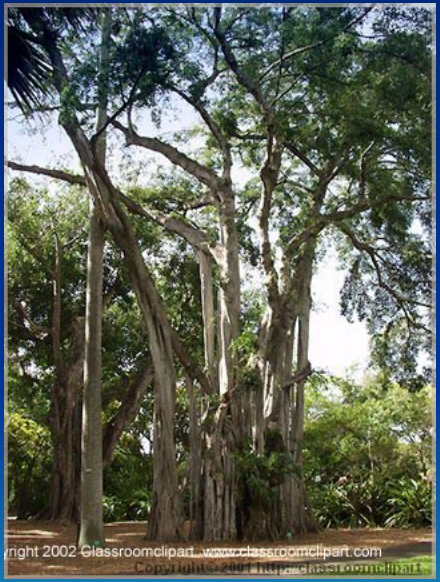
Trees



Free Trees

Free Tree/Tree (for short)

A connected, acyclic (no cycle), undirected graph.

Forest

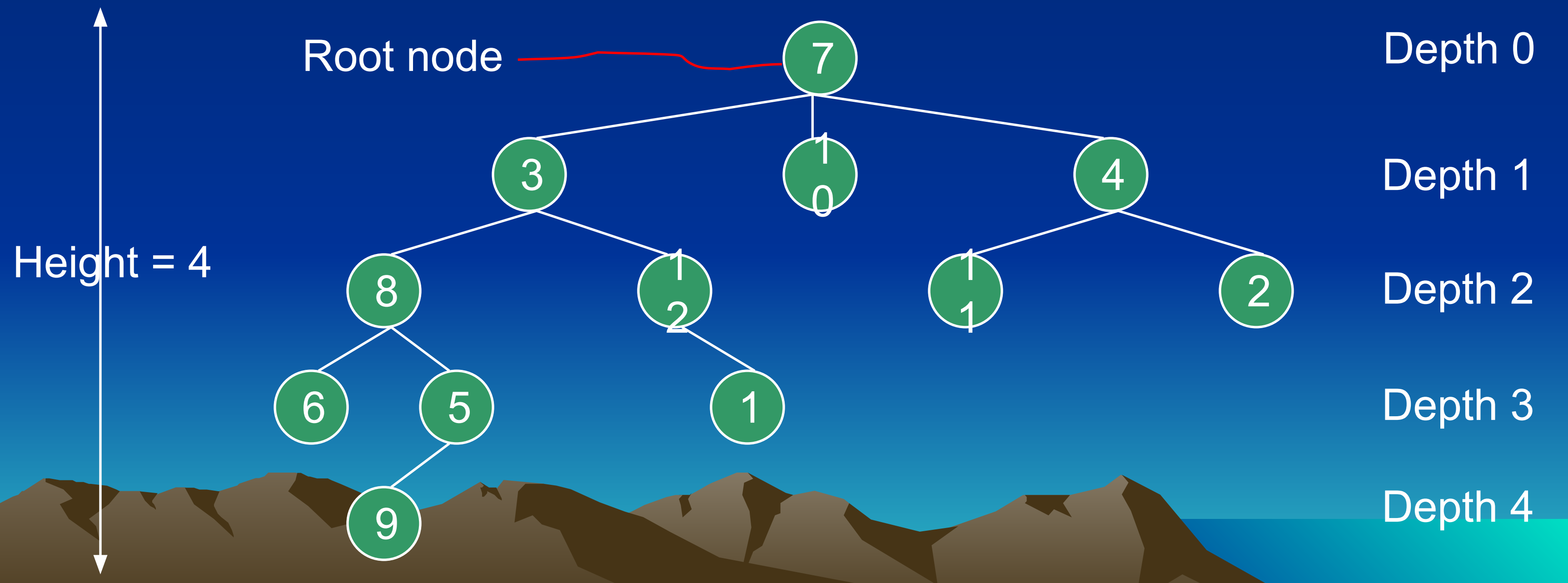
An undirected graph that is acyclic but possibly disconnected.

Free Trees (cont'd.)

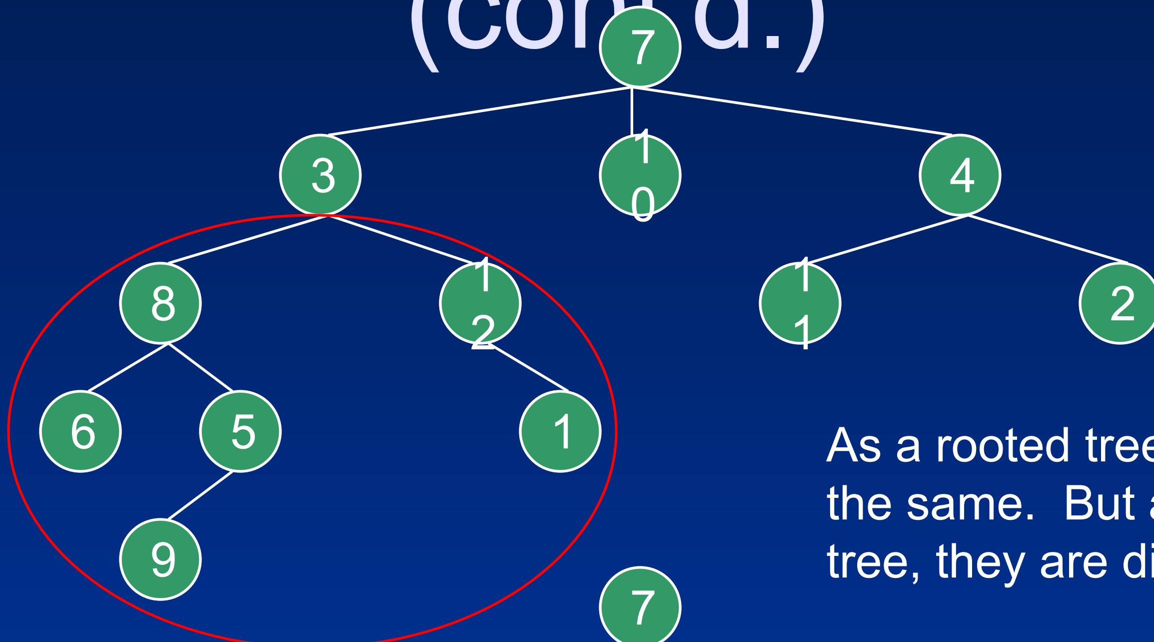
A tree is, therefore, a forest, but a forest is not a tree because it is not connected.

Rooted and Ordered Trees

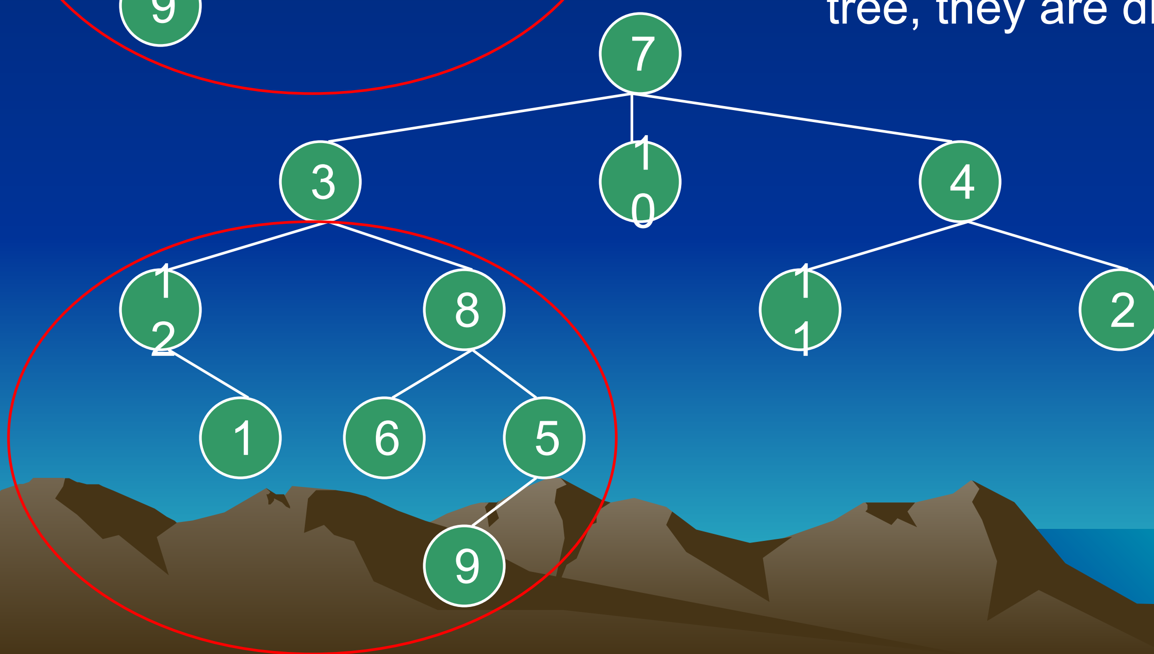
A free tree in which one of the vertices/nodes is distinguished from the others. The distinguished vertex is called the root of the tree.



Rooted and Ordered Trees (cont'd.)



As a rooted tree, both are the same. But as an ordered tree, they are different!



Rooted and Ordered Trees (cont'd.)

A finite set of one or more nodes such that

a. there is a specially designated node called root; and

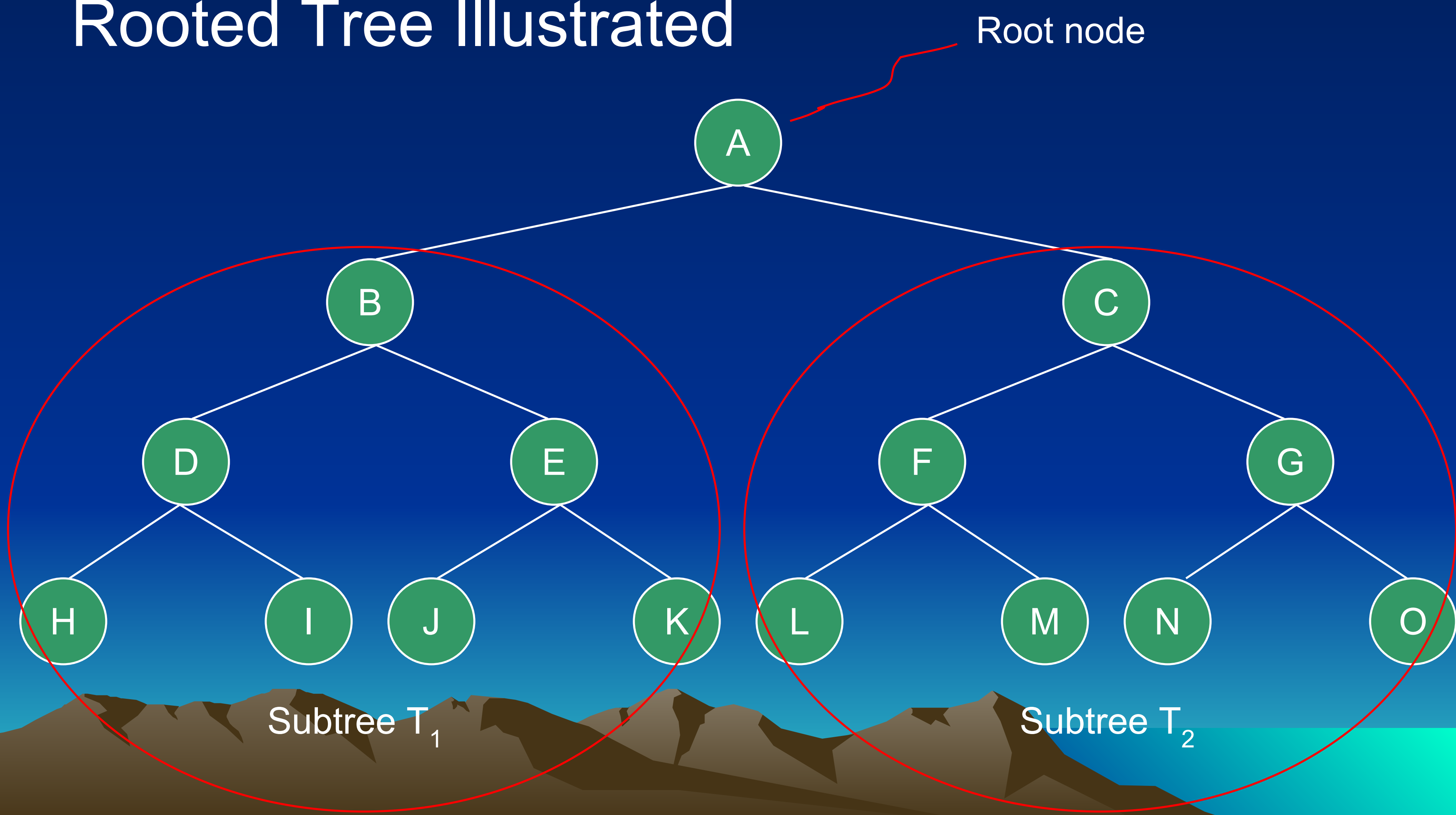
b. the remaining nodes are partitioned into $n \geq 0$ disjoint sets $T_1, T_2, \ldots, T_n$ where each set is a tree called subtrees of the root.

Rooted and Ordered Trees (cont'd.)

That means, your tree must contain at least 1 node (i.e. the root node). A tree can have no subtrees at all.

Rooted and Ordered Trees (cont'd.)

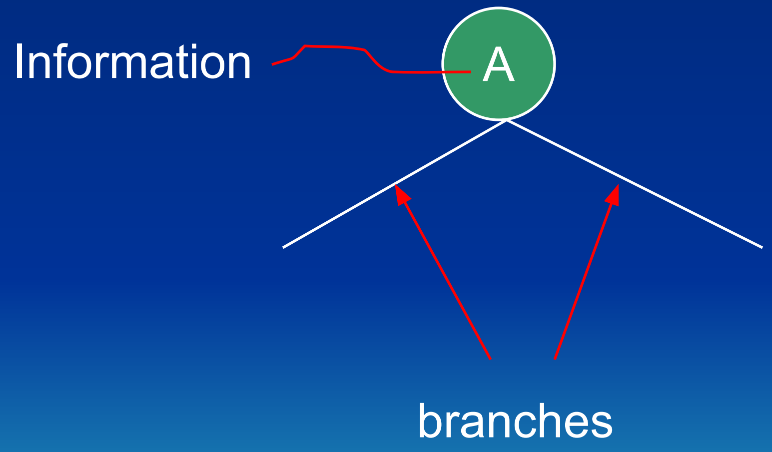
Rooted Tree Illustrated



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

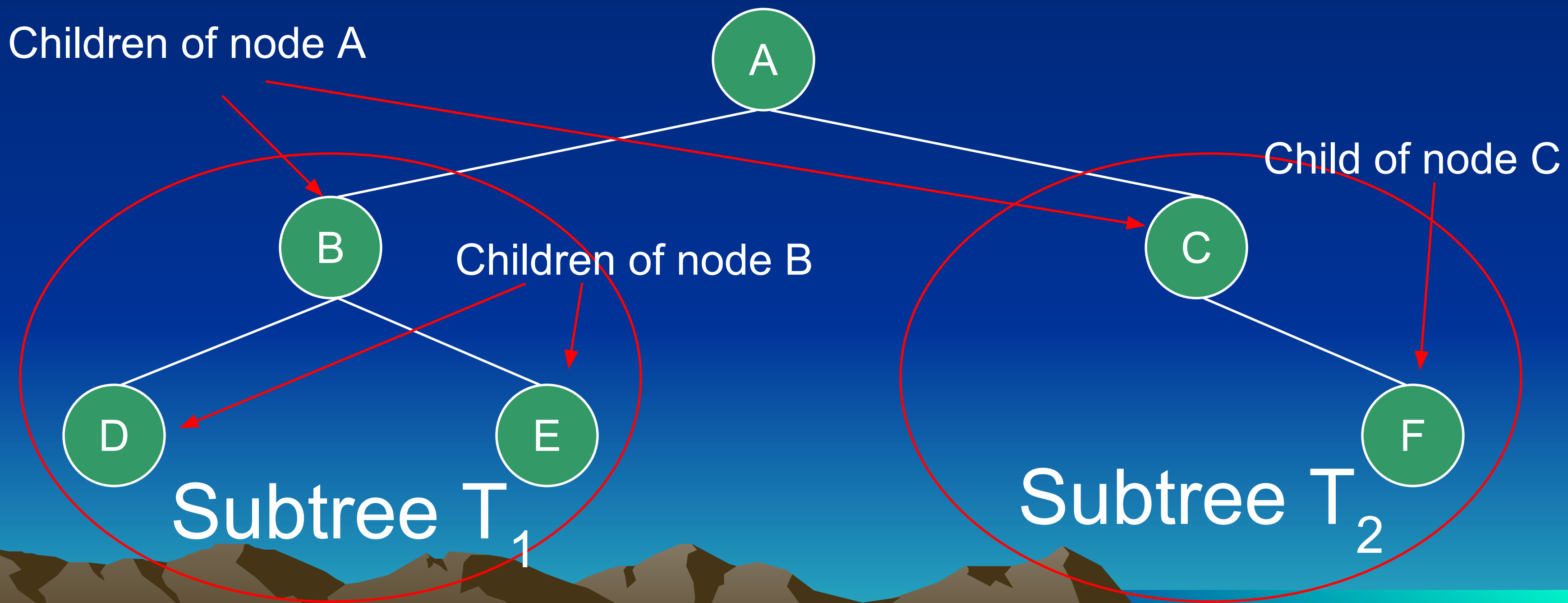
Vertex/Node – item of information plus the branches to other items



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

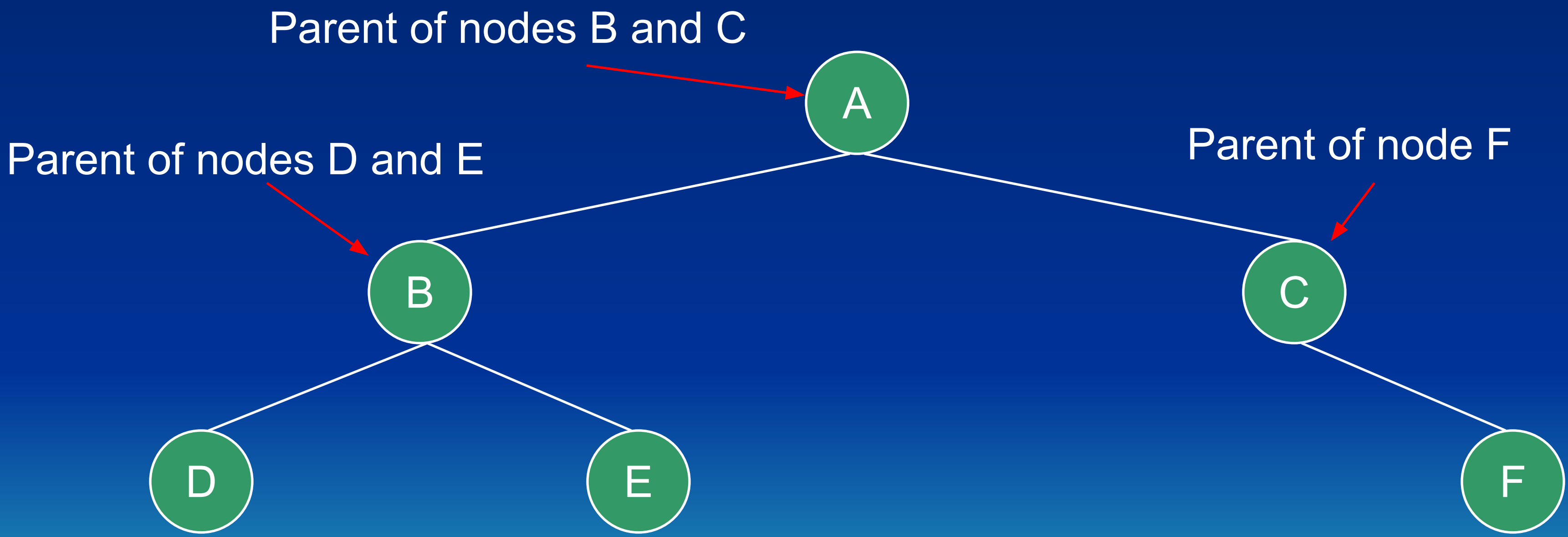
Children of a node – roots of the subtrees of a node



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

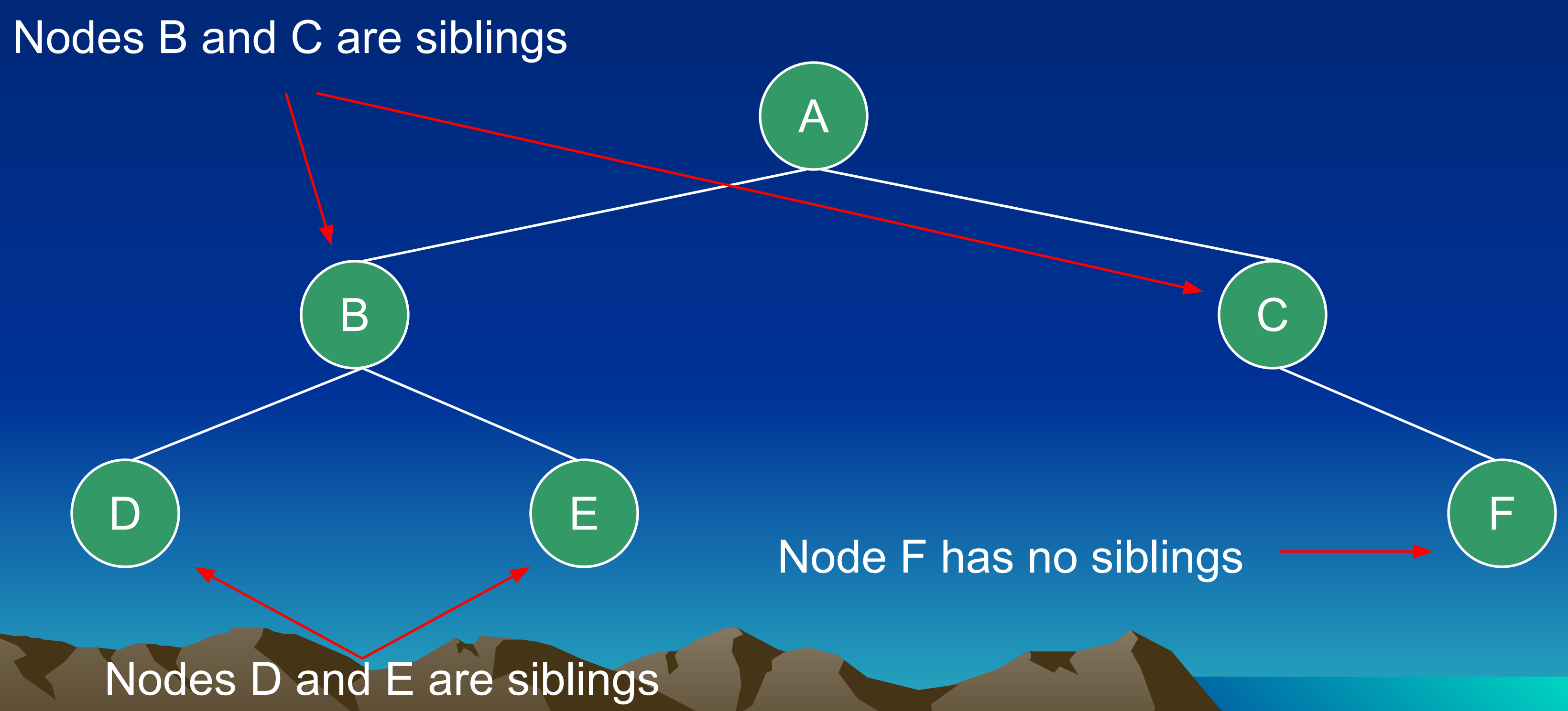
Parent of a node – immediate root of a node



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

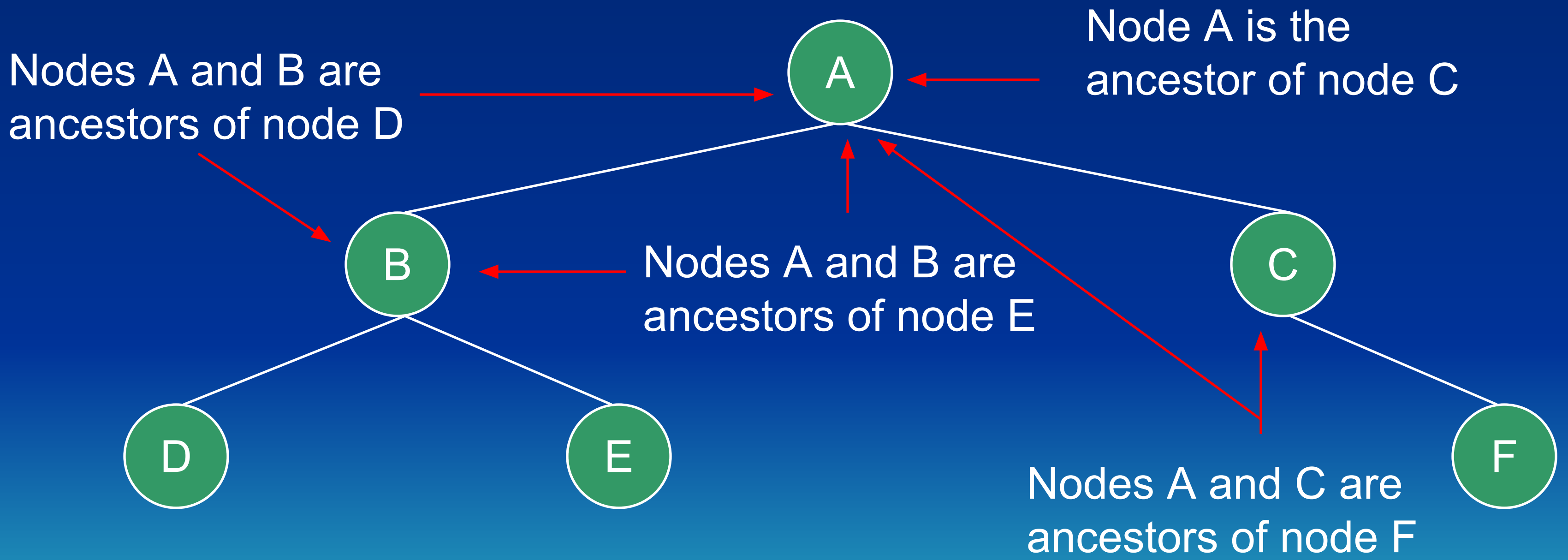
Siblings – children of the same parent



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

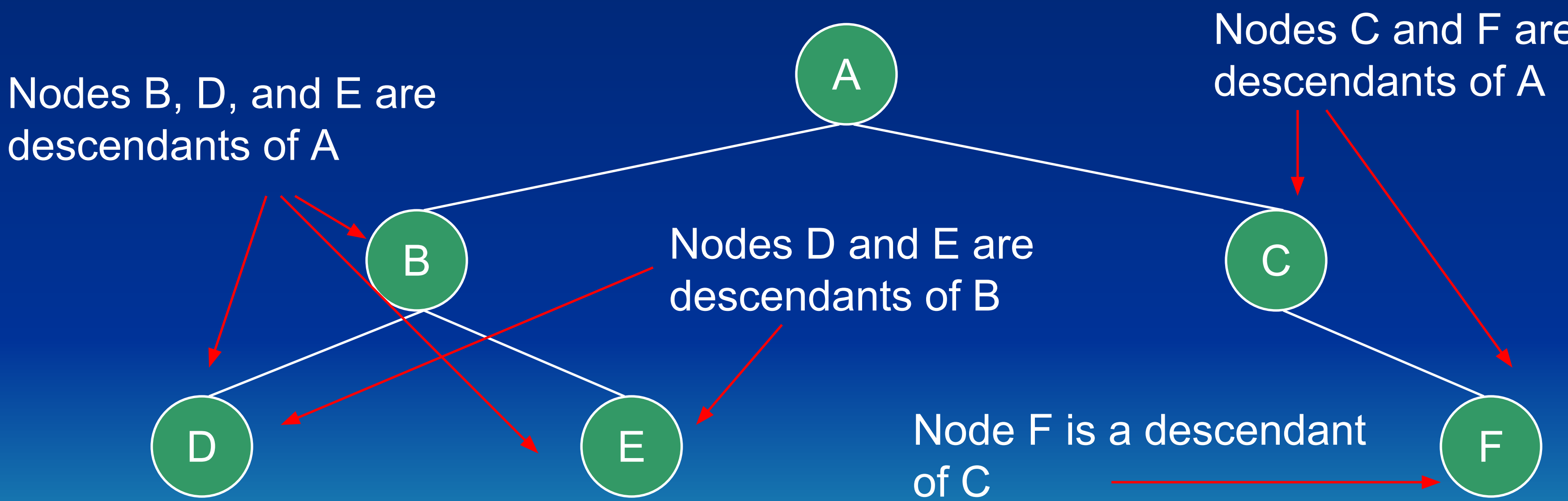
Ancestors of a node – all the nodes along the path from the root to that node



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

Descendants of a node – all the nodes of the subtrees of a node



Rooted and Ordered Trees (cont'd.)

Note

Every node is both an ancestor and a descendant of itself.

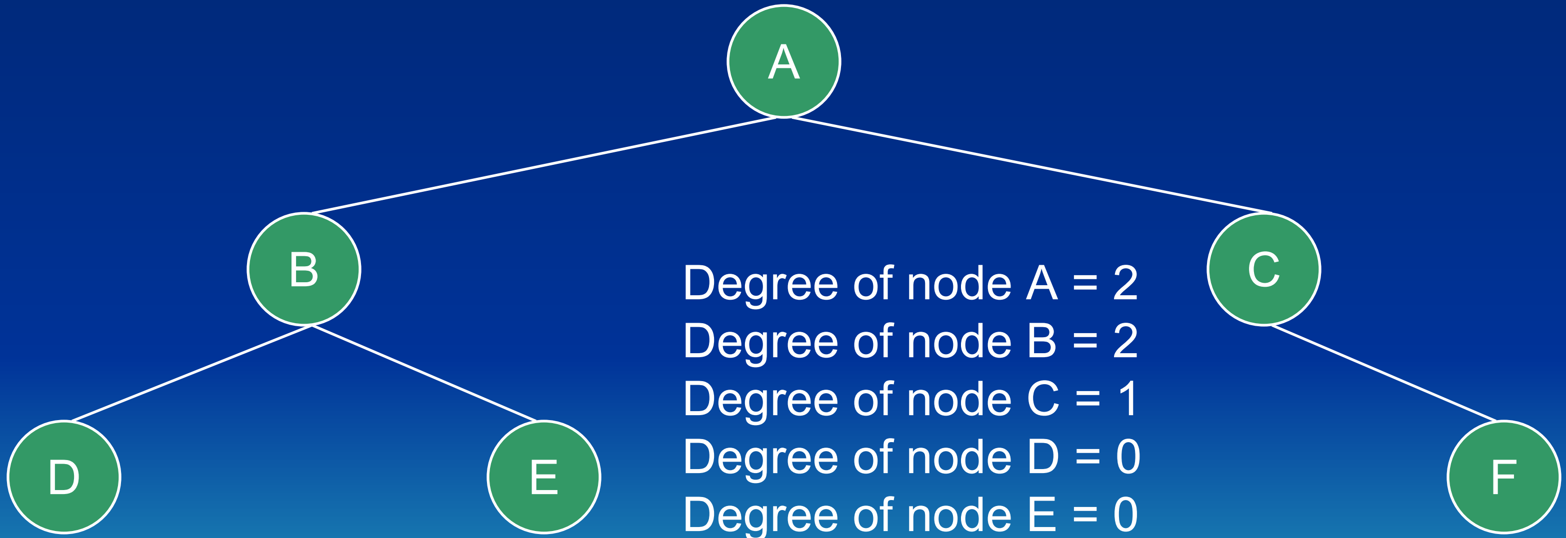
If y is an ancestor of x and $x \neq y$, then y is a proper ancestor of x and x is a proper descendant of y.

The subtree rooted at x is the tree induced by descendants of x, rooted at x.

Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

Degree of a node – number of children/subtrees of a node



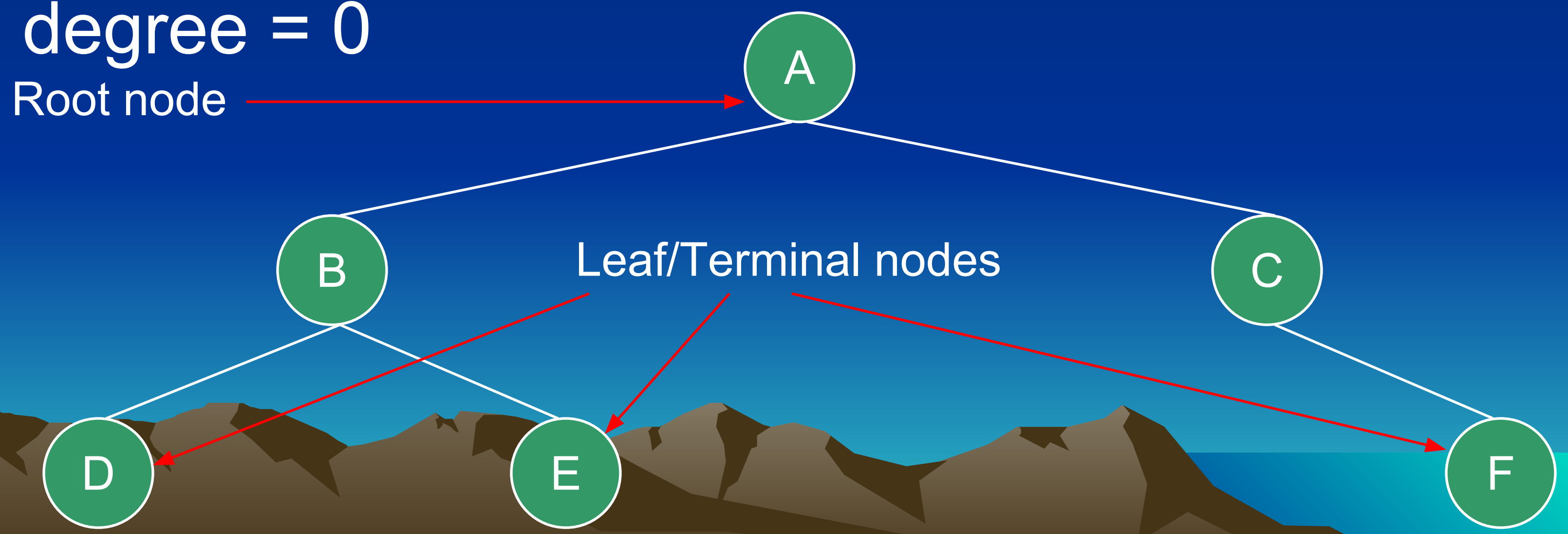
Degree of node A = 2
Degree of node B = 2
Degree of node C = 1
Degree of node D = 0
Degree of node E = 0

Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

Root node – the only node in a tree with no parent or proper ancestors

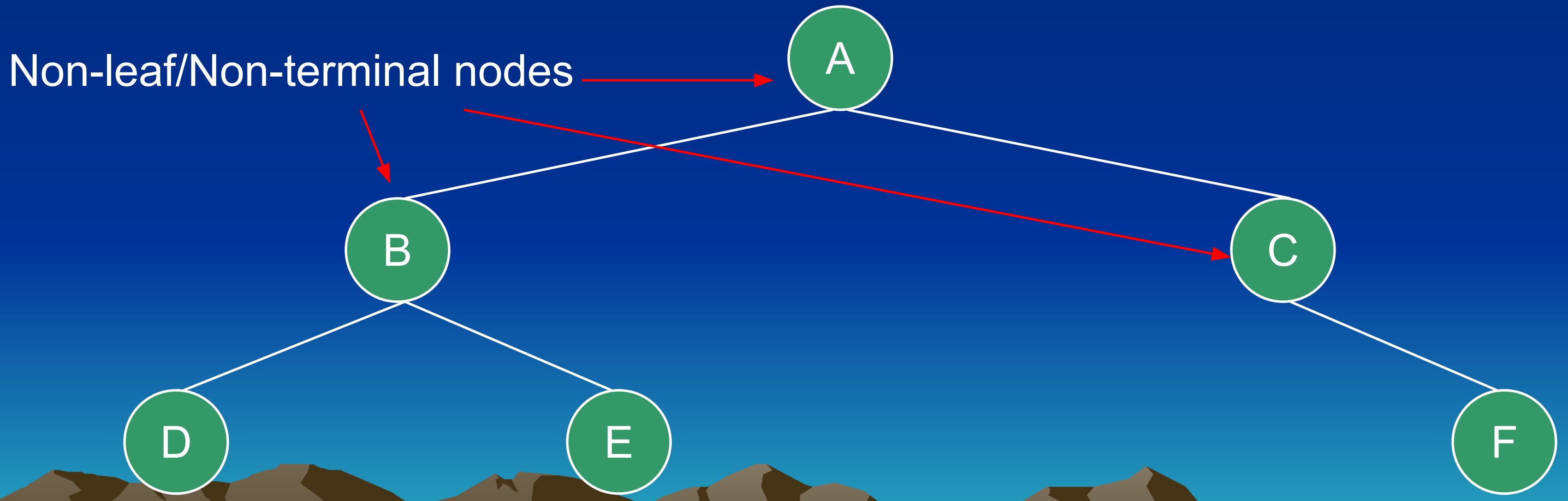
Leaf/Terminal/External node – node with no children or proper descendants node with degree = 0



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

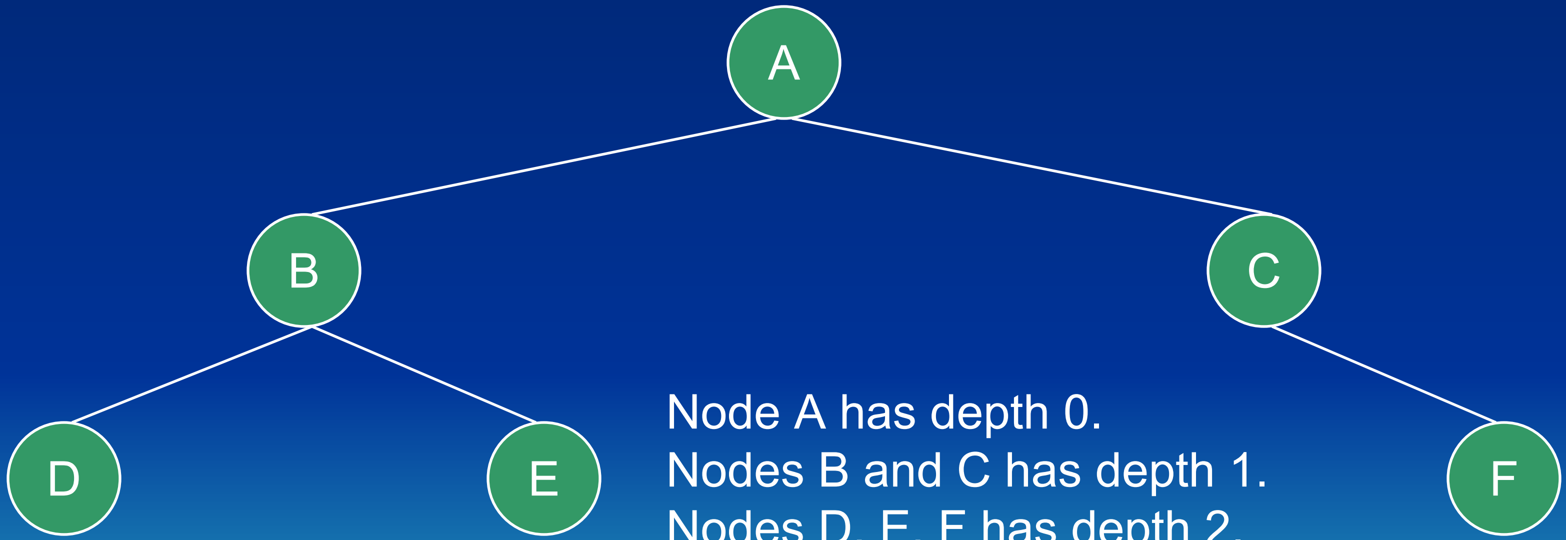
Non-leaf/Non-terminal/Internal node – has at least one child or descendant node with degree not equal to 0



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

Depth of a node – the length of the path from the root to that node

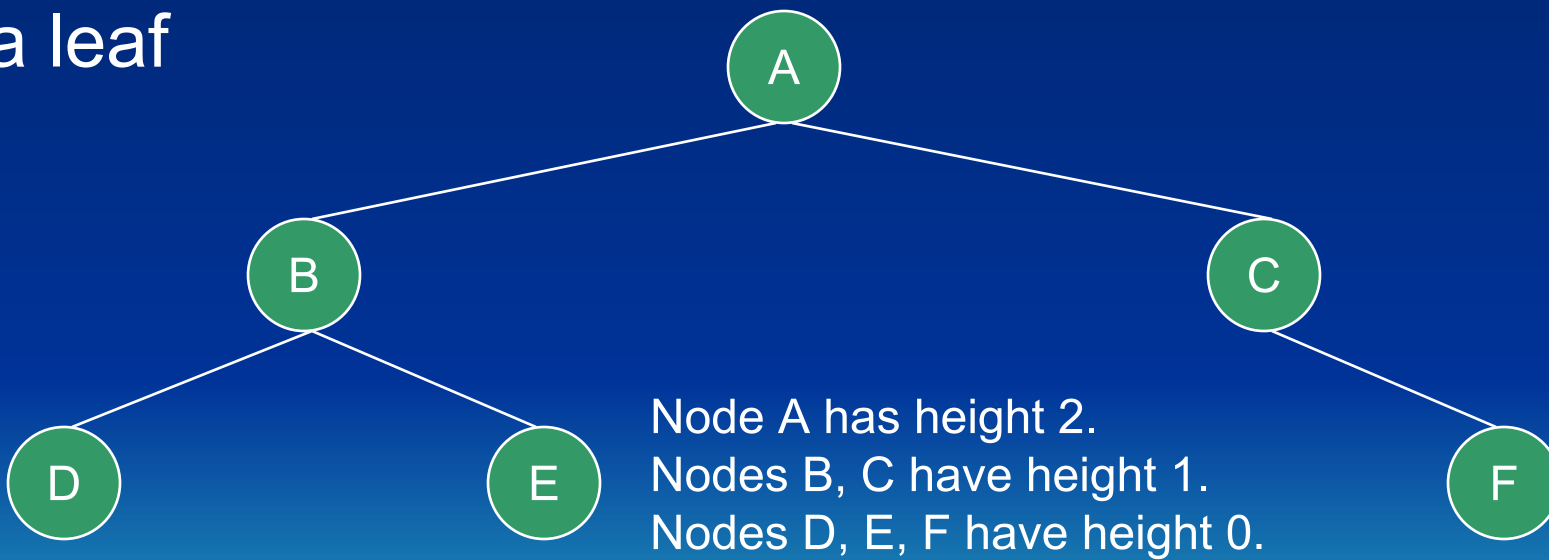


Node A has depth 0.
Nodes B and C has depth 1.
Nodes D, E, F has depth 2.

Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

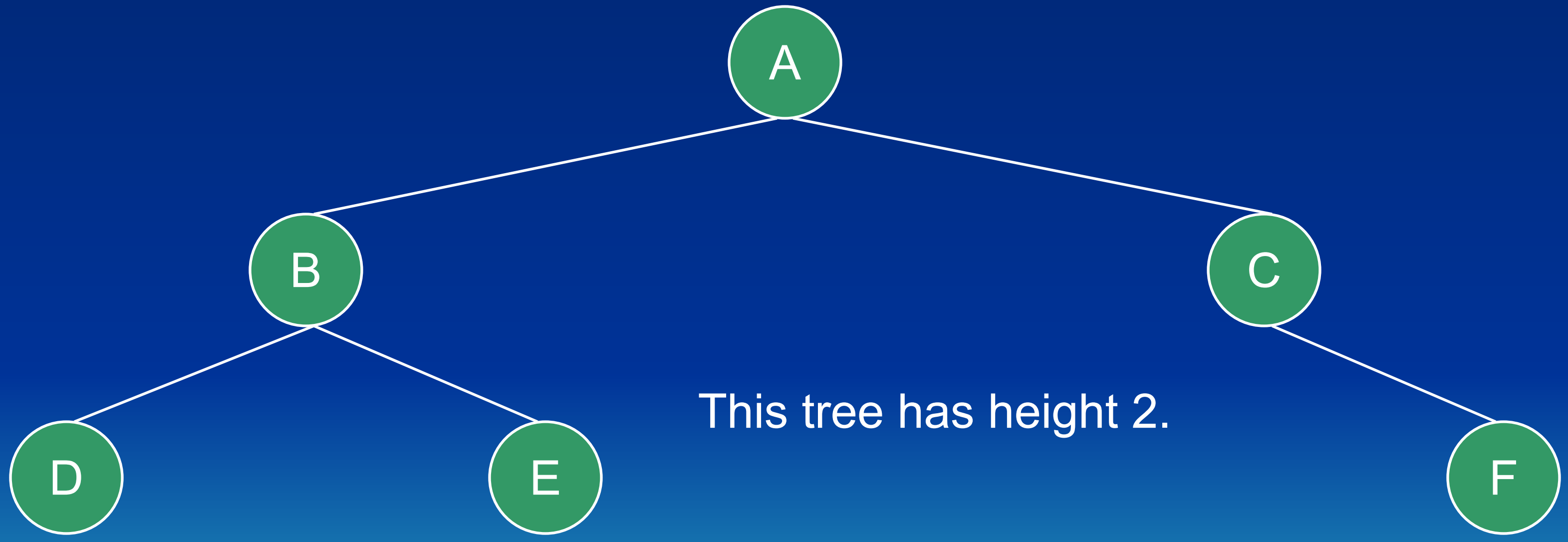
Height of a node – the number of edges on the longest simple downward path from the node to a leaf



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

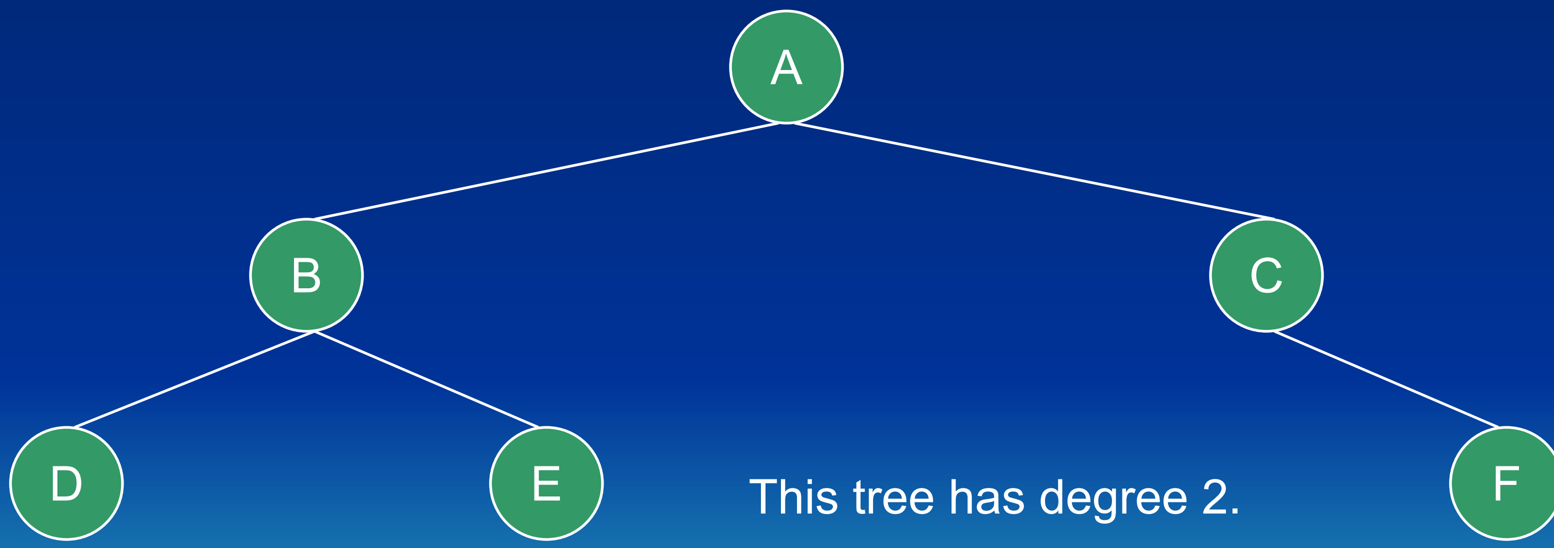
Height of a tree – the height of the root; equal to the largest depth of any node in the tree



Rooted and Ordered Trees (cont’d.)

Concepts and Terminologies

Degree of a tree – max(degree of nodes)

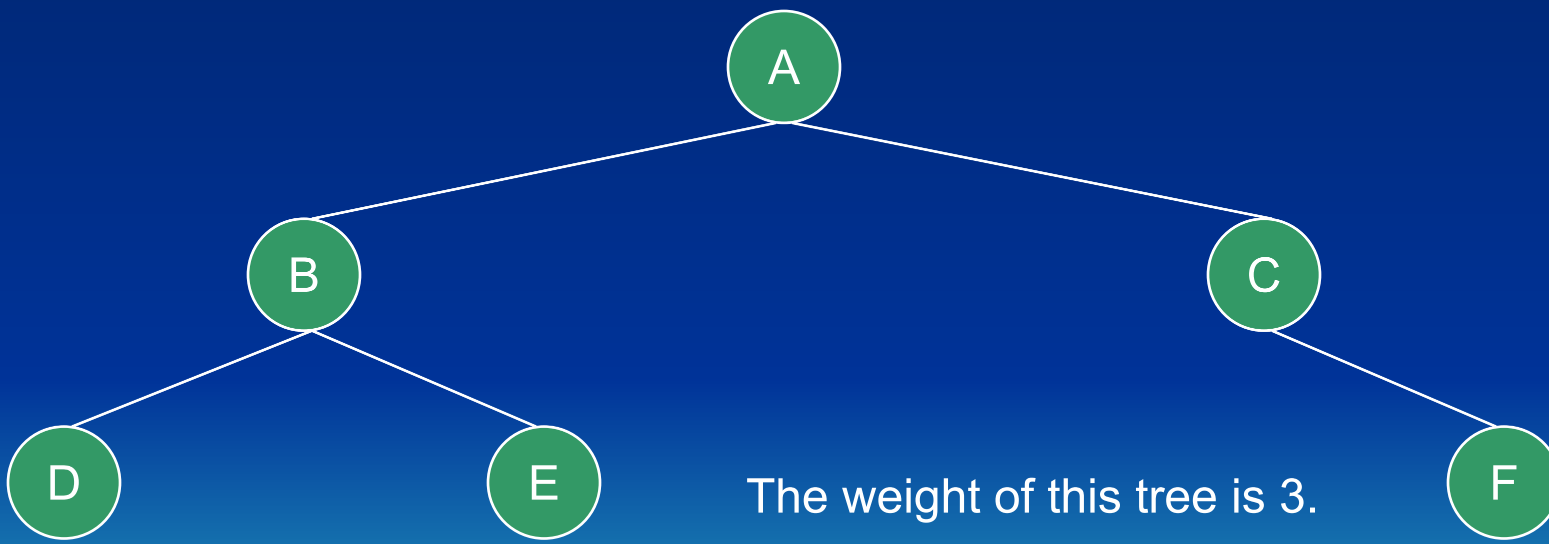


This tree has degree 2.

Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

Weight of a tree – number of leaf nodes in the tree



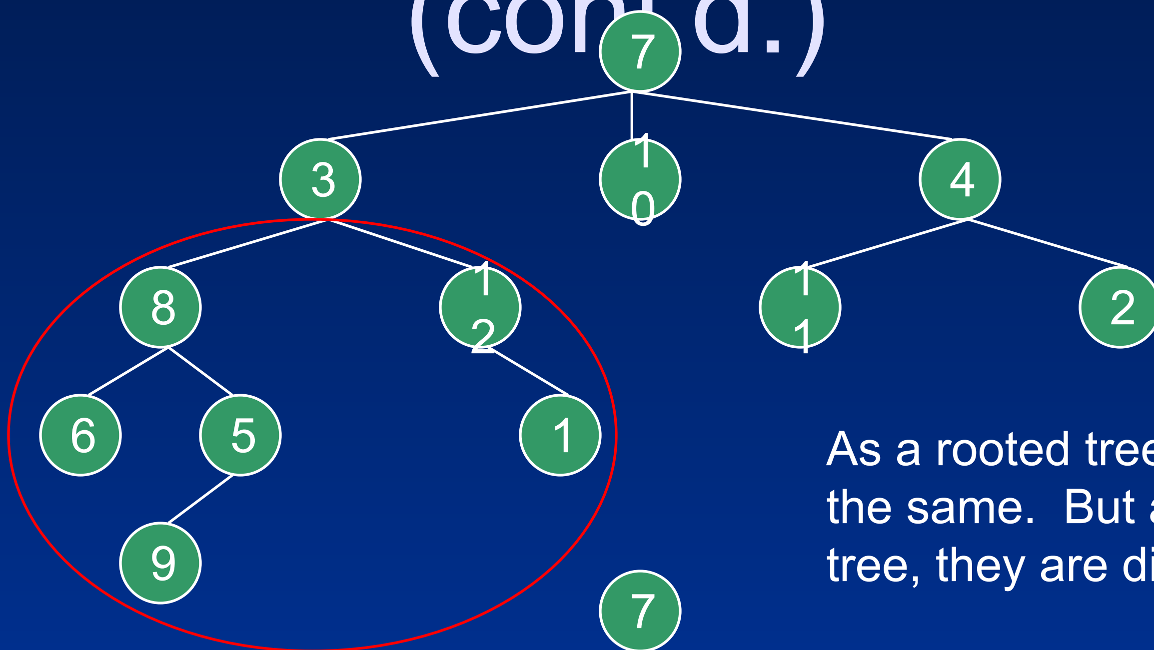
The weight of this tree is 3.

Rooted and Ordered Trees (cont'd.)

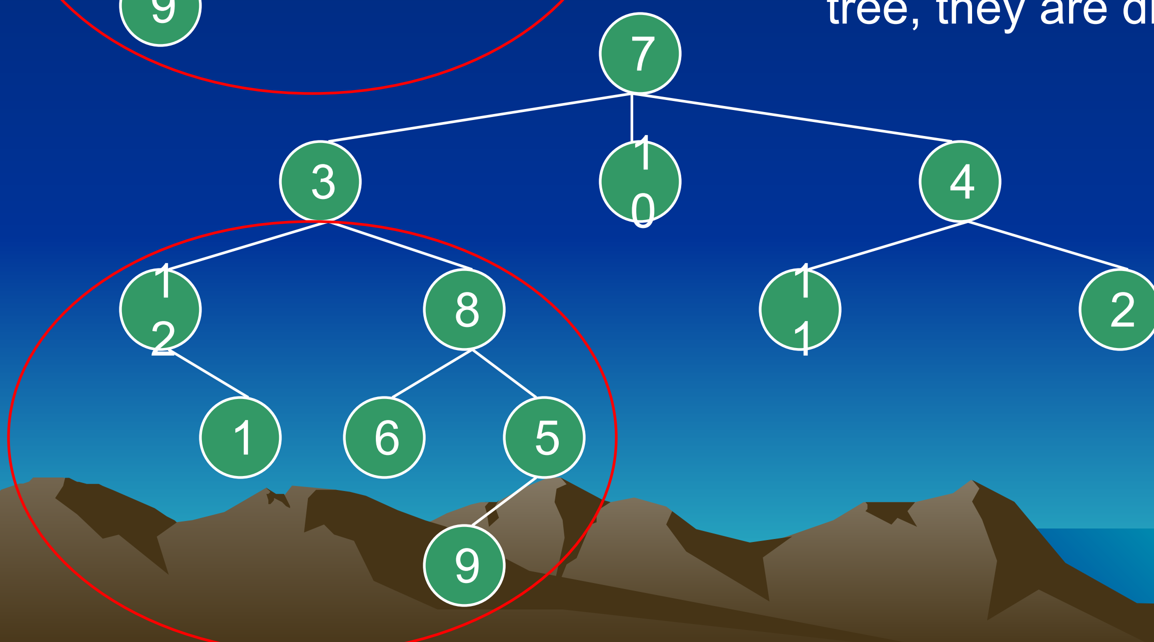
Concepts and Terminologies

Ordered tree – a rooted tree in which the children of each node are ordered, i.e., if a node has k children, then there is a first child, a second child, …, and a kth child

Rooted and Ordered Trees (cont'd.)



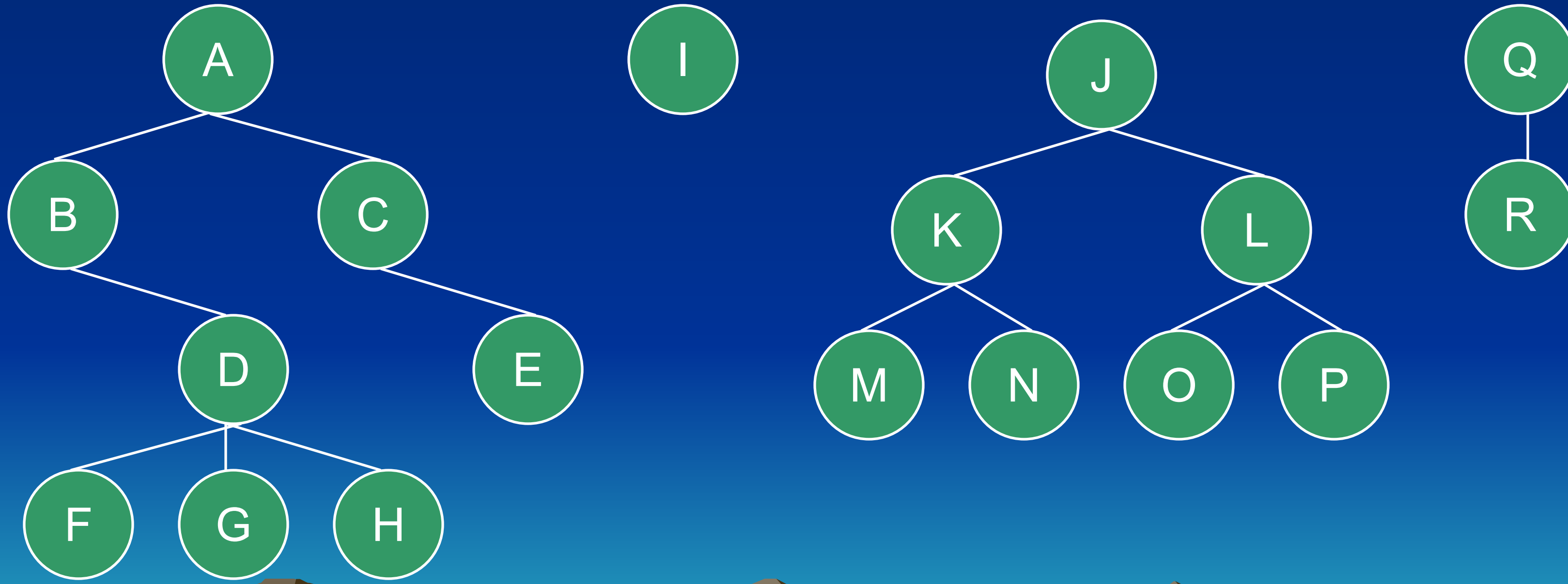
As a rooted tree, both are the same. But as an ordered tree, they are different!



Rooted and Ordered Trees (cont'd.)

Concepts and Terminologies

Forest – set of disjoint trees



Binary and Positional Trees

Binary trees are defined recursively. A binary tree *T* is a structure defined on a finite set of nodes that either

- contains no nodes; or
- is composed of three disjoint sets of nodes; a root node, a binary tree called its left subtree, and a binary tree called its right subtree.

Therefore, a binary tree is an ordered tree with degree = 2.

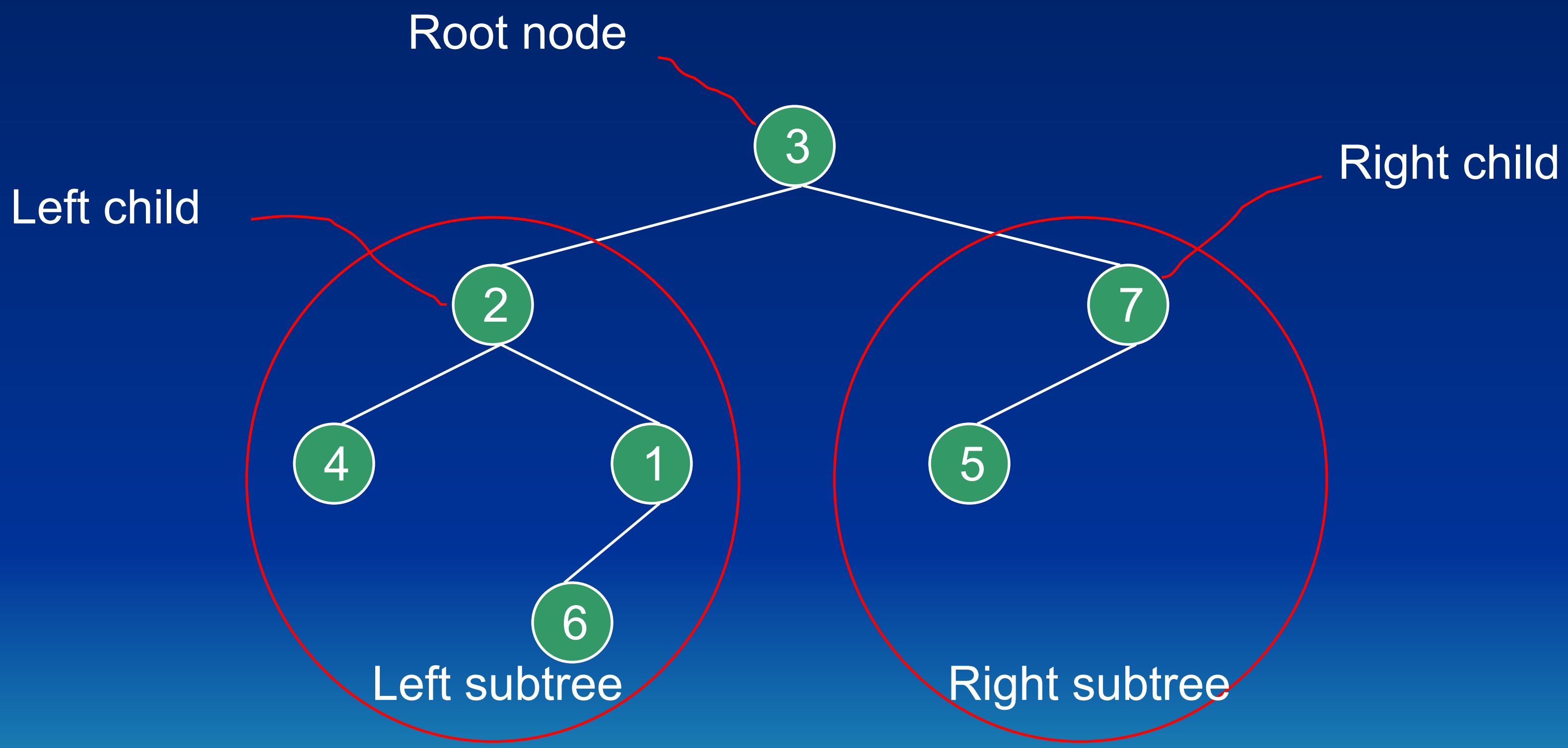
Binary and Positional Trees (cont'd.)

Empty Tree/Null Tree

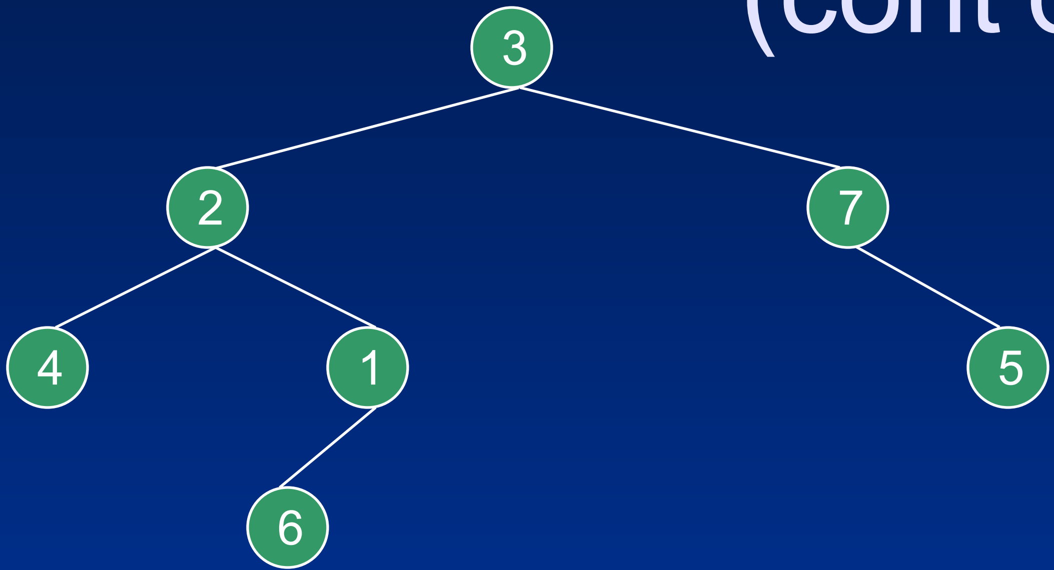
- binary tree that contains no nodes
- denoted by NIL

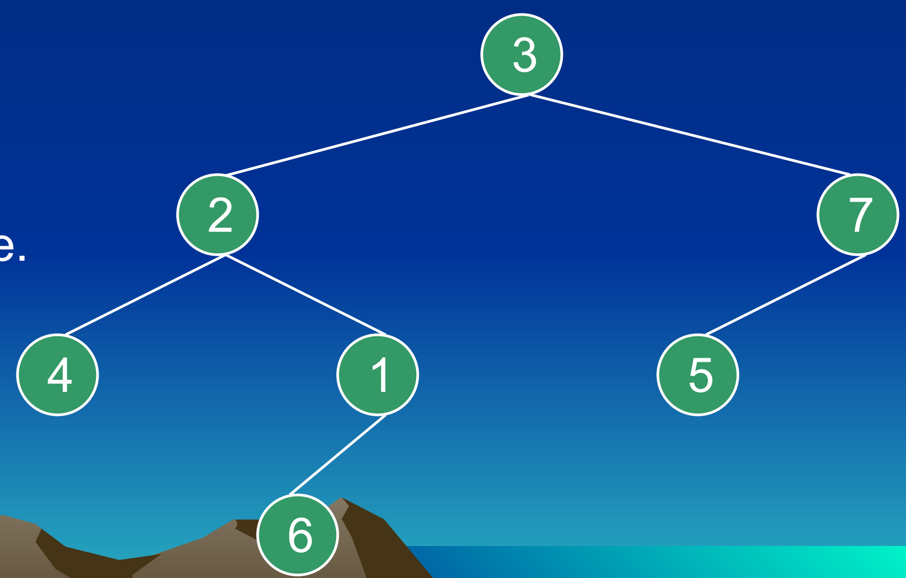
Binary and Positional Trees (cont'd.)

Binary Tree Illustrated



Binary and Positional Trees (cont'd.)



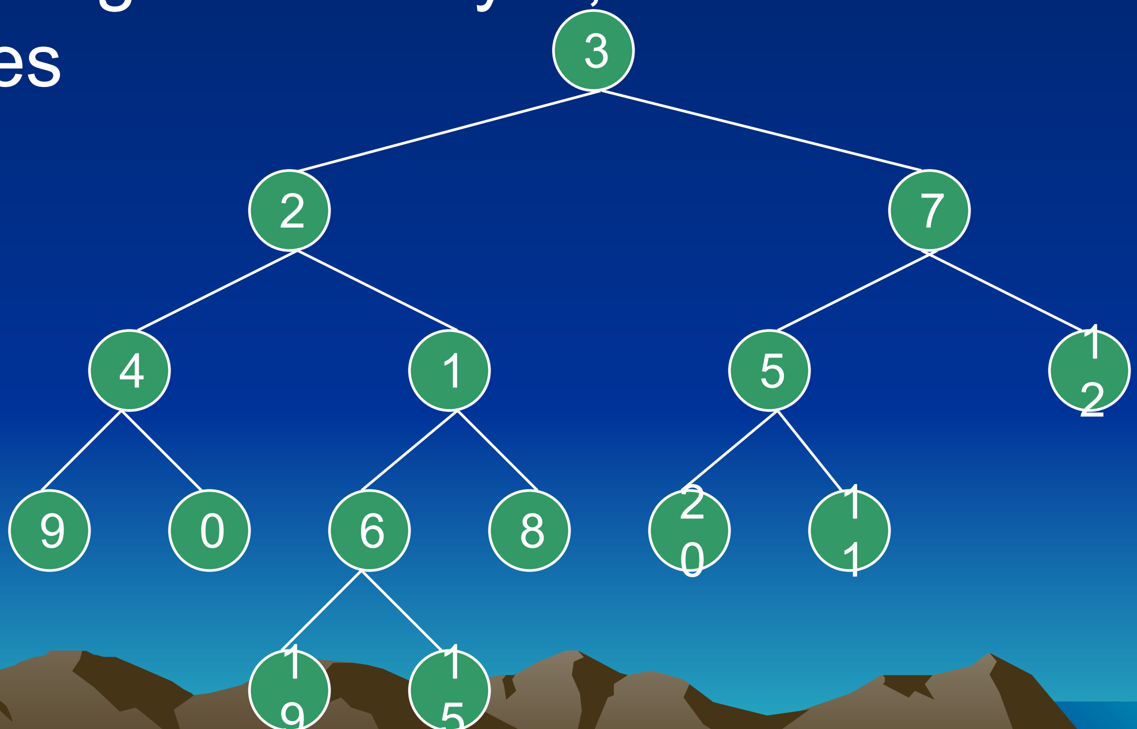


As ordered trees, these trees are the same.
As binary trees, they are distinct.

Binary and Positional Trees (cont'd.)

Full Binary Tree

- a binary tree where each node is either a leaf or has degree exactly 2, there are no degree – 1 nodes



Binary and Positional Trees (cont'd.)

Positional Tree

- the children of a node are labeled with distinct positive integers, the *i*th child of a node is absent if no child is labeled with integer *i*

Binary and Positional Trees (cont'd.)

k-ary Tree

- a positional tree in which for every node, all children with labels greater than k are missing

Note:

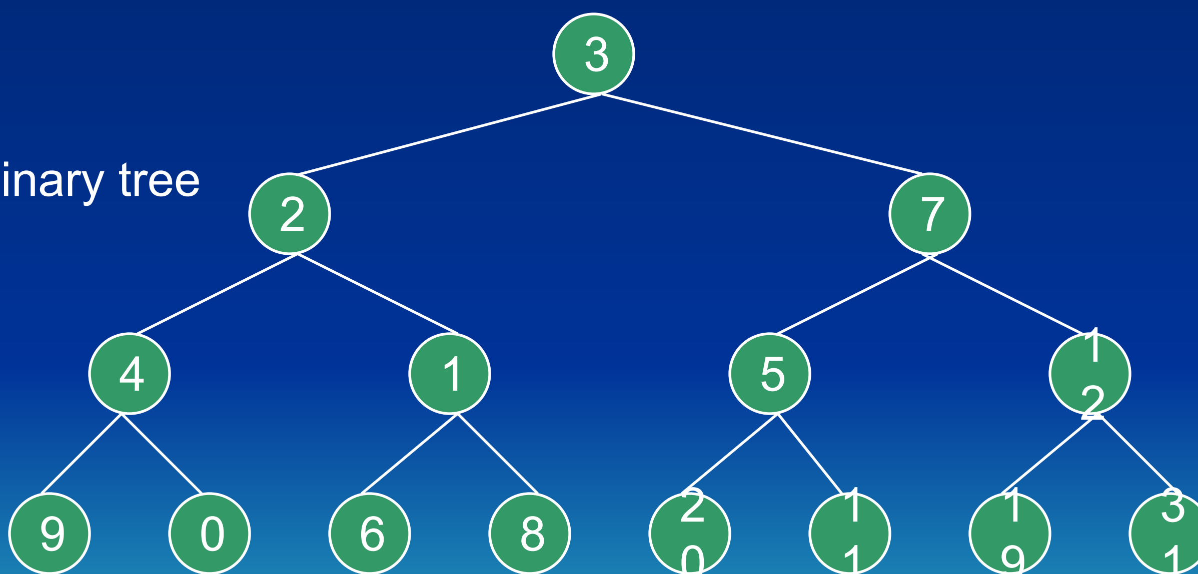
A binary tree is a k-ary tree with $k = 2$.

Binary and Positional Trees (cont'd.)

Complete *k*-ary Tree

- a *k*-ary tree in which all leaves have the same depth and all internal nodes have degree *k*

Complete binary tree of height 3



Binary and Positional Trees (cont'd.)

How many leaves does a complete k-ary tree of height h have?

Answer: k^h.

What is the height of a complete k-ary tree with n leaves?

Answer: $\log_k n$.

Binary and Positional Trees (cont'd.)

What is the number of internal nodes of a complete *k*-ary tree of height *h*?

Answer:

$$1+k+k^2+\ldots+k^{h-1}=\sum_{i=0}^{h-1}k^i=\frac{k^h-1}{k-1}$$

Therefore, a complete binary tree has $2^h - 1$ internal nodes.

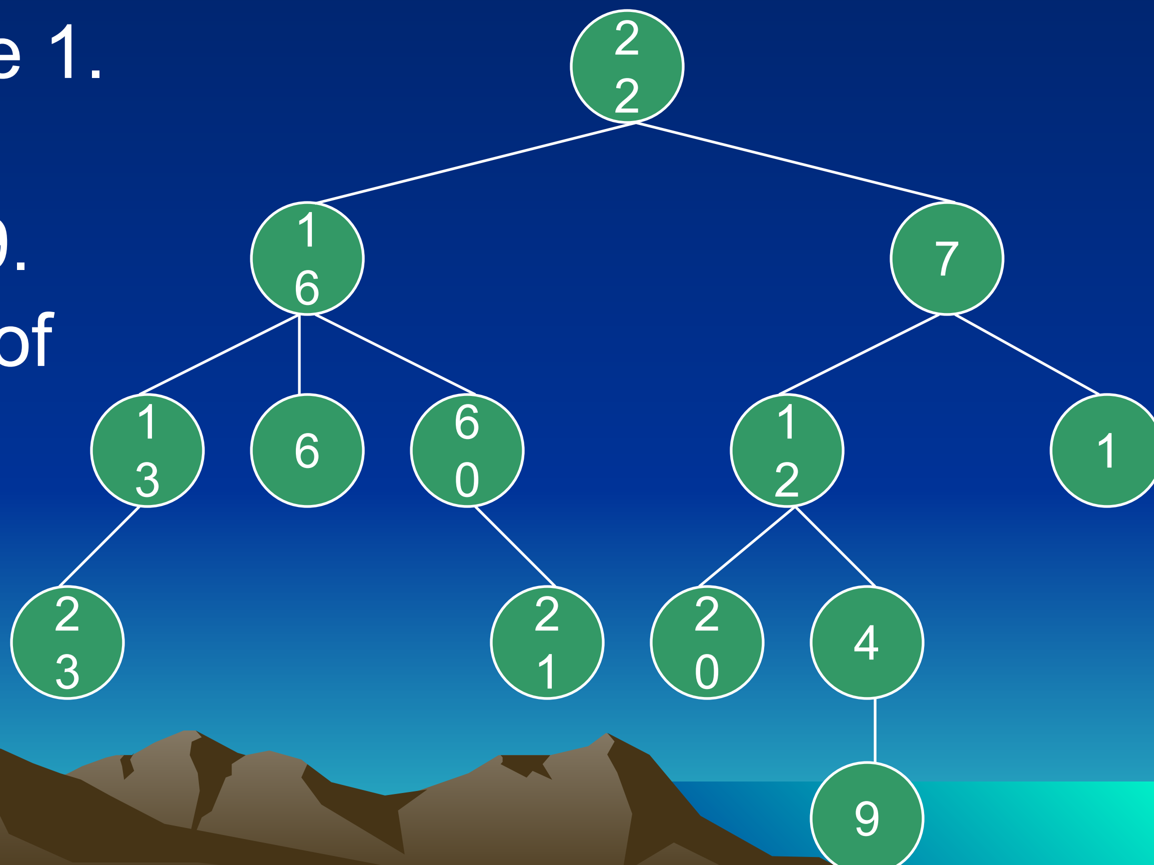
Exercises on Trees

1. Name the three properties of a tree.
2. Is a tree a forest?
3. What do you call the special designated node in a tree?
4. What is the minimum number of nodes in a tree?
5. Can a tree have no subtrees at all?

Short Quiz on Trees (cont'd.)

Given the tree to the right, identify the ff.:

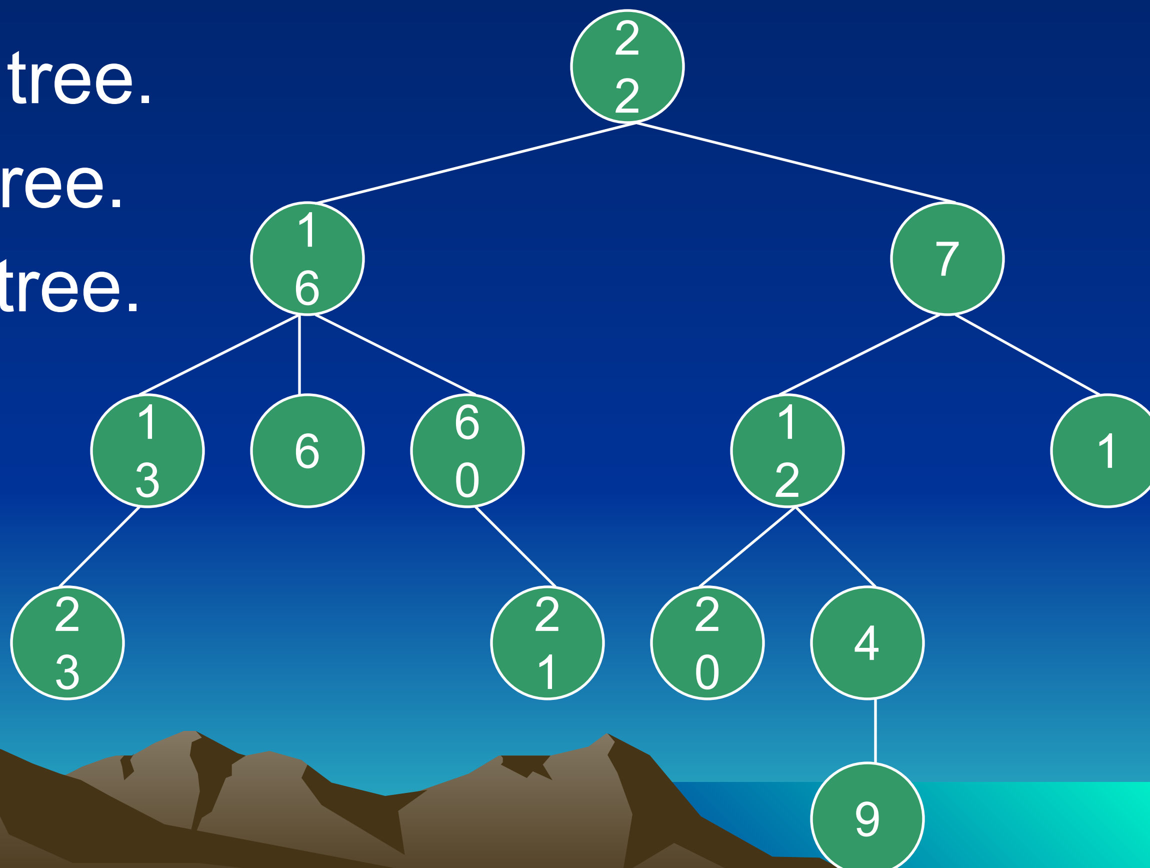
6. Children of node 16.
7. Parent of node 1.
8. Siblings of 23.
9. Ancestors of 9.
10. Descendants of 16.
11. Leaves.
12. Non-leaves.



Short Quiz on Trees (cont'd.)

Given the tree to the right, identify the ff.:

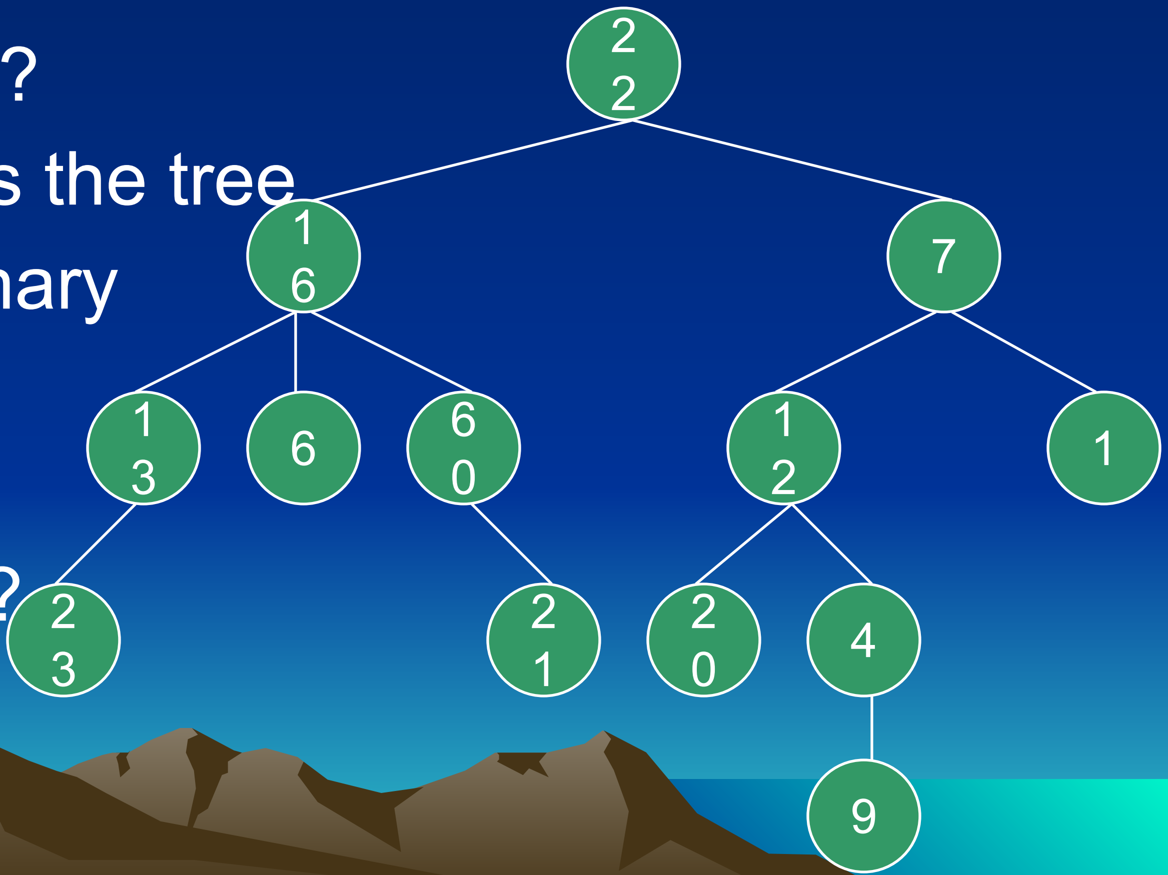
13. Depth of node 4.
14. Degree of the tree.
15. Height of the tree.
16. Weight of the tree.
17. Is the tree a binary tree?



Short Quiz on Trees (cont'd.)

Given the tree to the right, identify the ff.:

18. Removing 6, is the tree a full binary tree?
19. Removing 6, is the tree a complete binary tree?
20. Is a full binary tree complete?



Short Quiz on Trees (cont’d.)

Given the tree to the right, identify the ff.:

21. Is a complete binary tree full?
22. How many leaves does a complete *n*-ary tree of height *h* have?
23. What is the height of a complete *n*-ary tree with *m* leaves?
24. What is the number of internal nodes of a complete *n*-ary tree of height *h*?
25. What is the total number of nodes a complete *n*-ary tree of height *h* have?