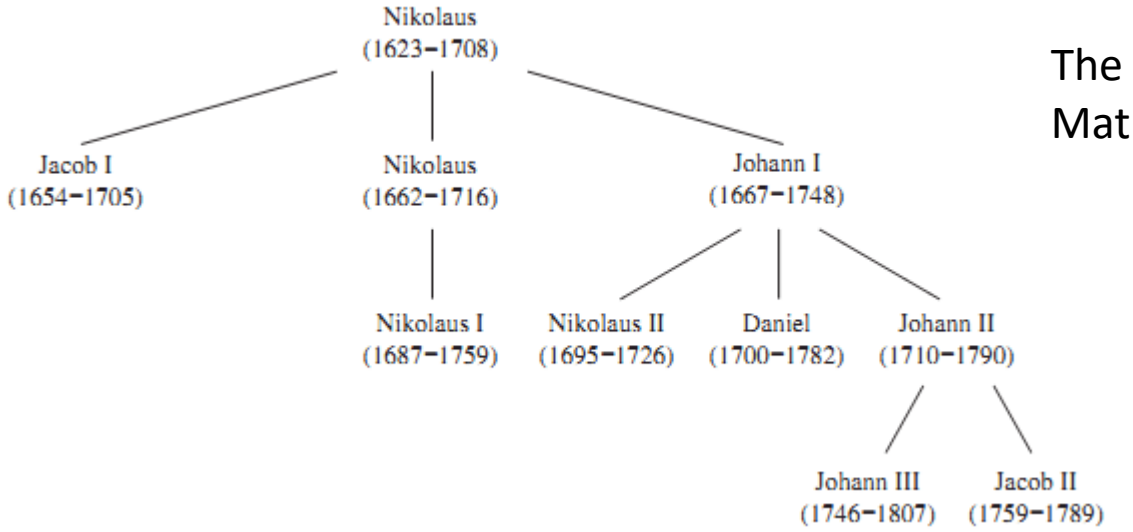
Trees (Part1)

In this lecture we will focus on a particular type of graph called a tree, so named because such graphs resemble trees.

For example, family trees are graphs that represent genealogical charts.

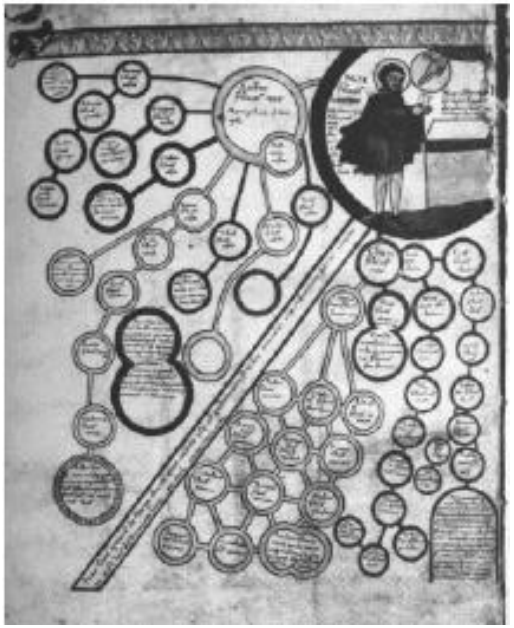
Family trees use vertices to represent the members of a family and edges to represent parent–child relationships.

The family tree of the male members of the Bernoulli family of Swiss mathematicians is shown bellow.



The Bernoulli Family of Mathematicians.

Ancient family trees



DEFINITION 1

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

OR: A tree is a connected acyclic undirected graph

Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops.

Therefore any tree must be a simple graph.

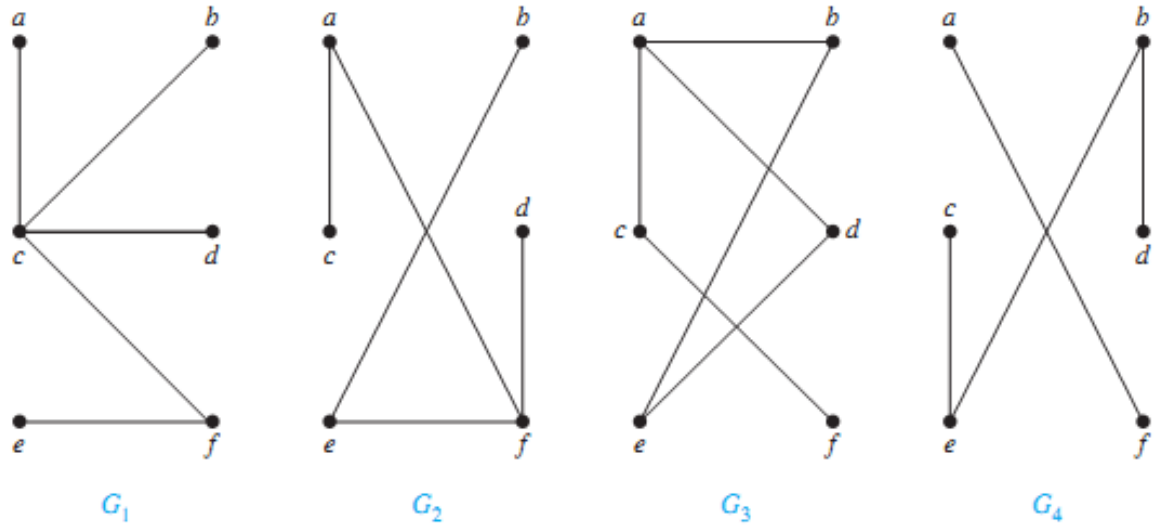
EXAMPLE

Which of the graphs G_1, G_2, G_3 , G_4 are trees?

Solution: G_1 and G_2 are trees, because both are connected graphs with no simple circuits.

G_3 is not a tree because *e, b, a, d, e* is a simple circuit in this graph.

Finally, G_4 is not a tree because it is not connected.



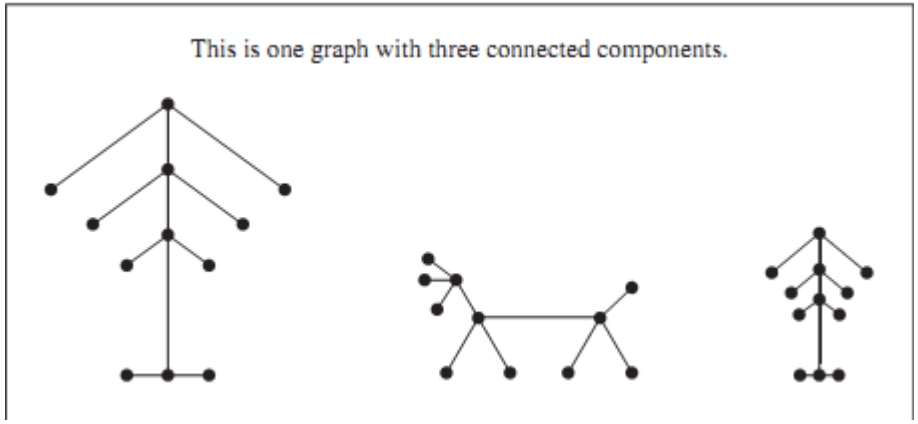
Forest (лес)

Any connected graph that contains no simple circuits is a tree.

What about acyclic graphs that are not necessarily connected?

These graphs are called forests (лес) and have the property that each of their connected components is a tree.

Trees are often defined as undirected graphs with the property that there is a unique simple path between every pair of vertices.



Example of a Forest.

THEOREM 1 An undirected graph is a tree ⇔ there is a unique simple path between any 2 of its vertices.

Proof:

⇒ First assume that *T* is a tree.

Then *T* is a connected graph with no simple circuits.

Let *x* and *y* be 2 vertices of *T* .

Because *T* is connected, by Theorem 1 of Lecture 4 there is a simple path between *x* and *y*.

Moreover, this path must be unique, for if there were a second such path, the path formed by combining the first path from **x** to *y* followed by the path from *y* to *x* obtained by reversing the order of the second path from *x* to *y* would form a circuit.

This implies that there is a simple circuit in *T* .

Hence, there is a unique simple path between any 2 vertices of a tree.

<= Now assume that there is a unique simple path between any two vertices of a graph *T* .

Then *T* is connected, because there is a path between any two of its vertices.

Furthermore, *T* can have no simple circuits.

To see that this is true, suppose *T* had a simple circuit that contained the vertices *x* and *y*.

Then there would be 2 simple paths between *x* and *y*, because the simple circuit is made up of a simple path from *x* to *y* and a second simple path from *y* to *x*.

Hence, a graph with a unique simple path between any two vertices is a tree.

Rooted Trees

In many applications of trees, a particular vertex of a tree is designated as the root (корень).

Once we specify a root, we can assign a direction to each edge as follows.

Because there is a unique path from the root to each vertex of the graph (by Theorem 1), we direct each edge away from the root.

Thus, a tree together with its root produces a directed graph called a rooted tree (корневое дерево).

Rooted Trees

DEFINITION 2 A *rooted tree* is a tree in which one vertex has been designated as the root and every edge is directed away from the root.

Rooted trees can also be defined recursively.

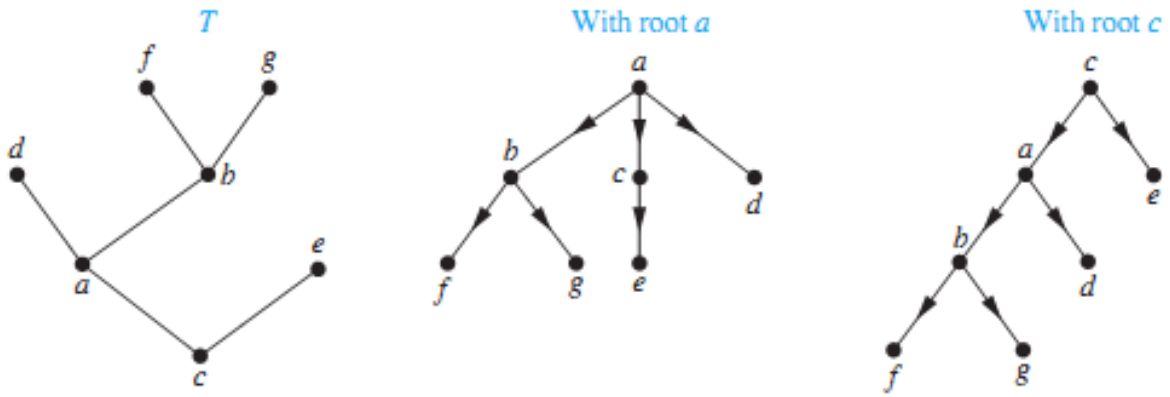
We can change an unrooted tree into a rooted tree by choosing any vertex as the root.

Note that different choices of the root produce different rooted trees.

For instance, the rooted trees formed by designating *a* to be the root and *c* to be the root, respectively, in the tree *T* are shown bellow.

We usually draw a rooted tree with its root at the top of the graph.

The arrows indicating the directions of the edges in a rooted tree can be omitted, because the choice of root determines the directions of the edges.



A Tree and Rooted Trees Formed by Designating 2 Different Roots.

The terminology for trees

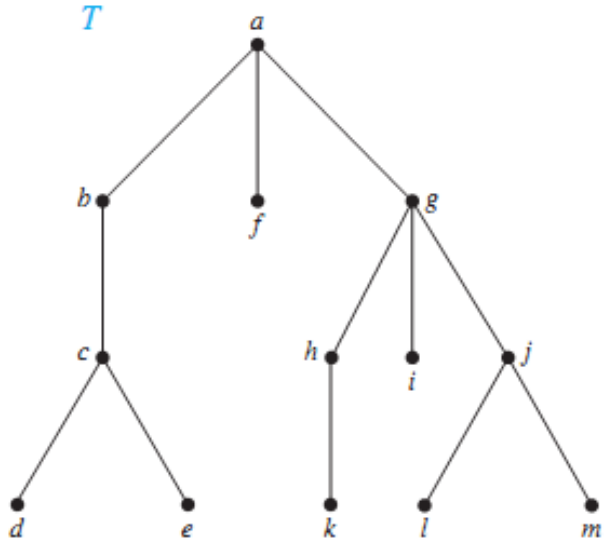
The terminology for trees has botanical and genealogical origins.

Suppose that *T* is a rooted tree.

If *v* is a vertex in *T* other than the root, the parent (отец) of *v* is the unique vertex *u* such that there is a directed edge from *u* to *v* (such a vertex is unique).

When *u* is the parent of *v*, v is called a child (сын) of *u*.

Vertices with the same parent are called siblings (братья).



Example

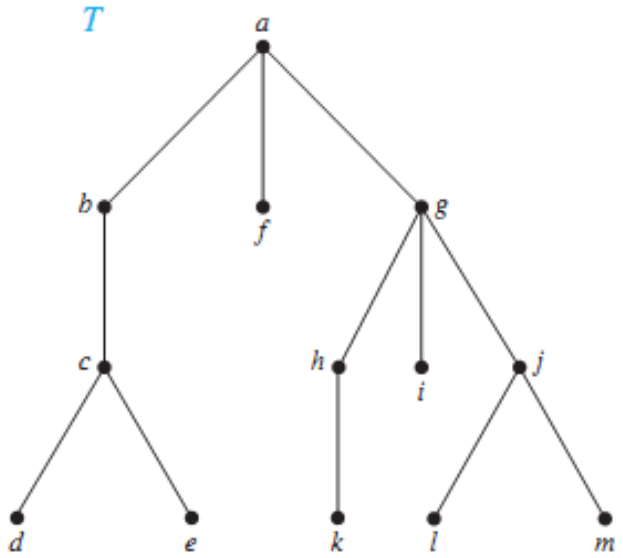
The parent of *c* is *b*.

The children of *g* are *h, i*, and *j* .

The siblings of *h* are *i* and *j* .

The ancestors (предки) of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root (that is, its parent, its parent's parent, and so on, until the root is reached).

The descendants(потомки) of a vertex *v* are those vertices that have *v* as an ancestor.



Example

The ancestors of *e* are *c*, *b*, and *a*.

The descendants of *b* are *c*, *d*, and *e*.

A vertex of a rooted tree is called a leaf (лист) if it has no children.

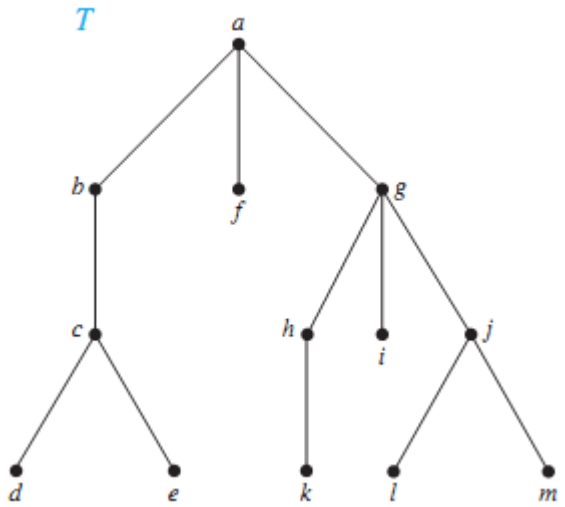
Vertices that have children are called internal vertices (внутренние вершины).

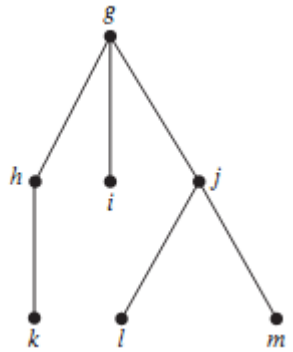
The root is an internal vertex unless it is the only vertex in the graph, in which case it is a leaf.

If *a* is a vertex in a tree, the subtree (поддерево) with *a* as its root is the subgraph of the tree consisting of *a* and its descendants and all edges incident to these descendants.

Example: The internal vertices are *a, b, c, g, h*, and *j* .

The leaves are *d, e, f , i, k, l,* and *m*.





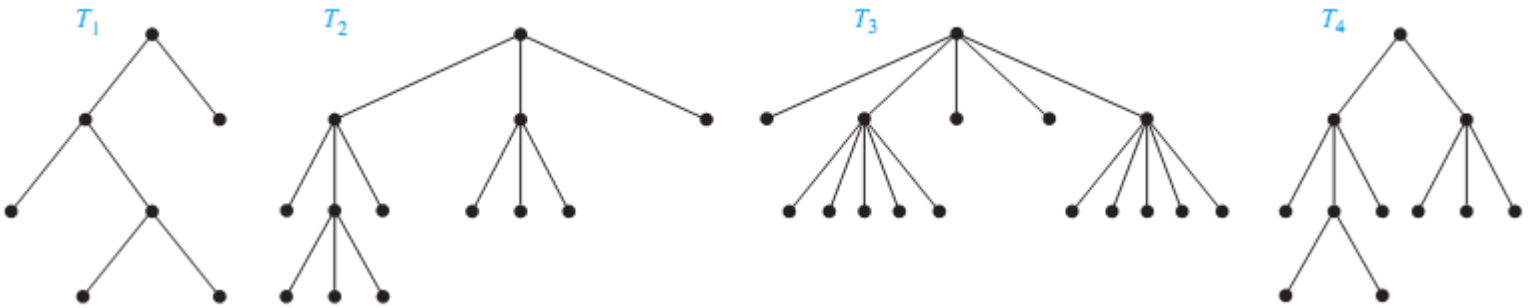
The Subtree Rooted at *g*.

DEFINITION 3 A rooted tree is called an *m*-ary tree (*m*- арное дерево) if every internal vertex has no more than *m* children.

The tree is called a full *m*-ary tree if every internal vertex has exactly *m* children.

An *m*-ary tree with *m* = 2 is called a binary tree (бинарное дерево).

EXAMPLE Are the rooted trees T_1, T_2, T_3, T_4 full m-ary trees for some positive integer m?



Solution: T_1 is a full binary tree because each of its internal vertices has 2 children.
T_2 is a full 3-ary tree because each of its internal vertices has 3 children.
In T_3 each internal vertex has 5 children, so T_3 is a full 5-ary tree.
T_4 is not a full m-ary tree for any m because some of its internal vertices have 2 children and others have 3 children.

ORDERED ROOTED TREES

An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered.

Ordered rooted trees are drawn so that the children of each internal vertex are shown in order from left to right.

Note that a representation of a rooted tree in the conventional way determines an ordering for its edges.

In an ordered binary tree (usually called just a binary tree), if an internal vertex has 2 children, the first child is called the left child (левый сын) and the second child is called the right child (правый сын).

The tree rooted at the left child of a vertex is called the left subtree of this vertex, and the tree rooted at the right child of a vertex is called the right subtree of the vertex.

Note that for some applications every vertex of a binary tree, other than the root, is designated as a right or a left child of its parent.

This is done even when some vertices have only 1 child.

We will make such designations whenever it is necessary, but not otherwise.

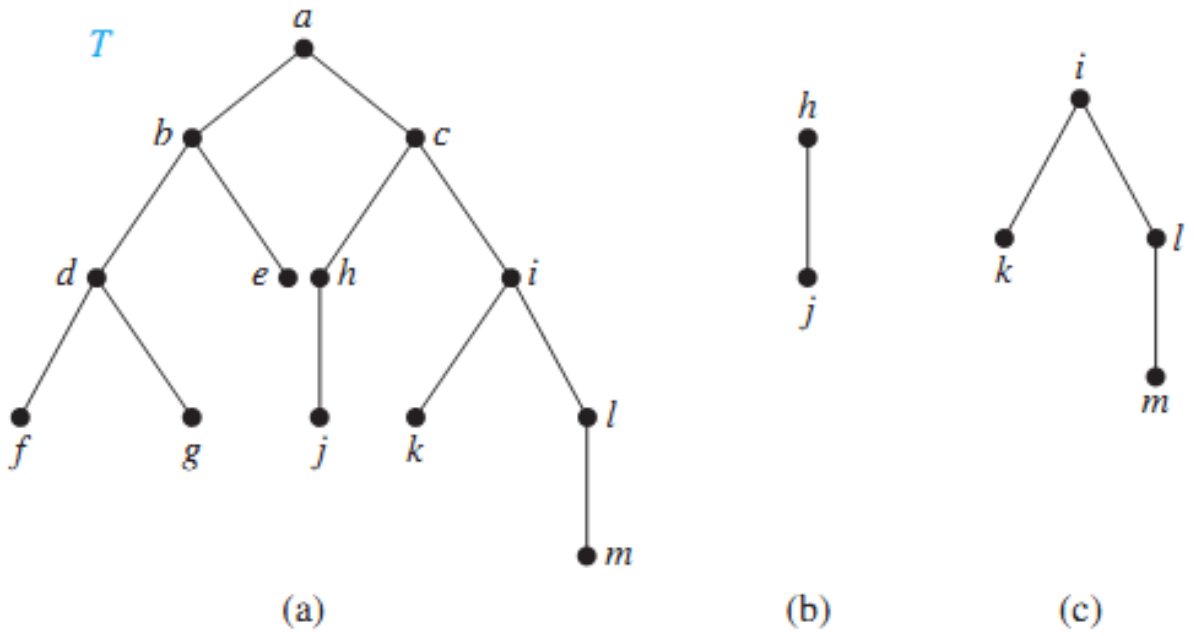
Ordered rooted trees can be defined recursively.

EXAMPLE What are the left and right children of *d* in the binary tree *T* (where the order is that implied by the drawing)?

What are the left and right subtrees of *c*?

Solution: The left child of *d* is *f* and the right child is *g*.

The left and right subtrees of *c* are shown bellow (b) and (c), respectively.



Trees as Models

Trees are used as models in such diverse areas as computer science, chemistry, geology, botany, and psychology.

EXAMPLE: Saturated Hydrocarbons and Trees

Graphs can be used to represent molecules, where atoms are represented by vertices and bonds between them by edges.

The English mathematician Arthur Cayley discovered trees in 1857 when he was trying to enumerate the isomers of compounds of the form C_nH_{2n+2}, which are called *saturated hydrocarbons*.

In graph models of saturated hydrocarbons, each carbon atom is represented by a vertex of degree 4, and each hydrogen atom is represented by a vertex of degree 1.

There are $3n + 2$ vertices in a graph representing a compound of the form C_nH_{2n+2}.

The number of edges in such a graph is 1/2 the sum of the degrees of the vertices.

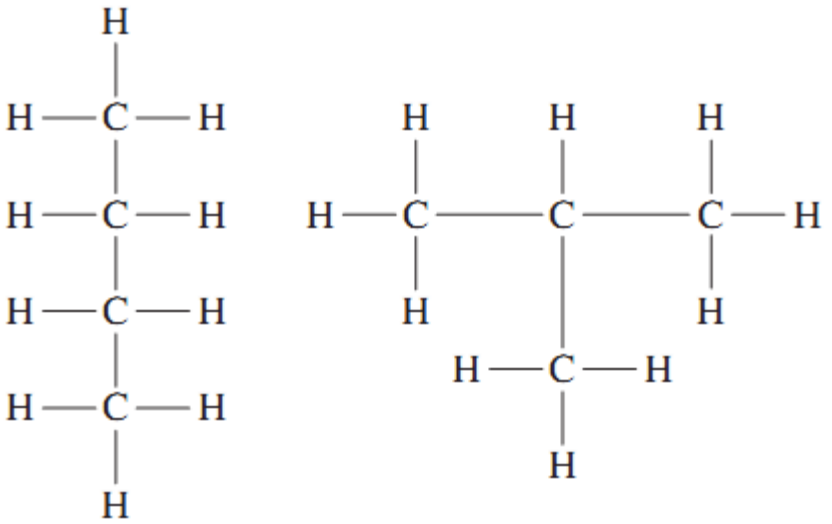
Hence, there are $(4n + 2n + 2)/2 = 3n + 1$ edges in this graph.

Because the graph is connected and $e = v-1$, it must be a tree.

The nonisomorphic trees with n vertices of degree 4 and $2n + 2$ of degree 1 represent the different isomers of C_nH_{2n+2}.

For instance, when $n = 4$, there are exactly 2 nonisomorphic trees of this type.

Hence, there are exactly 2 different isomers of C_4H_{10}. These isomers are called butane and isobutane.

butane  isobutane.

EXAMPLE Representing Organizations

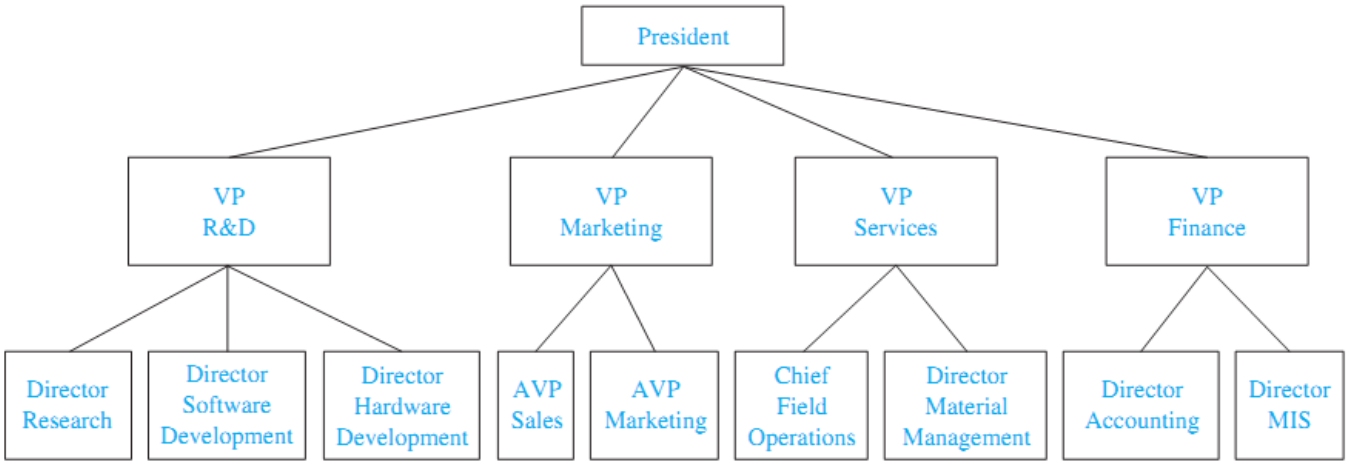
The structure of a large organization can be modeled using a rooted tree.

Each vertex in this tree represents a position in the organization.

An edge from one vertex to another indicates that the person represented by the initial vertex is the (direct) boss of the person represented by the terminal vertex.

The graph shown bellow displays such a tree.

In the organization represented by this tree, the Director of Hardware Development works directly for the Vice President of R&D.



EXAMPLE Computer File Systems

Files in computer memory can be organized into directories.

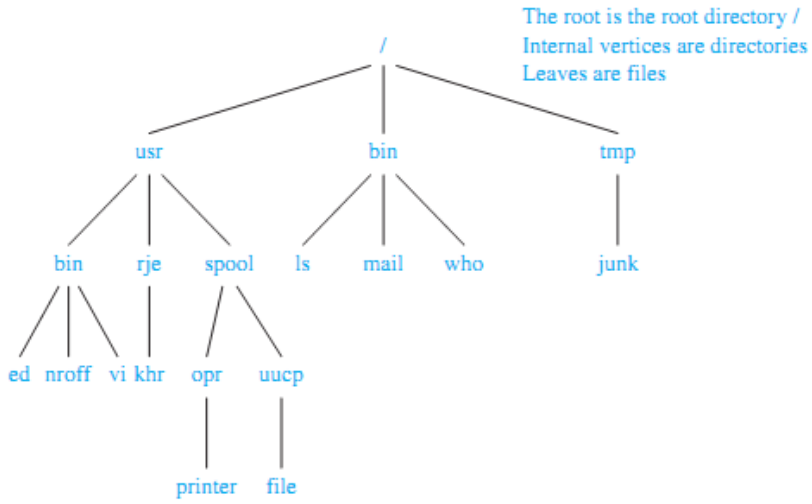
A directory can contain both files and subdirectories.

The root directory contains the entire file system.

Thus, a file system may be represented by a rooted tree, where the root represents the root directory, internal vertices represent subdirectories, and leaves represent ordinary files or empty directories.

One such file system is shown bellow.

In this system, the file khr is in the directory rje.



A Computer File System.

EXAMPLE Tree-Connected Parallel Processors

A tree-connected network is another important way to interconnect processors.

The graph representing such a network is a complete (!!!) binary tree, that is, a full binary tree where every leaf is at the same level.

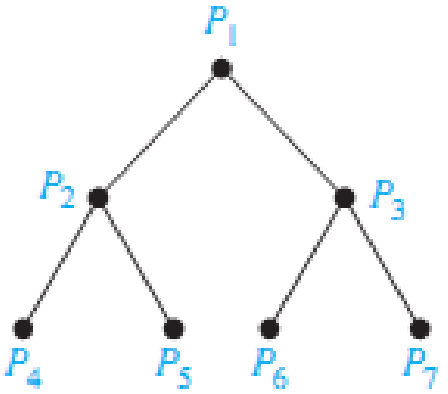
Such a network interconnects $n = 2^k - 1$ processors, where k is a positive integer.

A processor represented by the vertex v that is not a root or a leaf has three two-way connections—1 to the processor represented by the parent of v and 2 to the processors represented by the 2 children of v.

The processor represented by the root has 2 two-way connections to the processors represented by its 2 children.

A processor represented by a leaf v has a single two-way connection to the parent of v.

A tree-connected network with 7 processors is shown bellow.



A Tree-Connected Network of 7 Processors.

We now illustrate how a tree-connected network can be used for parallel computation.

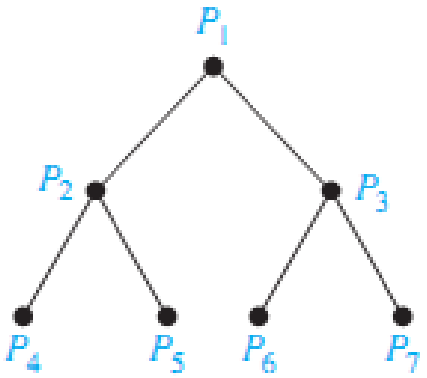
In particular, we show how the processors bellow can be used to add 8 numbers, using 3 steps.

In the first step, we add x_1 and x_2 using P_4, x_3 and x_4 using P_5, x_5 and x_6 using P_6, and x_7 and x_8 using P_7.

In the second step, we add $x_1 + x_2$ and $x_3 + x_4$ using P_2 and $x_5 + x_6$ and $x_7 + x_8$ using P_3.

Finally, in the third step, we add $x_1 + x_2 + x_3 + x_4$ and $x_5 + x_6 + x_7 + x_8$ using P_1.

The 3 steps used to add 8 numbers compares favorably to the 7 steps required to add 8 numbers serially, where the steps are the addition of one number to the sum of the previous numbers in the list.



A Tree-Connected Network of 7 Processors.

Properties of Trees

We will often need results relating the numbers of edges and vertices of various types in trees.

THEOREM 2 A tree with n vertices has $n - 1$ edges.

Proof: We will use mathematical induction to prove this theorem.

Note that for all the trees here we can choose a root and consider the tree rooted.

BASIS STEP: When $n = 1$, a tree with $n = 1$ vertex has no edges.

It follows that the theorem is true for $n = 1$.

INDUCTIVE STEP: The inductive hypothesis states that every tree with k vertices has $k - 1$ edges, where k is a positive integer.

Suppose that a tree T has $k + 1$ vertices and that v is a leaf of T (which must exist because the tree is finite), and let w be the parent of v.

Removing from T the vertex v and the edge connecting w to v produces a tree T' with k vertices, because the resulting graph is still connected and has no simple circuits.

By the inductive hypothesis, T' has $k - 1$ edges.

It follows that T has k edges because it has one more edge than T', the edge connecting v and w.

This completes the inductive step.

Recall that a tree is a connected acyclic undirected graph.

So, when G is an undirected graph with n vertices, Theorem 2 tells us that the 2 conditions

(i) G is connected and

(ii) G is acyclic,

imply (iii) G has $n - 1$ edges.

Also, when (i) and (iii) hold, then (ii) must also hold, and when (ii) and (iii) hold, (i) must also hold.

That is, if G is connected and G has $n - 1$ edges, then G is acyclic, so that G is a tree

and if G *is acyclic* and G has $n - 1$ edges, then G is connected, and so is a tree.

Consequently, when 2 of (i), (ii), and (iii) hold, the third condition must also hold, and G must be a tree.

THEOREM 3 A full m-ary tree with i internal vertices contains $n = mi + 1$ vertices.

Proof: Every vertex, except the root, is the child of an internal vertex.

Because each of the i internal vertices has m children, there are mi vertices in the tree other than the root. Therefore, the tree contains $n = mi + 1$ vertices.

Suppose that T is a full m-ary tree.

Let i be the number of internal vertices and l the number of leaves in this tree.

Once one of n, i, and l is known, the other 2 quantities are determined.

THEOREM 4 A full m-ary tree with

(*1*) n vertices has $i = (n - 1)/m$ internal vertices and $l = [(m - 1)n + 1]/m$ leaves,

(*2*) i internal vertices has $n = mi + 1$ vertices and $l = (m - 1)i + 1$ leaves,

(*3*) l leaves has $n = (ml - 1)/(m - 1)$ vertices and $i = (l - 1)/(m - 1)$ internal vertices.

Proof: Let n represent the number of vertices, i the number of internal vertices, and l the number of leaves.

The 3 parts of the theorem can all be proved using the equality given in Theorem 3, that is $n = mi + 1$,

together with the equality $n = l + i$, which is true because each vertex is either a leaf or an internal vertex.

We will prove part (*1*) here.

The proofs of parts (*2*) and (*3*) are similar

Solving for i in $n = mi + 1$ gives $i = (n - 1)/m$.

Then inserting this expression for i into the equation $n = l + i$ shows that $l = n - i = n - (n - 1)/m = [(m - 1)n + 1]/m$.

EXAMPLE

Suppose that someone starts a chain letter.

Each person who receives the letter is asked to send it on to 4 other people.

Some people do this, but others do not send any letters.

How many people have seen the letter, including the first person, if no one receives more than 1 letter and if the chain letter ends after there have been 100 people who read it but did not send it out?

How many people sent out the letter?

Solution: The chain letter can be represented using a 4-ary tree.

The internal vertices correspond to people who sent out the letter, and the leaves correspond to people who did not send it out.

Because 100 people did not send out the letter, the number of leaves in this rooted tree is $l = 100$.

Hence, part (*3*) of Theorem 4 shows that the number of people who have seen the letter is $n = (ml - 1)/(m - 1) = (4 \cdot 100 - 1)/(4 - 1) = 133$.

Also, the number of internal vertices is $133 - 100 = 33$, so 33 people sent out the letter.

Balanced m-ary trees

It is often desirable to use rooted trees that are "balanced" so that the subtrees at each vertex contain paths of approximately the same length.

Some definitions will make this concept clear.

The level of a vertex v in a rooted tree is the length of the unique path from the root to this vertex.

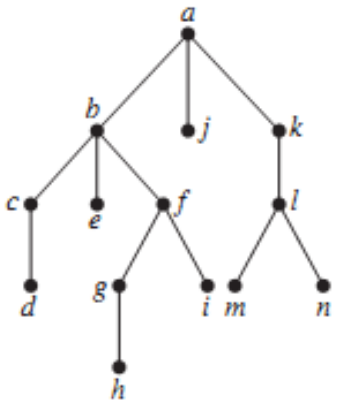
The level of the root is defined to be zero.

The height of a rooted tree is the maximum of the levels of vertices.

In other words, the height of a rooted tree is the length of the longest path from the root to any vertex.

EXAMPLE

Find the level of each vertex in the rooted tree *T*.
What is the height of this tree?
Solution: The root a is at level 0.
Vertices *b*, *j* , and *k* are at level 1.
Vertices *c*, *e*, *f* , and *l* are at level 2.
Vertices *d*, *g*, *i*, *m*, and *n* are at level 3.
Finally, vertex *h* is at level 4.
Because the largest level of any vertex is 4, this tree has height 4.



T

A rooted m-ary tree of height h is balanced if all leaves are at levels h or $h - 1$.

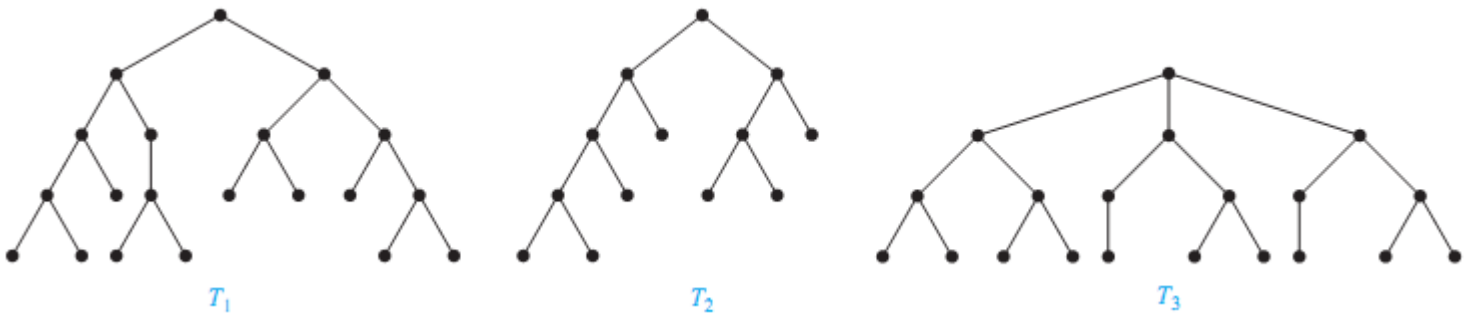
EXAMPLE 11

Which of the rooted trees T_1 , T_2 , T_3 are balanced?

Solution: T_1 is balanced, because all its leaves are at levels 3 and 4.

However, T_2 is not balanced, because it has leaves at levels 2, 3, and 4.

Finally, T_3 is balanced, because all its leaves are at level 3.



THEOREM 5 There are $\leq m^h$ leaves in an m-ary tree of height h.

Proof: The proof uses mathematical induction on the height.

First, consider m-ary trees of height 1.

These trees consist of a root with $\leq m$ children, each of which is a leaf.

Hence, there are $\leq m^1 = m$ leaves in an m-ary tree of height 1.

This is the basis step of the inductive argument.

A BOUND FOR THE NUMBER OF LEAVES IN AN *m*-ARY TREE

Now assume that the result is true for all m-ary trees of height $< h$; this is the inductive hypothesis.

Let T be an m-ary tree of height h.

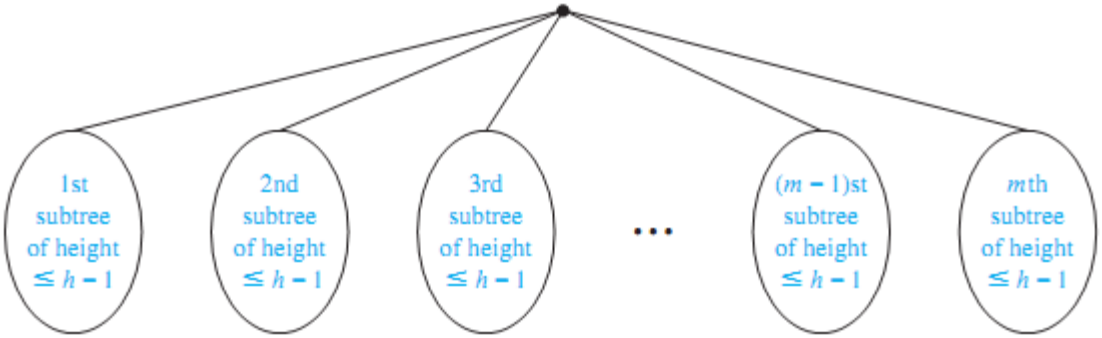
The leaves of T are the leaves of the subtrees of T obtained by deleting the edges from the root to each of the vertices at level 1, as shown in Figure .

Each of these subtrees has height $\leq h - 1$.

So by the inductive hypothesis, each of these rooted trees has $\leq m^{h-1}$ leaves.

Because there are at most m such subtrees, each with a maximum of m^{h-1} leaves, there are $\leq m \cdot m^{h-1} = m^h$ leaves in the rooted tree.

This finishes the inductive argument.



The Inductive Step of the Proof.

COROLLARY 1 If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$.
If the m-ary tree is full and balanced, then $h = \lceil \log_m l \rceil$.
(We are using the ceiling function here.
Recall that $\lceil x \rceil$ is the smallest integer $\geq x$.)
Proof: We know that $l \leq m^h$ from Theorem 5.
Taking logarithms to the base m shows that $\log_m l \leq h$.
Because h is an integer, we have $h \geq \lceil \log_m l \rceil$.
Now suppose that the tree is balanced.
Then each leaf is at level h or $h - 1$, and because the height is h, there is at least 1 leaf at level h.
It follows that there must be more than m^{h-1} leaves .
Because $l \leq m^h$, we have $m^{h-1} < l \leq m^h$.
Taking logarithms to the base m in this inequality gives $h - 1 < \log_m l \leq h$.
Hence, $h = \lceil \log_m l \rceil$.