

# CSC 212: Data Structures and Abstractions

## Trees

Marco Alvarez

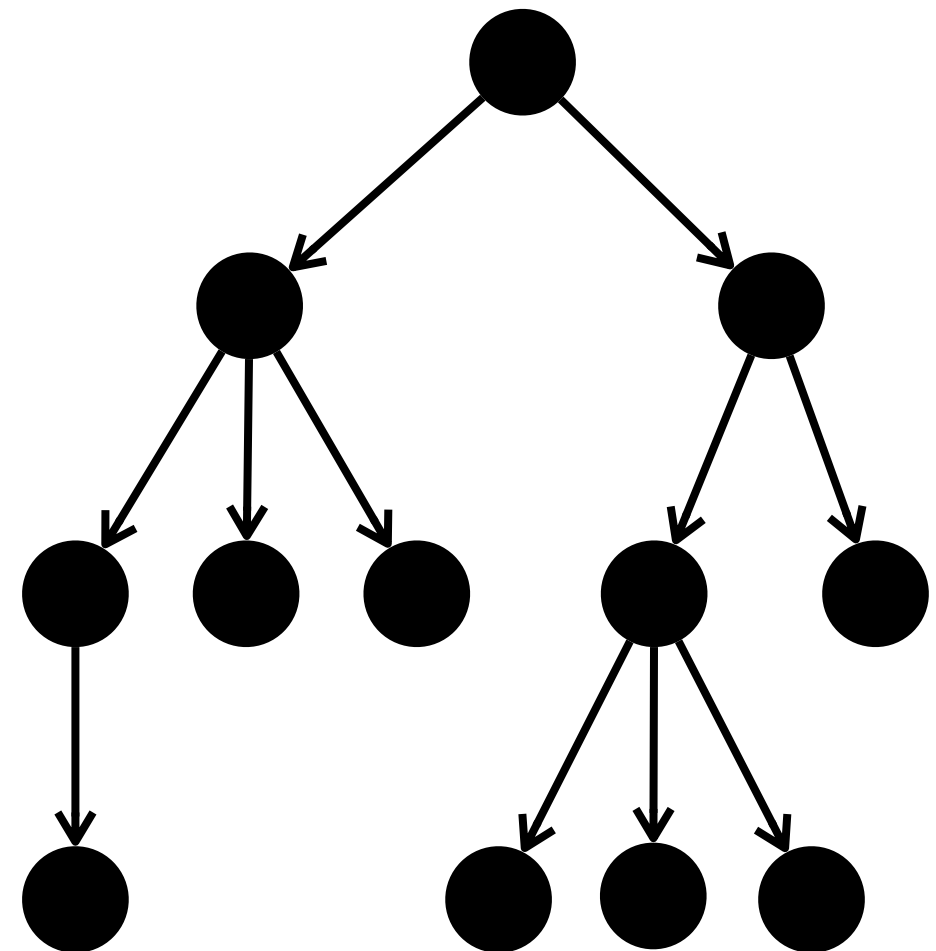
Department of Computer Science and Statistics  
University of Rhode Island

Fall 2020



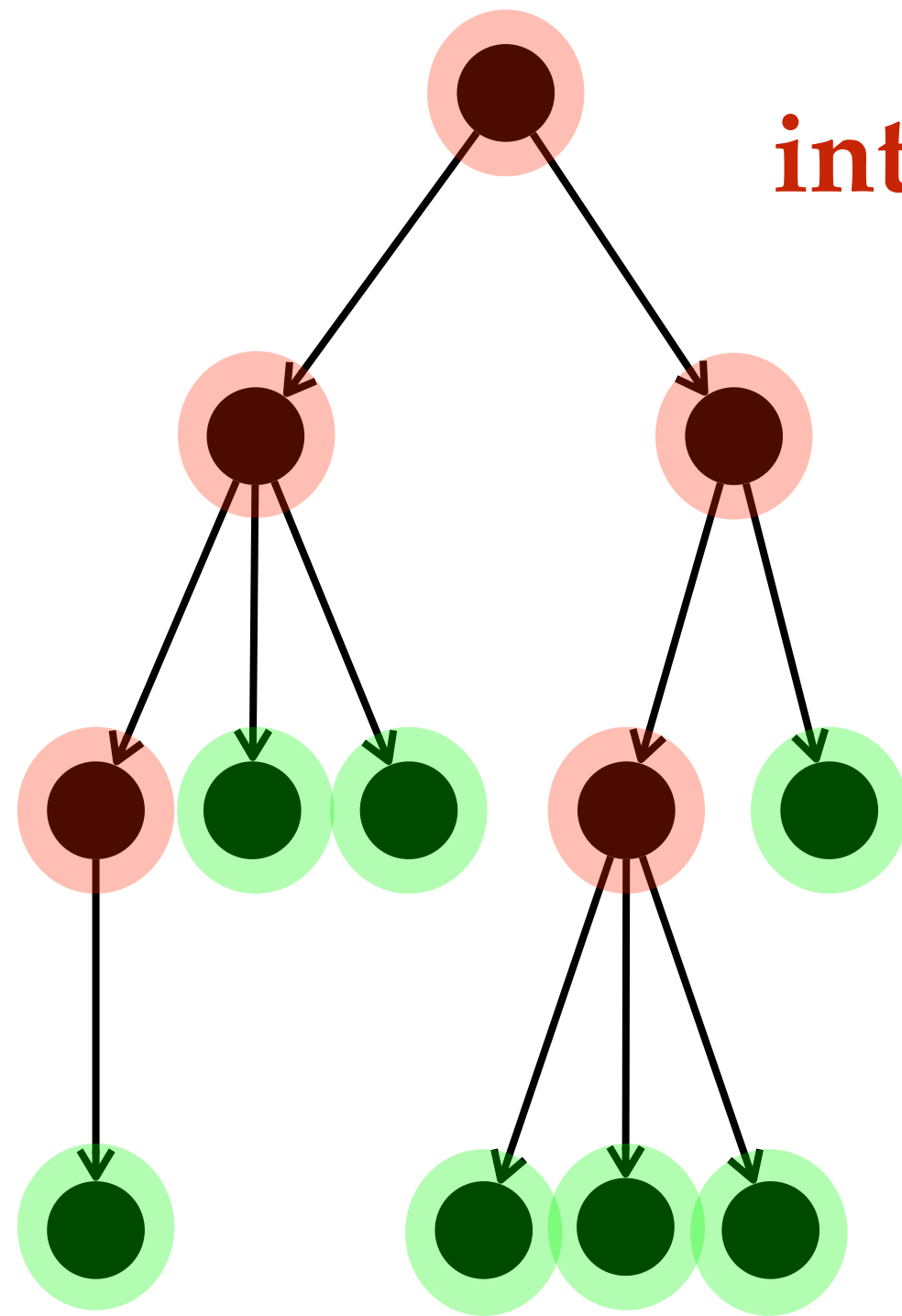
# Trees

- Lists, Stacks, Queues are **linear data structures**
- Trees allow for **hierarchical** relationships
  - ✓ nodes have **parent-child** relation



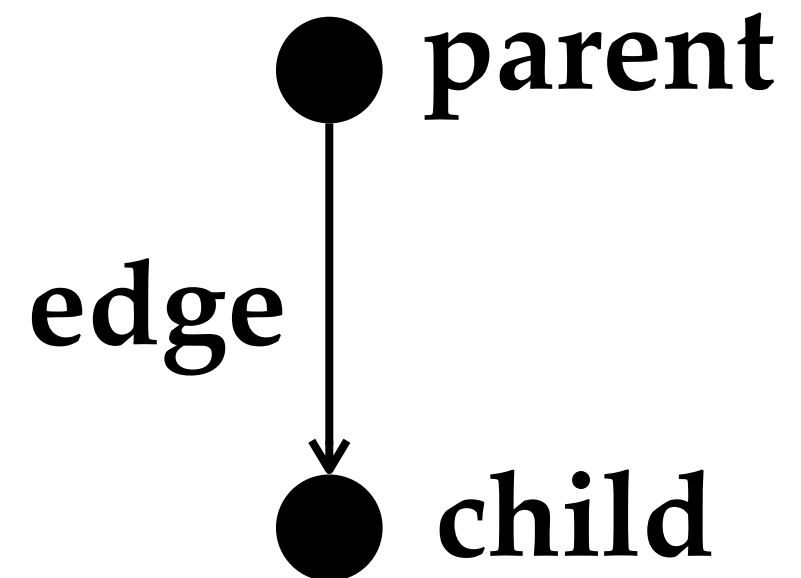
# Trees (jargon)

---



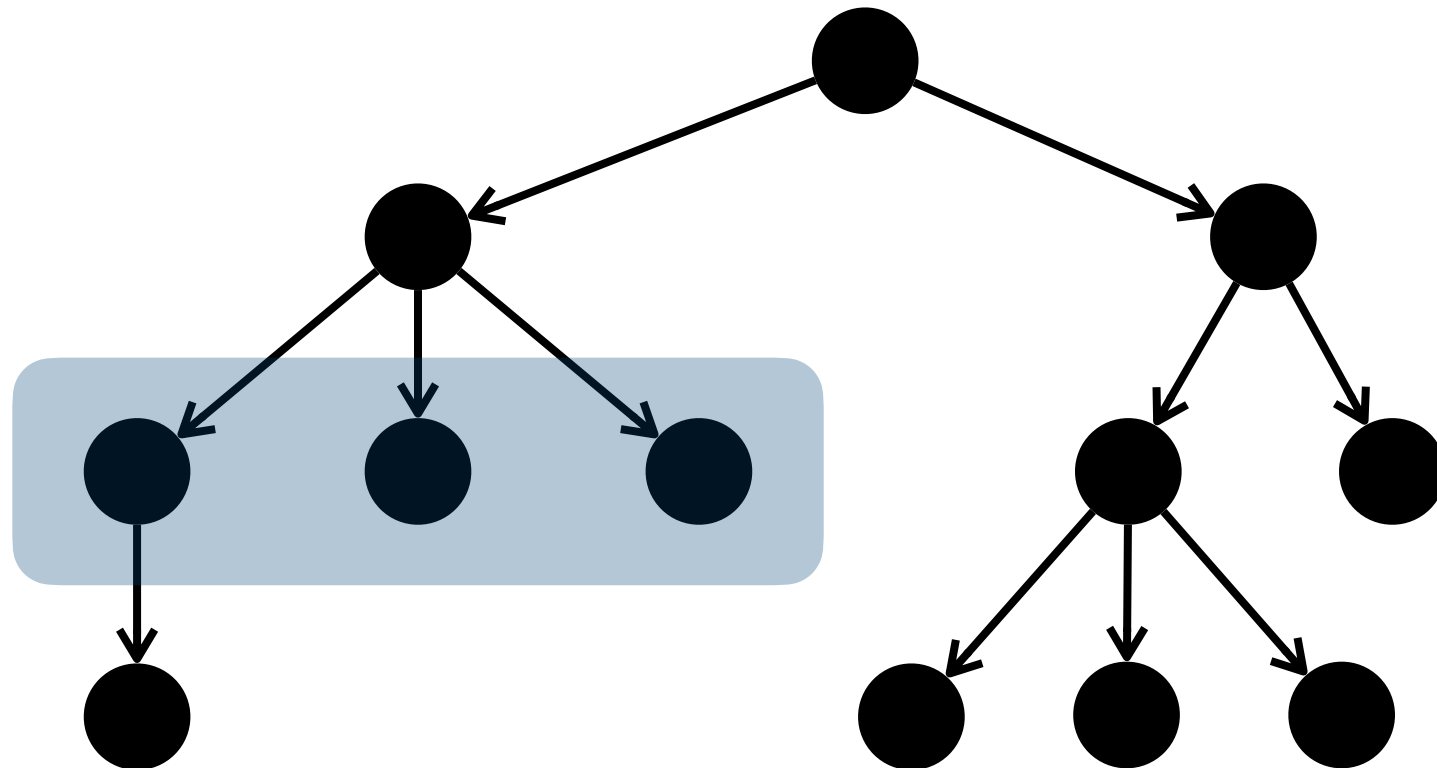
**internal nodes**

**leaves**

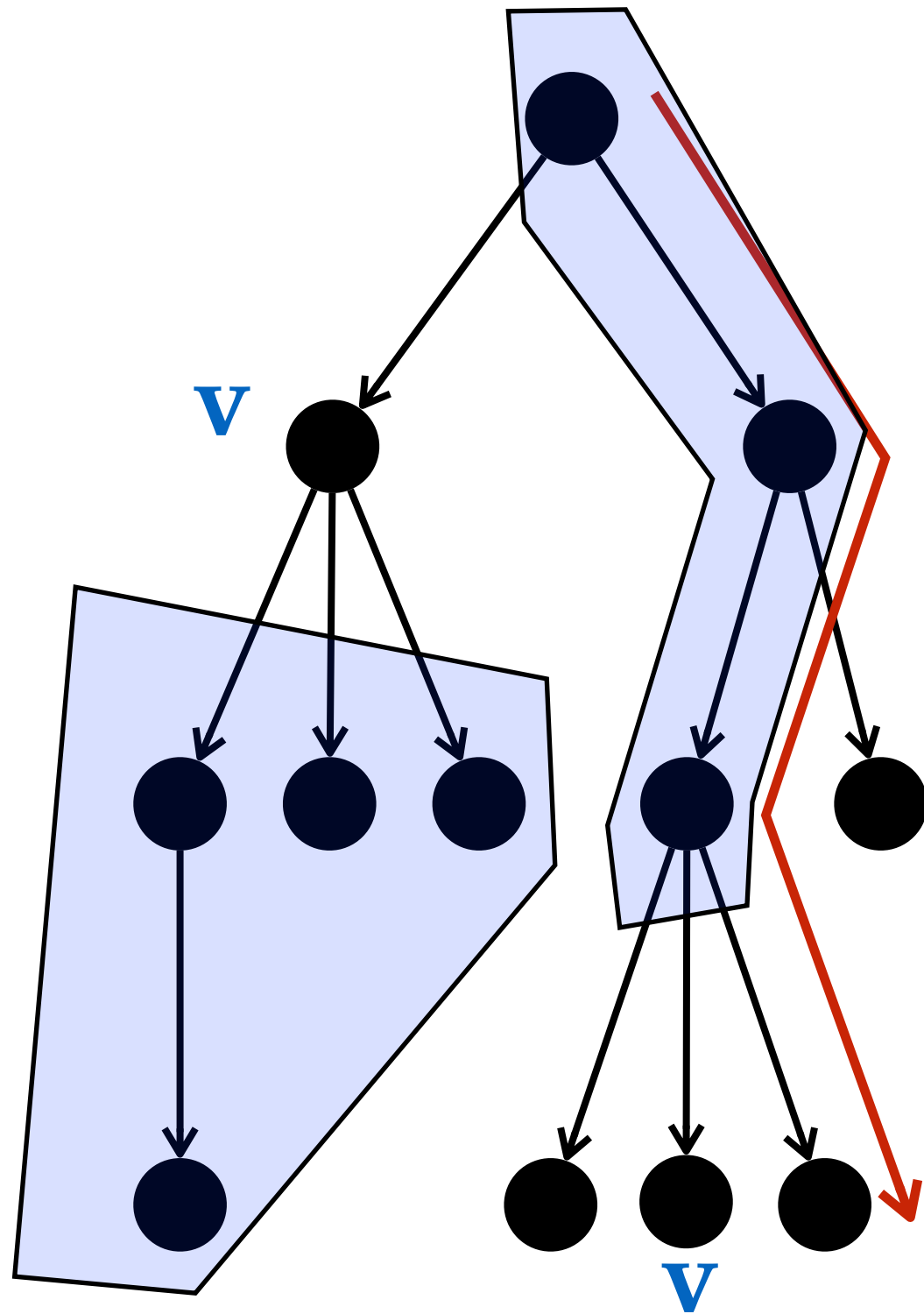


# Trees (jargon)

- Each node is either a **leaf** or an **internal node**
  - ✓ an internal node has one or more children
  - ✓ a leaf node (external node) has no children
- Nodes with the same parent are **siblings**



# Paths

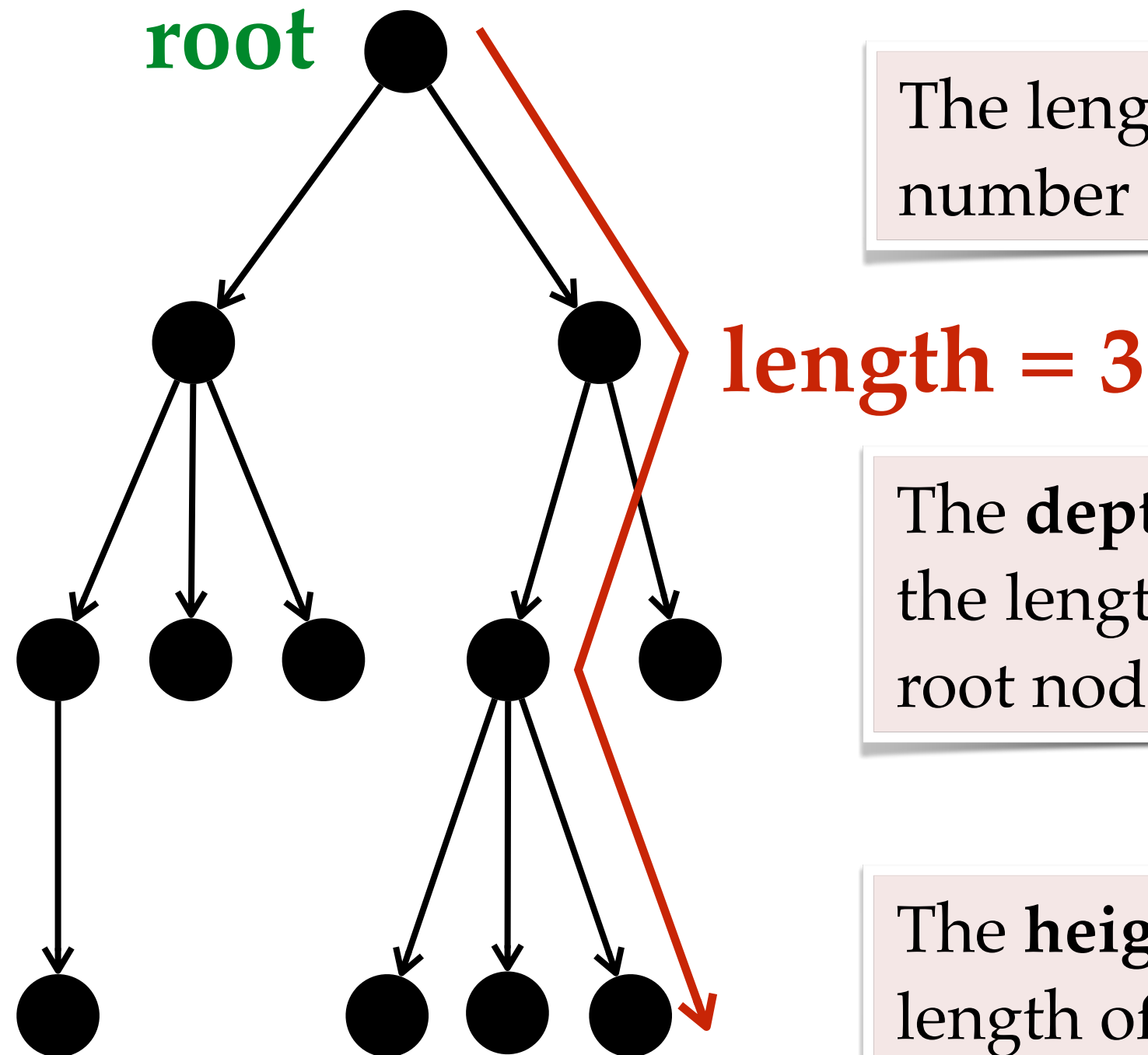


A **path** from node  $v_0$  to  $v_n$  is a sequence of nodes  $v_0, v_1, v_2, \dots, v_n$ , where there is an edge from one node to the next

The **descendants** of a node  $v$  are all nodes reached by a path from node  $v$  to the leaf nodes

The **ancestors** of a node  $v$  are all nodes found on the path from the root to node  $v$

# Depth and Height

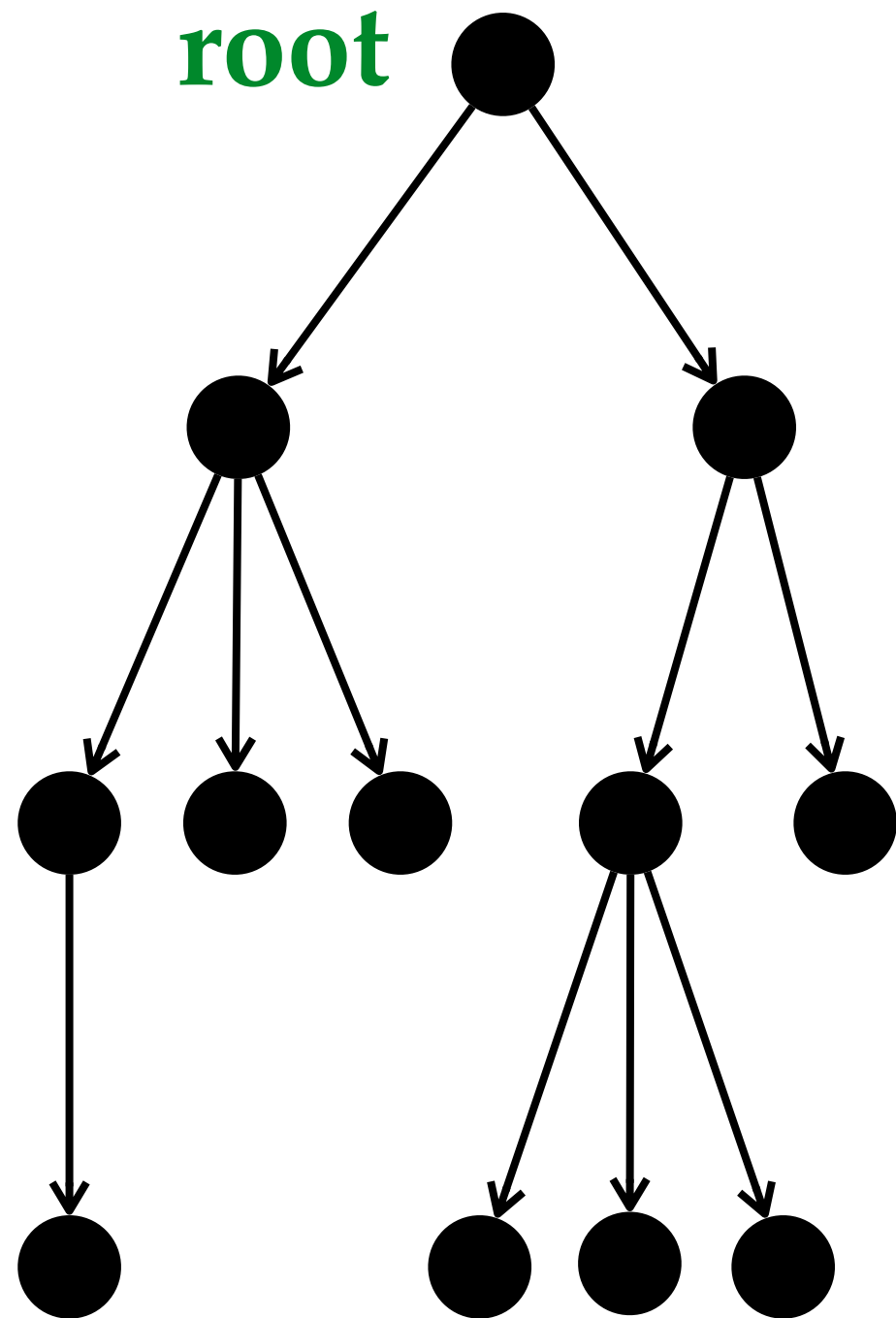


The length of a **path** is the number of edges in the path

The **depth** (level) of a node  $v$  is the length of the path from the root node to  $v$

The **height** of a node  $v$  is the length of the path from  $v$  to its deepest descendant

# Tree Properties



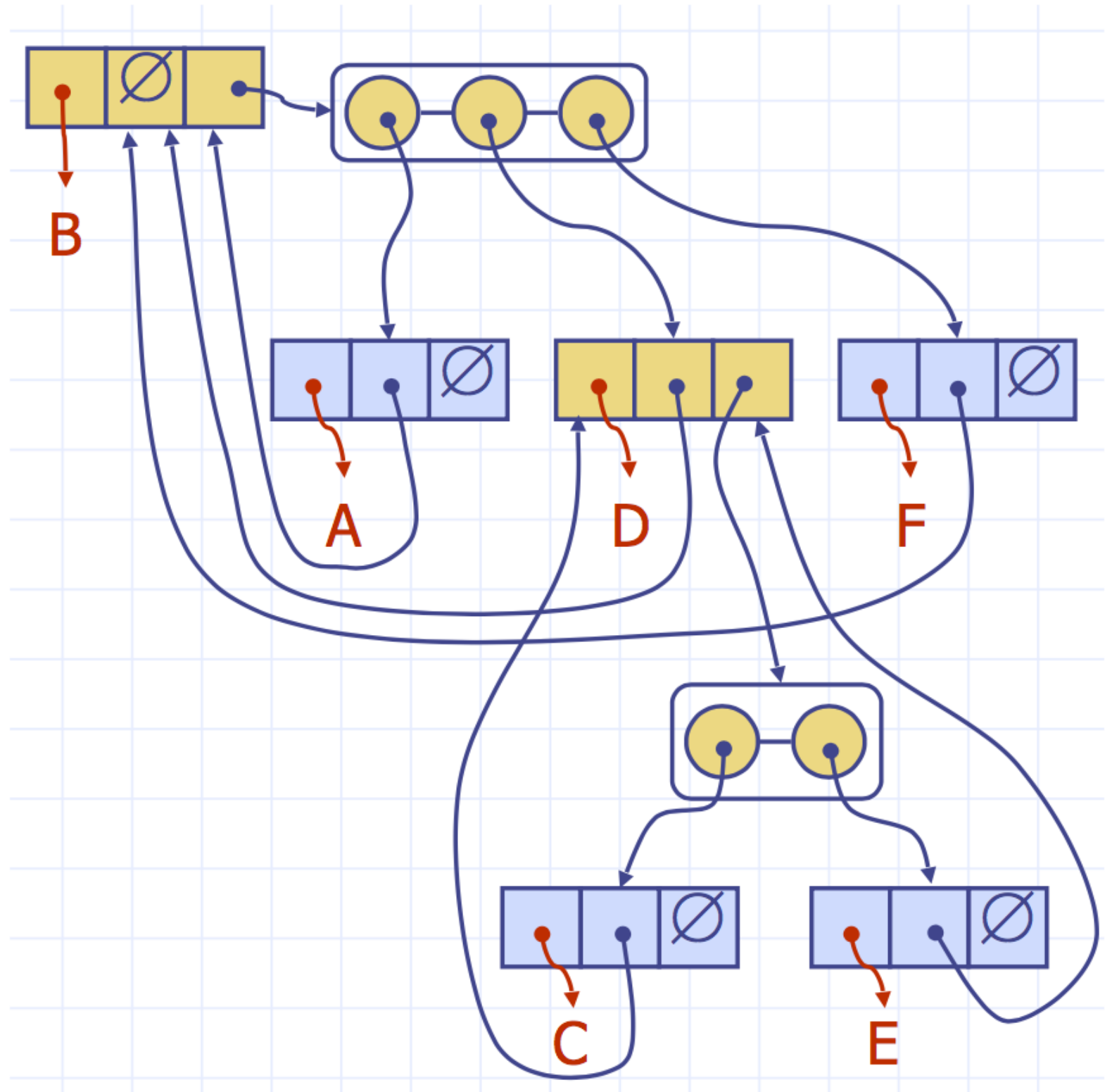
The **depth** of the tree is the depth of deepest node

The **height** of the tree is the height of the root

# Implementing general trees?

**Node:**

data  
parent  
children []



From Algorithm Design and Applications, Goodrich & Tamassia