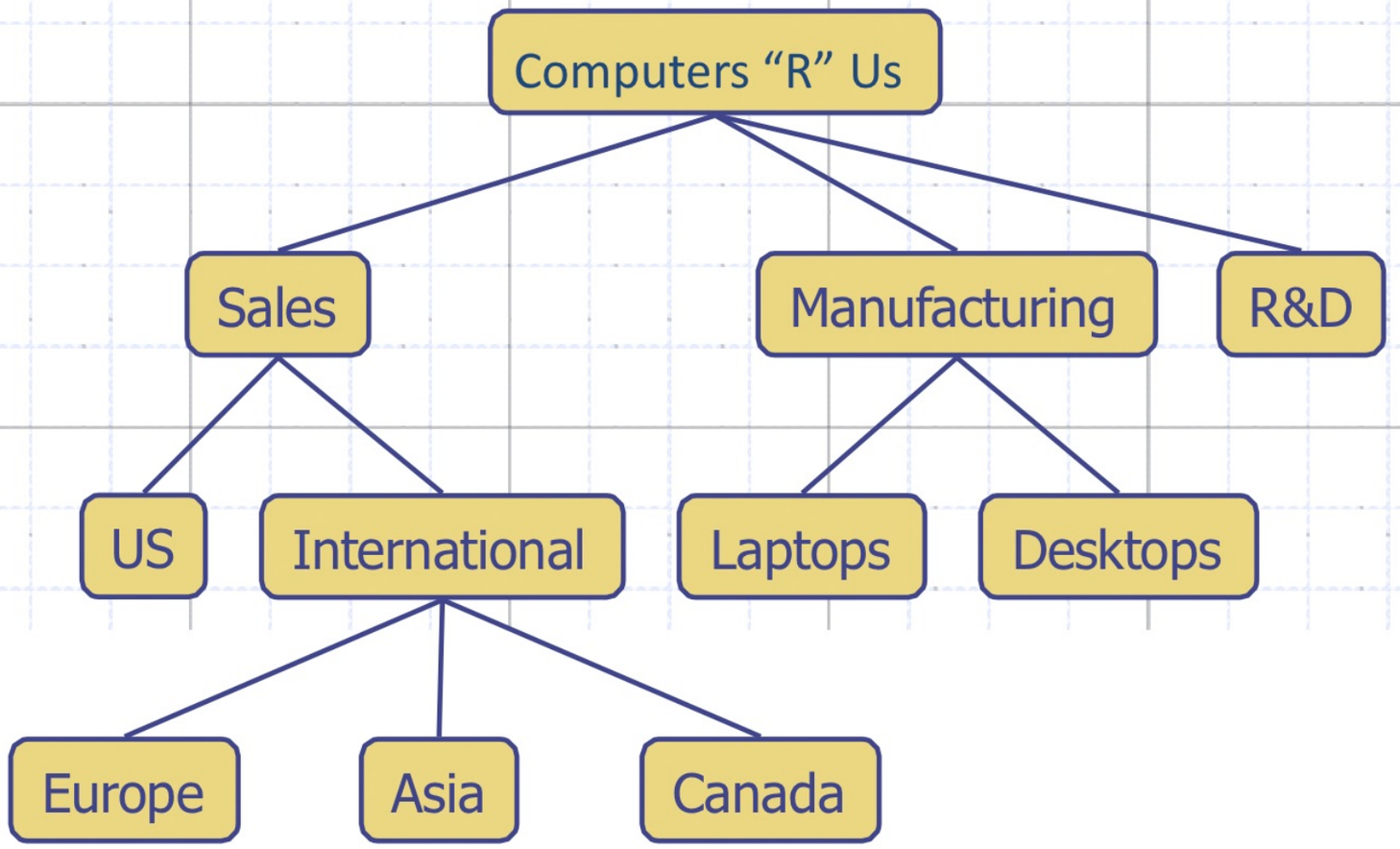
TREES

What is a Tree?

A tree is an abstract model of a hierarchical structure

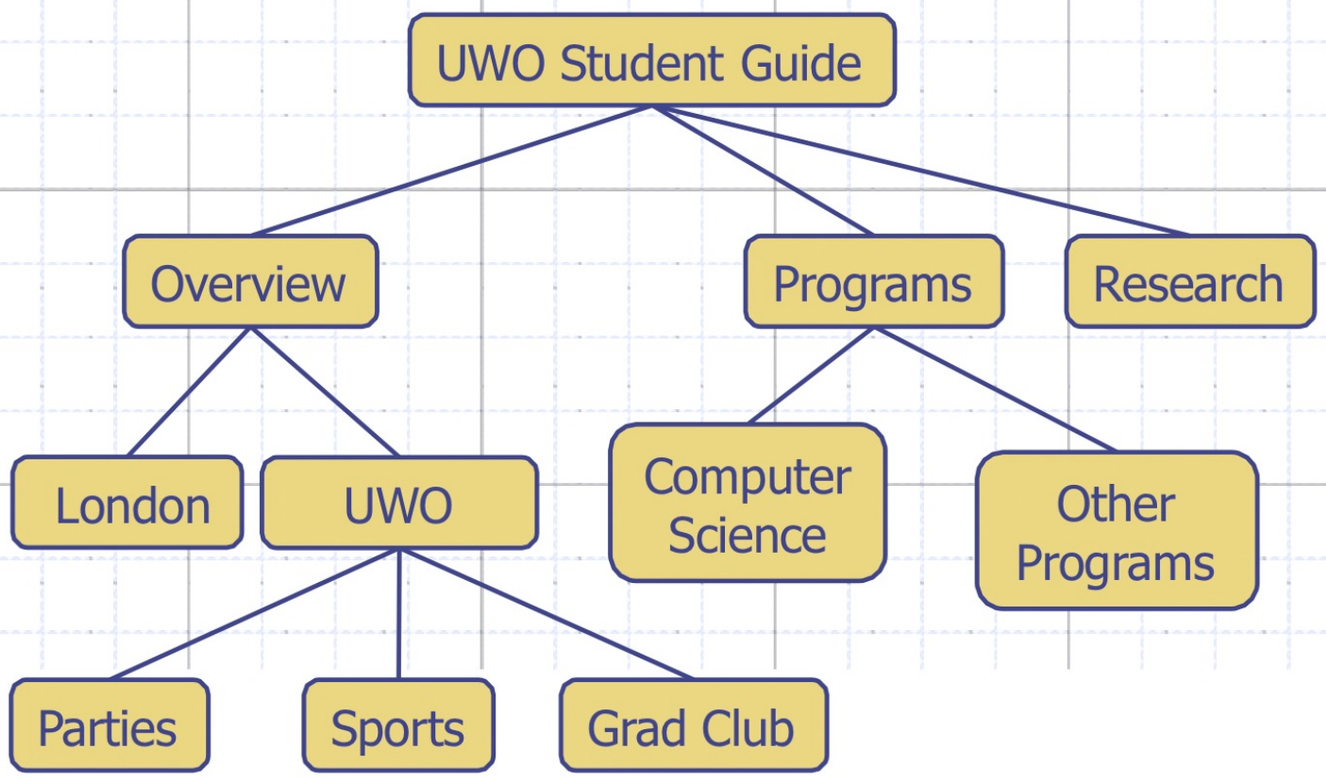
Applications

Organization of a company



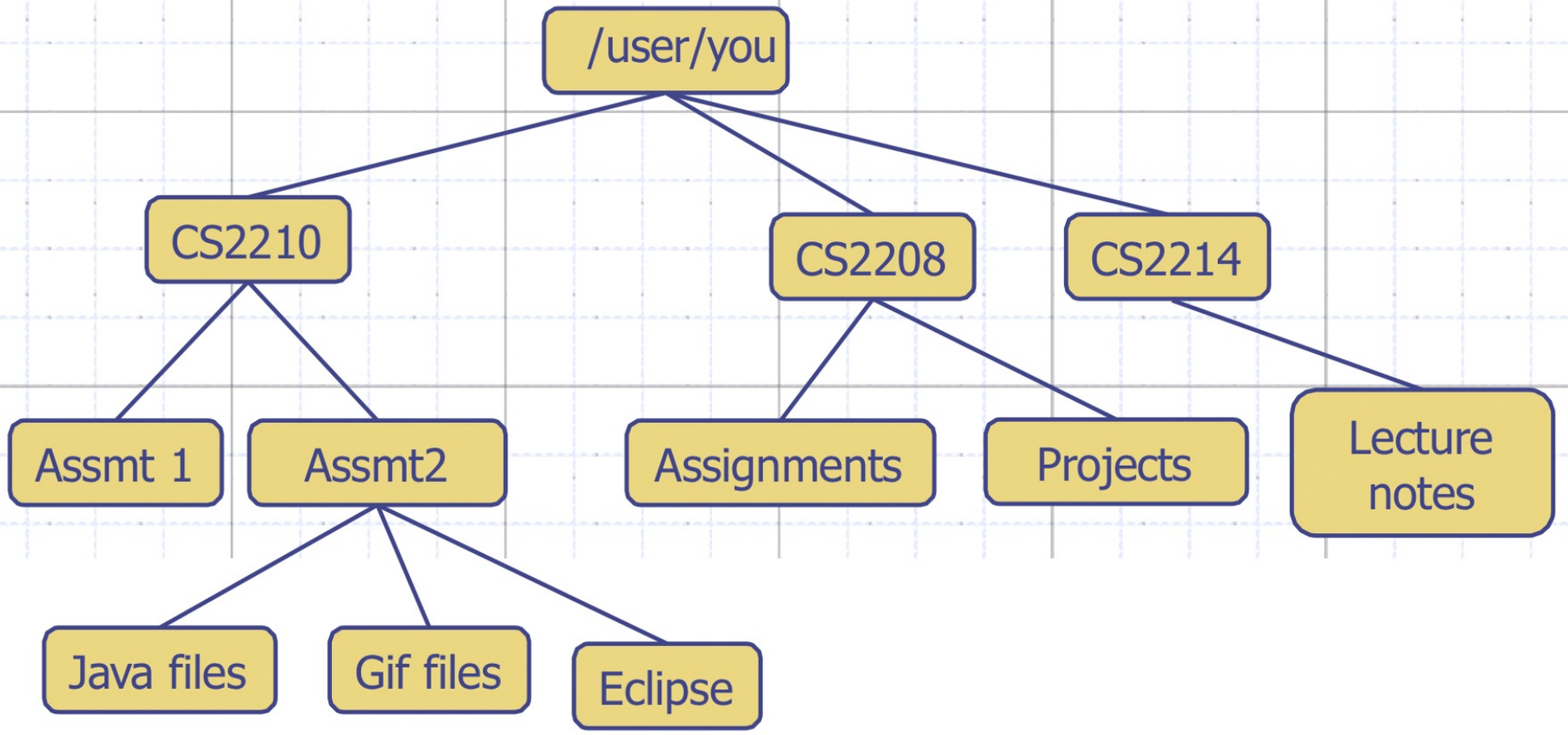
Applications

Table of contents of a book



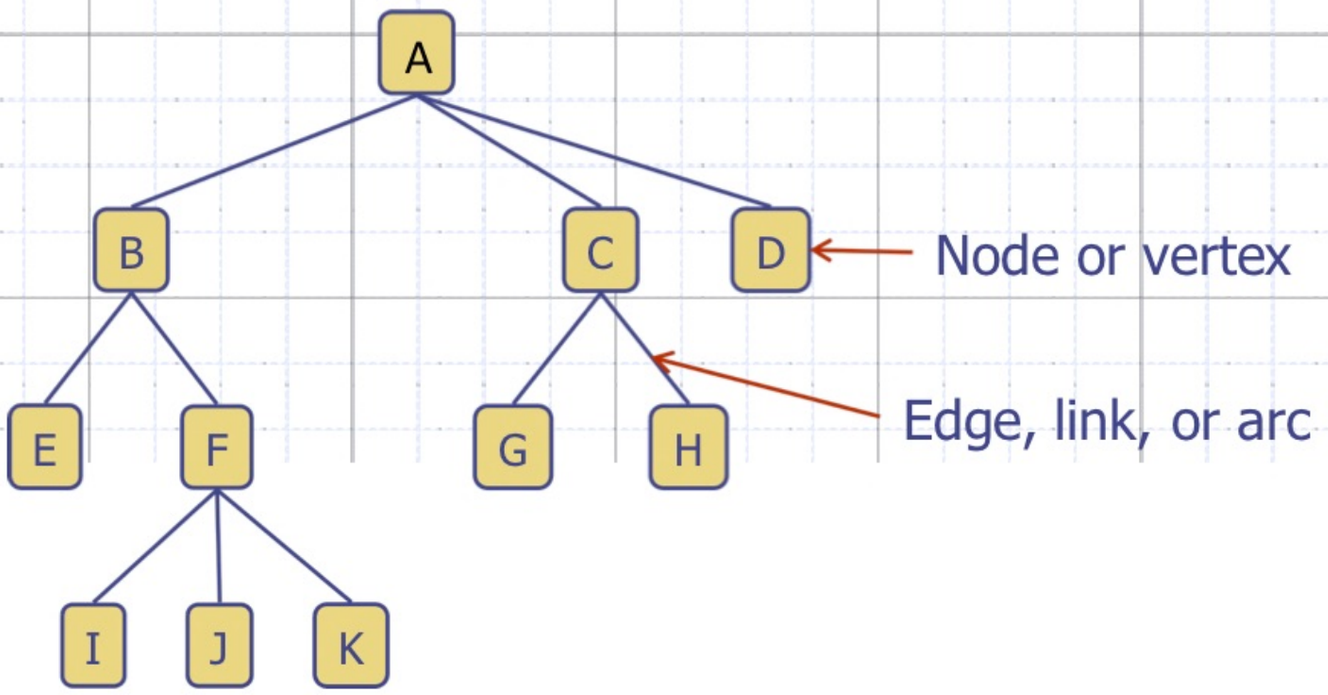
Applications

File system



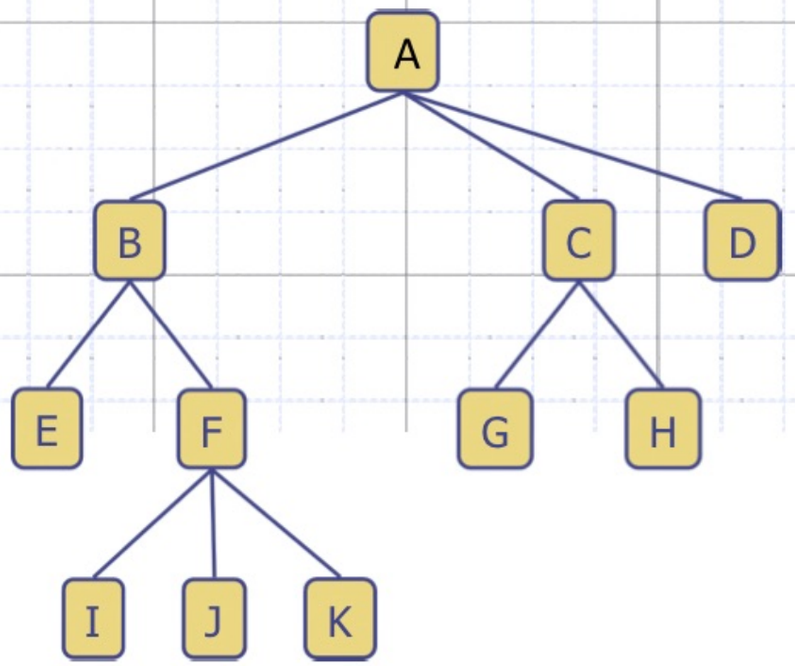
Tree Terminology

A is the root node
B is the parent of E and F
C is the sibling of B and D
B, C, and D are children of A



Tree Terminology

E, F, I, J, K are descendants of B
All nodes, except A, are descendants of A
A, B, F are ancestors of J
A has no ancestors



Tree Terminology

Internal node:

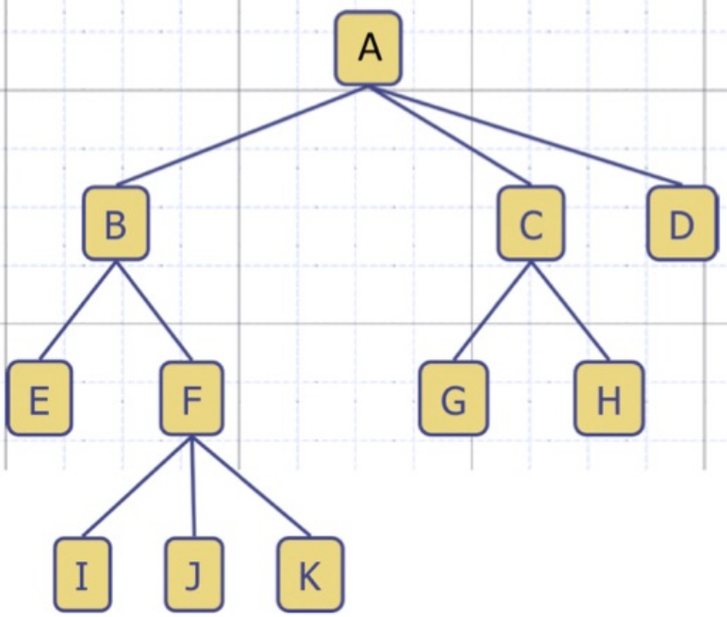
node with at least one child (A, B, C, F)

External node or leaf:

node without children (E, I, J, K, G, H, D)

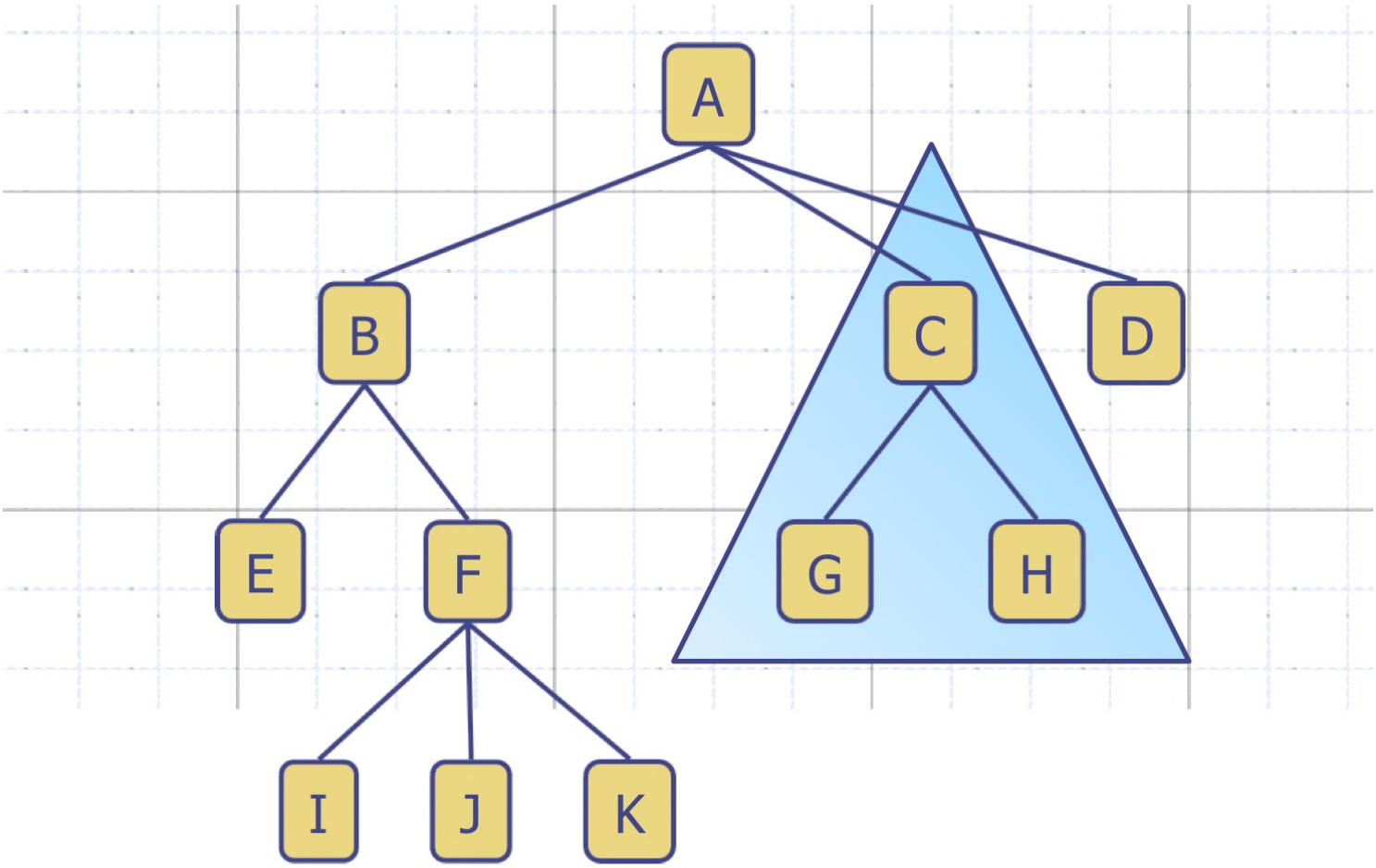
Degree of a node:

number of children. Degree of F is 3.



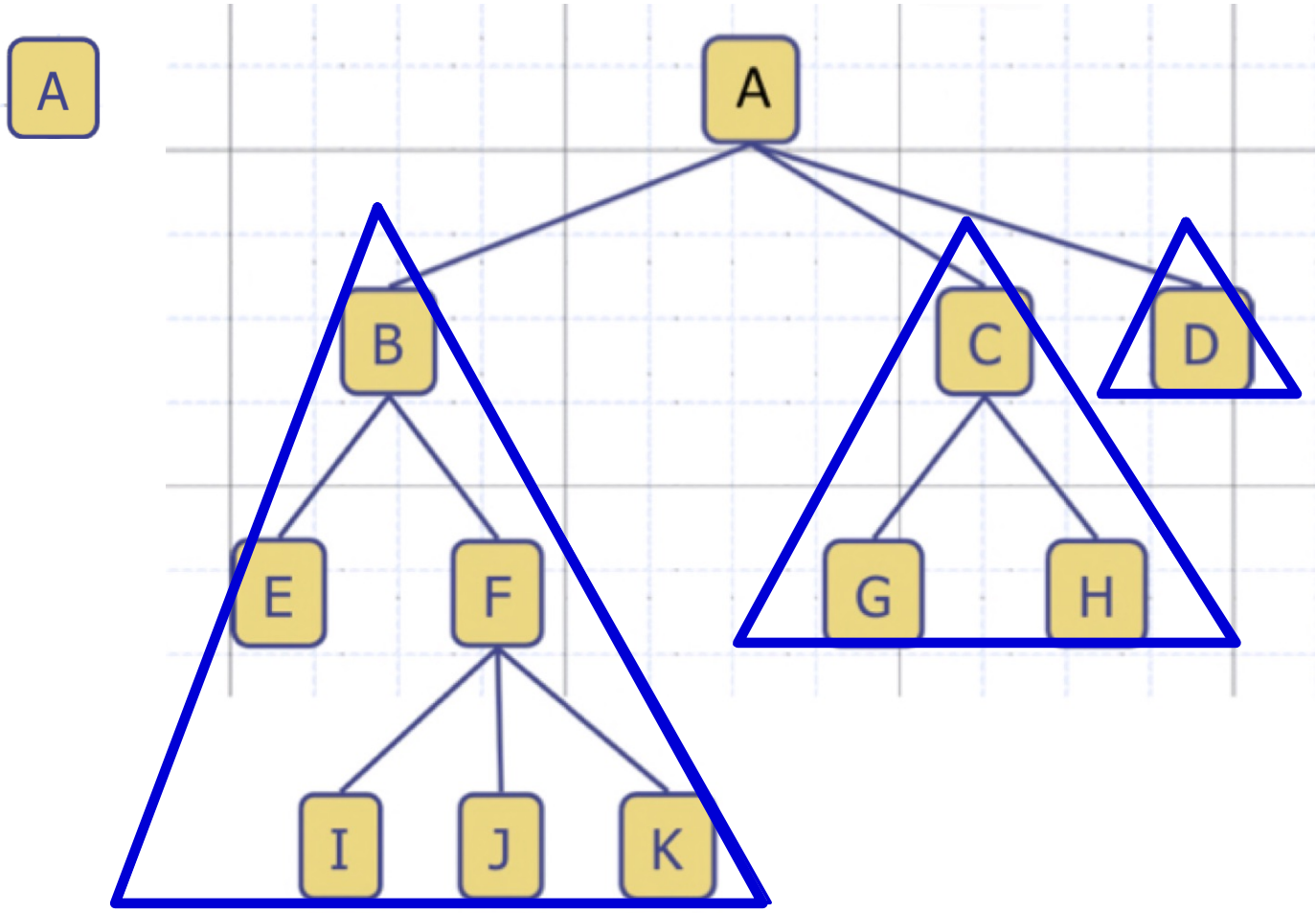
Tree Terminology

A subtree is a tree consisting of a node and all its descendants



Recursive Definition of a Tree

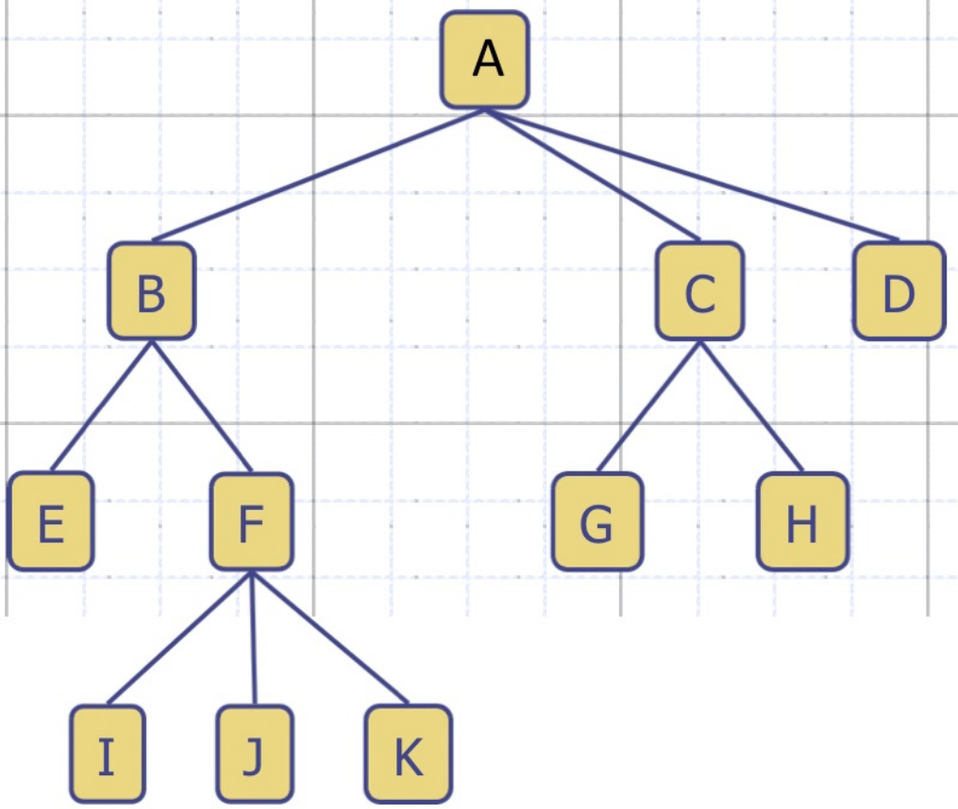
- A tree is a node
- A tree is a node whose children are the roots of subtrees



Since trees are recursive structures, recursive algorithms are efficient for manipulating them.

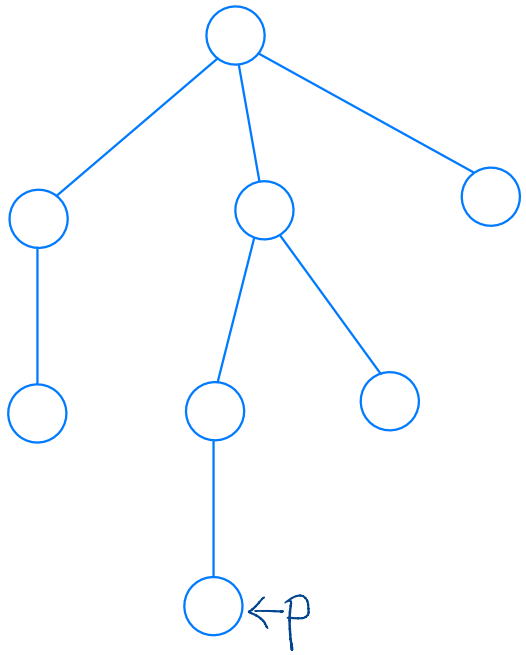
Tree Terminology

Depth or level of a node:
number of ancestors. Depth of E is 2.



Computing the Level of a Node

Definition of level:
level(p) = 0 if p is the root
level(p) = 1 + level(p.parent) if p is not the root



Algorithm level(p)
In: Node p of a tree
Out: level of p

```
if p.parent = null then return 0
else return 1 + level(p.parent)
```

Correctness of the algorithm follows from the recurrence equation defining the notion of level.

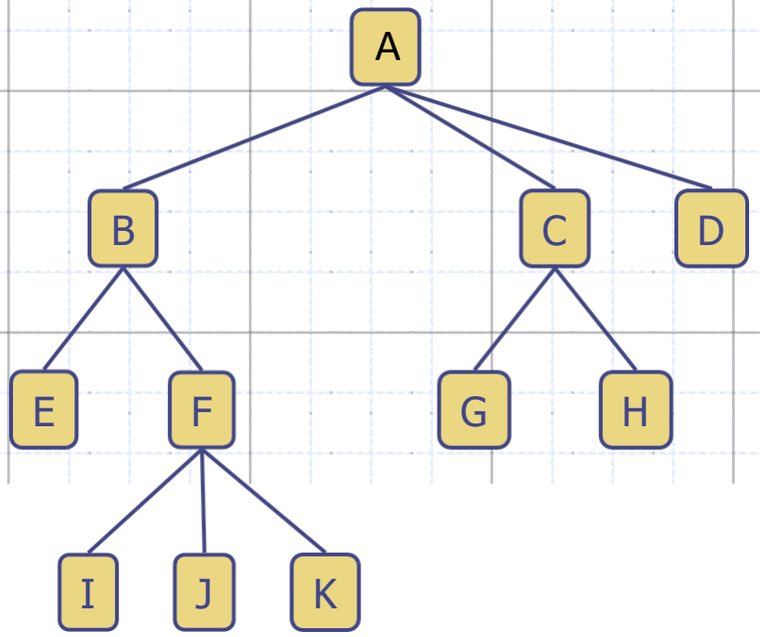
Tree Terminology

Depth or level of a node:

number of ancestors. Depth of E is 2.

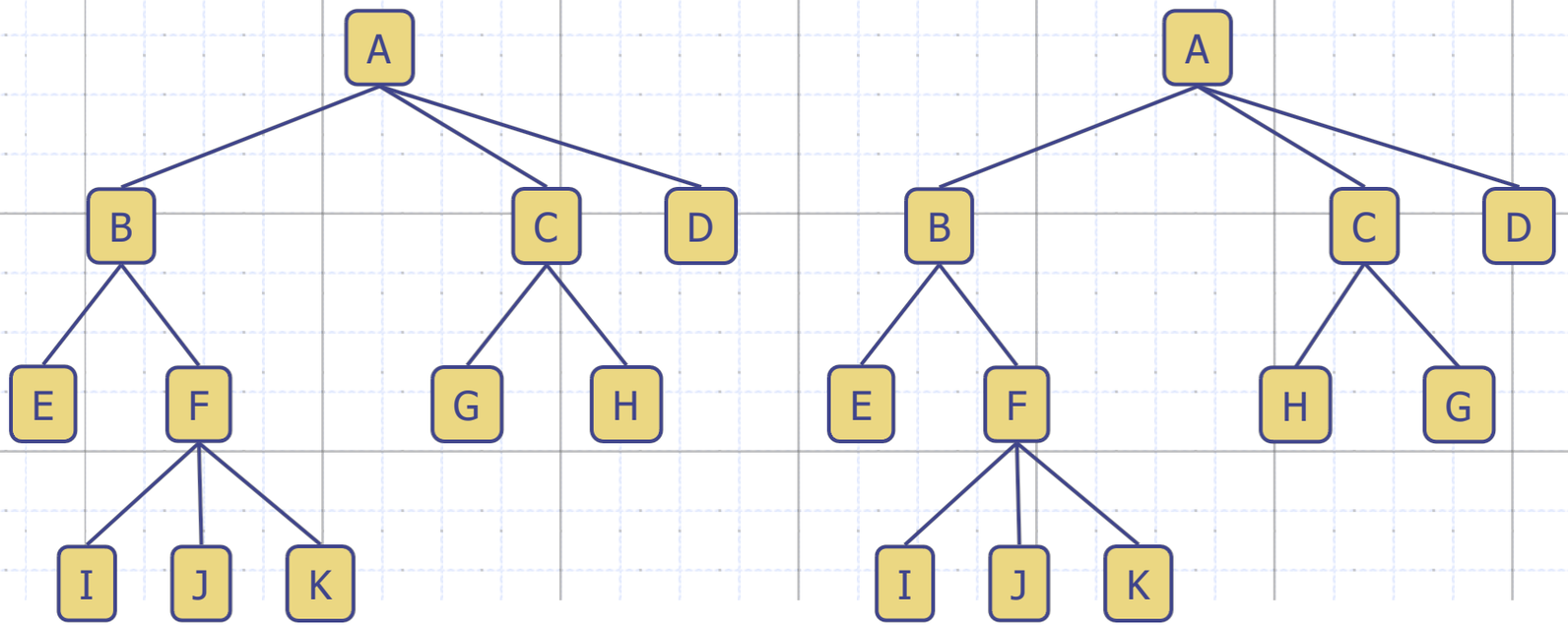
Height of tree

maximum depth of any node. Tree has height 3.



Tree Terminology

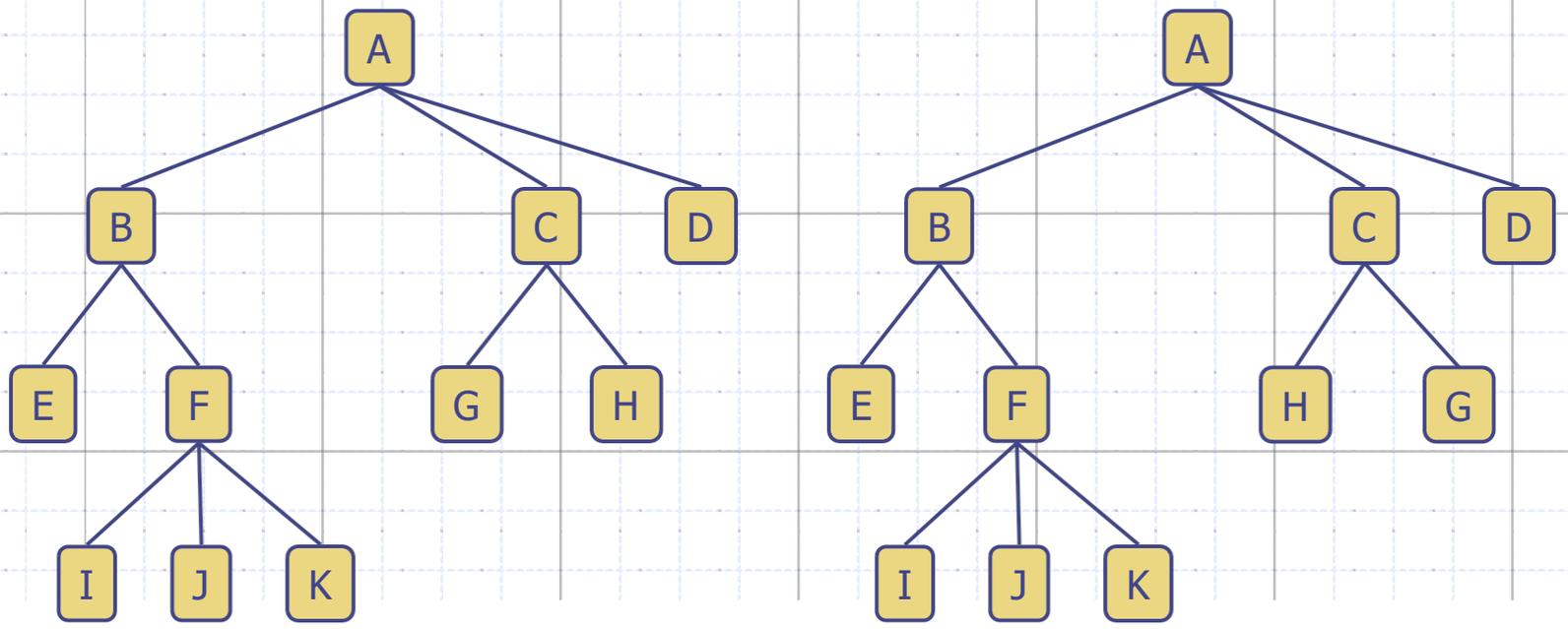
Ordered tree: The children of a node are ordered.



The above two orders trees are not identical.

Tree Terminology

Un-Ordered tree: The children of a node are not ordered.

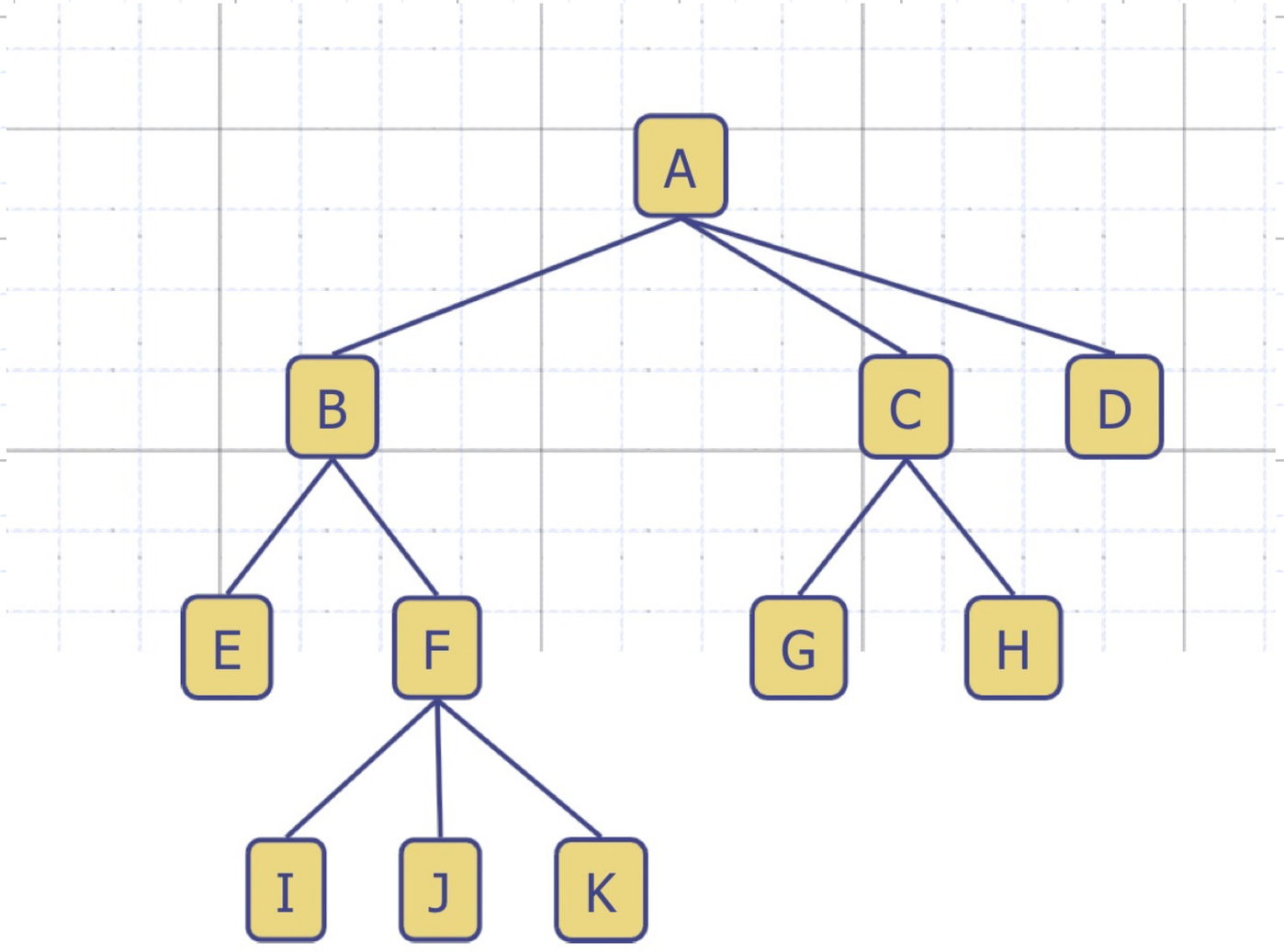


The two above unordered trees are identical.

Tree Properties

Number of nodes = Number of edges + 1

Proof. Glue every node to the edge connecting it to its parent. The root is not glued to any edges.

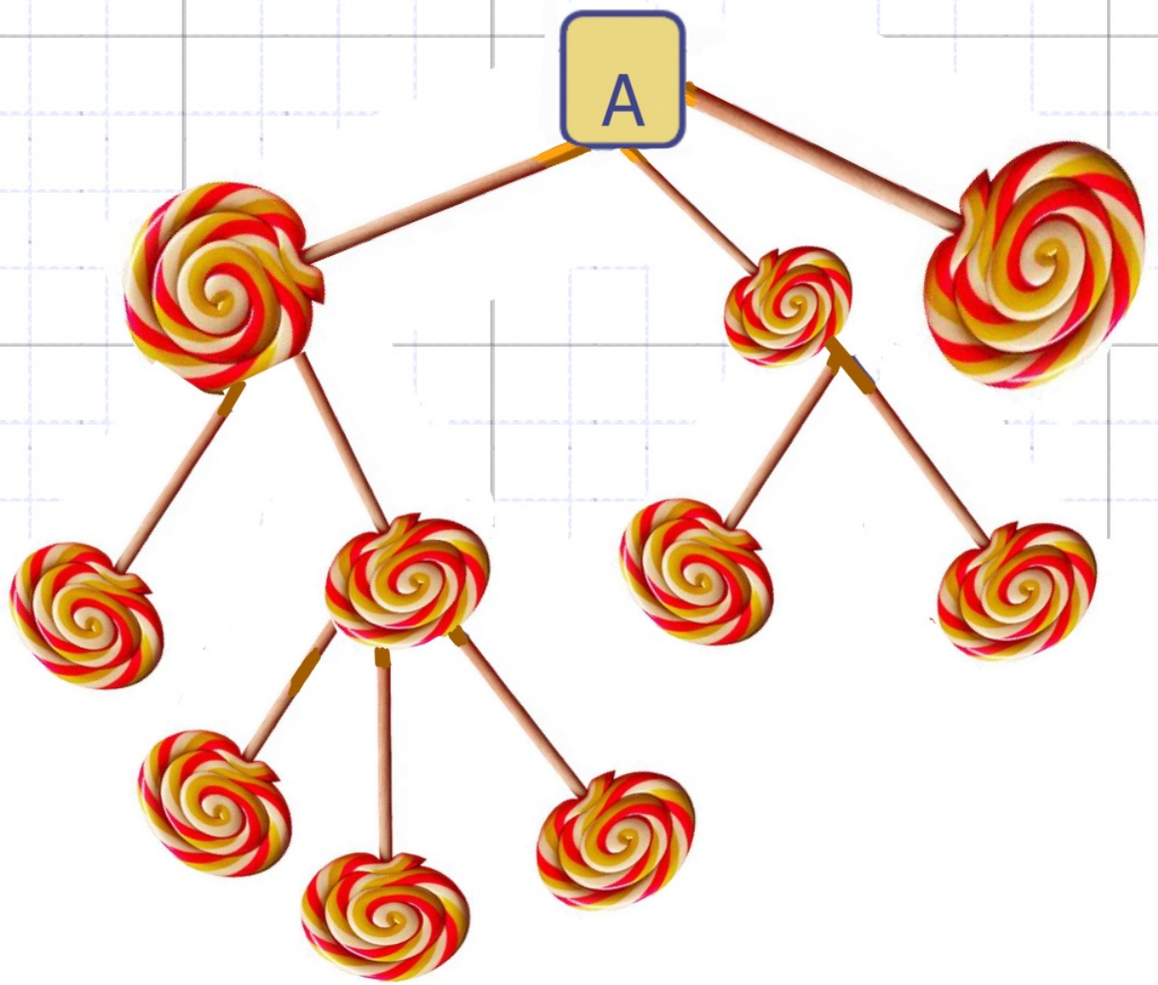


Tree Properties

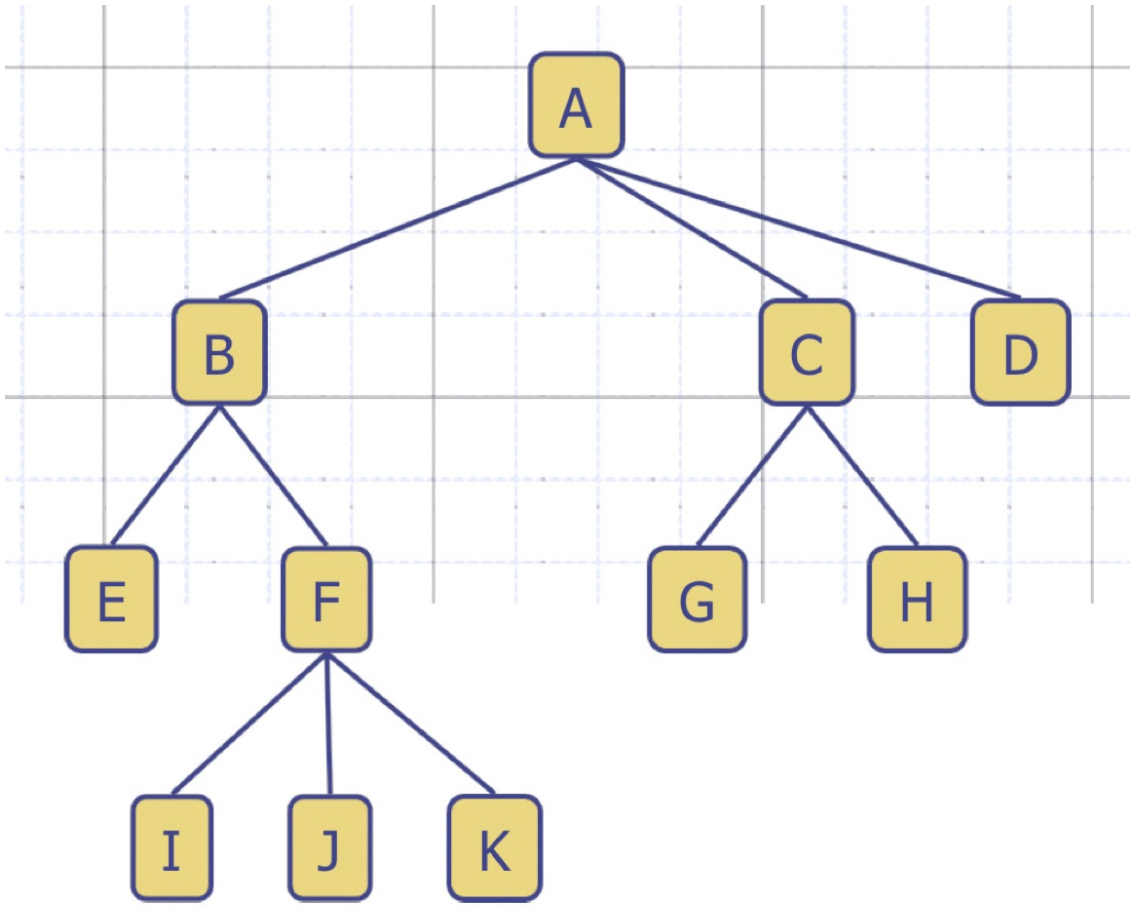
n = Number of nodes

Number of nodes = Number of edges + 1

Proof. Glue every node to the edge connecting it to its parent. The root is not glued to any edges.



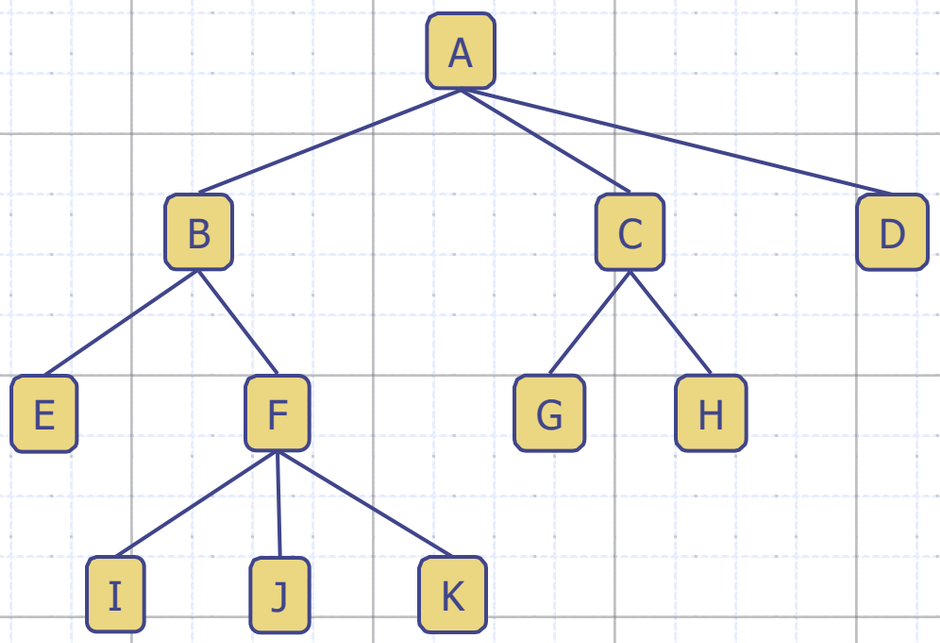
Tree Properties



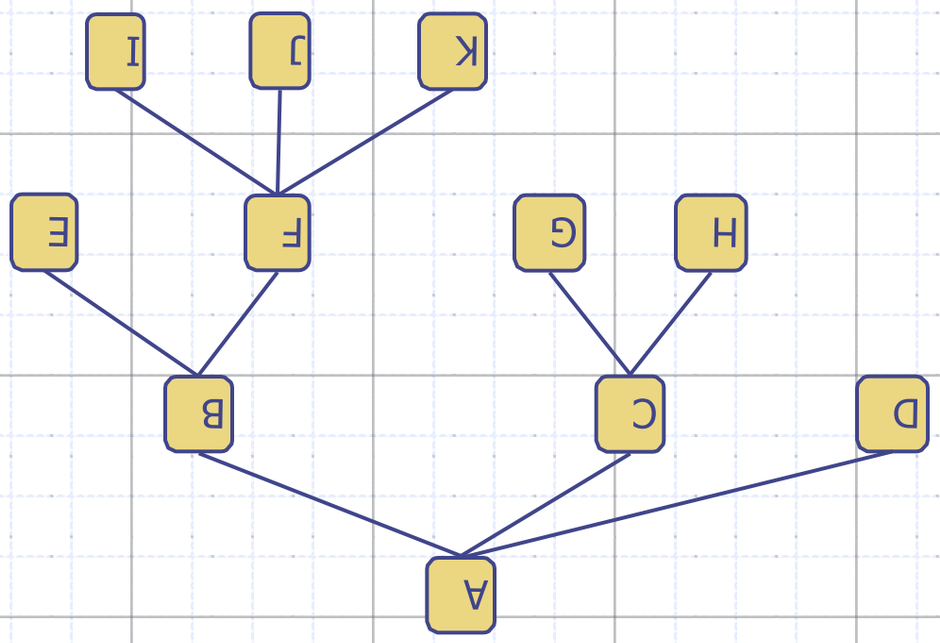
$$\sum_{\substack{\text{Nodes} \\ u}} \text{degree}(u) = \#\text{edges} = n - 1$$

n = number of nodes

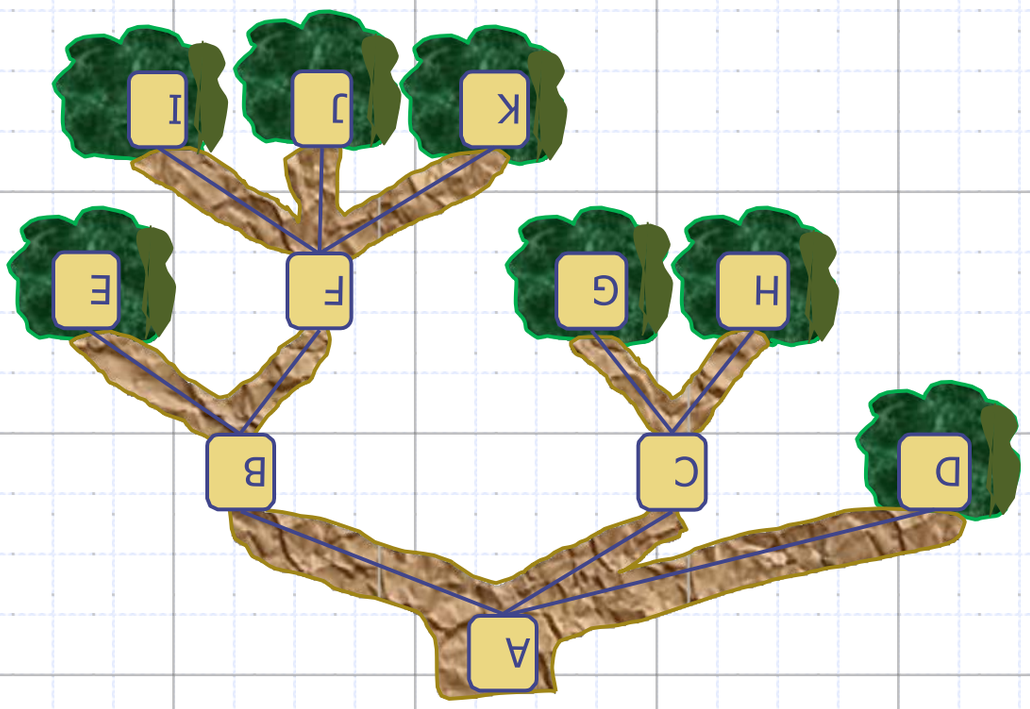
Why is this a Tree?



Why is this a Tree?



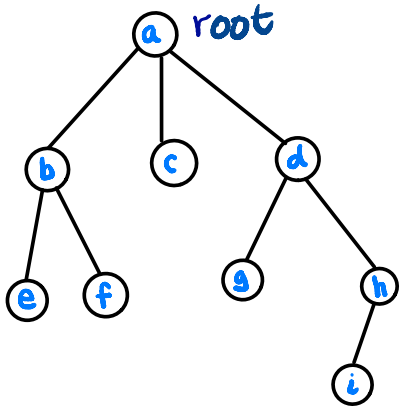
Why is this a Tree?



Computing the Height of a Tree

Recursive definition of the height of a tree with root r:

height(r) = 0 if r is a leaf
height(r) = 1 + max {height(c) | c is a child of r} if r is not a leaf

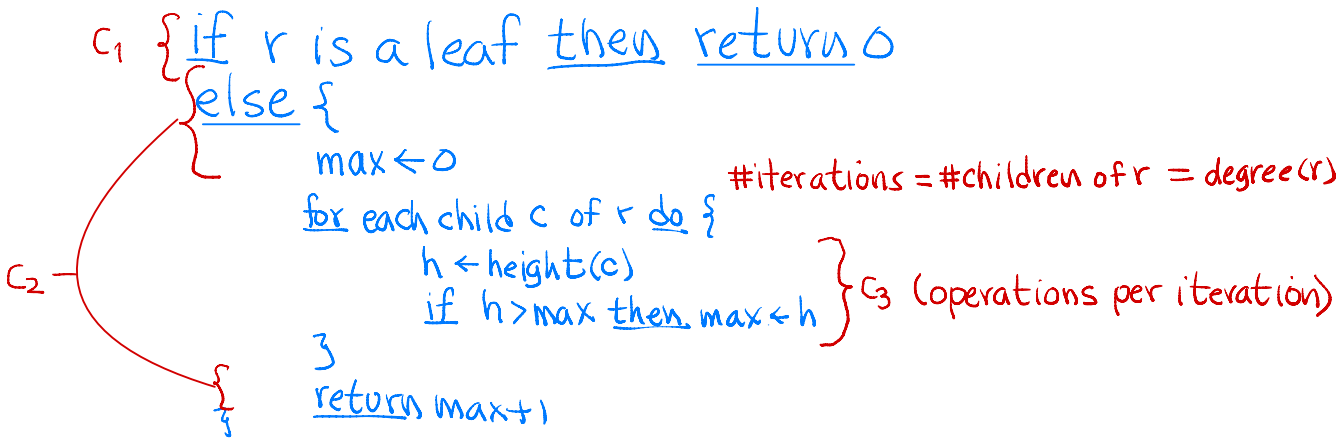


Algorithm height(r)
In: root r of a tree
Out: height of the tree

```
if r is a leaf then return 0
else {
    max ← 0
    for each child c of r do {
        h ← height(c)
        if h > max then max ← h
    }
    return max+1
}
```

Time Complexity of Algorithm height

Algorithm height(r)
In: root r of a tree
Out: height of the tree



Since the algorithm has loops and recursive calls, to simplify the analysis we split it into 3 parts:

1. Compute the complexity ignoring recursive calls (complexity of one call)
   - c_1 if r is a leaf
   - $c_2 + c_3 \times \text{degree}(r)$ if r is not a leaf

   c_1, c_2, and c_3 are the constant number of operations performed in the parts of the algorithm marked above

2. Compute the number of recursive calls

   One call per node (so the total is n)

3. Combine (1) and (2) to have the total number of operations

$$f(n) = \sum_{\text{leaves}} c_1 + \sum_{\substack{\text{internal} \\ \text{nodes } u}} (c_2 + c_3 \times \text{degree}(u)) = c_1 \times \#\text{leaves} + \sum_{\substack{\text{internal} \\ \text{nodes } u}} c_2 + c_3 \underbrace{\sum_{\substack{\text{internal} \\ \text{nodes } u}} \text{degree}(u)}_{n-1}$$

$$= \underbrace{c_1 \times \#\text{leaves} + c_2 \times \#\text{internal}}_{n} + c_3(n-1)$$

is $O(n)$