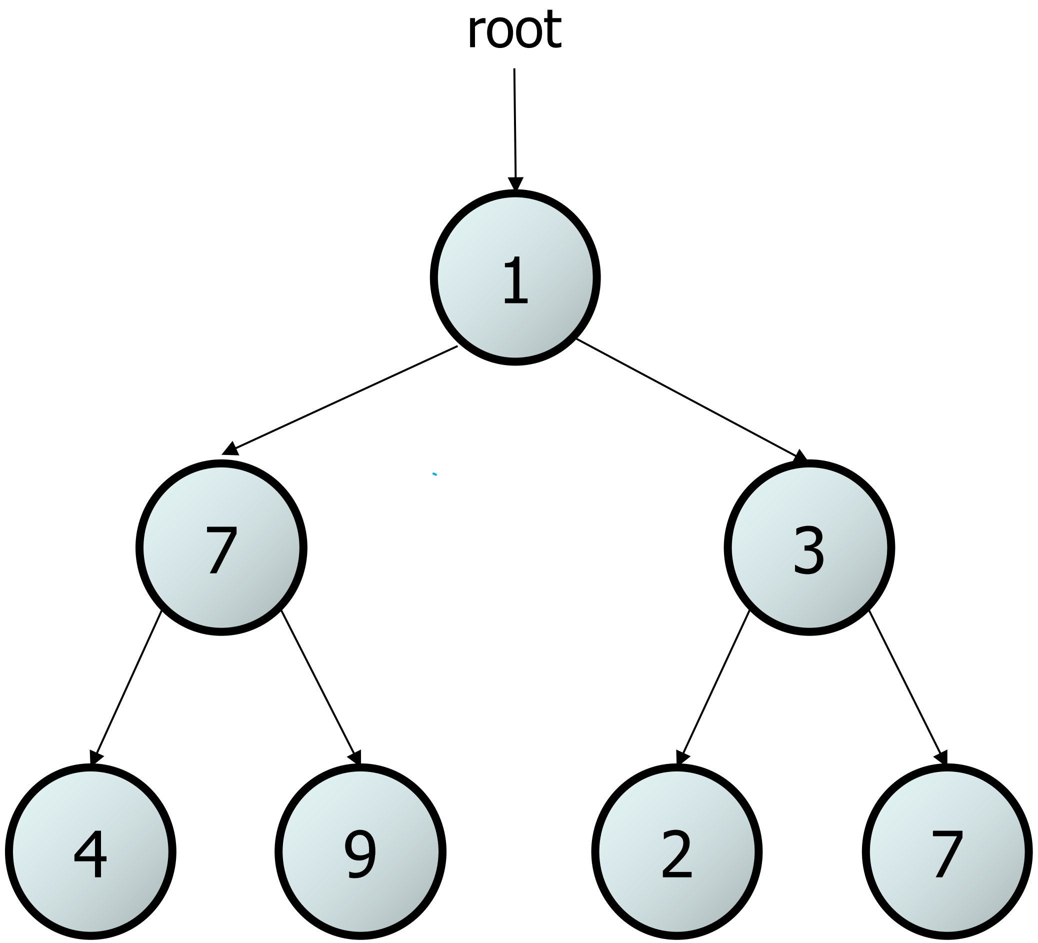
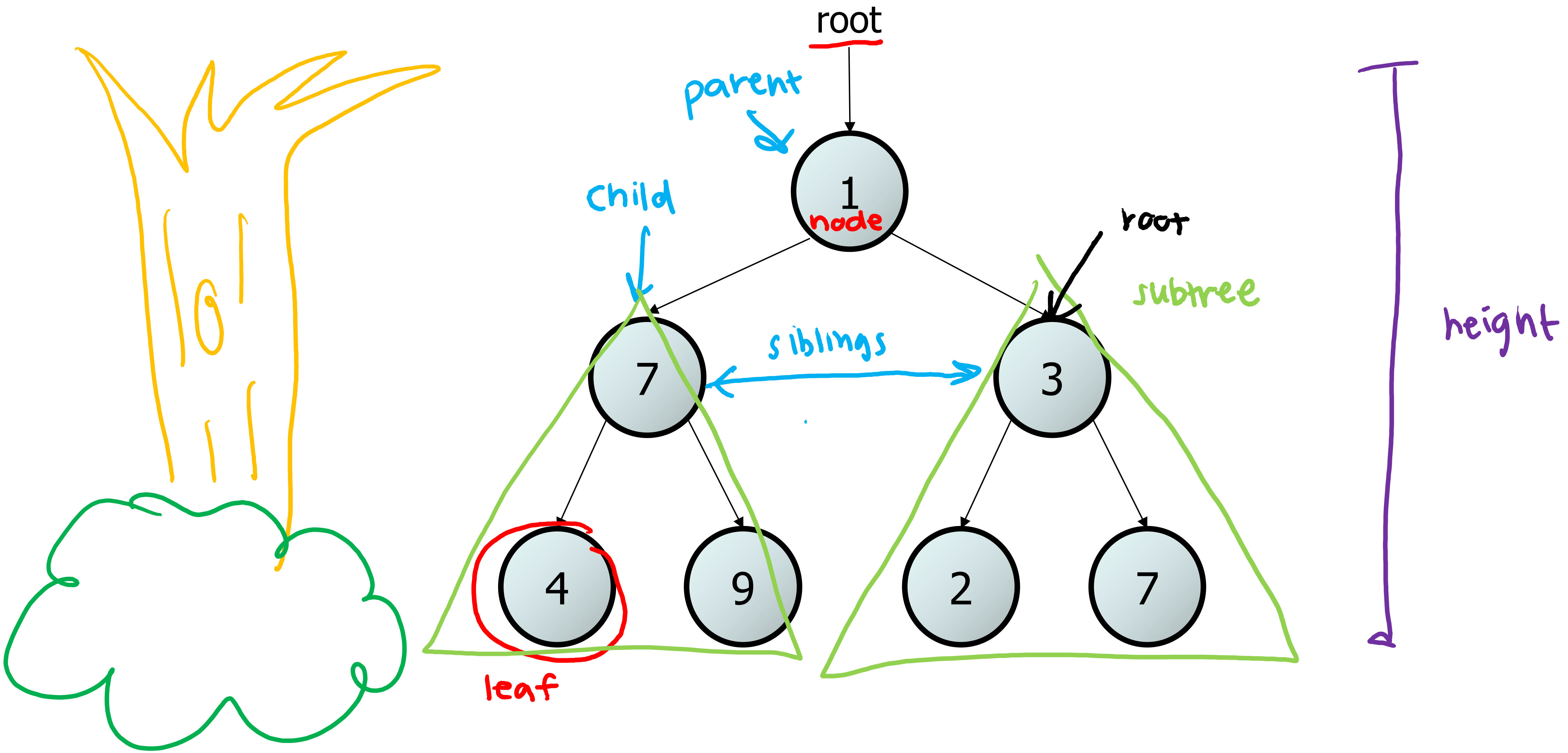
Lecture 16: Binary Trees

08/01/22

Binary Tree



Binary Tree


root
parent
1
node
child
root
subtree
height
siblings
7
3
4
9
2
7
leaf

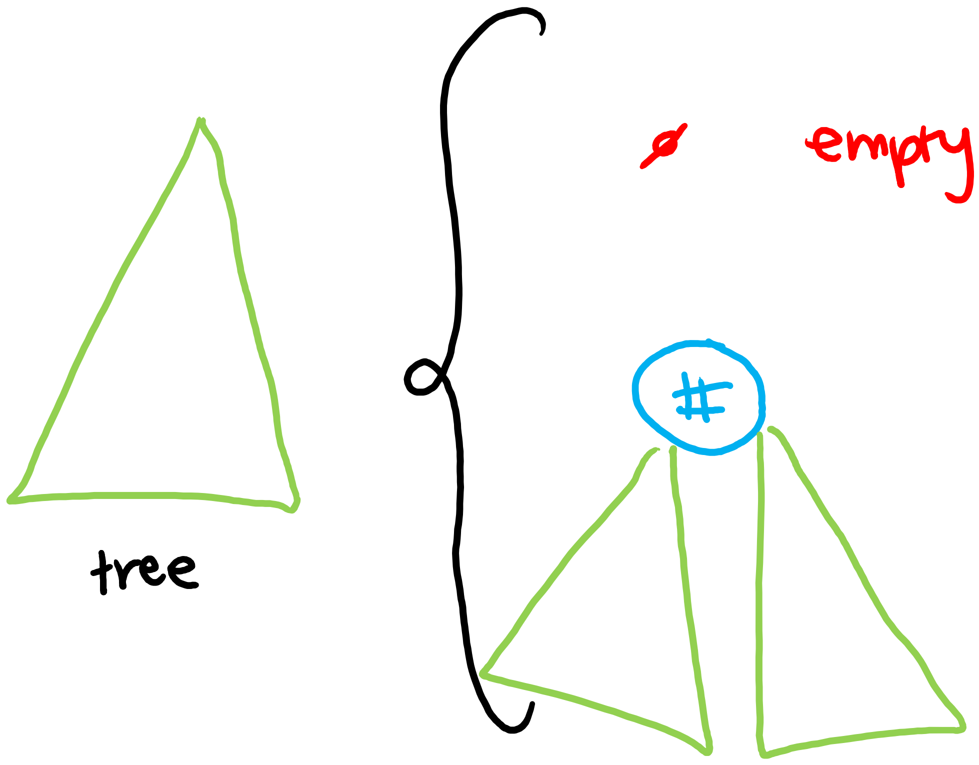
Binary Tree Terminology

- **node:** an object containing a data value and left/right children
- **root:** topmost node of a tree
- **leaf:** a node that has no children
- **branch:** any internal node; neither the root nor a leaf
- **parent:** a node that refers to this one
- **child:** a node that this node refers to
- **sibling:** a node with a common parent
- **subtree:** the smaller tree of nodes on the left or right of the current node
- **height:** length of the longest path from the root to any node

(Recursive) Definition of a Tree

A tree is either:

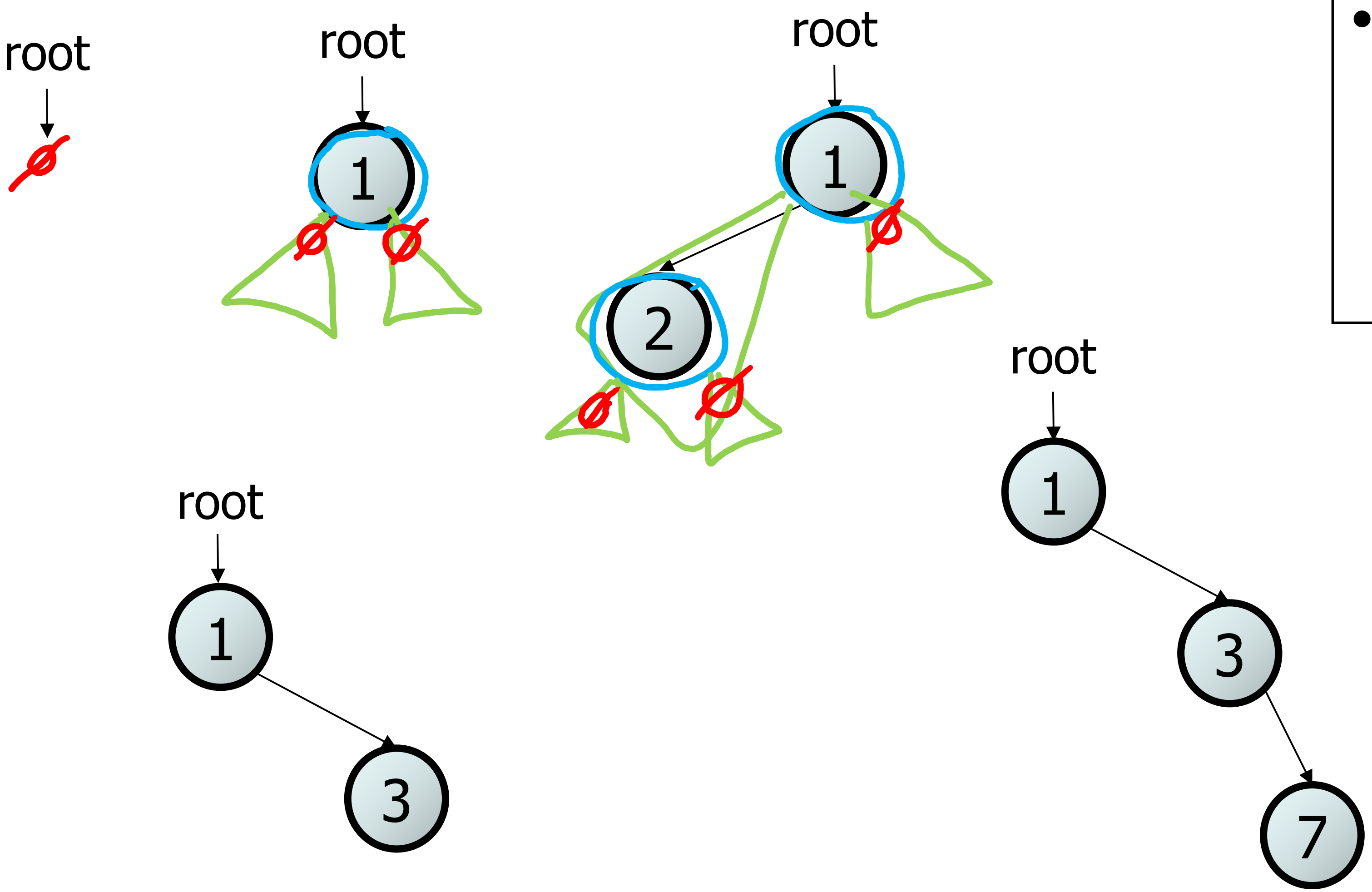
- An empty tree (null), *or*
- A root node with:
  - Data
  - A left subtree
  - A right subtree



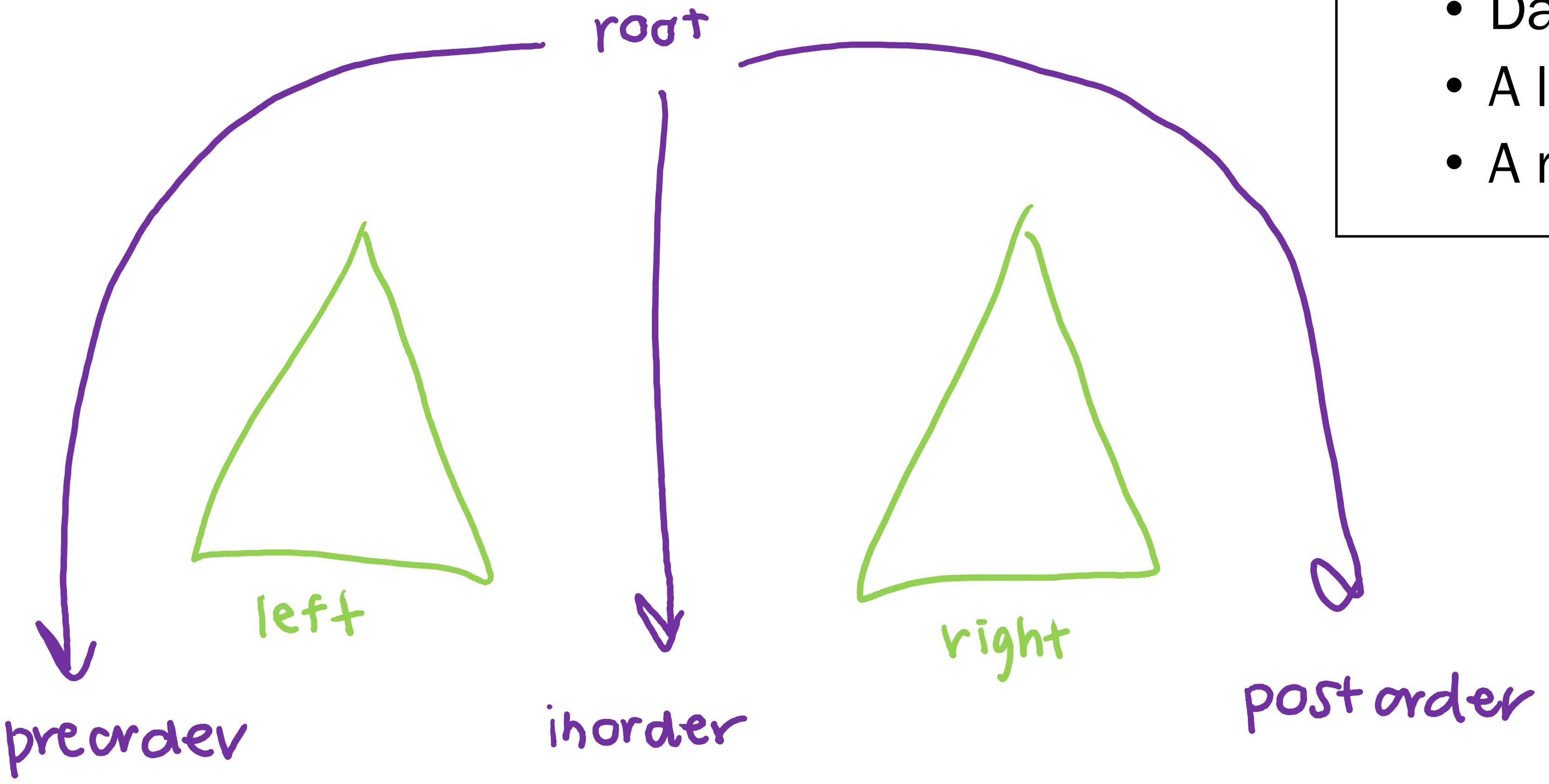
Many different trees!

A tree is either:

- An empty tree (null), *or*
- A root node with:
  - Data
  - A left subtree
  - A right subtree



Traversals



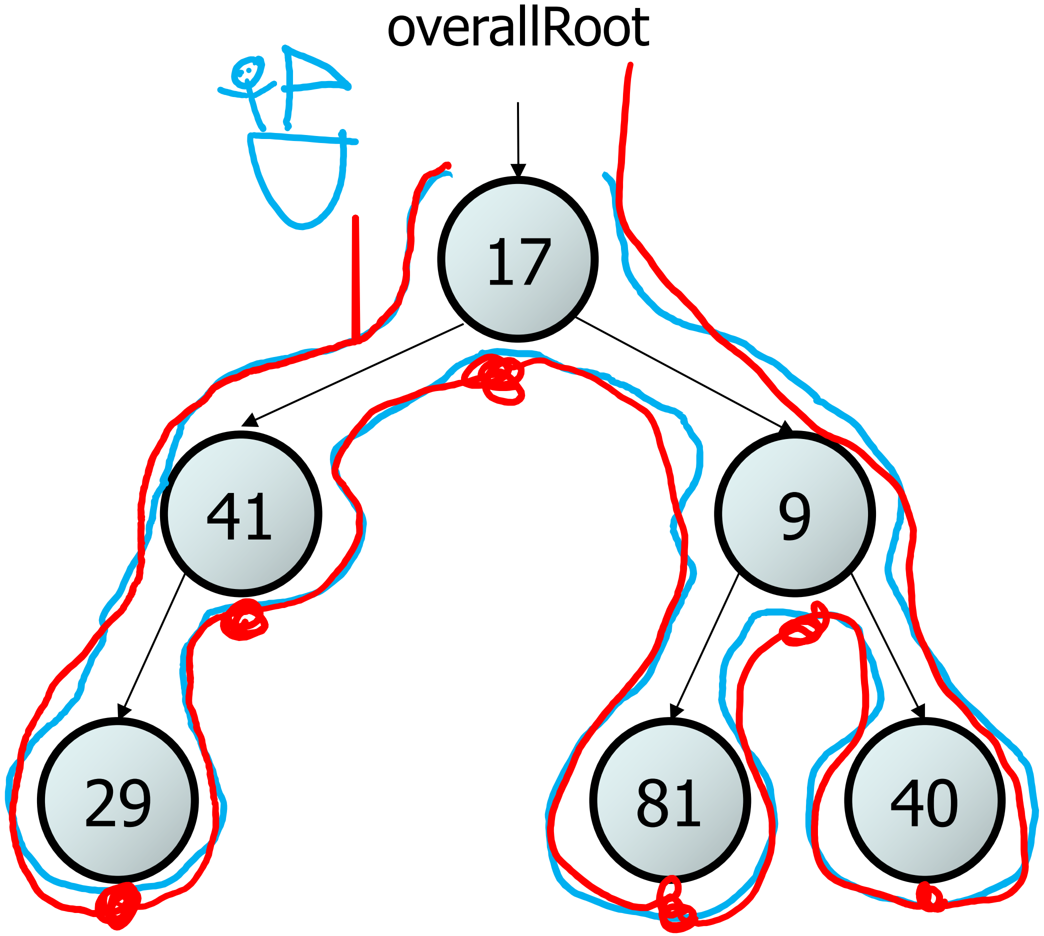
A tree is either:

- An empty tree (null), *or*
- A root node with:
  - Data
  - A left subtree
  - A right subtree

Traversals

- **pre-order:** process root node, then left subtree, then right subtree
- **in-order:** process left subtree, then root node, then right subtree
- **post-order:** process left subtree, then right subtree, then root node

Traversal Example

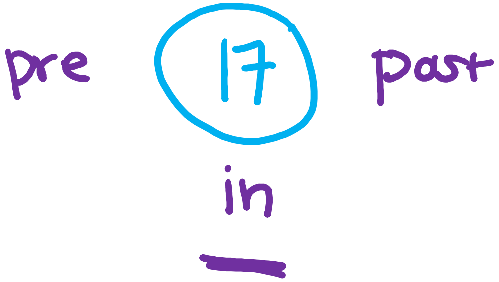


pre: root, left, right

17 41 29 9 81 40

in: left, root, right

29 41 17 81 9 40


pre
17
post
in