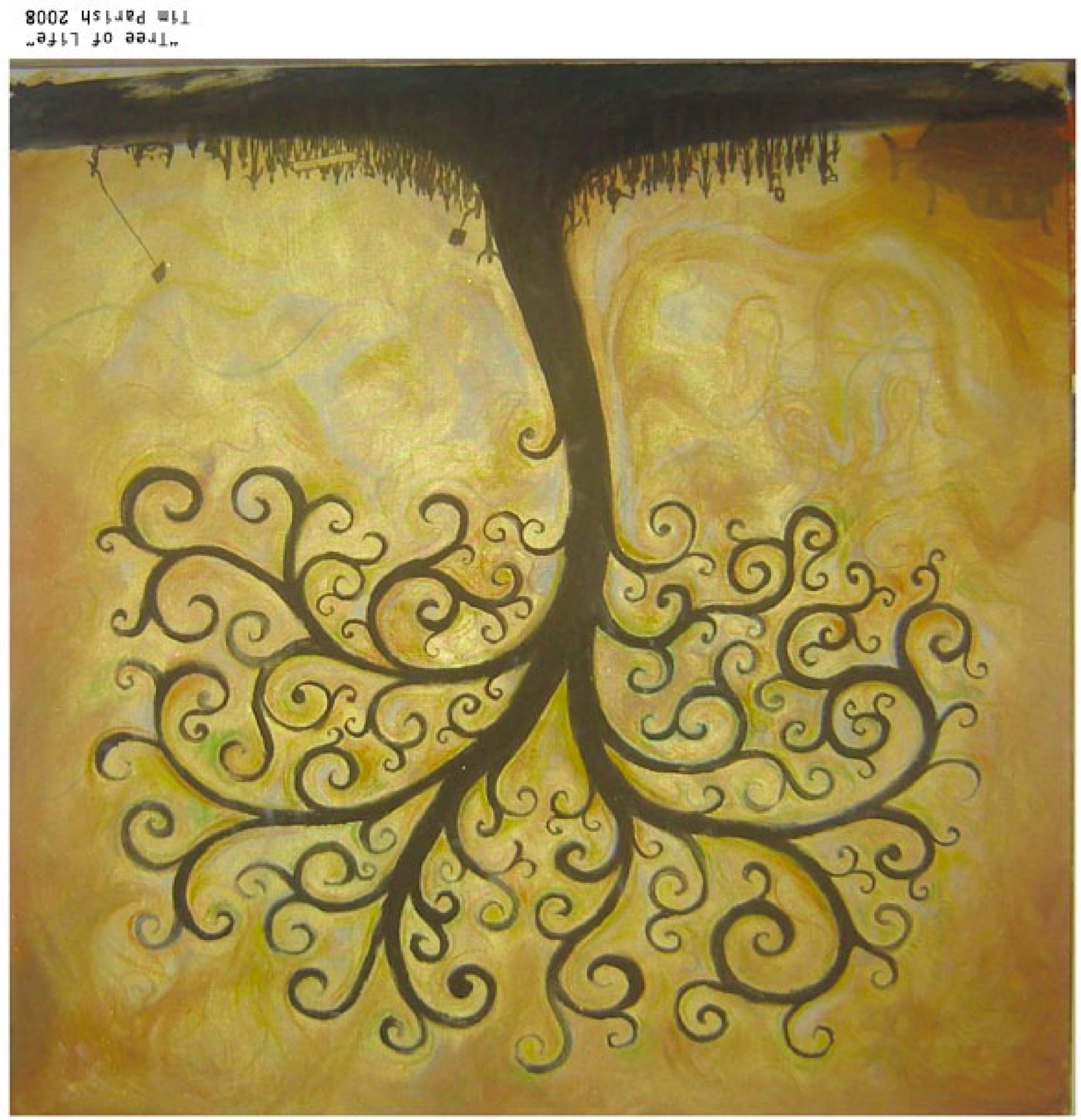
Announcements

MP4 available, due , 11:59p. EC due , 11:59p.

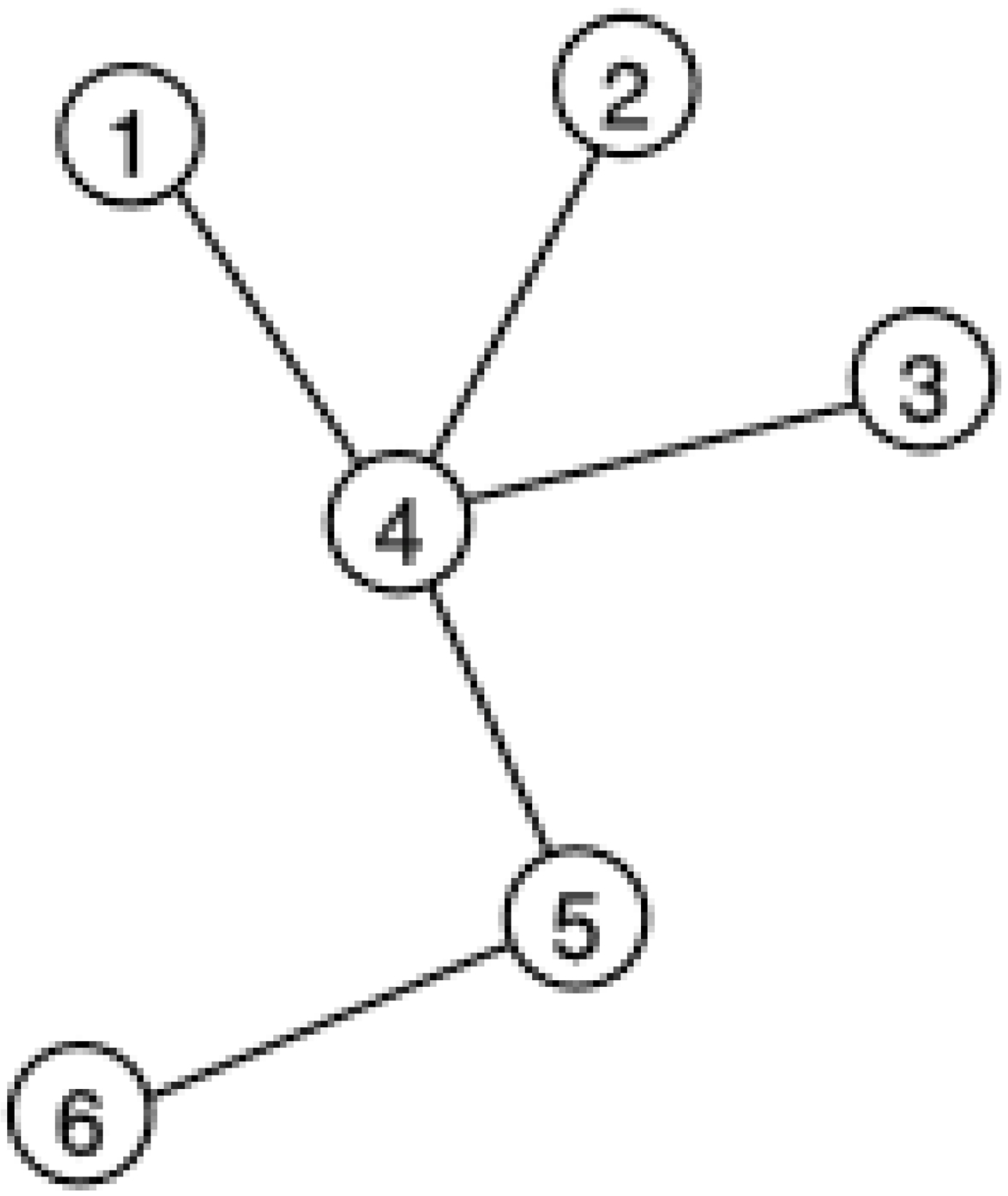


Trees:

"… most important nonlinear structure in computer science."

-- Donald Knuth, *Art of Computer Programming Vol 1*

A tree: ____



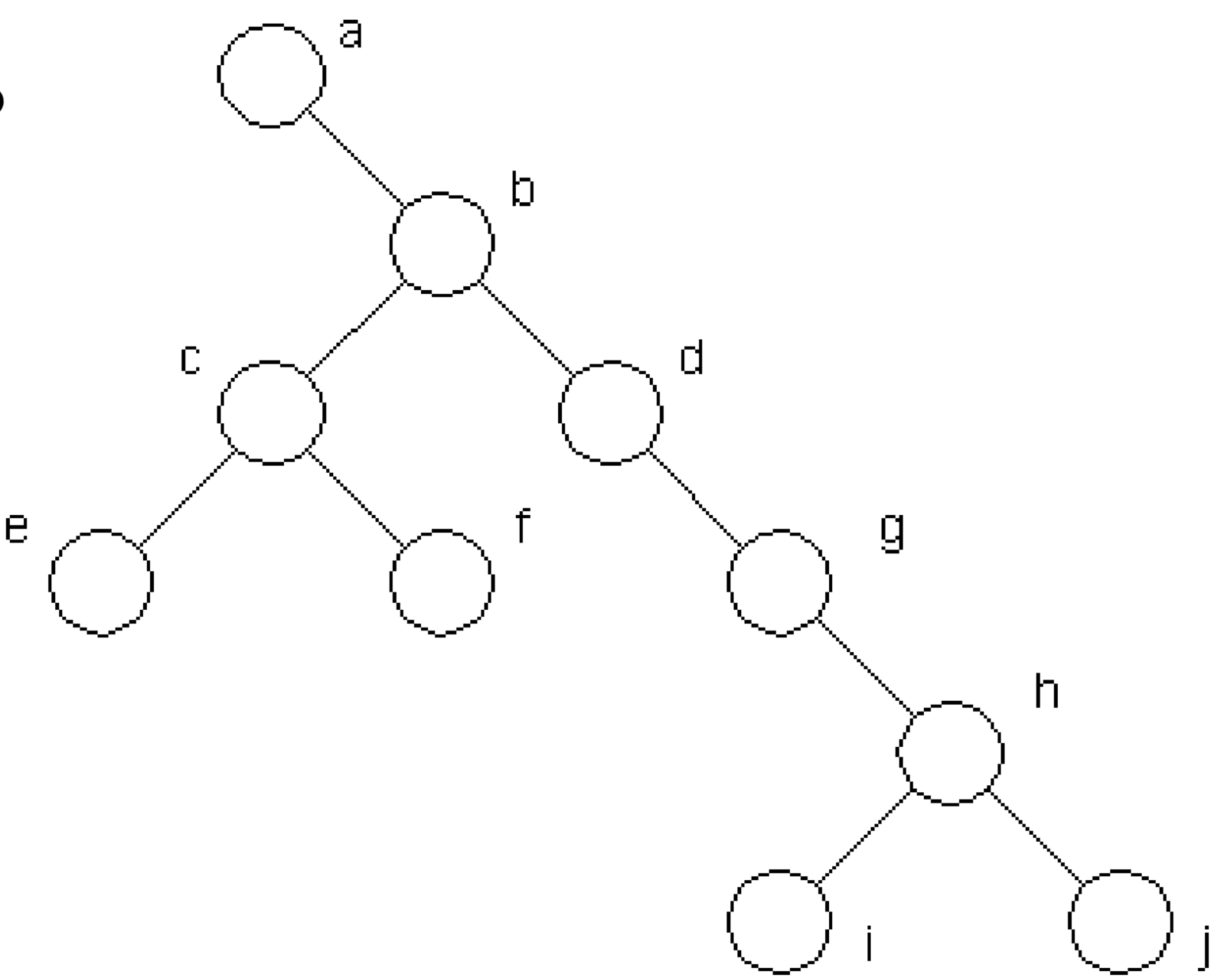
We'll study more specific trees:

Tree terminology:

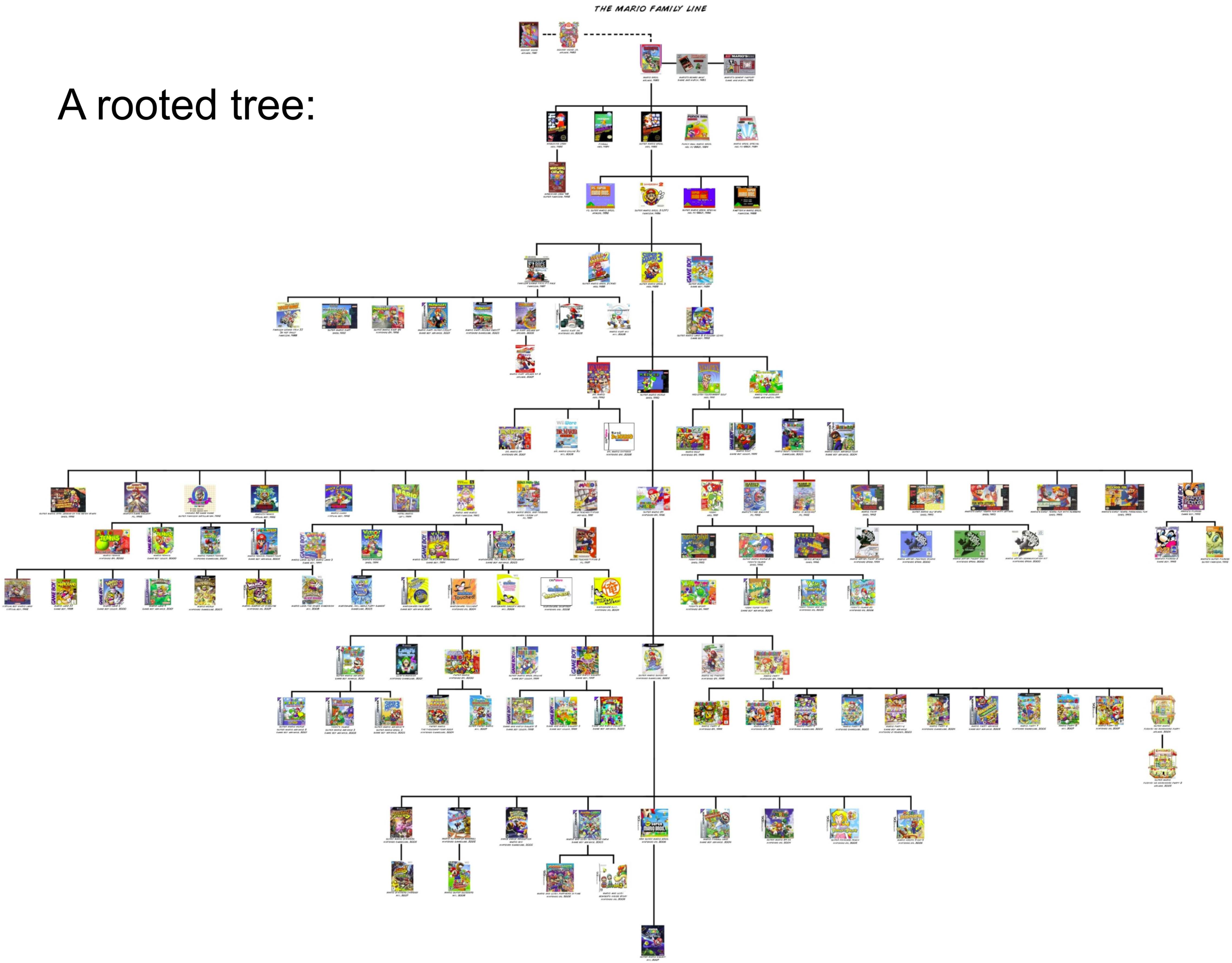
- What's the longest English word you can make using the vertex labels in the tree (repeats allowed)?
- Find an edge that is not on the longest path in the tree. Give that edge a reasonable name.

For the rest of the exercises, assume the tree is rooted.

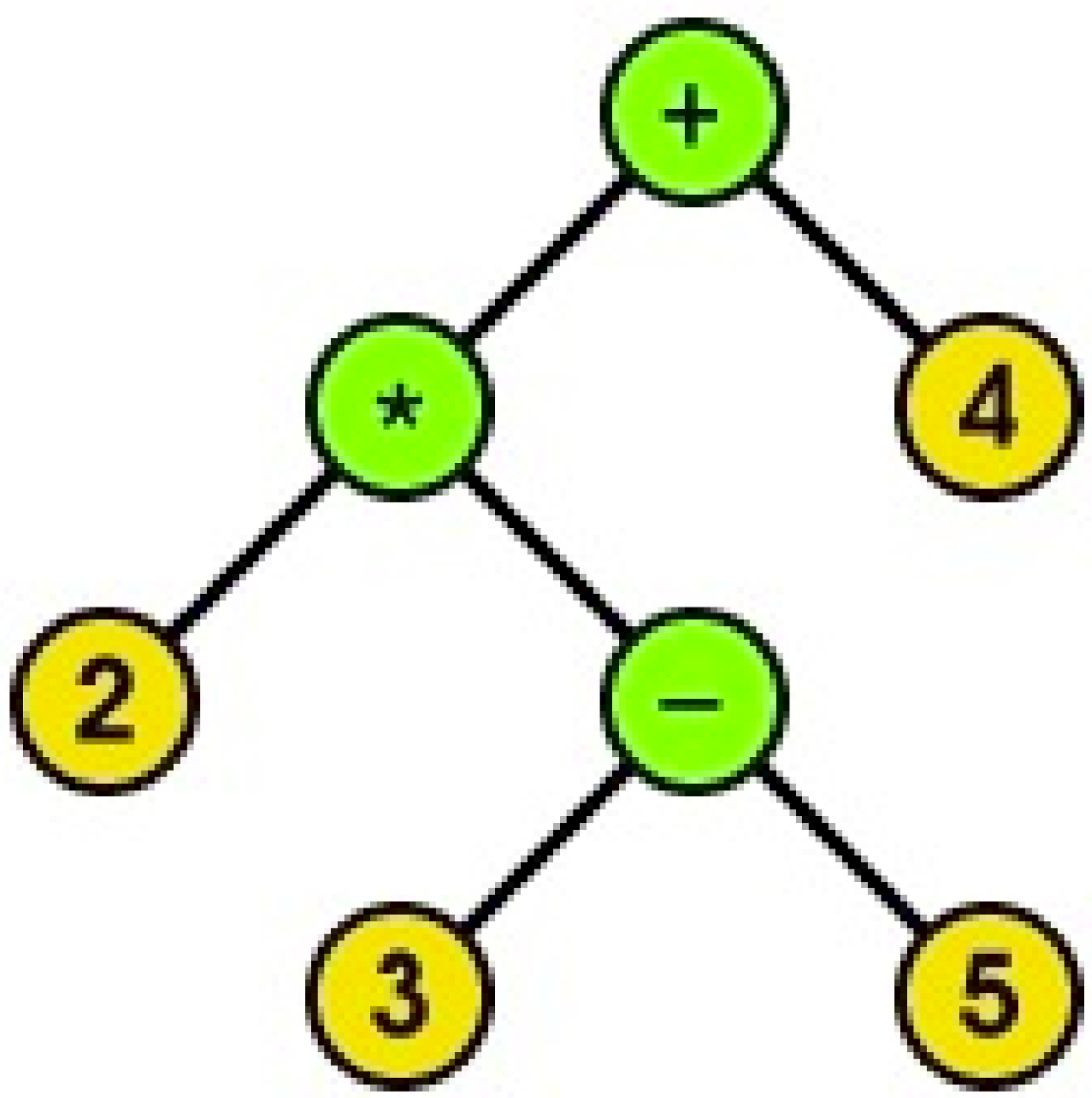
- One of the vertices is called the "root" of the tree. Guess which one it is.
- Make an English word containing the names of the vertices that have a parent but no sibling.
- How many parents does each vertex have?
- Which vertex has the fewest children?
- Which vertex has the most ancestors?
- Which vertex has the most descendants?
- List all the vertices is b's left subtree.
- List all the leaves in the tree.



A rooted tree:



Binary tree, recursive definition:



A *binary tree* T is either

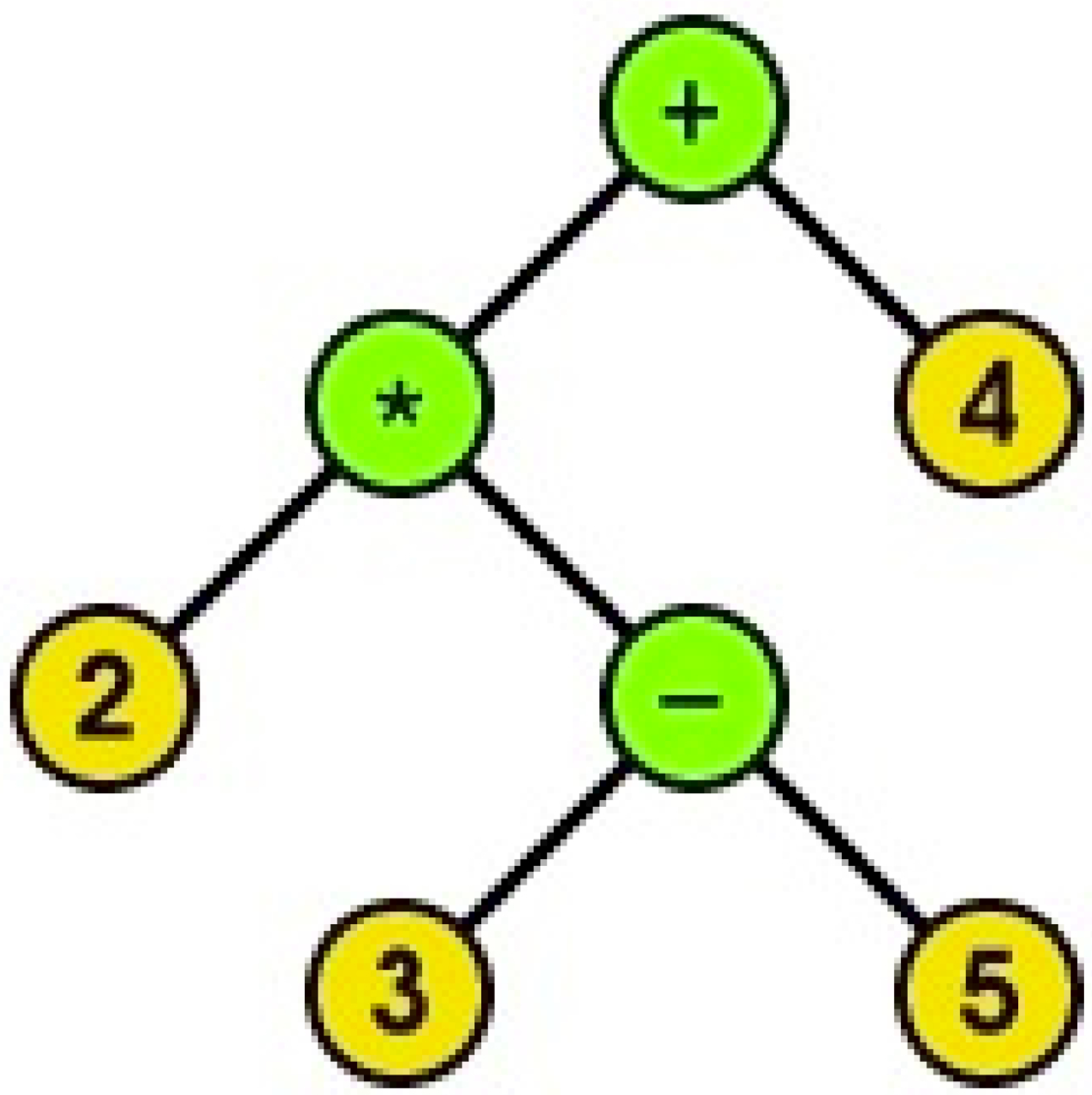
-

OR

-

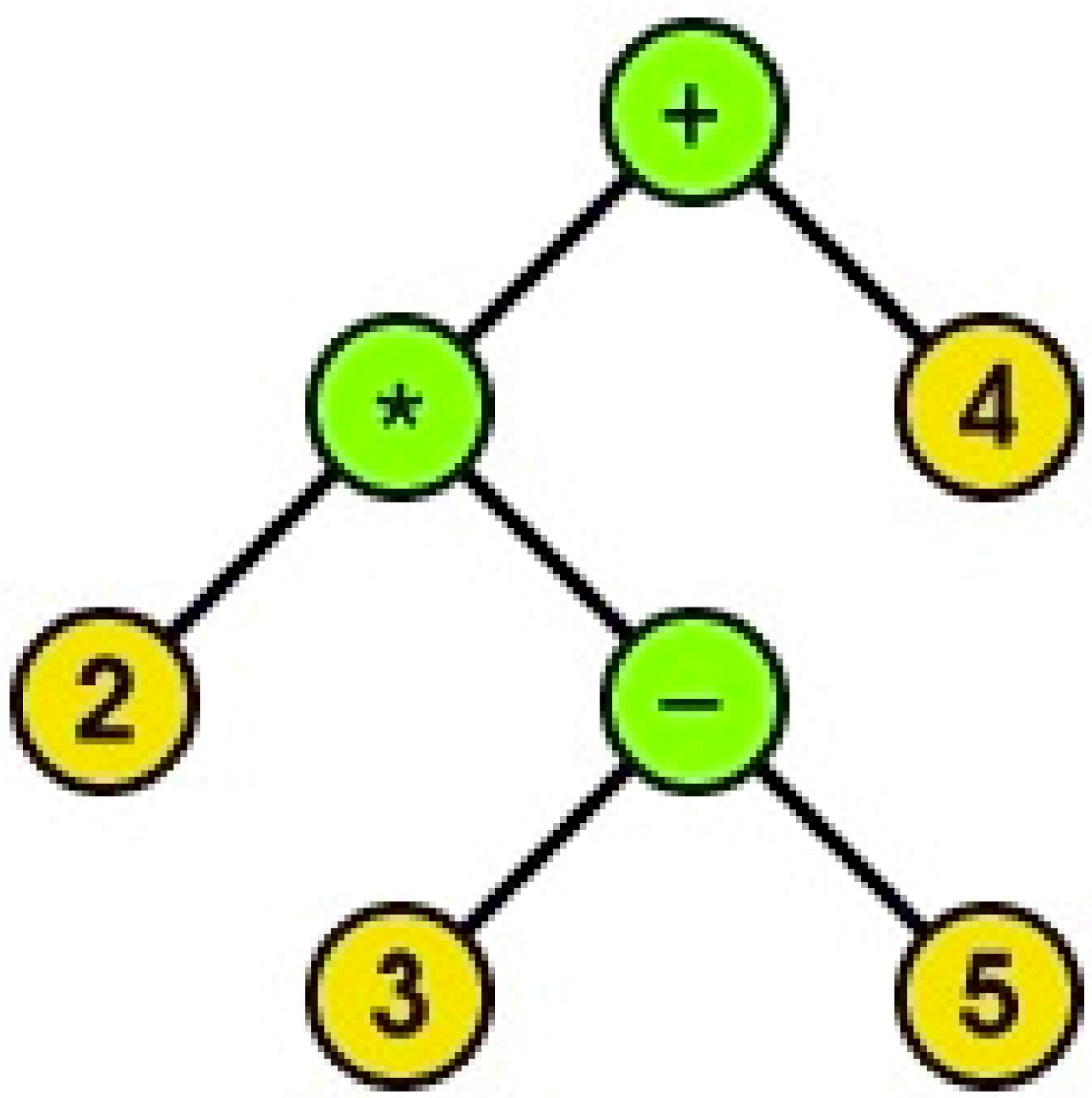
An (important) example of a function on a binary tree:

height(t) -- length of longest path from root to a leaf



Given a tree T, write a recursive defn of the height of T, height(T):

Full Binary tree: a tree in which every node has 2 or 0 children



F is a full binary tree if and only if:

- F={}   OR,
- F={r, T_L, T_R}, and

Perfect Binary tree:

Perfect tree of height h, P_h:

- P_{-1} is an empty tree
- if h > -1, then P_h is $\{r, T_L, T_R\}$,

where T_L and T_R are P_{h-1}.

P_0:

P_2:

P_1:

Check for understanding:

How many nodes in a perfect tree of height h?