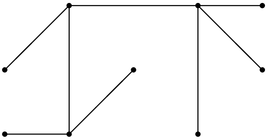
EECS/MATH 1019
Section 11.1:
Trees – Introduction

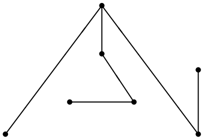
April 6, 2023

Definition: A tree is a connected undirected graph with no simple circuits.

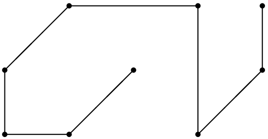
E.g.



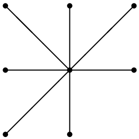
or



or



or

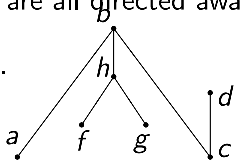


Observe that in a tree, for any two vertices u and v, there is a unique simple path from u to v.
Conversely, suppose a graph G has the property that for any two vertices u and v, there is a unique simple path from u to v. Then G is a tree. (G is obviously connected; and if u and v were on a simple circuit, then there would be two simple paths from u to v.)

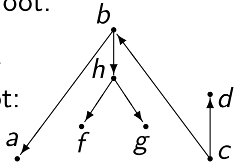
(The above is Theorem 1, p. 782.)

In a rooted tree, one vertex is designated to be the root and all the edges are all directed away from the root.

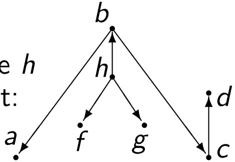
E.g.



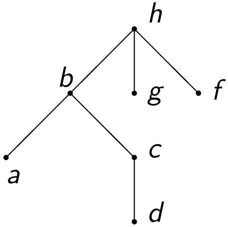
Choose c to be root:



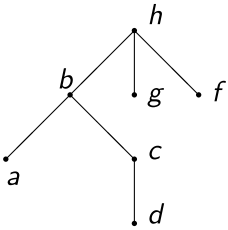
Or choose h to be root:



Customarily, we draw a rooted tree with the root at the top, and all directions of edges pointing down. Then we can omit the arrows, as in above example with root h:



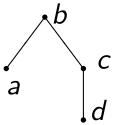
Terminology for rooted trees:



b is the parent of c and of a
a and c are the children of b
a is a sibling of c (same parent)
Also, b, g, and f are siblings.
The descendants of b are a, c, and d.
The ancestors of c are b and h.

A leaf in a rooted tree is a vertex with no children. Here, the leaves are a, d, g, and f. The internal vertices are the vertices with children: here, b, c, and h.

The subtree with root b of the above tree is



In an ordered rooted tree, the left-to-right order of the drawing matters. For example, considering the above tree as an ordered rooted tree, the first child of h is b, the second child of h is g, and the third child of h is f.

More definitions:

A binary tree is a rooted tree in which each vertex has at most 2 children. A full binary tree is a rooted tree in which every internal vertex has exactly 2 children.

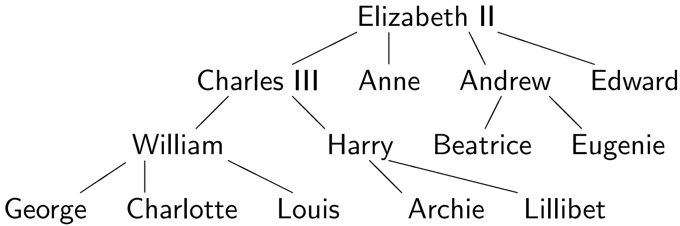
If we change "2 children" to "3 children" in the above definitions, then we get ternary trees and full ternary trees.

If we change "2 children" to "m children" in the above definitions, where m is any integer greater than 1, then we get m-ary trees and full m-ary trees. (Not "mary trees.")

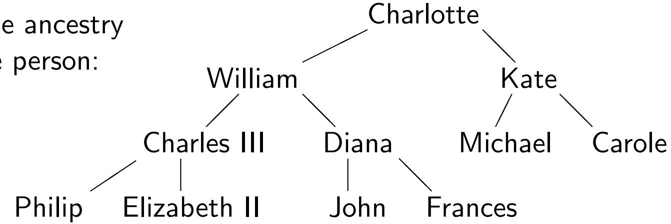
In an **ordered binary tree**, a vertex v with two children has a left child and a right child, which in turn are the roots of the left subtree and right subtree of v, respectively.

Trees can describe many situations: (see also Sec. 11.1.2)

1. Genealogy – Family trees: The descendants of one person:



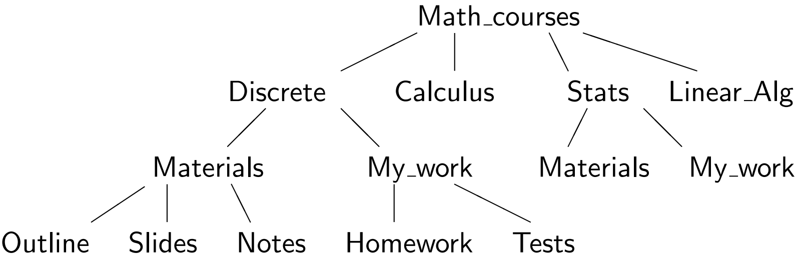
Or, the ancestry of one person:



(The second example should be an ordered full binary tree, in principle. In this picture, the "left child" of a vertex is that person's father, and the "right child" is their mother.(!))

2. Evolutionary trees:
Each vertex is a species, or a genus, or a family, depending on the level of resolution.

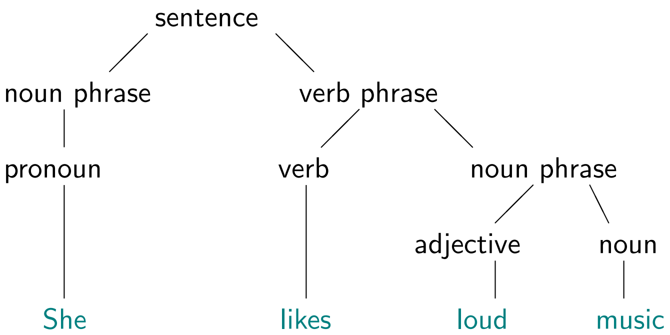
3. Directory structure in a computer file system: Files are contained in folders, which are contained in other folders,. . .



Each file is a leaf. Each (nonempty) folder is an internal vertex.

4. Natural language analysis: Parsing a sentence

For example, "She likes loud music" is parsed as follows:



5. Sorting or classification trees: E.g. books are sorted by the Library of Congress classification system.

. . . P is language and literature; Q is Science; R is Medicine; . . .

QA is Math; QB is Astronomy; QC is Physics; . . .

. . . QA75.5–76.95 is Computer Science; . . . QA150–272.5 is Algebra; . . . QA273-280 is Probability & Statistics . . .

6. Decision trees: Each vertex represents a situation that can arise. Each child of a vertex v represents a possible event that could happen following situation v, or else a decision that could be made.

E.g. for chess-playing program: The root is the opening position. Its children are the possible positions after one move. (If computer plays first, then these are decisions; otherwise, these are events.) These children's children are the positions after the second move. And so on.

In chess, the whole tree is too big to consider. A good player (or program) needs different strategies.

Sec. 11.1.3: Properties of Trees

Terminology: A “leaf” in a (rooted or unrooted) tree is a vertex of degree 1.

Recall that a tree is defined to be a connected graph with no simple circuits. Every example of a tree that we have drawn has leaves. Is it obvious that every tree has a leaf?
There is at least one tree with no leaves: a tree with one vertex.
Any other exceptions? We need a proof.

Lemma 11.A: Every tree with more that one vertex has at least one leaf.

Lemma 11.A: Every tree with more that one vertex has at least one leaf.

Proof: Assume $n \geq 2$. Let T be a tree with n vertices.
Let $\mathcal{P}$ be the set of all paths in T that do not visit any vertex twice. So every path in $\mathcal{P}$ has length at most $n-1$.
Let L be the length of a longest path in $\mathcal{P}$, and let $v_0, v_1, \ldots, v_{L-1}, v_L$ be such a path.
We shall use contradiction to prove that v_L is a leaf.

Assume that v_L is not a leaf. It has a neighbour v_{L-1}, so it must have another neighbour, which we can call w. (And $w \neq v_{L-1}$).
The path $v_0, v_1, \ldots, v_{L-1}, v_L, w$ has length $L+1$, so it is not in $\mathcal{P}$.
Therefore it must visit some vertex more than once. Since $v_0, v_1, \ldots, v_{L-1}, v_L$ is a path in $\mathcal{P}$, these vertices are all different, so it must be that w equals v_i for some i (with $i < L-1$). Then $v_i, v_{i+1}, \ldots, v_{L-1}, v_L, w$ is a circuit with no repeated vertices (except $v_i = w$), which implies that it is a simple circuit in T.
This a contradiction because T is a tree. Therefore, v_L is a leaf.

We showed that any tree T with n vertices must have a leaf ($n \geq 2$). This proves the lemma. Q.E.D.

Theorem 2: Every tree with n vertices has $n - 1$ edges.

Proof: We shall use induction on n. Let $P(n)$ be the statement that every tree with n vertices has $n - 1$ edges.
Basis step: $n = 1$: A tree with 1 vertex is just a vertex. It has no edges. So $P(1)$ is true.

Inductive step: Let $k \in \mathbb{Z}^+$. Assume that $P(k)$ is true.
Let T be a tree with $k + 1$ vertices.
Let m be the number of edges in T.
By Lemma 11.A, T has a leaf, which we can call w.

Let S be the graph obtained by removing w from T. This operation also removes the single edge of T incident on w, so S has $m - 1$ edges and $(k + 1) - 1$ vertices.

Observe that S is a tree (no simple circuits, and connected). And S has k vertices, so S must have $k - 1$ edges, by $P(k)$.
Therefore $m - 1 = k - 1$, i.e. $m = k$. Therefore T has $(k + 1) - 1$ edges. Since T was an arbitrary tree with $k + 1$ vertices, we have proved that $P(k + 1)$ is true.

By induction, we conclude that $P(n)$ is true for every $n \in \mathbb{Z}^+$.
Q.E.D.

Theorem 3: Let T be a full m-ary tree with i internal vertices. Then T has $mi + 1$ vertices.

Proof: Every internal vertex has exactly m children. So there are im children in T (in trees, a child only has one parent!).

Every vertex of the tree is the child of exactly one internal vertex (except the root is not the child of anything). So the number of vertices in T is one more than the number of children. That is, T has $mi + 1$ vertices. Q.E.D.

E.g. In a full binary tree with i internal vertices, the total number of vertices is $2i + 1$, and hence the number of leaves is $(2i + 1) - i = i + 1$.
We used the property that

number of vertices = number of leaves + number of internal vertices

which holds in any rooted tree.

To summarize the last slide: In a full m-ary tree, let

$$\begin{aligned} n &= \text{number of vertices,} \\ L &= \text{number of leaves, and} \\ i &= \text{number of internal vertices.} \end{aligned}$$

Then $\quad n = mi + 1 \quad$ and $\quad n = L + i\,.$

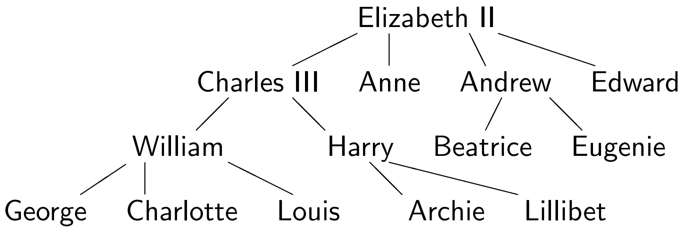
For a given m-ary tree (with m known), if we know any one of the quantities n or L or i, then the above two equations allow us to solve for the other two.
The formulas are given explicitly in Theorem 4, but they are easy to produce when needed if we remember $n = mi + 1$ and $n = L + i$.

Example: A full 3-ary tree has 13 leaves. How many vertices and internal vertices does it have?

We know $n = 3i + 1$ and $n = 13 + i$. Solving these two equations for i and n gives

$$3i + 1 = 13 + i\,; \quad \therefore 2i = 12\,; \quad \therefore i = 6 \text{ and } n = 13 + 6 = 19.$$

More terminology for rooted trees: The level of a vertex is the length of the path from that vertex to the root.



E.g. The level of Edward is 1, of Harry is 2, and of George is 3. Elizabeth has level 0 (the root). In this example, level counts the number of generations after Elizabeth.

The height of a rooted tree is the largest level in the tree.
The above example has height 3.

Poll: In an 5-ary tree, what is the largest possible number of vertices that could be at level 3?

(A) 5	(B) 15	(C) 25
(D) 75	(E) 125	(F) 375

Answer: (E).

For vertices at different levels of a 5-ary tree:
The first level has at most 5 vertices.
The second level has at most 5×5 $(= 25)$ vertices.
The third level has at most 25×5 $(= 125)$ vertices.

$\vdots$

The k^{th} level has at most 5^k vertices.

In an 5-ary tree of height 3, what is the largest possible number of leaves?
If all leaves are at level 3, then there are at most 125 leaves.
Poll: Can a 5-ary tree of height 3 have more than 125 leaves if some leaves are at levels 1 and 2?
(A) Yes (B) No

Answer: (B).

Theorem 5: An m-ary tree of height h has at most m^h leaves.

Proof is by mathematical induction. See page 790.

Please visit the course evaluation web site:
https://courseevaluations.yorku.ca

The Connect assignment is due Monday April 10.

Some exam information is on the eClass page now. More will be there by Monday, including my office hours for the exam period.