Introduction to Trees

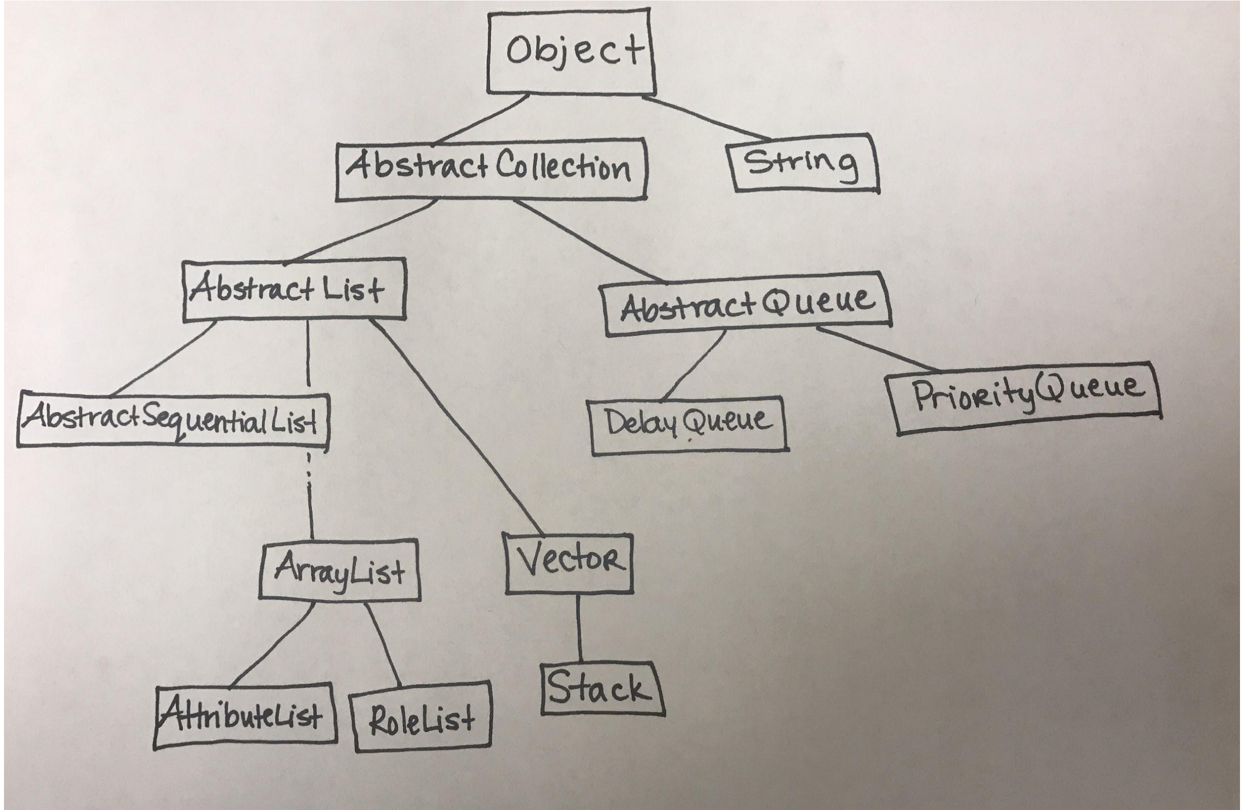
- Description and terminology
- Operations
- Types of trees
- Traversals
- Properties

What is a (rooted) tree?

What is a (rooted) tree?

- An abstract data structure (ADT) that models a hierarchical structure (parent-child relationship)
- Terminology: node, root, parent, child, sibling, internal/external node, leaf, ancestors, descendants, depth, height, subtree, etc.

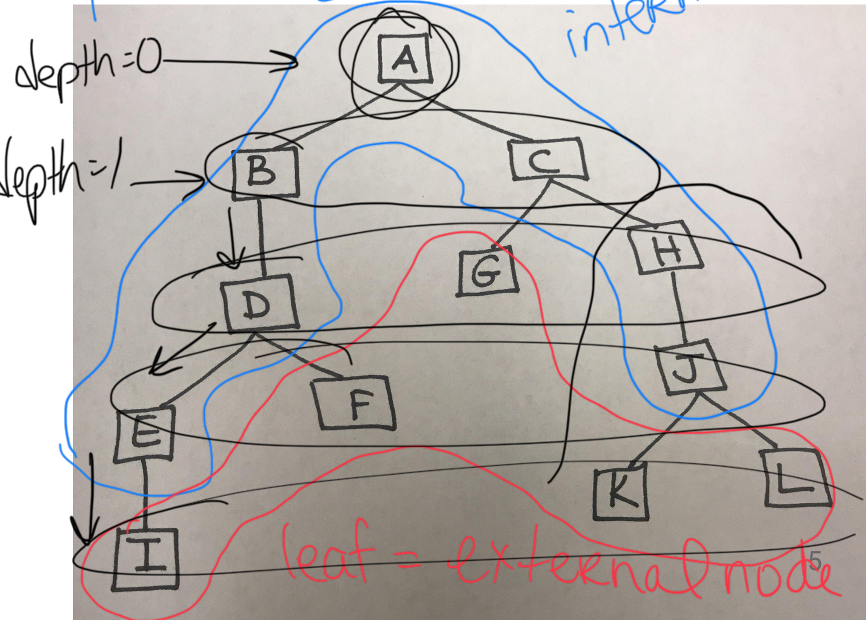
Example: Java classes and inheritance



Some terminology

- *size*: # of nodes
- *root*: the node at level 0
- *parent*: the node directly above another node
- *child:* the node directly beneath another node
- *internal node:* a node with at least one child
- *external node/leaf*: a node with no children
- *height of a node:* the longest distance from the node to a leaf node
- *height of the tree*: the height of the root
- *depth of a node*: the distance from the node to the root

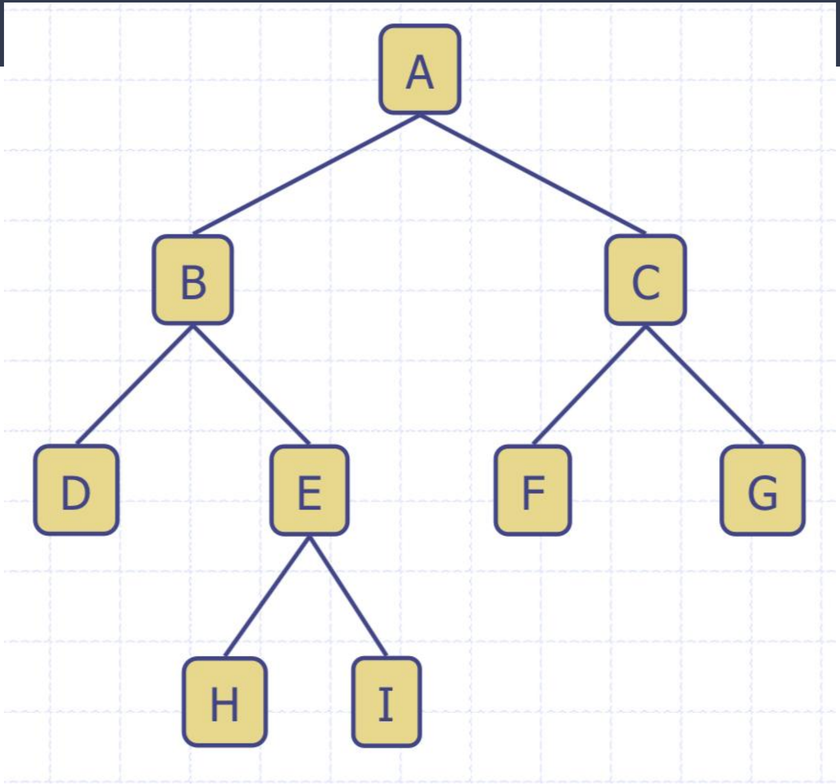
size = 12
root = A
Parent(B) = A



child(D) = E, F
height(B) = 3 height(J) = 4

Binary Tree

- Every node has at most 2 children (right and left), an ordered pair
- New terminology: ***leftChild***, ***rightChild***, ***sibling***

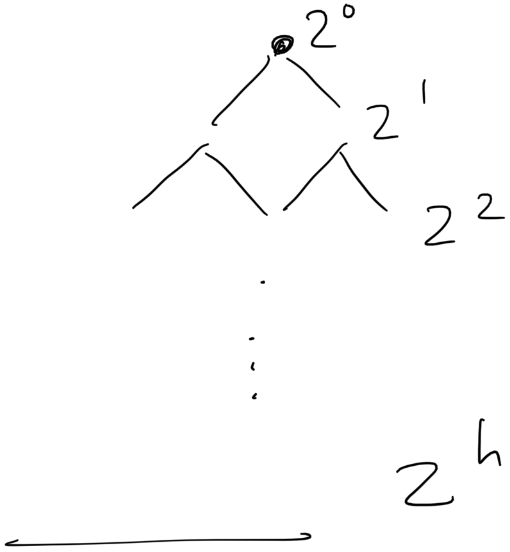


Picture from [2]

Binary Tree

Given a binary tree of height *h*, what is the maximum number of leaves it can have?

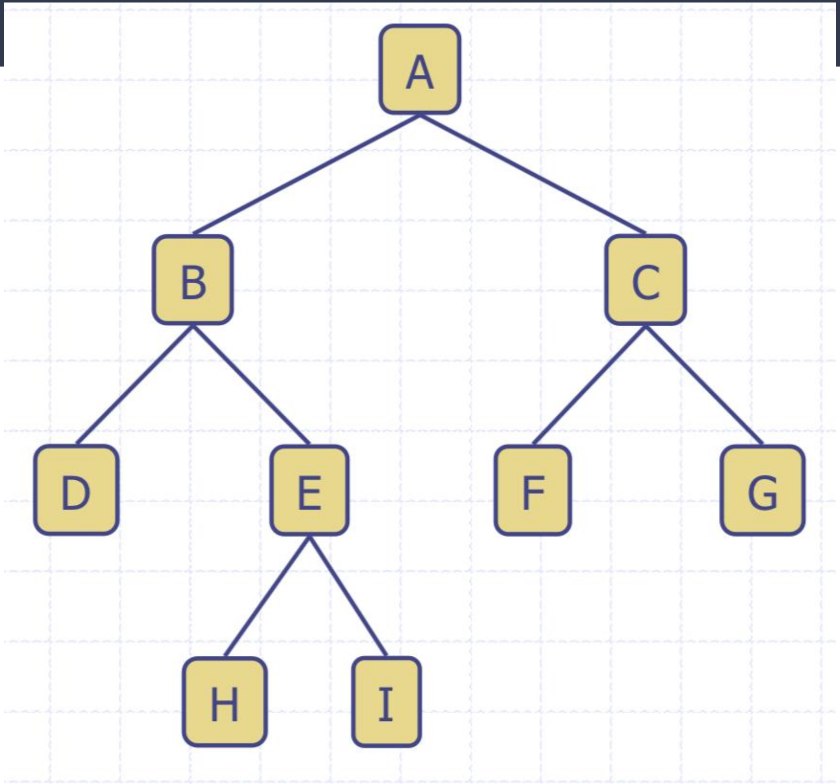
$$2^h$$



Full Binary Tree

- Every node other than the leaves has two children.
- Is this a full binary tree?

yes

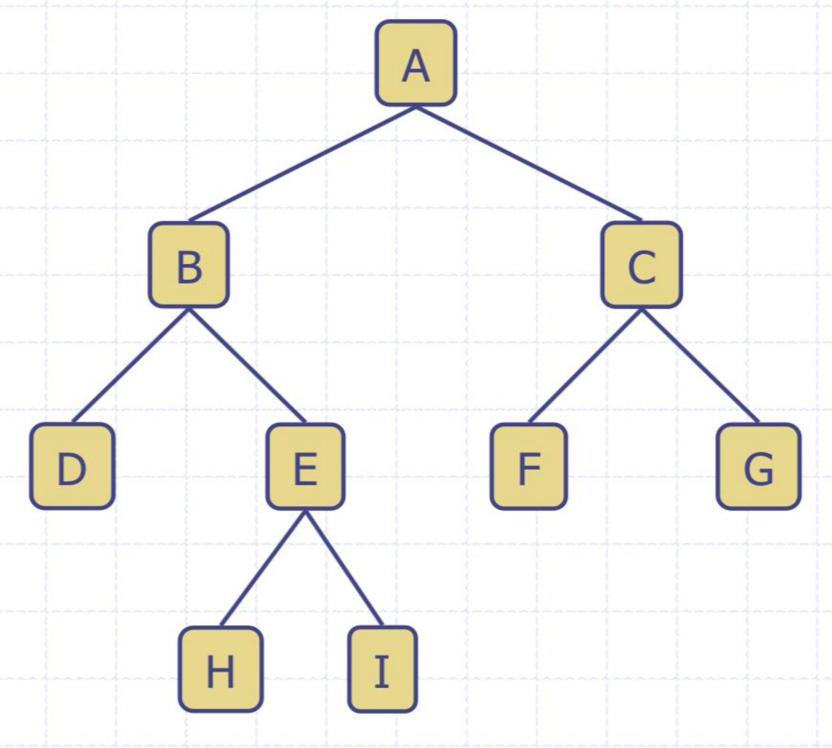


Picture from [2]

Properties of a full binary tree...

Given ***e*** external nodes, ***i*** internal nodes, ***n*** total nodes, and ***h*** the height of the tree:

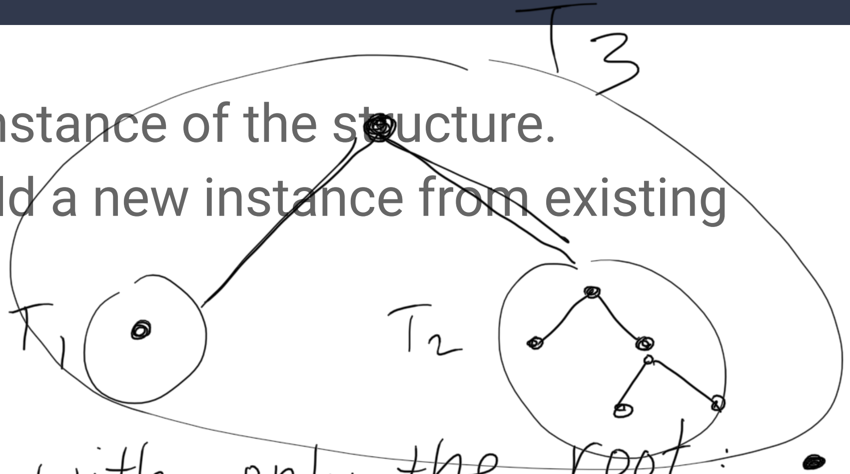
- $e = i + 1$
- $e <= 2^h$
- $n = 2e - 1$
- $h <= i$
- $h <= (n-1)/2$
- $h >= \log_2 e$
- $h >= \log_2(n + 1) - 1$



Picture from [2]

Recursive Definitions of Structures

- Base Case: Consider the smallest instance of the structure.
- Inductive Step: Consider how to build a new instance from existing smaller instances.



Example (full binary tree):

Base Case: A tree with only the root:

Inductive Step: Assume T_1 and T_2 are FBTs.

Then $T_3 = \underbrace{T_1 \bullet T_2}_{\text{join } T_1 \text{ and } T_2 \text{ w/ a new root.}}$ is a FBT.

Definition. Give a recursive definition for the total number of external nodes $e(T)$ of a full binary tree.

Definition. Give a recursive definition for the total number of external nodes $e(T)$ of a full binary tree.

Basis Step. When T is just one node, then $e(T) = 1$.
Inductive Step. Let T_1 and T_2 be full binary trees. You can form a new binary tree T_3 by making a new node and letting T_1 and T_2 be children of that node, which would then be the root of the new tree.

Then $e(T_3) = e(T_1) + e(T_2)$.

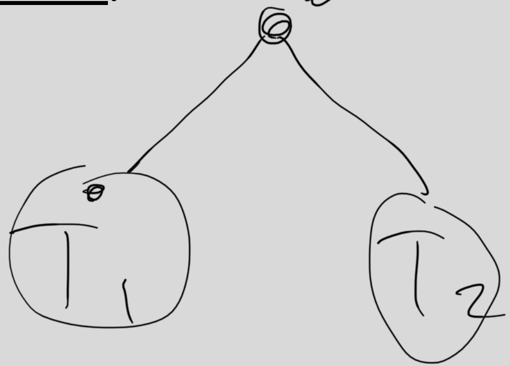
Definition. Give a recursive definition for the the number of internal nodes $i(T)$ of a full binary tree.

Definition. Give a recursive definition for the the number of internal nodes $i(T)$ of a full binary tree.

Basis Step. When T is just one node, then $i(T) = 0$.
Inductive Step. Let T_1 and T_2 be full binary trees. You can form a new binary tree T_3 by making a new node and letting T_1 and T_2 be children of that node, which would then be the root of the new tree.

Then $i(T_3) = i(T_1) + i(T_2) + 1$.

+1



Definition. Give a recursive definition for the total number of nodes $n(T)$ of a full binary tree.

Definition. Give a recursive definition for the total number of nodes $n(T)$ of a full binary tree.

Basis Step. When T is just one node, then $n(T) = 1$.

Inductive Step. Let T_1 and T_2 be full binary trees. You can form a new binary tree T_3 by making a new node and letting T_1 and T_2 be children of that node, which would then be the root of the new tree.

Then $n(T_3) = n(T_1) + n(T_2) + 1$.

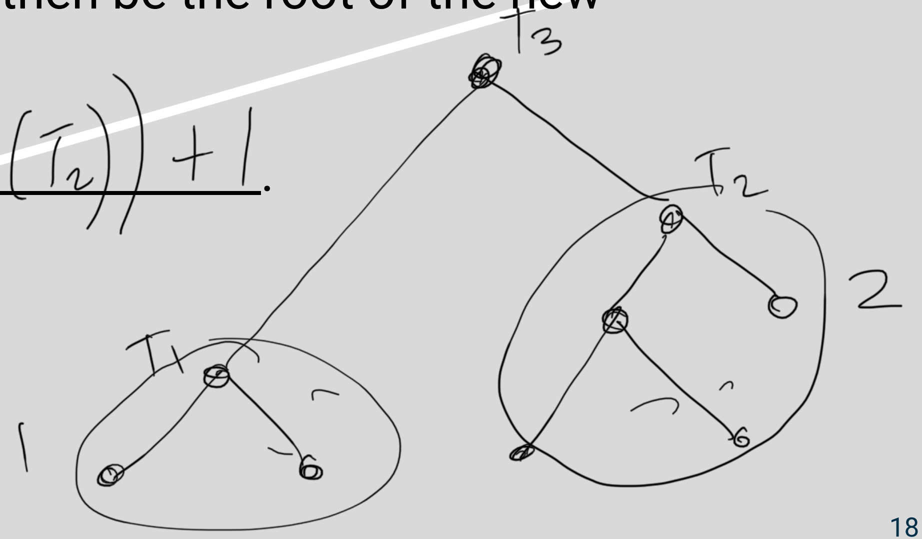
Definition. Give a recursive definition for the height $h(T)$ of a full binary tree.

Definition. Give a recursive definition for the height $h(T)$ of a full binary tree.

Basis Step. When T is just one node, then $h(T) = 0$.

Inductive Step. Let T_1 and T_2 be full binary trees. You can form a new binary tree T_3 by making a new node and letting T_1 and T_2 be children of that node, which would then be the root of the new tree.

Then $h(T_3) = \max(h(T_1), h(T_2)) + 1$.



Proof. Prove by structural induction that a full binary tree with *e* external nodes has $2e-1$ total nodes.

Basis Step. Let T_0 be an FBT with just one node.
Then $n(T_0) = 1$ and $e(T_0) = 1$ and $1 = 2(1) - 1 = 1$.

Inductive Step. Let T_1 and T_2 be FBTs and $T_3 = T_1 \cdot T_2$.
Assume that $n(T_1) = 2e(T_1) - 1$ and $n(T_2) = 2e(T_2) - 1$.

$$n(T_3) = n(T_1) + n(T_2) + 1$$
$$= 2e(T_1) - 1 + 2e(T_2) - 1 + 1 \quad \text{by the IH}$$
$$= 2e(T_1) + 2e(T_2) - 1$$
$$= 2(e(T_1) + e(T_2)) - 1$$
$$= 2e(T_3) - 1$$

Proof. Prove by structural induction that if T is a full binary tree, then $n(T) <= 2^{h(T)+1} - 1$.

Basis Step. Let T_0 be an FBT with just one node.
Then $n(T_0) = 1$ and $h(T_0) = 0$ and $1 \le 2^{0+1} - 1 = 1$.

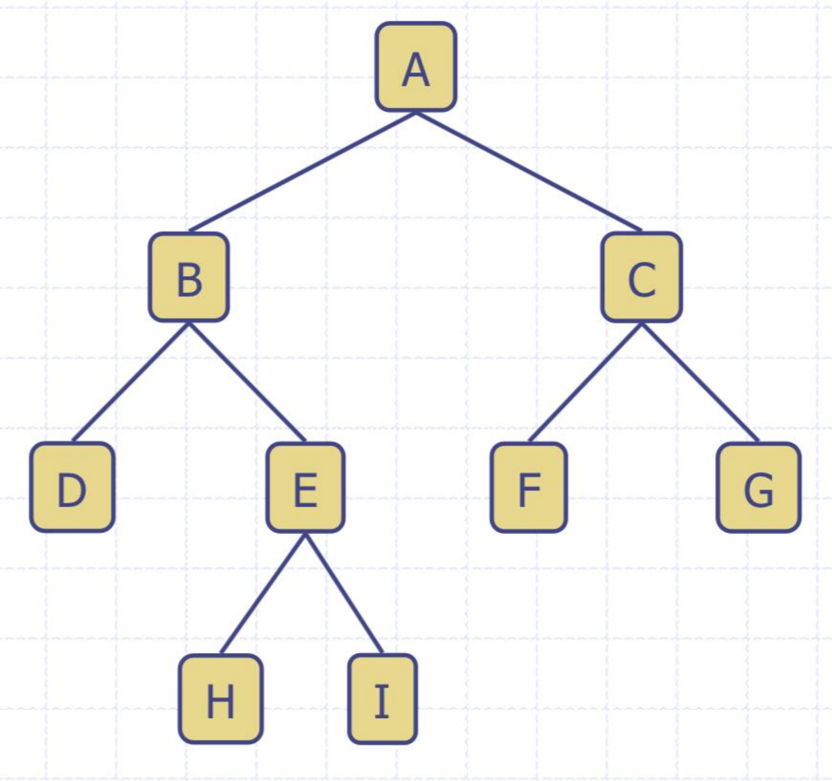
Inductive Step. Let T_1 and T_2 be FBTs and $T_3 = T_1 \cdot T_2$.
Assume that $n(T_1) \le 2^{h(T_1)+1} - 1$ and $n(T_2) \le 2^{h(T_2)+1} - 1$.

$$
\begin{aligned}
n(T_3) &= n(T_1) + n(T_2) + 1 \\
&\le 2^{h(T_1)+1} - 1 + 2^{h(T_2)+1} - 1 + 1 \quad \text{by the IH} \\
&= 2^{h(T_1)+1} + 2^{h(T_2)+1} - 1 \\
&\le 2 \cdot 2^{\max(h(T_1), h(T_2))+1} - 1 \\
&= 2 \cdot 2^{h(T_3)} - 1 \\
&= 2^{h(T_3)+1} - 1
\end{aligned}
$$

Complete Binary Tree

- Each level is completely filled up before going on to the next level.
- Is this a complete binary tree?

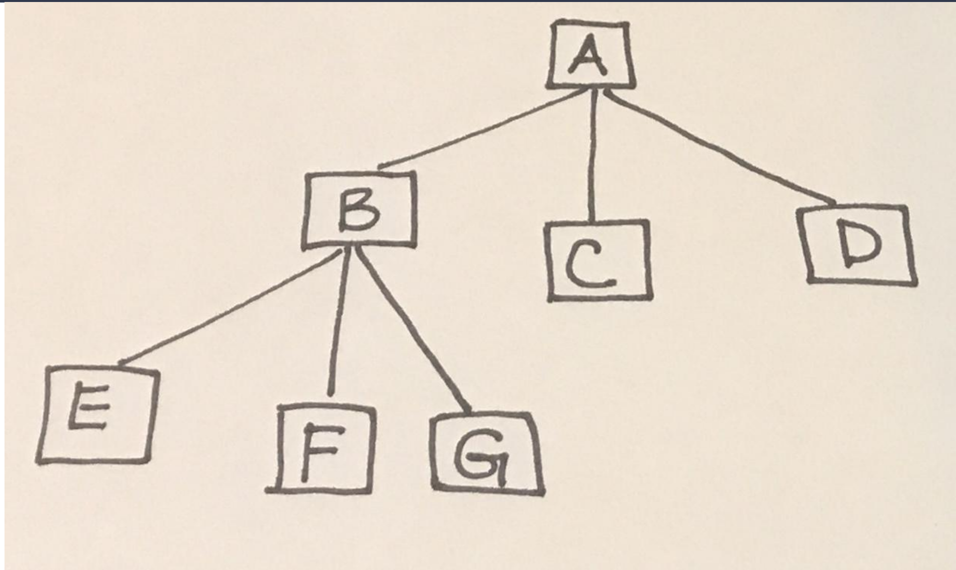
Yes



Picture from [2]

Ternary Tree

Every node has at most 3 children



Ternary Tree

Given a ternary tree of height h, what is the maximum number of leaves it can have?

3^h

Ternary Tree

Given a complete (meaning a level is complete before moving on to the next) ternary tree of height 7, how many internal nodes are there at level 3?

$3^3 = 27$

Ternary Tree

Given a complete (meaning a level is complete before moving on to the next) ternary tree with N external nodes, what is the maximum number of nodes the tree can have?

$$\sum_{k=0}^{\log_3 N} 3^k = \frac{3N-1}{2}$$

Basic Tree Traversals

- Preorder: Parents before Children—a node is visited before its descendants
- Postorder: Children before Parents—a node is visited after its descendants
- Inorder: A node is visited after its left subtree and before its right subtree
- DFS & BFS: These algorithms will be covered when we cover graphs

Preorder Traversal

Algorithm *preOrder*(v)

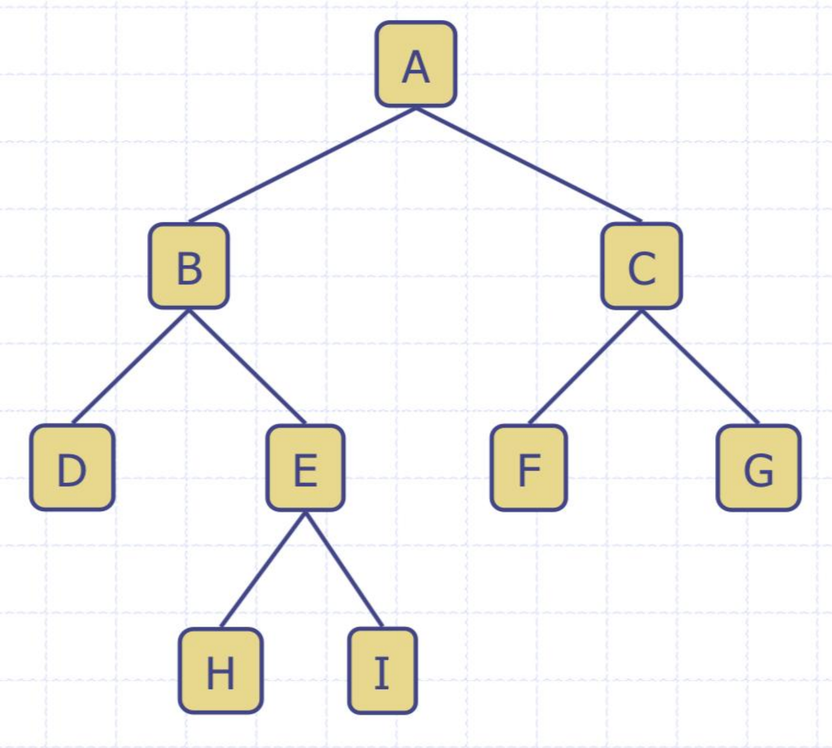
visit(v)

for each child w of v

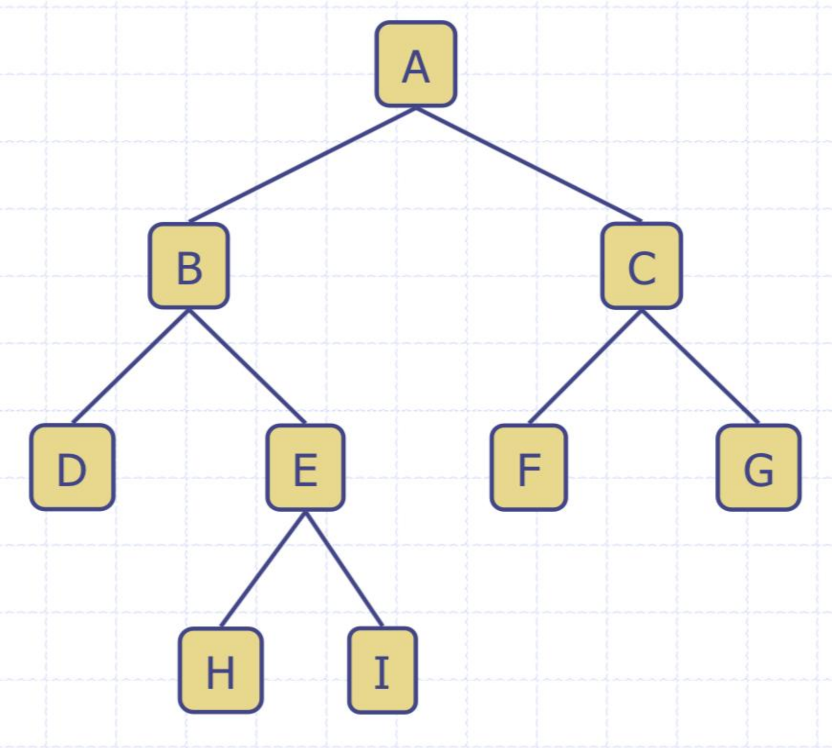
preorder (w)

Picture from [2]

Example: What's the pre-order traversal of this tree?



Example: What's the pre-order traversal of this tree?



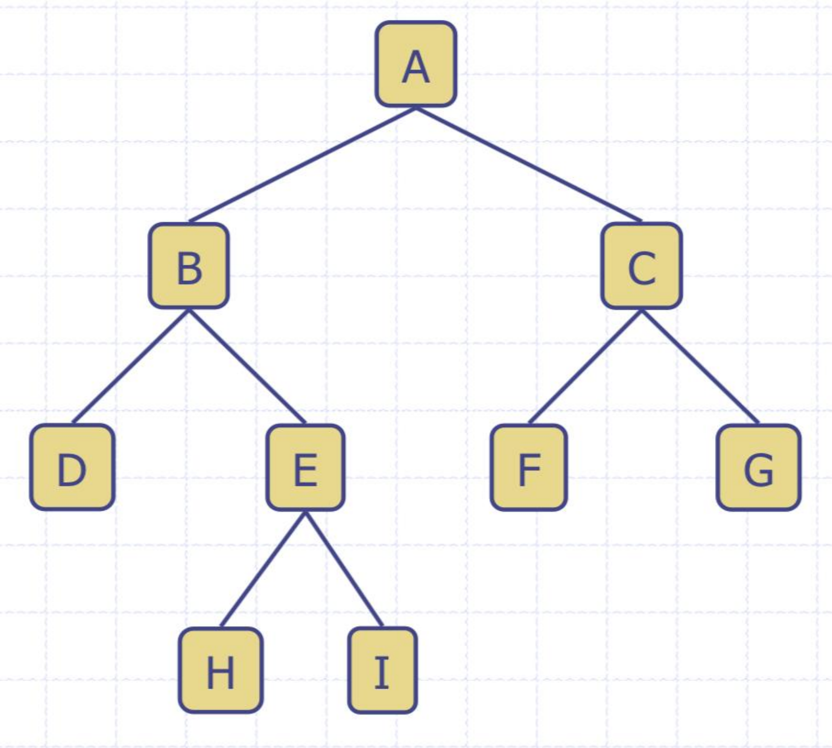
A-B-D-E-H-I-C-F-G

Postorder Traversal

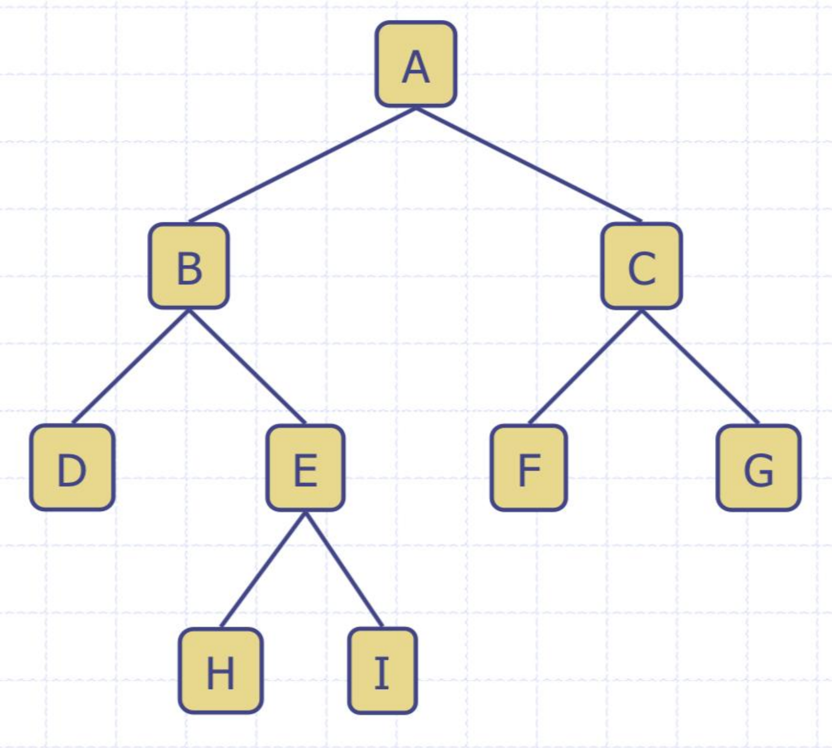
```
Algorithm postOrder(v)
    for each child w of v
        postOrder (w)
    visit(v)
```

Picture from [2]

Example: What's the post-order traversal of this tree?



Example: What's the post-order traversal of this tree?



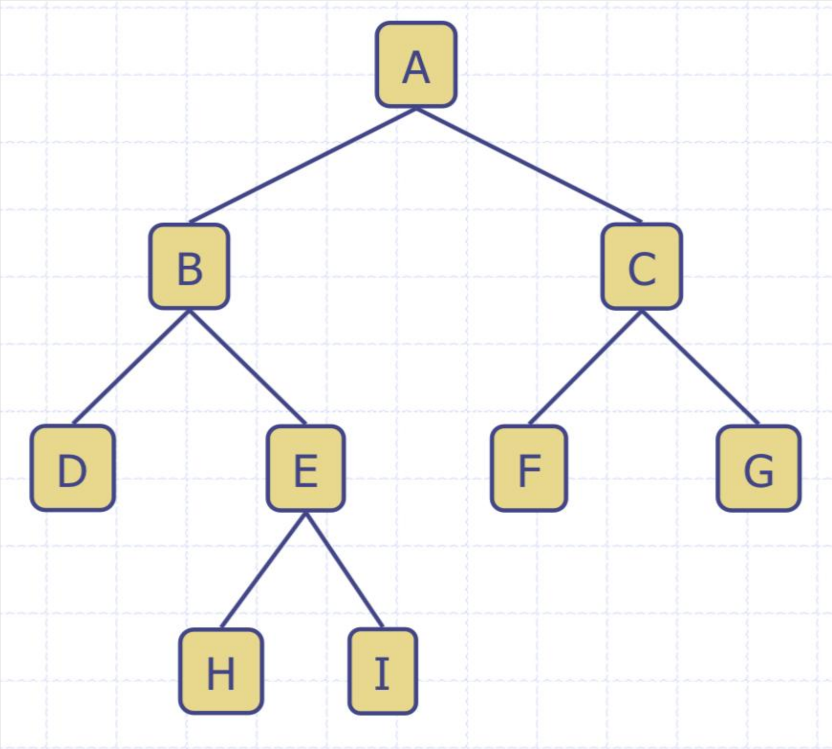
D-H-I-E-B-F-G-C-A

Inorder Traversal

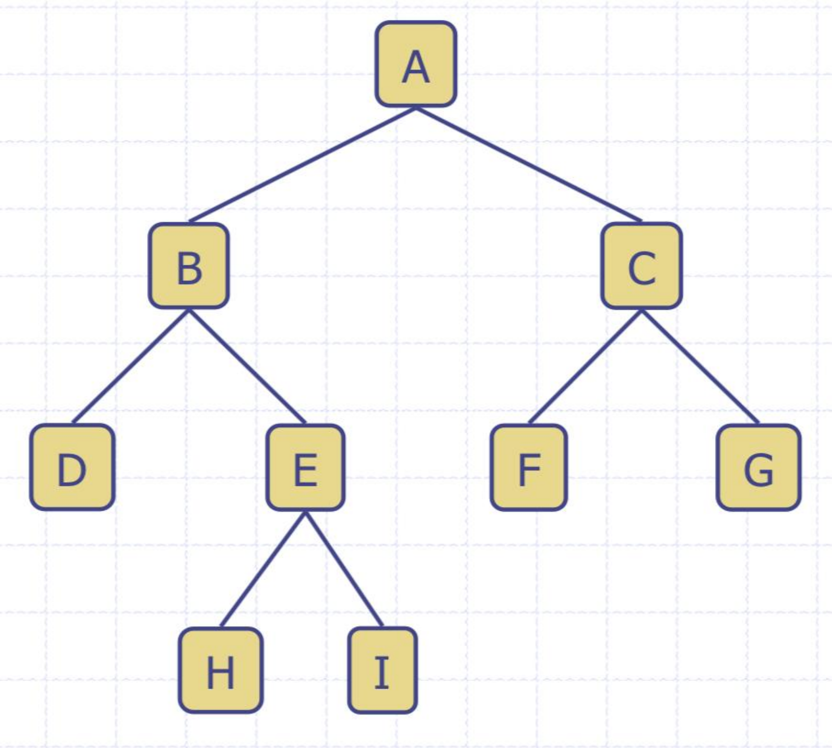
```
Algorithm inOrder(v)
    if isInternal (v)
        inOrder (leftChild (v))
    visit(v)
    if isInternal (v)
        inOrder (rightChild (v))
```

Picture from [2]

Example: What's the in-order traversal of this tree?

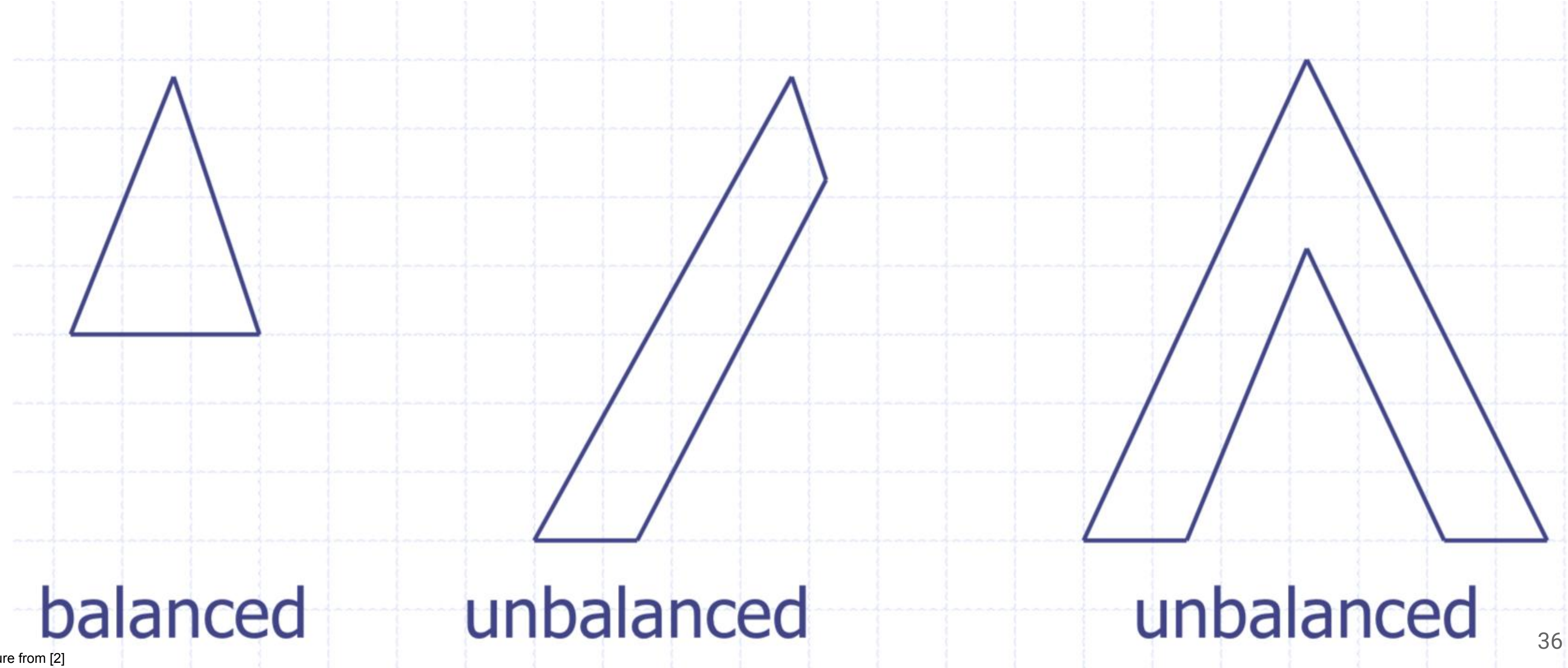


Example: What's the in-order traversal of this tree?



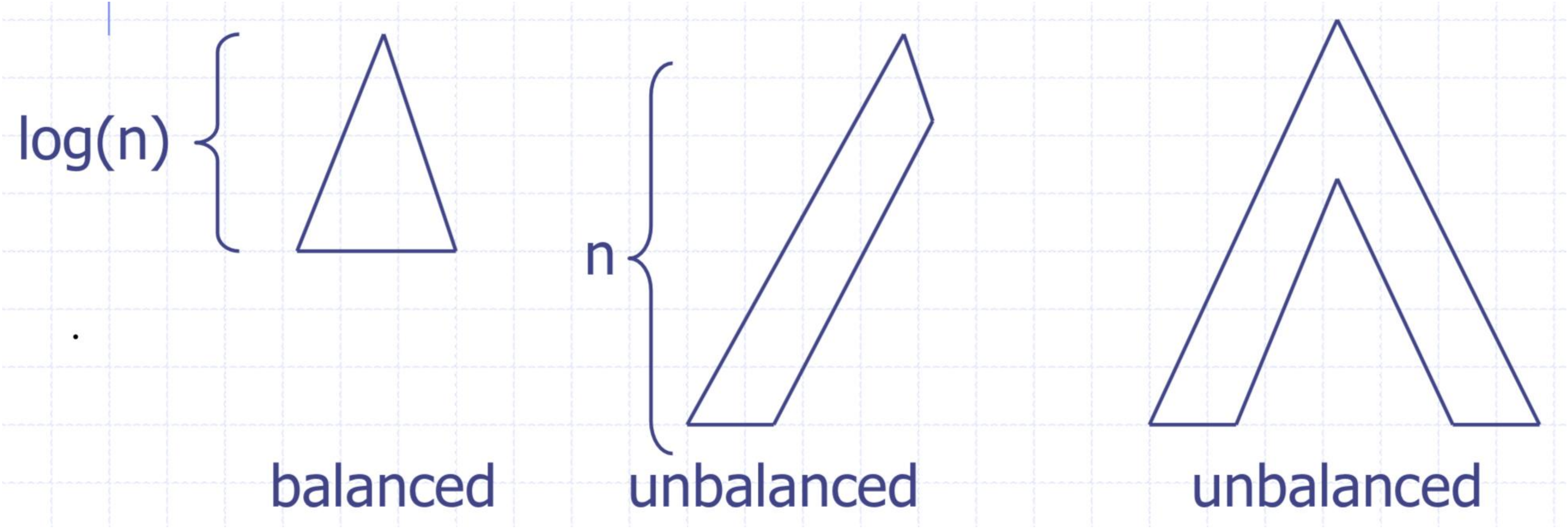
D-B-H-E-I-A-F-C-G

The Importance of Balance: How tall are these trees given *n* nodes?



Picture from [2]

The Importance of Balance: How tall are these trees given *n* nodes?



Why does it matter?

Picture from [2]

The Importance of Balance: Why does it matter?

- The operations associated with tree data structures often depend on traversing a path from root to leaf.
- If that is the case, then the runtime of such an operation will be proportional to the height of that tree (i.e. the maximum-length path from root to leaf).
- If the tree is unbalanced, that may be O(N).
- If the tree is balanced, that is only O(logN).

References

[1] *Algorithms, Fourth Edition;* Robert Sedgewick and Kevin Wayne (and associated slides)
[2] *Data Structures & Algorithms in Java,* Goodrich & Tamassia (and associated slides)