SCC120 Fundamentals of Computer Science
Unit 7: Trees (Terminology)

Jidong Yuan
yuanjd@bjtu.edu.cn

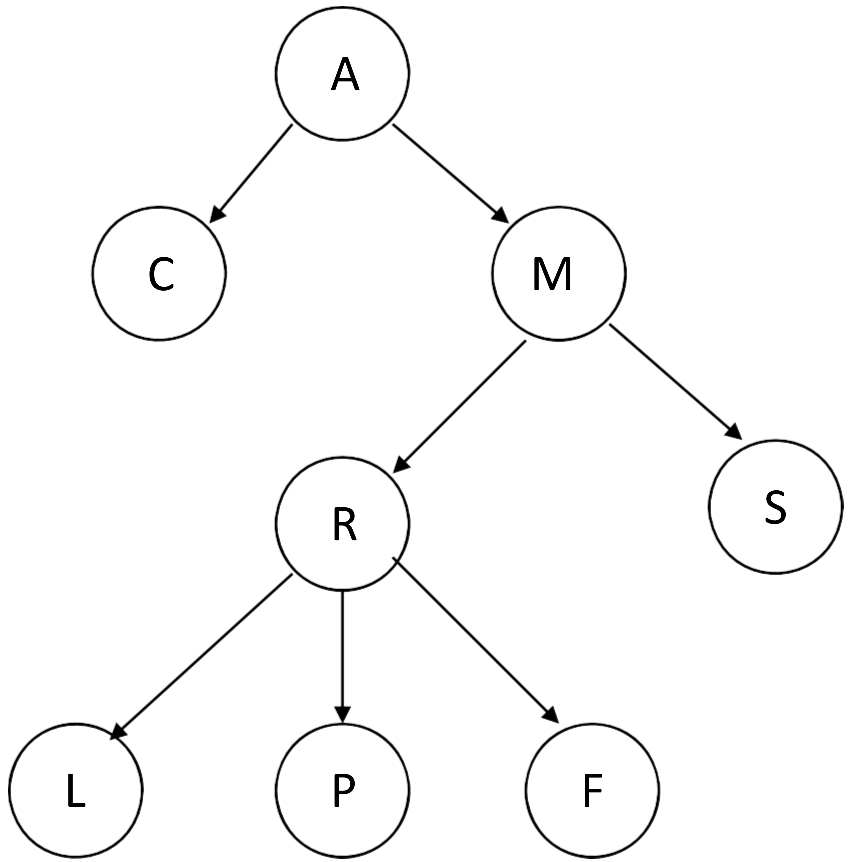
Overview

- Concept of a Tree
- Examples of Trees
- Terminology/Definitions

A Tree
(we’ve seen this when we looked at graphs)

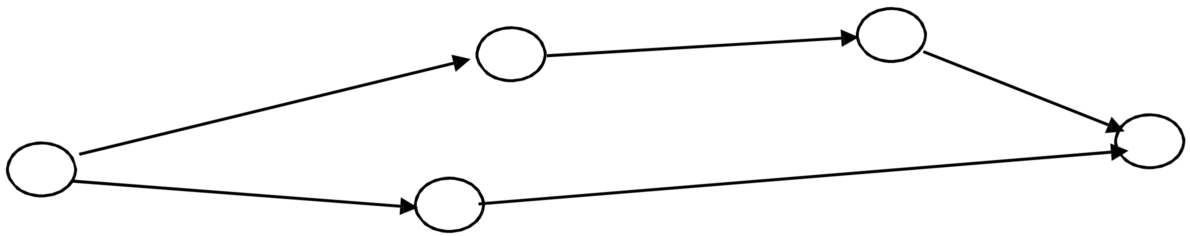
- A certain engineering company A is divided into a consultancy division C and a manufacturing division M
- The manufacturing division M is divided into a railway section R and a marine engine section S
- Section R is divided into three departments, building locomotives (L), passenger coaches (P), and freight wagons (F)

A Tree

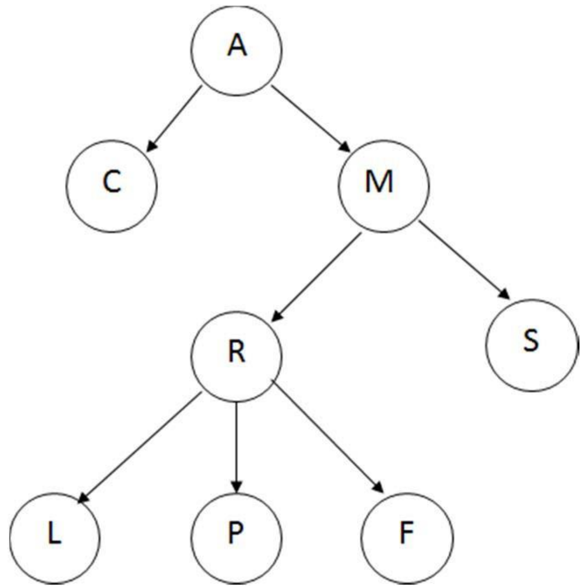


Graphs vs. Trees

- A directed or undirected graph:
  - has a set of **nodes**
  - and a set of **edges** connecting these nodes
  - may have loops

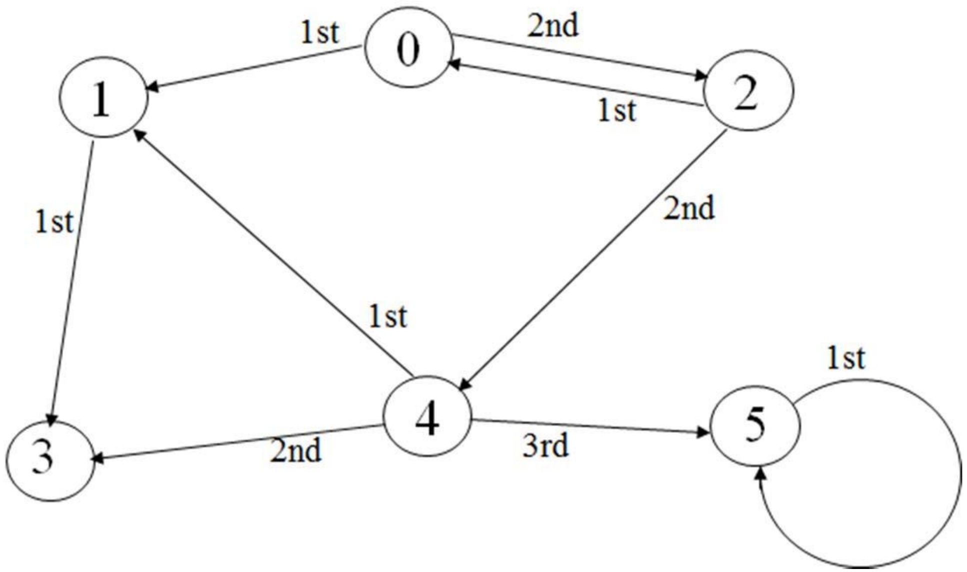


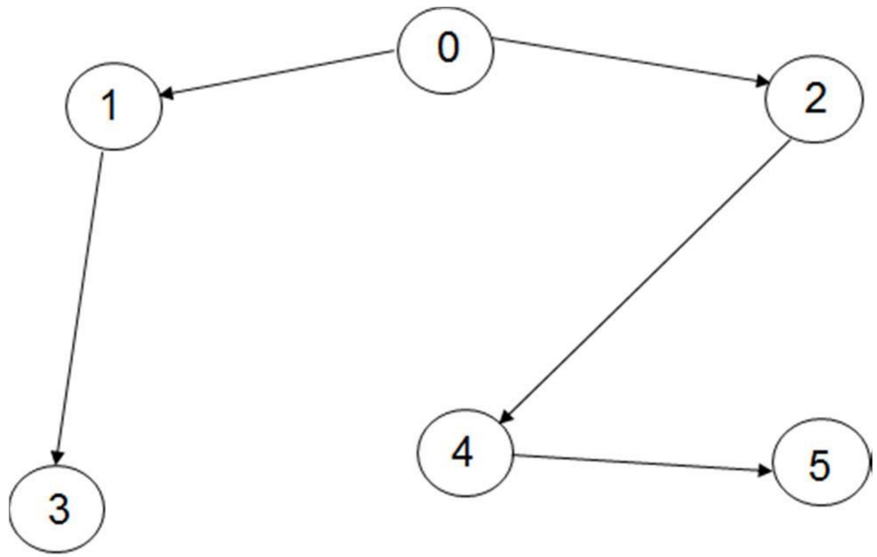
- A tree:
  - also has a set of nodes and edges
  - there are **no loops**
  - well-structured in terms of **levels** of a tree



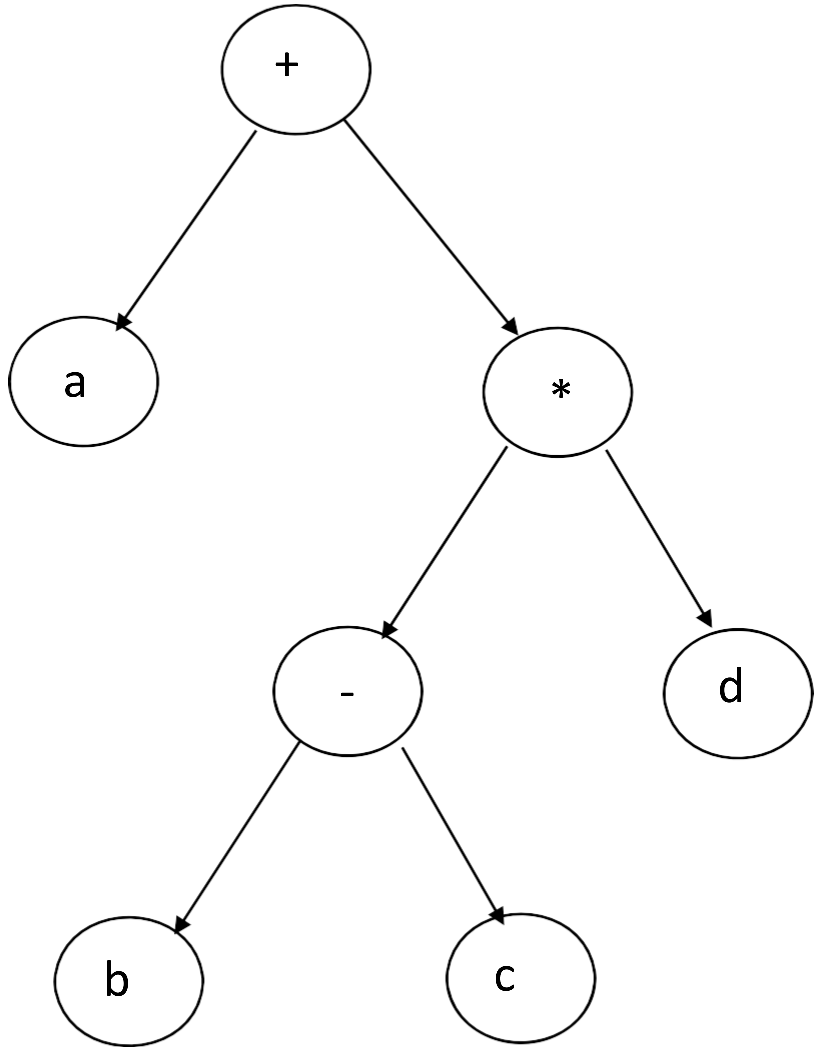
Graphs vs. Trees

- If we take a connected graph and delete edges until no loops remain, but the structure is still connected, we obtain a **spanning tree** of the graph

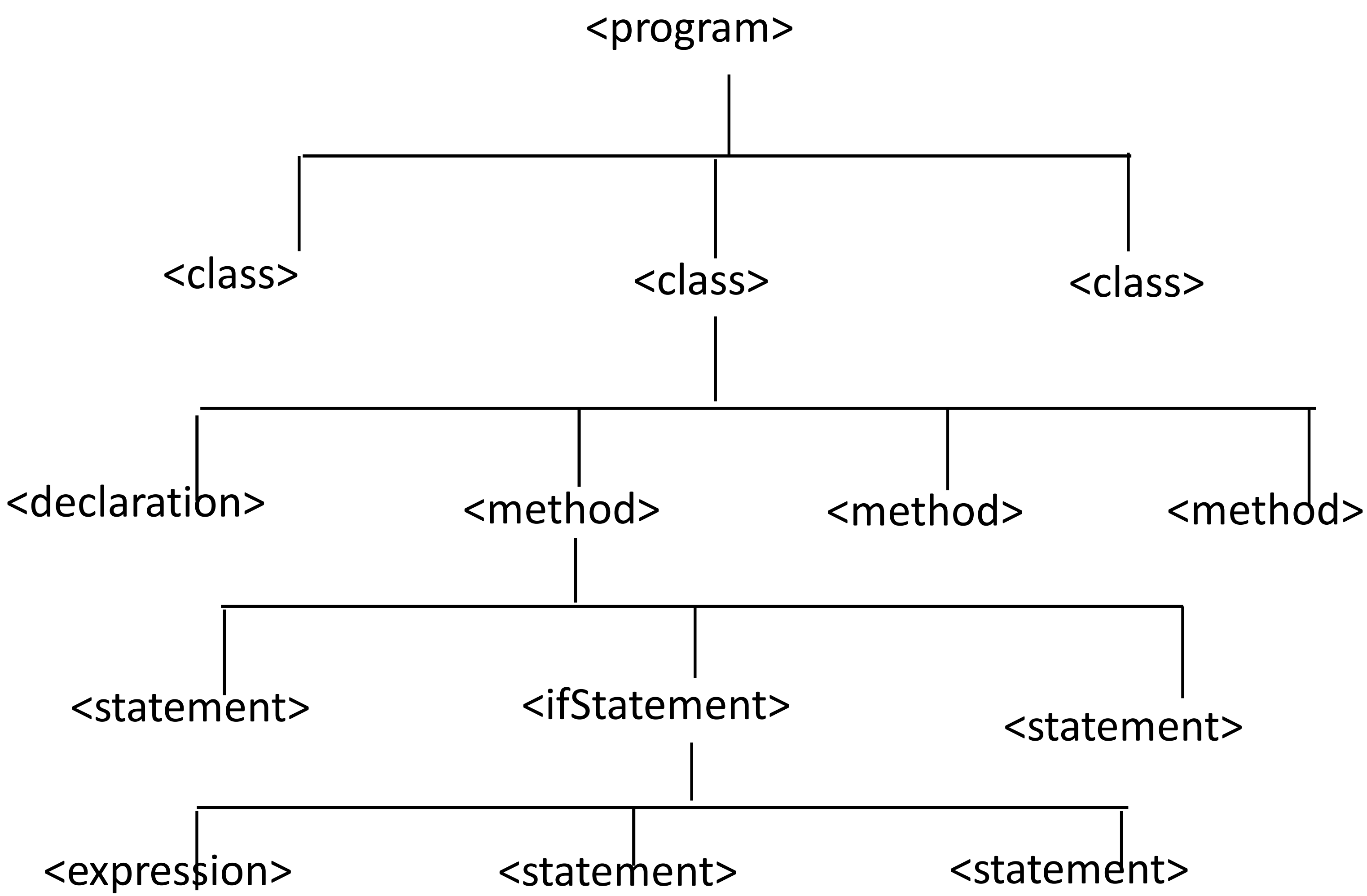




Examples of Trees: (Arithmetic) Expressions



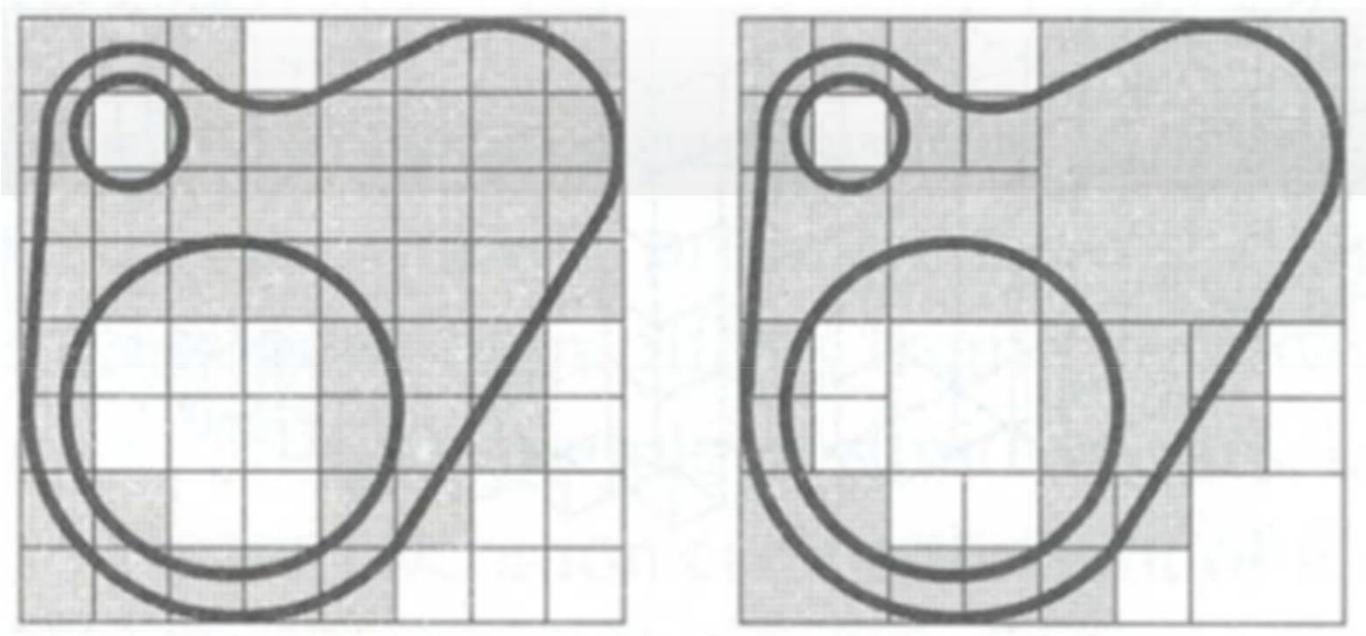
Examples of Trees: A Parse Tree



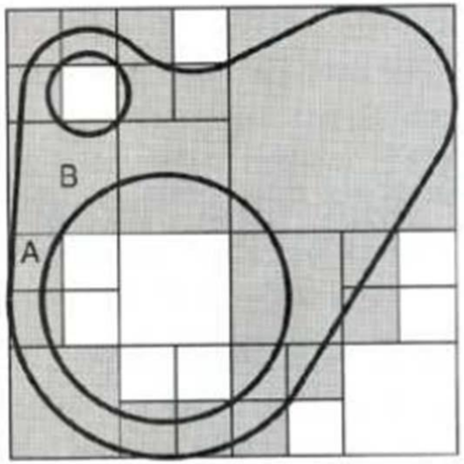
Examples of Trees: A Document

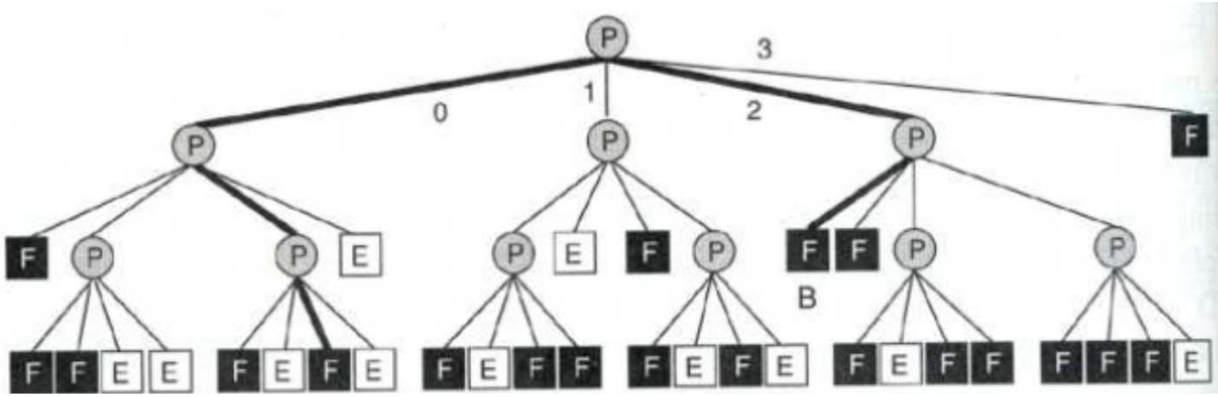
- A book, article, etc.
  - document = {frontMatter, body, backmatter}
  - body = {section, section, section …}
  - section = {chapter, chapter, chapter, … }
  - chapter = {title, para, para, para, …}
  - para = {sentence, sentence, sentence, …}
  - sentence = {word, word, word, …, punctuation}

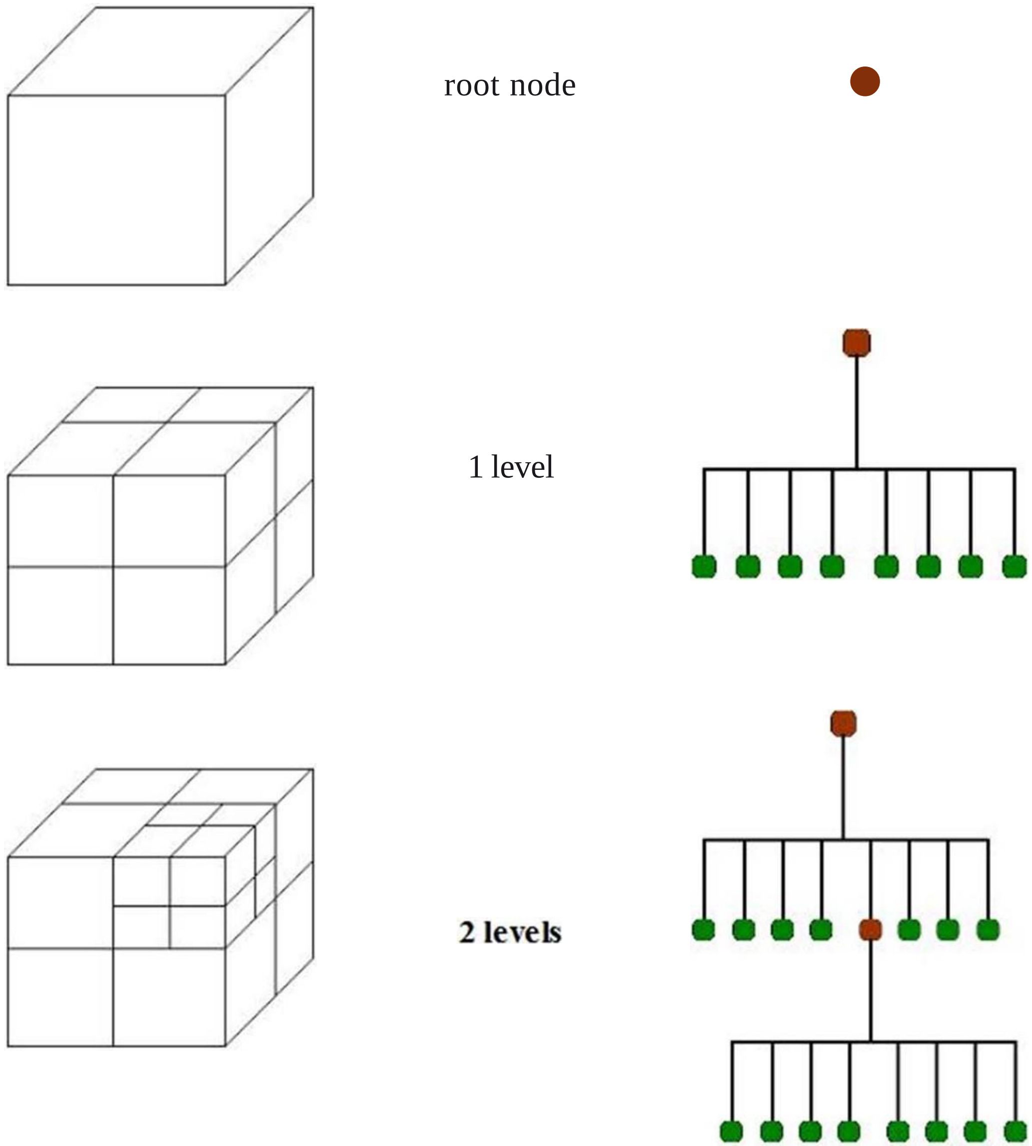
Examples of Trees: QuadTrees

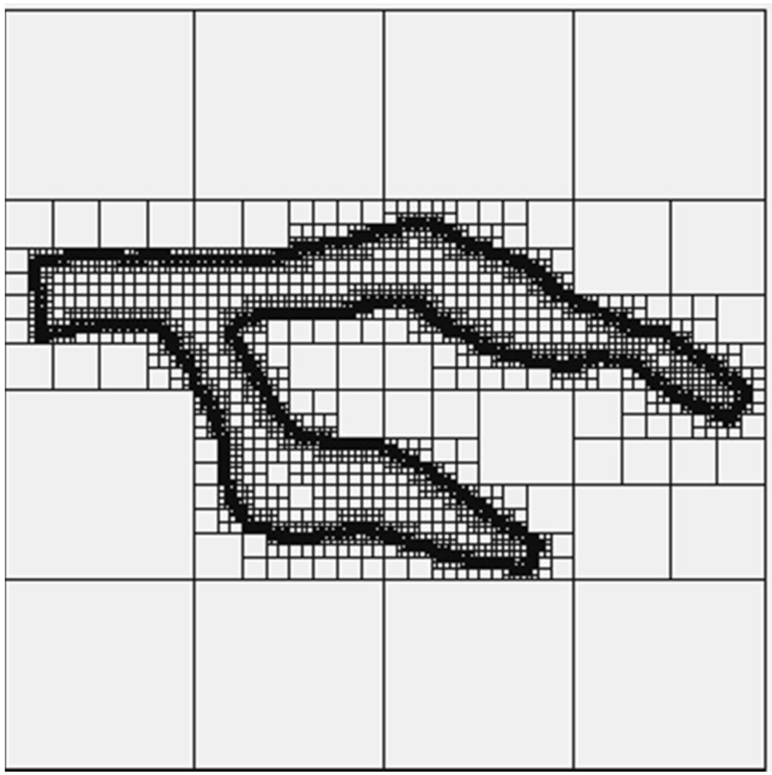


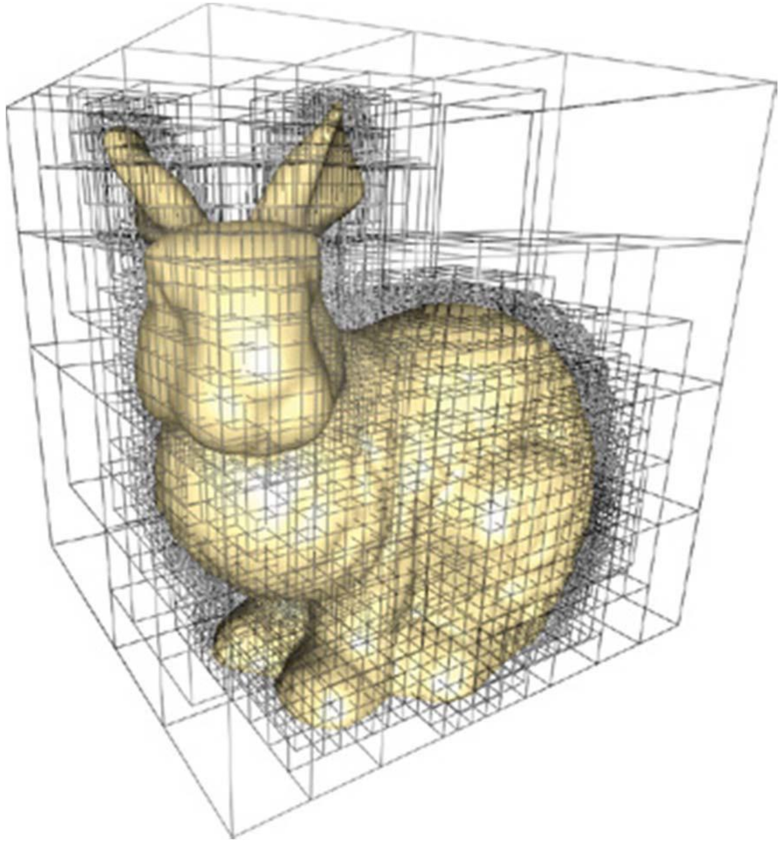
Uniform Voxels Quadtree


B
A


P
0
1
2
3
P
P
P
F
F
P
P
E
P
E
F
P
F
F
P
P
B
F F E E
F E F E
F E F F
F E F E
F E F F
F F F E

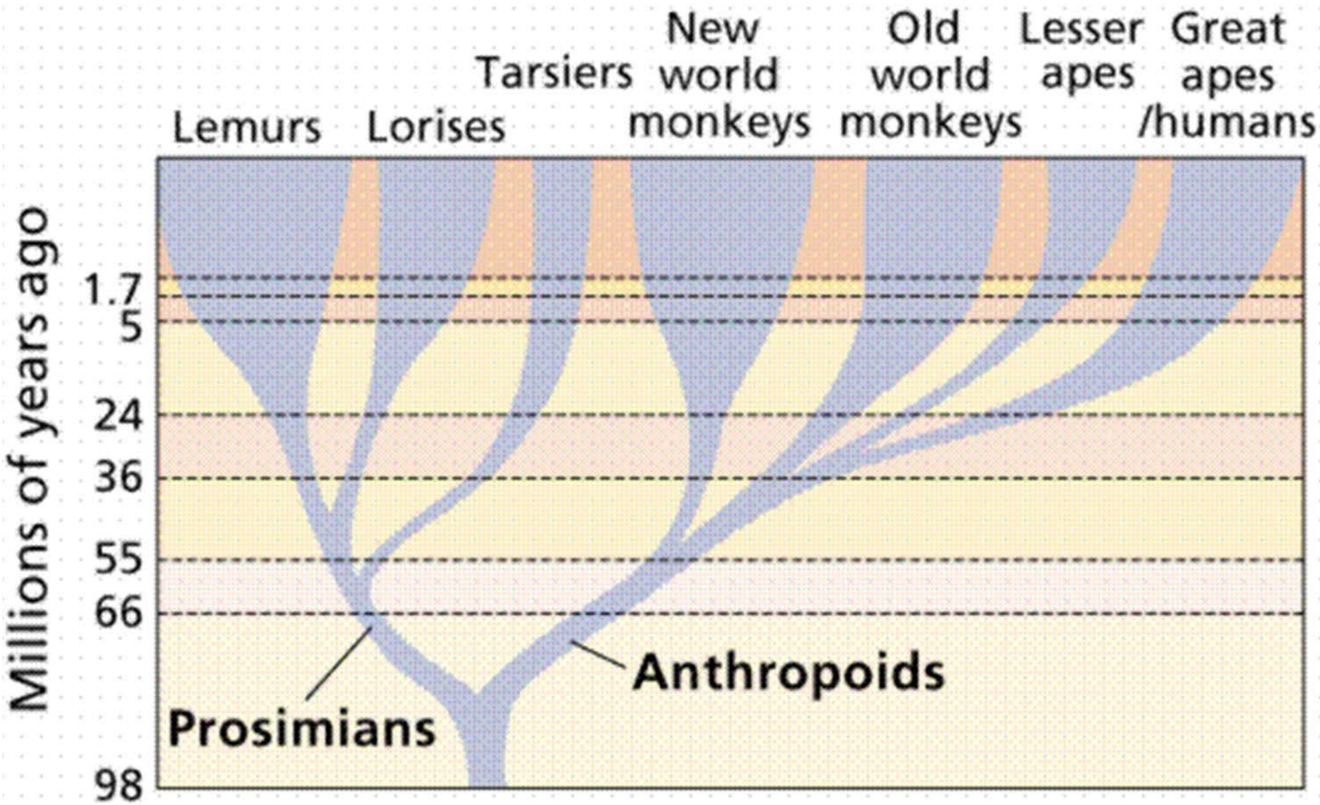

root node
1 level
2 levels





Examples of Trees: Evolutionary Tree

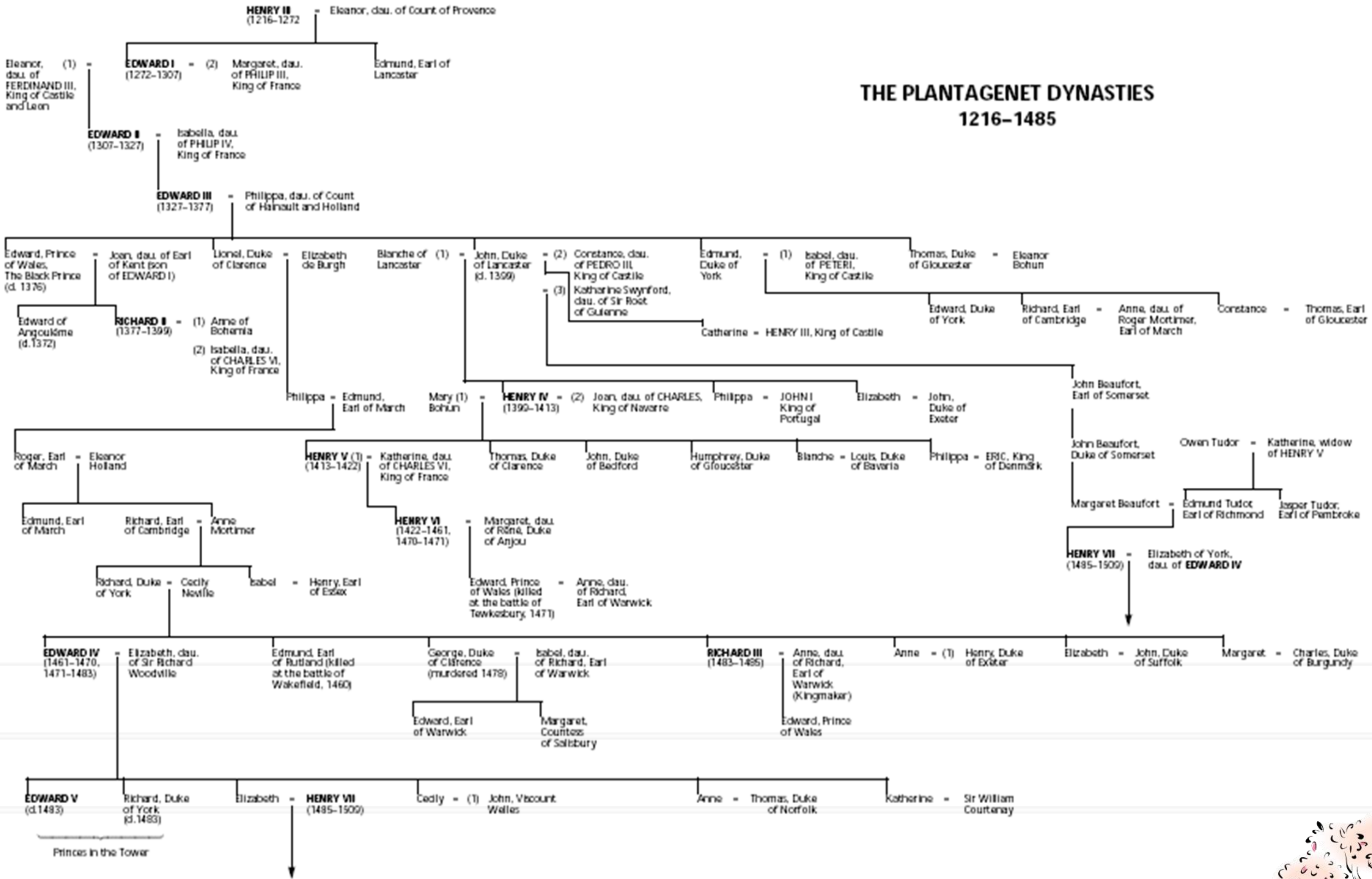
Example of an evolutionary tree (conventionally with the root at the bottom)



Examples of Trees: Family Trees

- Simple family trees are trees in the Computer Science sense
  - But marriage of cousins, etc., makes it a graph
  - The line of descent of the English kings and queens is close to being a tree in most representations

A Family Tree

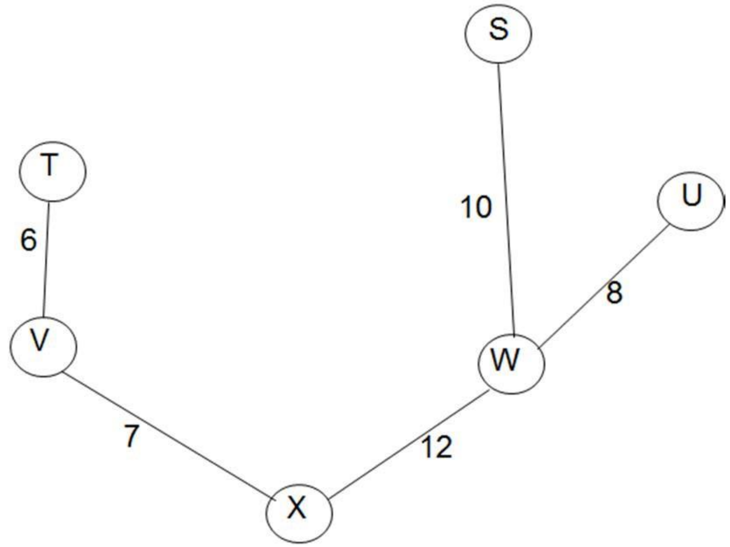


Overview

- Concept of a Tree
- Examples of Trees
- Terminology/Definitions

Non-Rooted vs. Rooted Trees

- A non-rooted tree is a connected non-directed graph without any loops
  - Example: Minimal Spanning Tree



- A rooted tree has a "root" node, at the "top" level of the tree
  - We focus on rooted trees (i.e. "tree" means rooted tree)

Definitions of Trees

- A tree is a node pointing to zero or more distinct trees
  - This definition is recursive
  - A tree contains smaller trees within it
- This definition also emphasizes the thought that a rooted tree represents a *hierarchy* of objects
  - Like the structure of a company

Definitions of Trees

- A tree is a directed graph containing one node of in-degree 0, and zero or more other nodes of in-degree 1
  - This focuses on the nodes of a tree
  - The node of in-degree zero is the *root* node

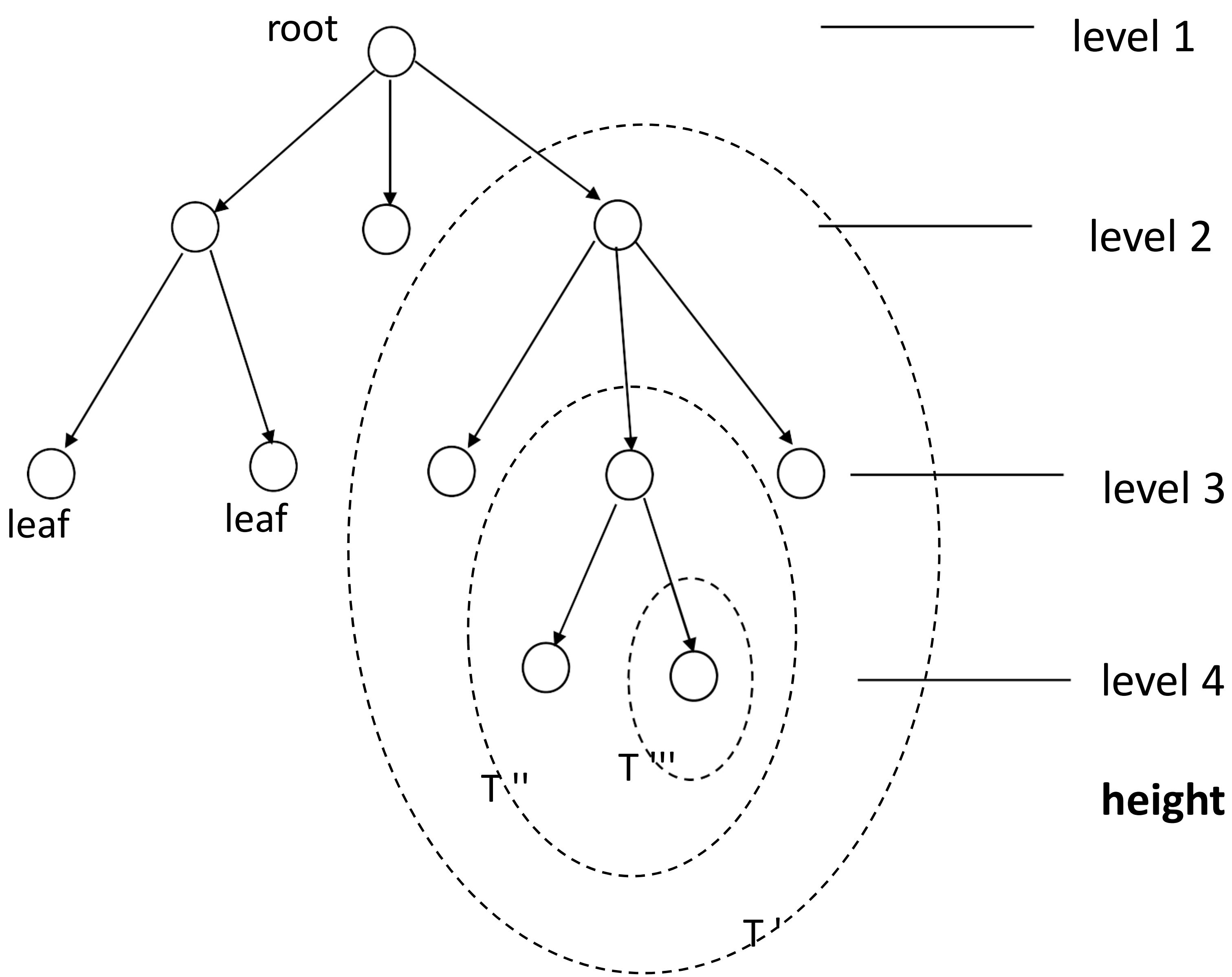
Empty Tree

- We can also define a tree which contains no nodes at all
- So we might want our definition on previous slide to start with “A tree is either empty with no nodes and no edges, or …”

Terminology

- A node of in-degree zero is called the *root* node of the tree
- Nodes of out-degree zero are called *leaf* nodes
- A node which is neither a root nor a leaf is called an *internal node*

An Example Tree T



height of tree = 3

Comments on the Tree T

- The *levels* of the nodes in the tree defined by 1 + (the number of connections between the node and the root).
- The *height* of a **node** is the number of edges on the longest path between that node and a leaf.
- The *height* of the tree is the height of its root node.
- The *depth* of a node is the number of edges from the tree's root node to the node.

Comments on the Tree T

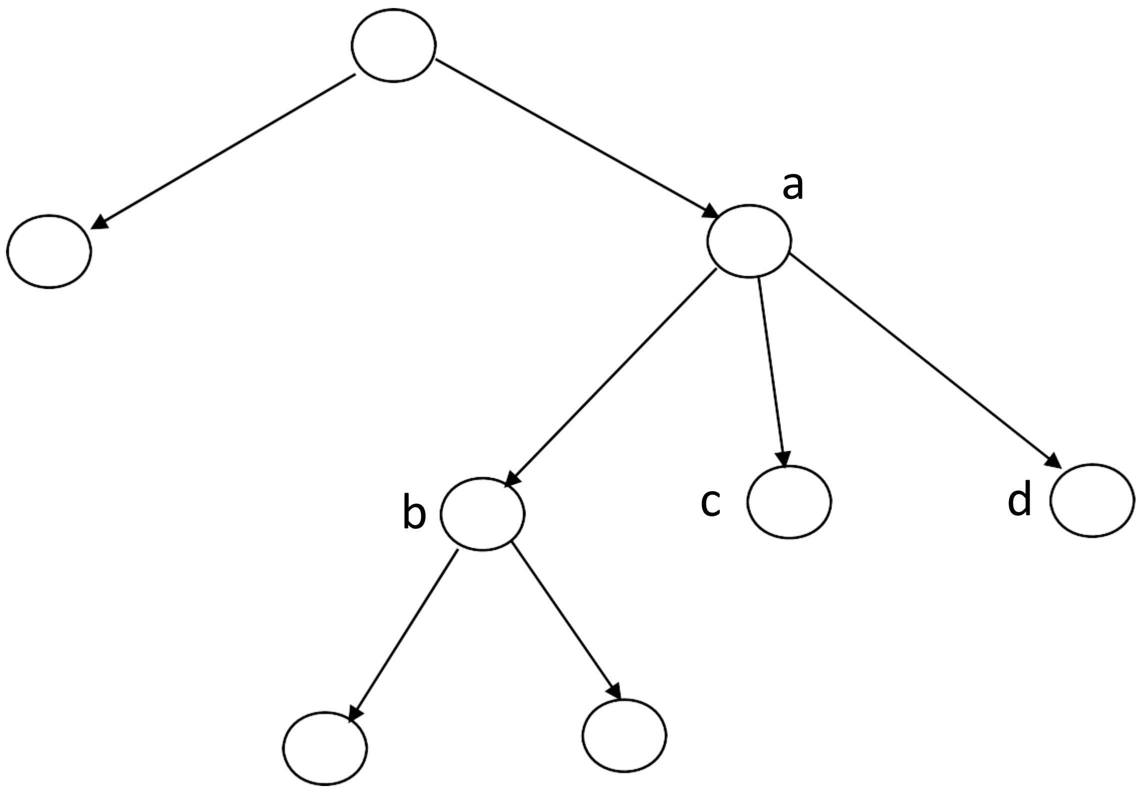
- A *subtree* consists of a node X from/within a tree, together with all the nodes and edges below X (if any)
- X is the *root* of the subtree
- In our Tree T:

  T' is a subtree of T

  T'' is a subtree of T'

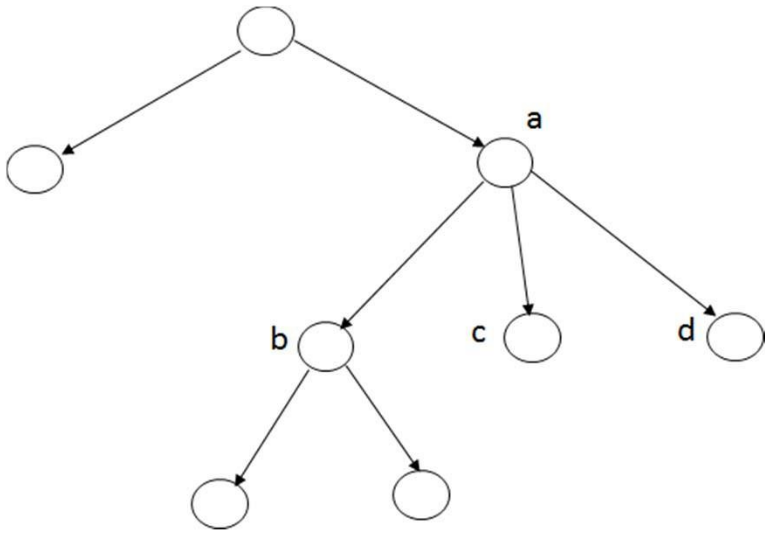
  T''' is a subtree of T''

An Alternative Metaphor

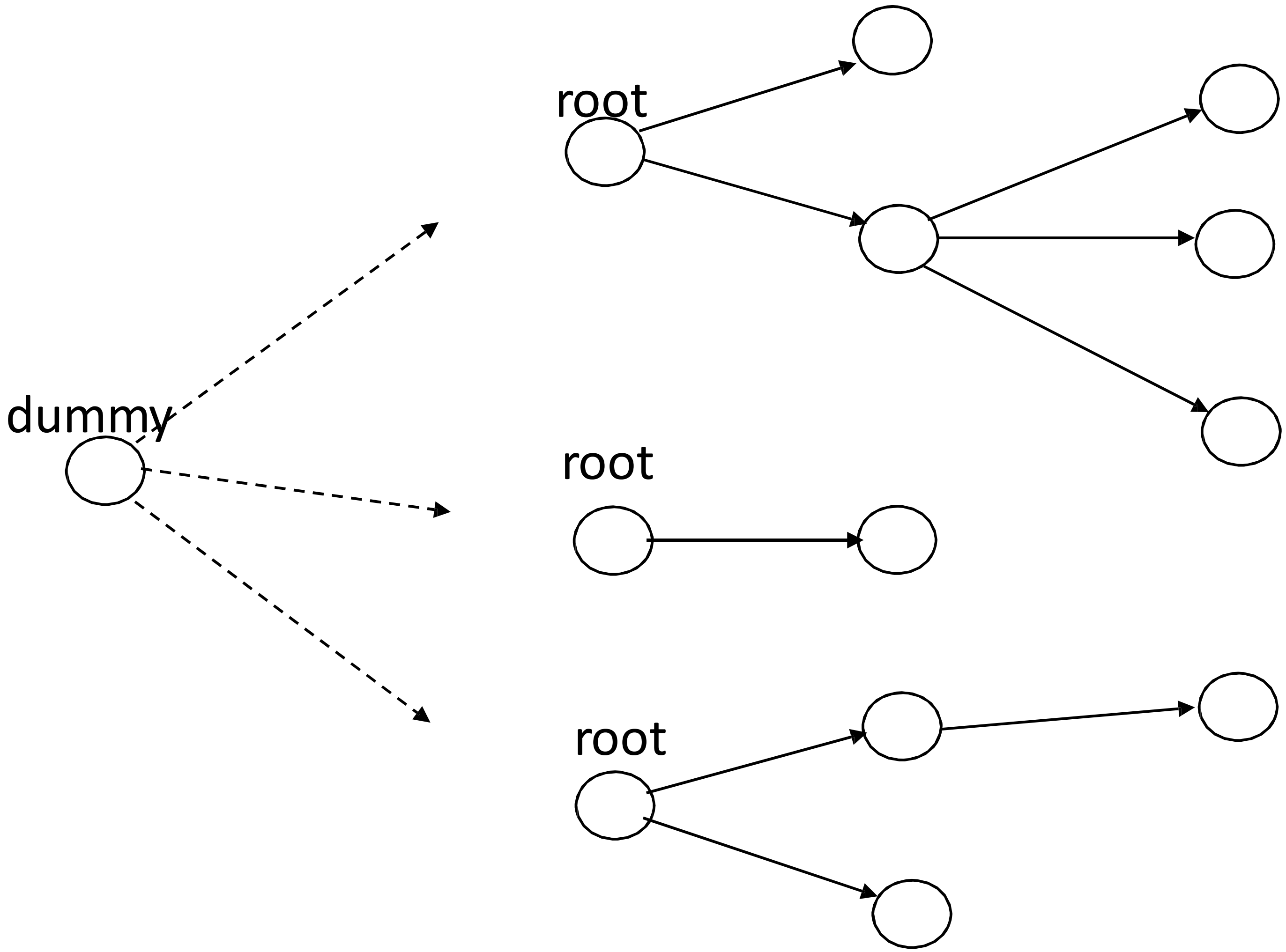


An Alternative Metaphor

- Instead of metaphors based on real trees (e.g. root and leaf), we can use family metaphors
- Node “a” is the *parent* of nodes “b, c and d”
  - The out-degree of “a” is the number of children it has (here 3)
- Nodes “b, c and d” are the *children* of “a”
- Nodes “b, c and d” are *siblings* of each another



A Forest of Three Trees



Formulas for Trees

- For a singly-rooted tree

  num of edges = num of nodes - 1

- For multiple roots

  num of edges = num of nodes - num of roots

- As with a directed graph, the number of nodes in a tree may be called the *weight*

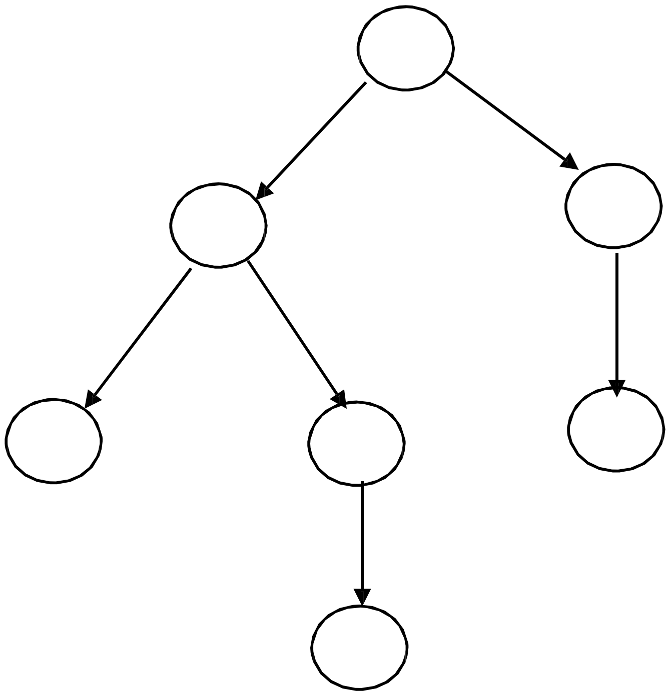
Restricted Forms of Trees

- Trees may be restricted in various ways
  - We have already seen the restriction to a single root
- These restrictions are generally to make the trees easier to represent and manipulate by computer

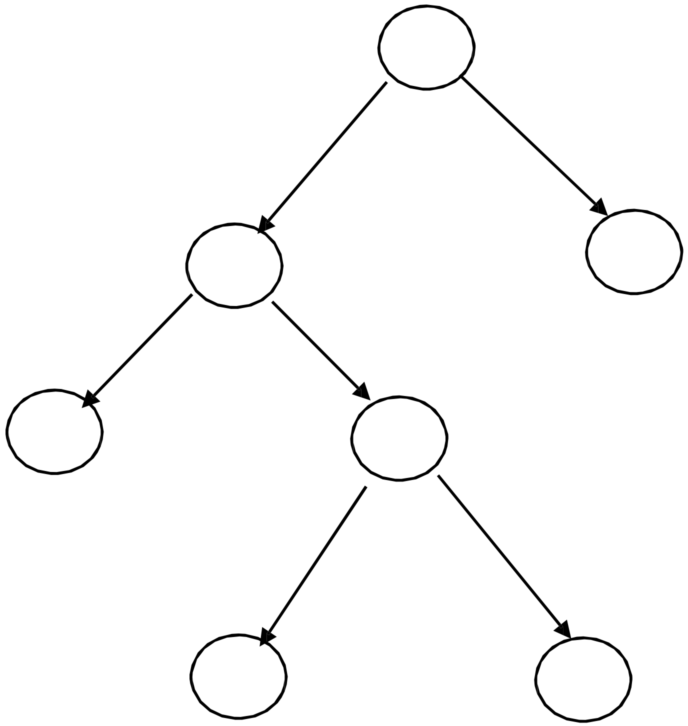
Node Degree

- A tree is said to have *limited out-degree d* if no node in the tree has an out-degree greater than d
- A tree is said to have *strict out-degree d* if the out-degree of every node is either 0 or d
- Specifically, for d=2:
  - In a *binary-limited tree* (or just *binary tree*), the nodes have a maximum out-degree of 2
  - In a *strict binary tree,* the nodes have an out-degree of 0 or 2 (so no out-degrees of 1)

Node Degree



binary-limited

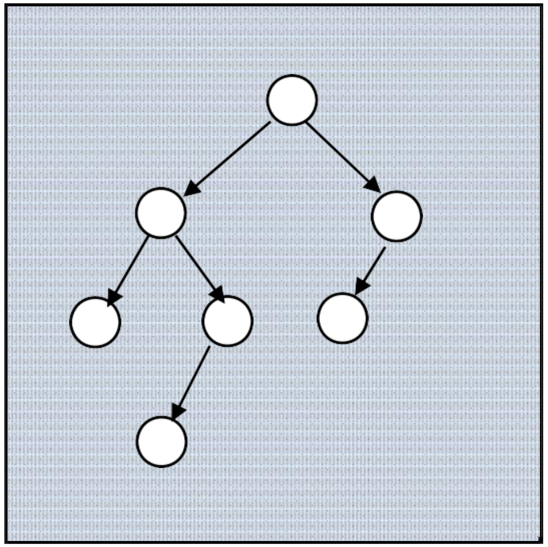


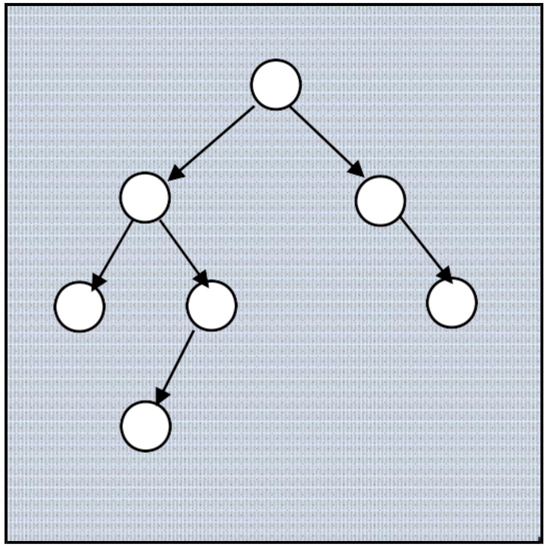
strict binary

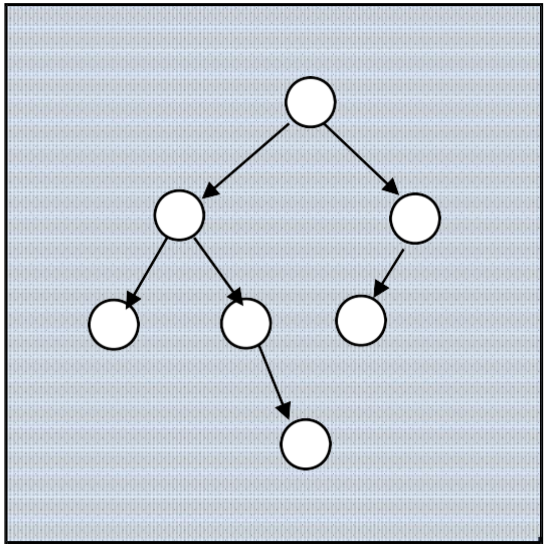
Orientation

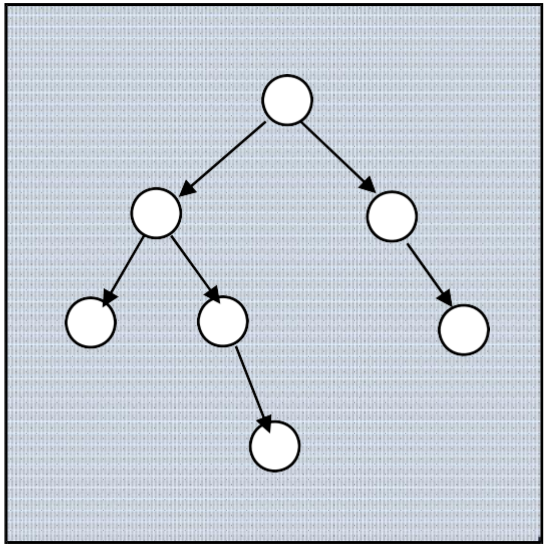
- In an *oriented* tree, we think of each node as having d distinct possible subnode positions
  - None, some or all of these positions may be occupied by a node
  - If only some of the node positions are occupied, the different ways in which subnode positions can be occupied or not are thought of as different trees
  - In other words, the subnodes of a node are *ordered*
- In an oriented *binary* tree, the *left* and *right* subnodes are distinct

Four Different Oriented Binary Trees





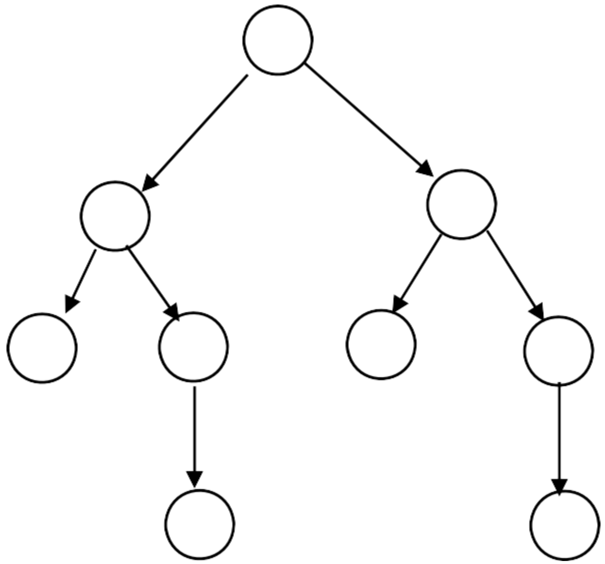




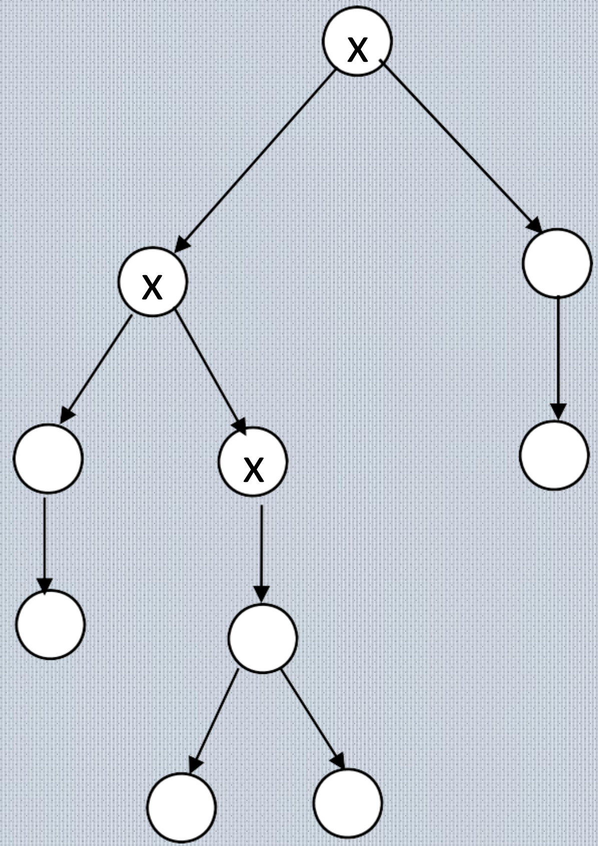
Balance

- In a balanced tree, the "weights" (numbers of nodes) of all the subtrees of a node are as nearly equal as possible
- In practice, this means that the weights are either equal, or differ by not more than one
  - Other definitions are possible, but this is a reasonable one

Balance



balanced

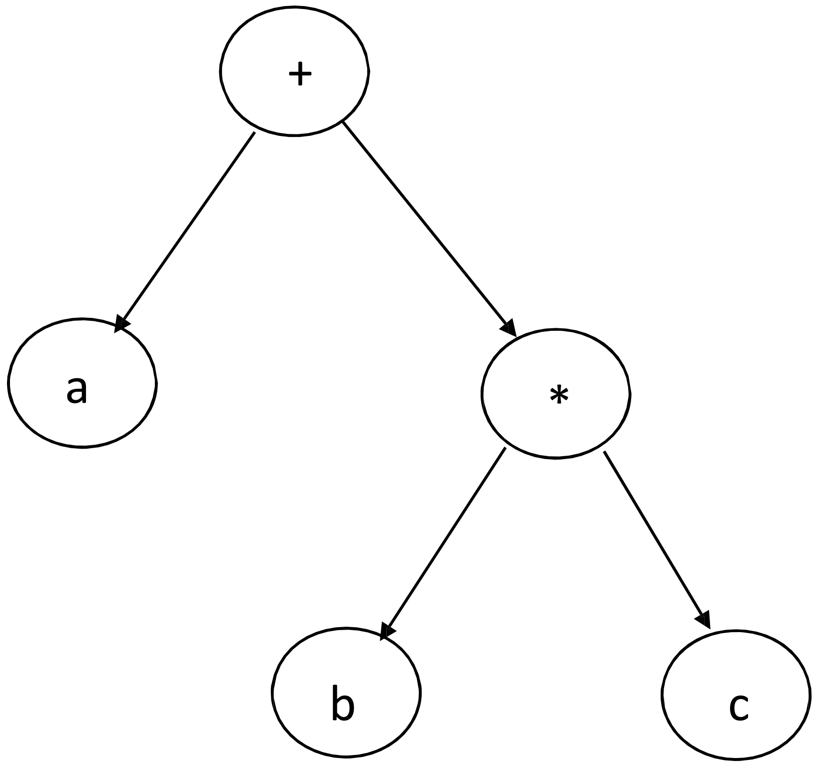


unbalanced at nodes marked “x”

Value Distribution

- In most applications of trees, the nodes (or most of them) hold *values*, while the edges rarely do
- The values may be distributed in different ways
  - All the nodes hold values
  - All the nodes except the root hold values
  - Only the leaf nodes hold values
  - Leaves and non-leaves hold values, but of two distinct types

Value Distribution



- This tree represents the expression: a+(b*c)
- The leaves hold variables (or their values) and the non-leaves hold operators

Other Restrictions

- The height of the tree (the number of edges on the longest path between root node and a leaf) may be limited
  - For example, maximum height = 5
- The leaves may all be at the same distance from the root: in this case, we say that the tree is *uniform*

SCC120 ADT (weeks 7-13)

• Week 7	Abstractions; Set Stack
• Week 8	Queues Priority Queues
• Weeks 9-11	Graphs (Terminology) Graphs (Traversals) Graphs (Representations)
• Week 12	Trees (Terminology)
• Week 13	