Foundations of Discrete Mathematics

Chapters 11 and 12

By Dr. Dalia M. Gil, Ph.D.

Trees

- Tree are useful in computer science, where they are employed in a wide range of algorithms.
- They are used to construct efficient algorithms for locating items in a list.

Trees

- Trees can be used to construct efficient code saving cost in data transmission and storage.
- Trees can be used to study games such as checkers and chess an can help determine winning strategies for playing these games.

Trees

- Trees can be used to model procedures carried out using a sequence of decisions.
- Constructing these models can help determine the computational complexity of algorithms based on a sequence of decisions, such a sorting algorithms.

Trees

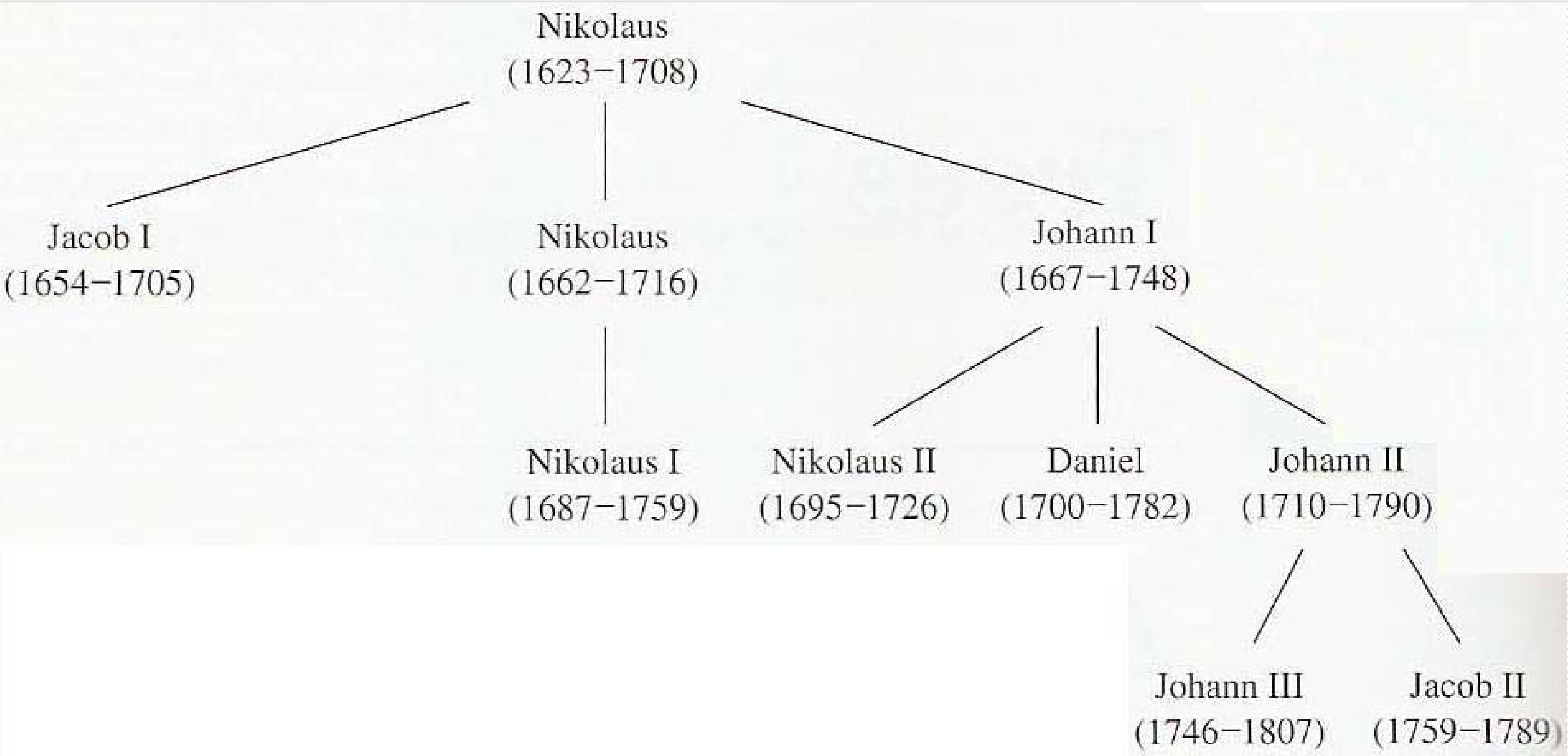
- Procedures for building trees including
- Depth-first search,
- Breadth-first search,

can be used to systematically explore the vertices of a graph.

Trees

- A tree is a connected undirected graph with no simple circuits.
- A tree cannot contain multiple edges or loops.
- Any tree must be a simple graph.

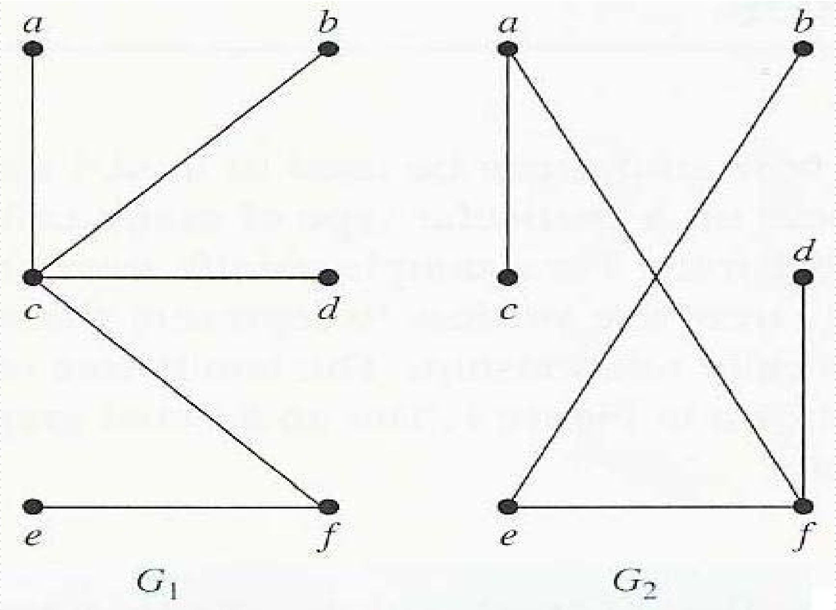
An Example of a Tree



The Bernoulli Family of Mathematicians

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 632

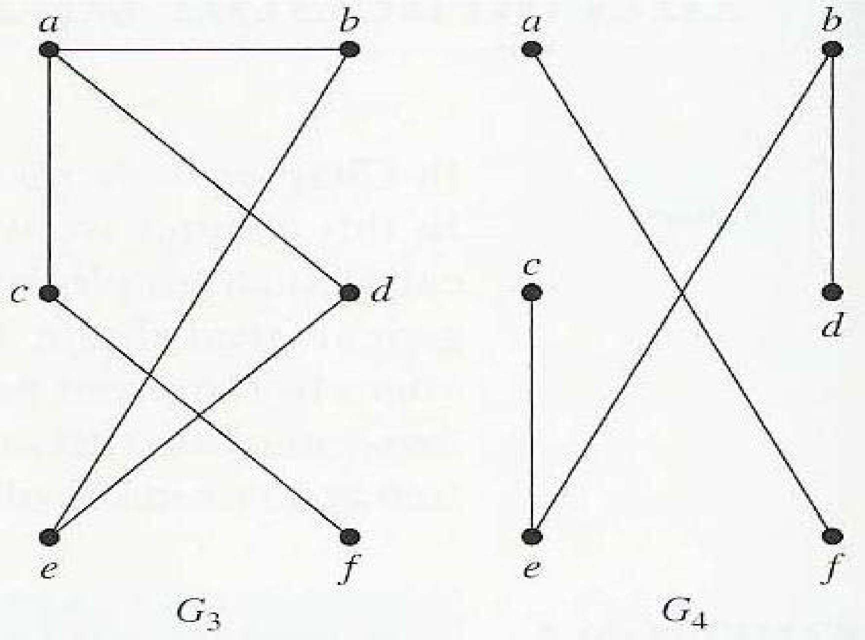
Example: Trees



- G_1 and G_2 are trees.
- Both are connected graph with no simple circuits.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 632

Example: Not Trees



- G_3 is not a tree.

  e, b, a, d, e is a simple circuit.

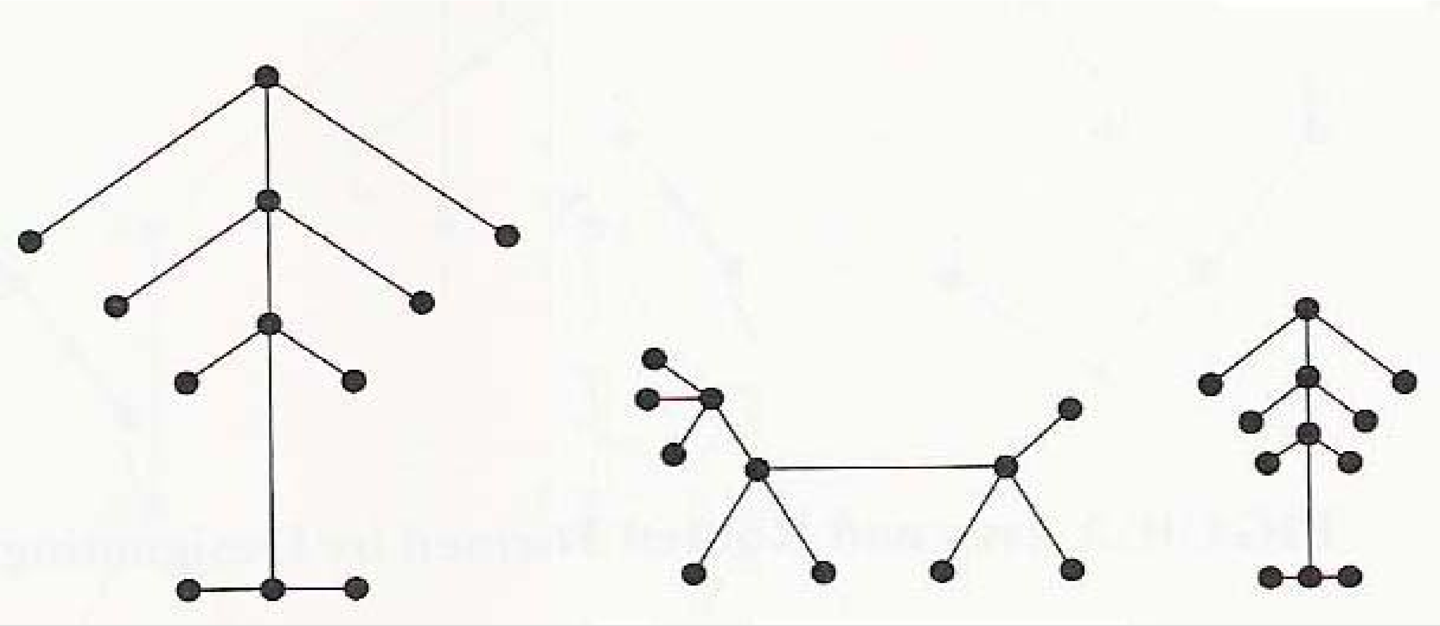
- G_4 is not a tree. It is not connected.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 632

Forest

- A Forest is a graph containing no simple circuits that are not necessarily connected.
- Forests have the property that each of their connected components is a tree.

Example: Forest



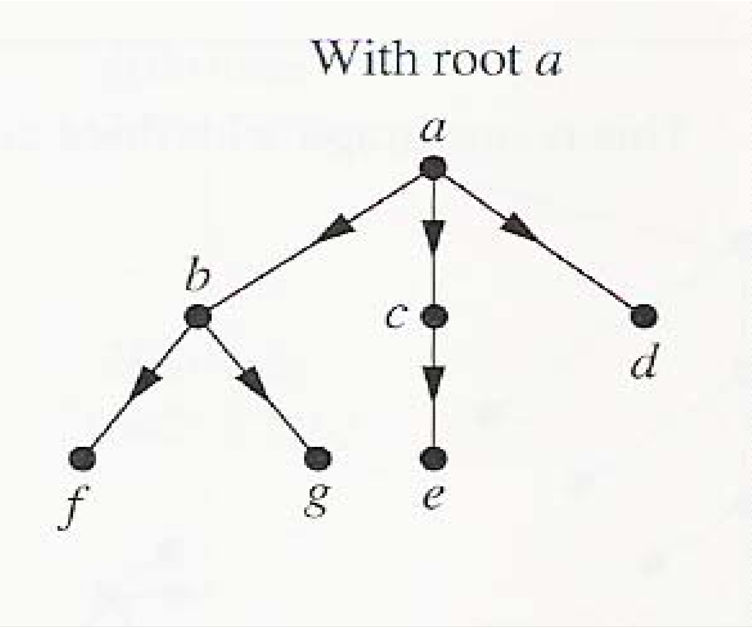
A one graph with three connected components

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 633

Theorem

- An undirected graph is a tree if and only if there is a unique simple path between any two vertices.

A Rooted Tree



- 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.

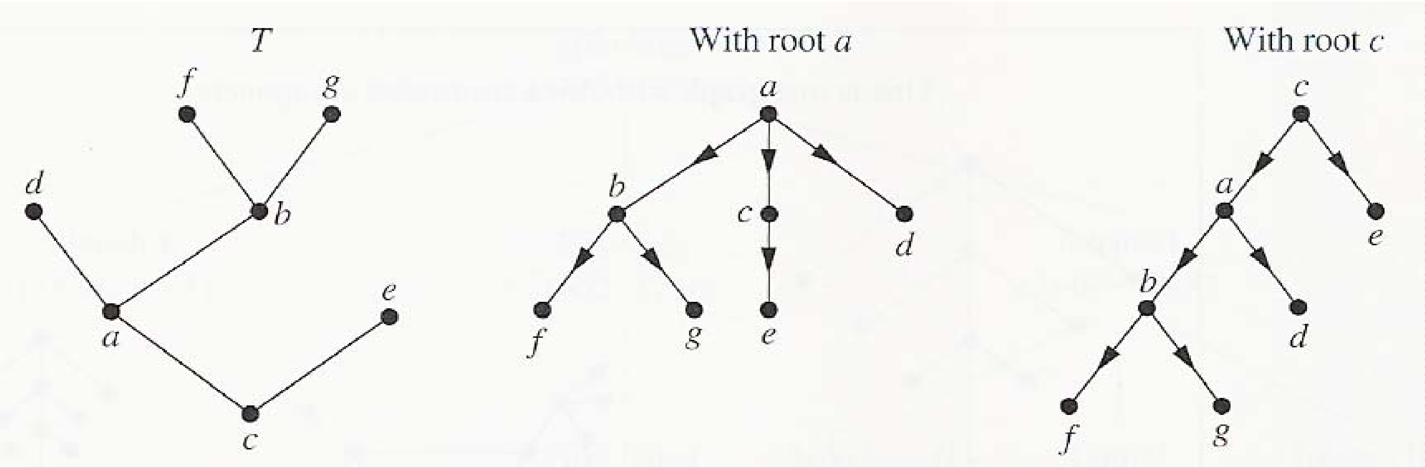
“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 634

Rooted Trees

- We can change an unrooted tree into a rooted tree by choosing any vertex as the root.
- Different choices of the root produce different trees.

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 578

Example: Rooted Trees



- The rooted trees formed by designating a to be the root and c to be the root, respectively, in the tree T.

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 634

The Terminology for Trees

- Suppose that T is a rooted tree. If v is a vertex in T other than the root.
- The **parent** of v is the unique vertex u such that there is a directed edge from u to v.

The Terminology for Trees

- When u is the parent of v, v is called a **child** of u.
- Vertices with the same parent are called **siblings**.

The Terminology for Trees

- The **ancestors** 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.
- The **descendants** of a vertex v are those that have v as an ancestor.

The Terminology for Trees

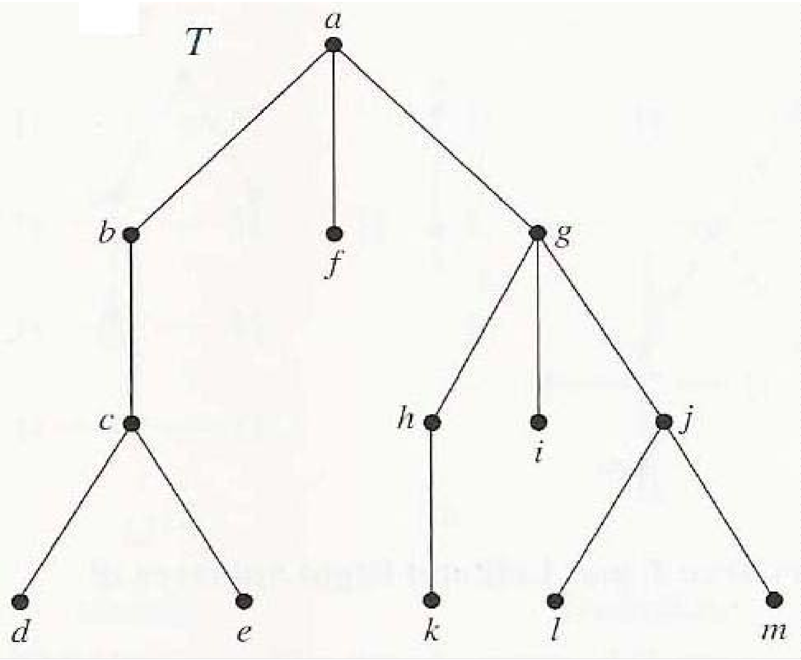
- A vertex of a tree is called a **leaf** if it has no children.
- Vertices that have children are called **internal vertices**.
- The root is an internal vertex unless it is the only vertex in the graph, in which case it is a leaf.

The Terminology for Trees

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

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 582

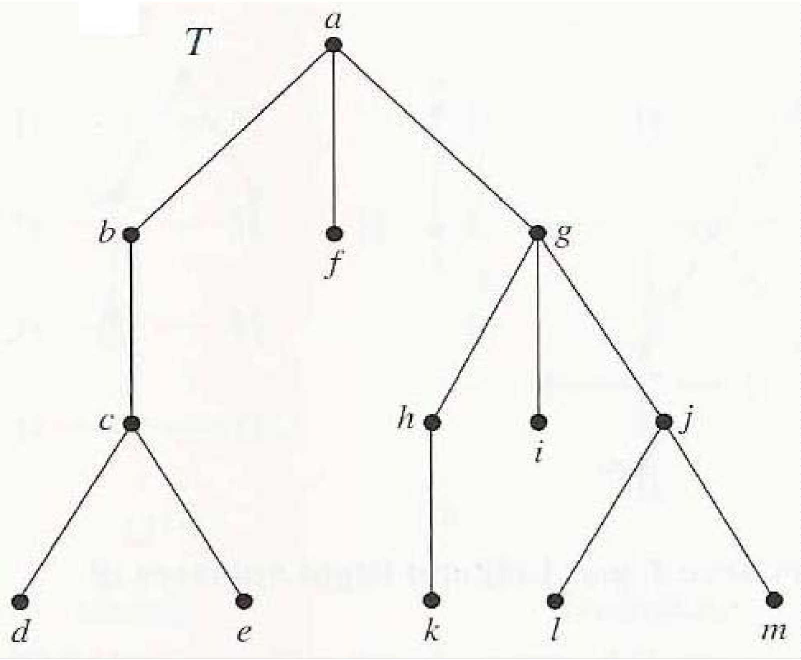
Example: Using Terminology



- T is a rooted tree with root a.
- The parent of vertex c is b.
- The children of g are h, i, and j.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

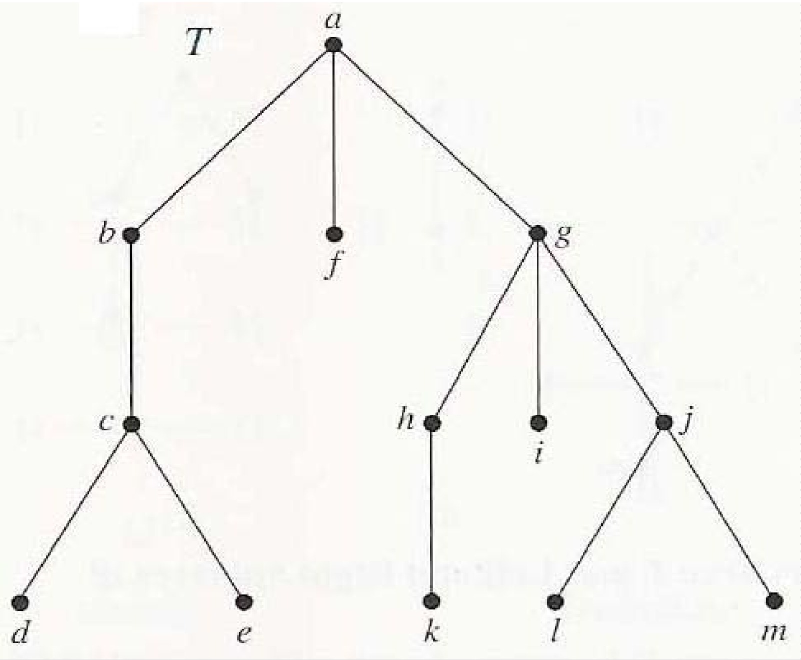
Example: Using Terminology



- The siblings of h are i and j.
- The ancestors of e are c, b, and a.
- The descendant of b are c, d, and e.

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

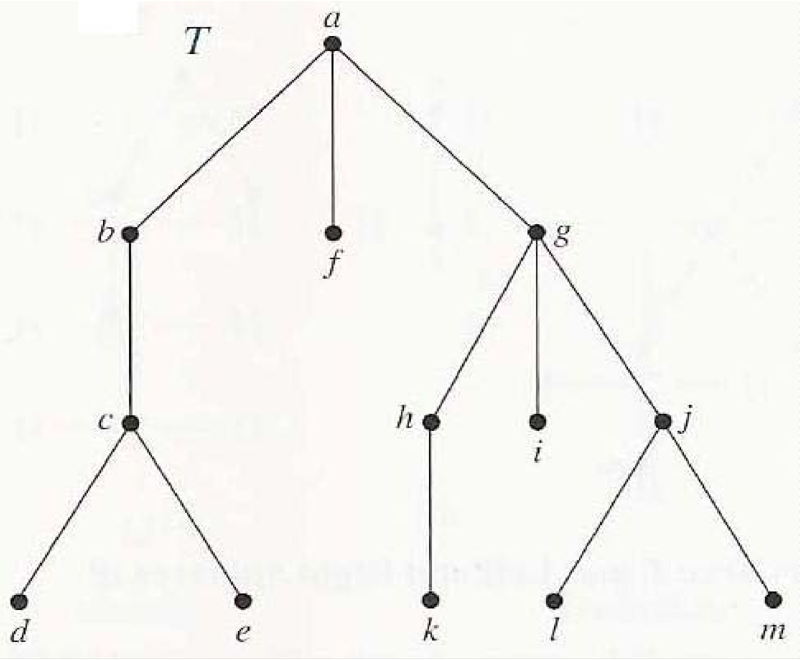
Example: Using Terminology



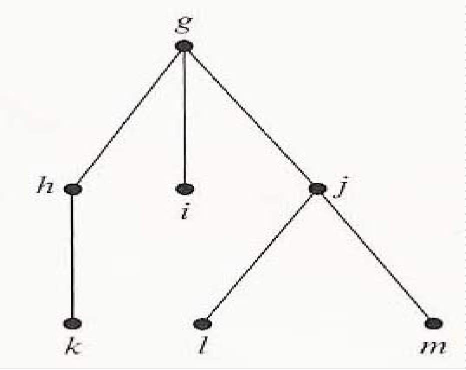
- ☐ The internal vertices are a, b, c, g, h, and j.
- ☐ The leaves are d, e, f, i, k, l, and m.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

Example: Using Terminology



- The subtree rooted at g is



“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

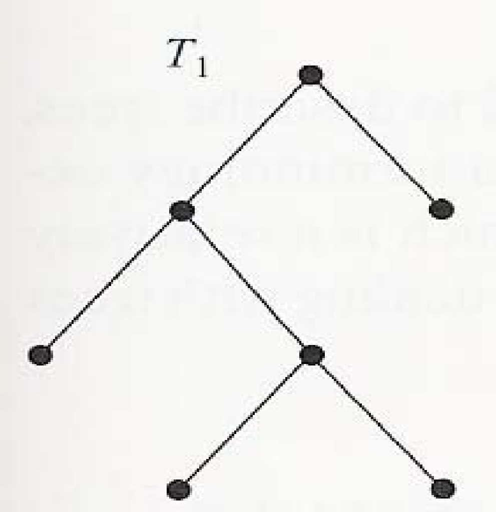
m-ary Tree

- A rooted tree is called m-ary tree 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.

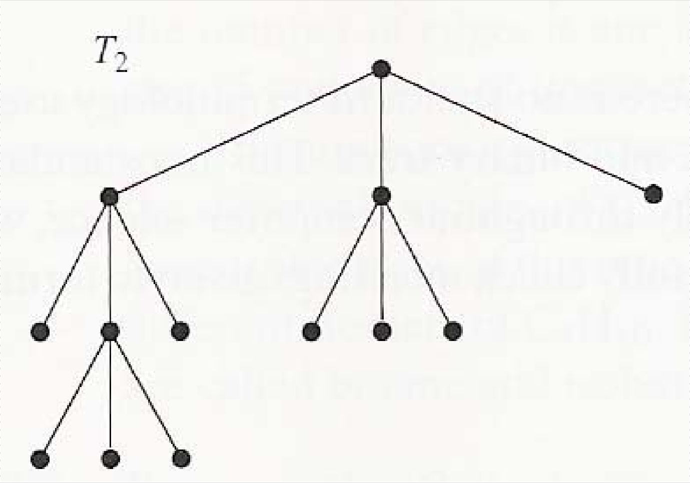
Example of m-ary tree



- T_1 is a full binary tree.
- Each of its internal vertices has two children

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

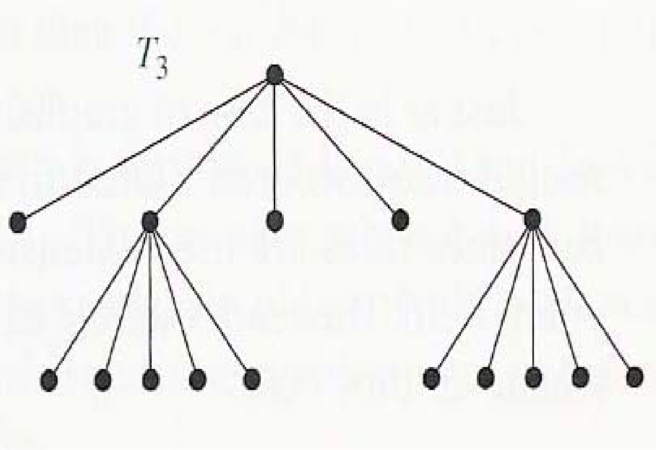
Example of m-ary tree



- T_2 is a full 3-ary tree.
- Each of its internal vertices has three children

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

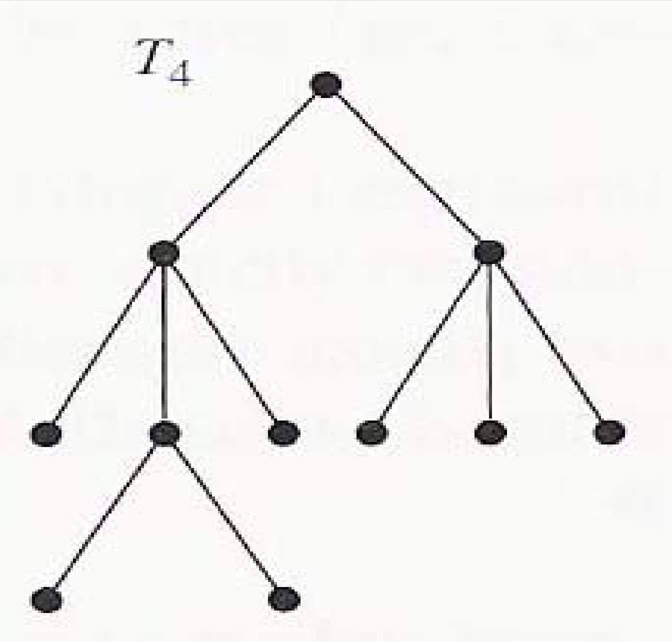
Example of m-ary tree



- T_3 is a full 5-ary tree.
- Each of its internal vertices has five children

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

Example of m-ary tree



- T_4 is not a full m-ary tree for any m.
- Some of its internal vertices has 2 children and others have 3.

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 635

Ordered Rooted Tree

- In an ordered rooted tree 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.

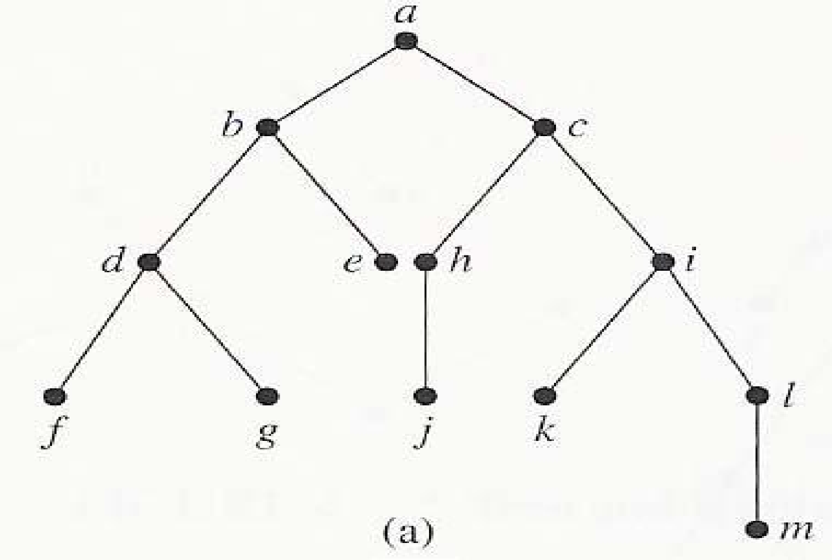
Ordered Binary Tree

- In ordered binary tree (a **binary tree**), an internal vertex has two children.
- The first child is called the **left child** and
- the second child is called the **right child**.

Ordered Binary Tree

- 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.

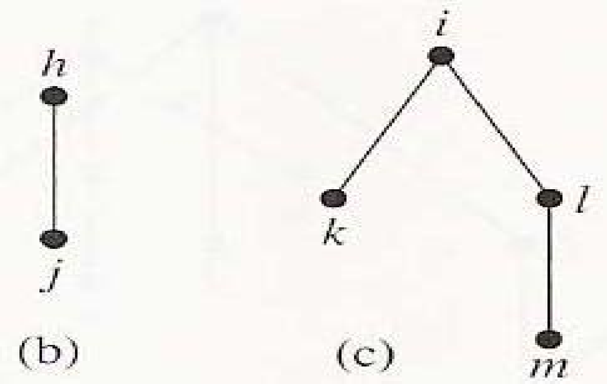
Example



(a)

- The left child of d is f and the right child is g.

- The left and right subtrees of c are



(b) (c)

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 636

Properties of Trees

- A tree with n vertices has n – 1 edges.
- A full m-ary tree with i internal vertices contains $n = m*i + 1$ vertices.
- There are at most m^h leaves in an m-ary tree of hight h.

Where n: vertices, i: internal vertices.

Properties of a Full m-ary Tree

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

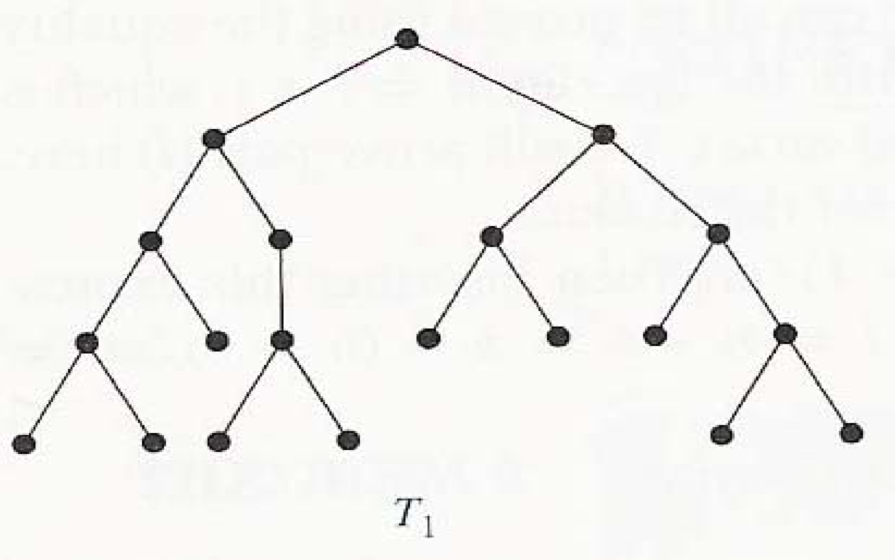
2. i internal vertices has $n = m*i + 1$ vertices and $l = (m - 1)i + 1$ leaves.

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

- l : leaves, m: children, n: vertices, i:int.vertices

Properties of Trees

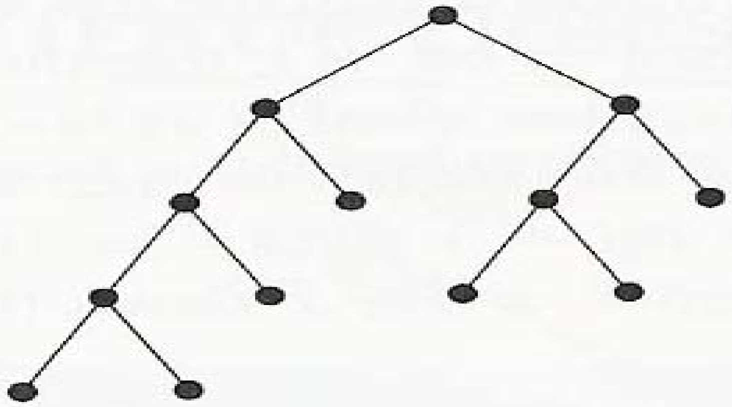
- A rooted m-ary tree of height ***h*** is balanced if all leaves are at levels h or h – 1.



- T_1 is **balanced.**
- All its leaves are at levels 3 and 4.

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 640

Properties of Trees

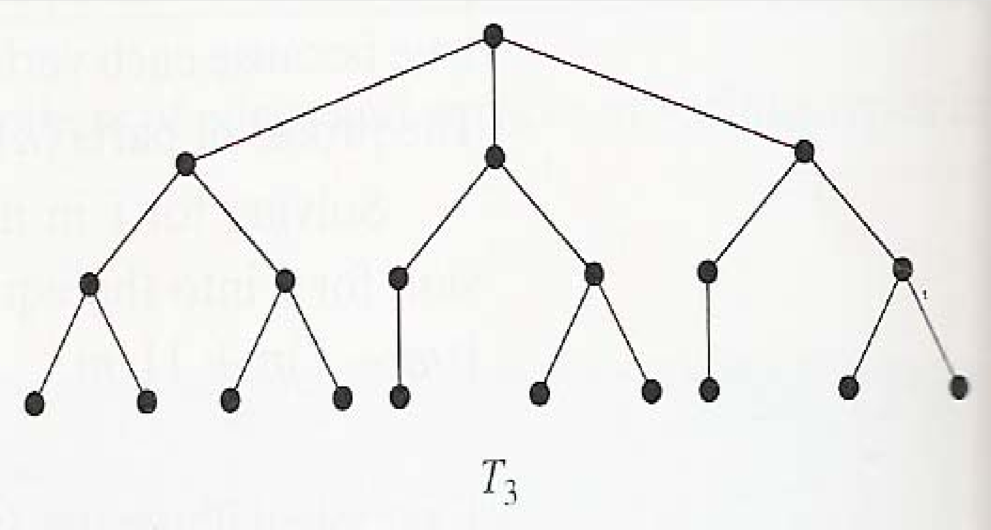


T_2

- T_2 is not **balanced**.
- It has leaves at levels 2, 3, and 4.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 640

Properties of Trees



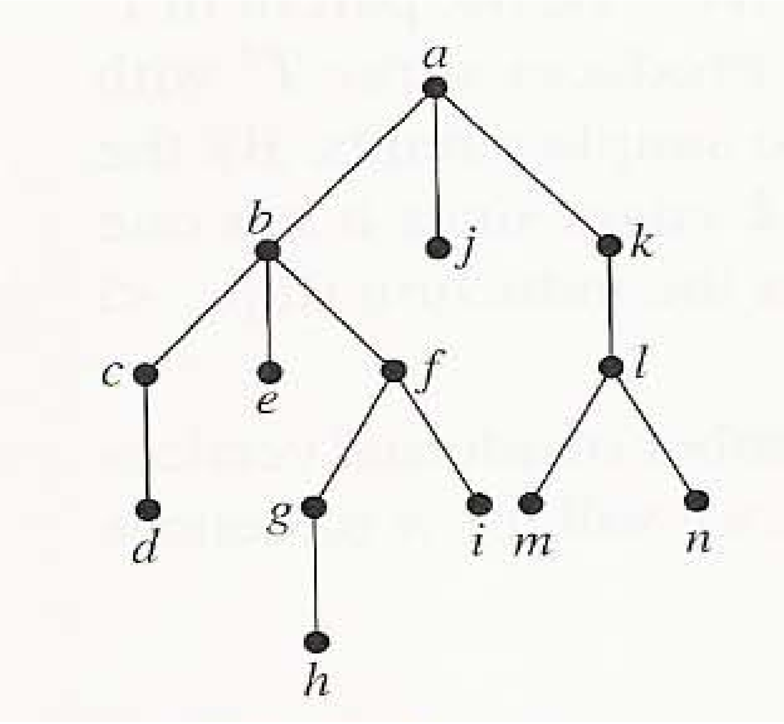
T_3

- T_3 is **balanced**.

  All its leaves are at level 3.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 640

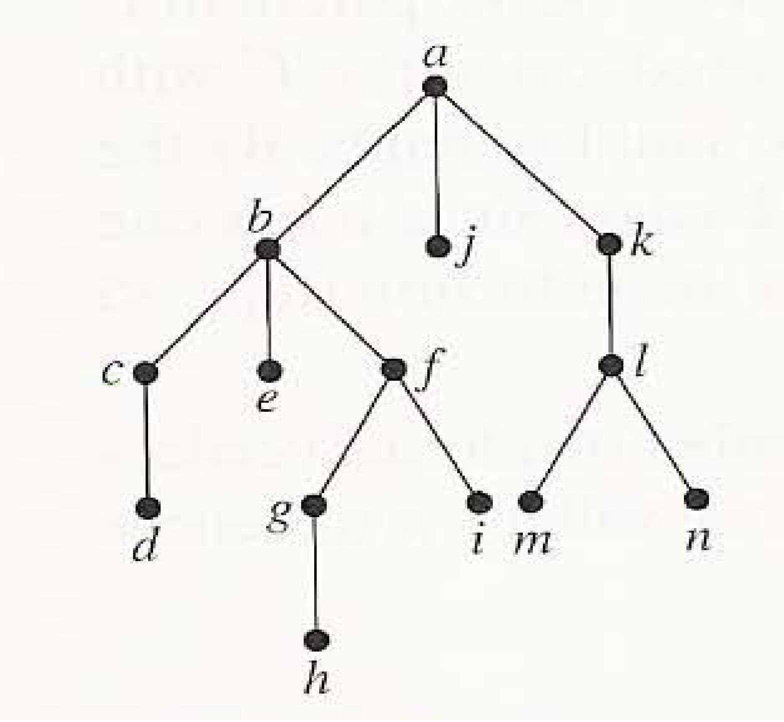
Example: Properties of Trees



- 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.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 640

Example: Properties of Trees



- Vertices d, g, i, m, and n are at level 3.
- Vertex h is at level 4.
- This tree has height 4.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 640

Spanning Trees

- A spanning tree of a connected graph G is a subgraph that is a tree and that includes every vertex of G.

- A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.

Prim’s Algorithm

- Prim’s algorithm constructs a minimum spanning tree.
- Successively add to the tree edges of minimum weight that are incident to a vertex already in the tree and not forming a simple circuit with those edges already in the tree.
- Stop when n – 1 edges have been adding.

Prim’s Algorithm

- Step 1: Choose any vertex v and let e_1 be and edge of least weight incident with v. Set k = 1.

- Step 2: While $k < n$

  If there exists a vertex that is not in the subgraph T whose edges are $e_1, e_2, \ldots, e_k$,

- Let e_{k+1} be an edge of least weight among all edges of the form ux, where u is a vertex of T and x is a vertex not in T;

Prim’s Algorithm (cont.)

- Let e_{k+1} be an edge of least weight among all edges of the form ux, where u is a vertex of T and x is a vertex not in T;
- Replace k by k + 1;

else output e_1, e_2, …, e_k and stop.

end while.

Example Using Prim’s Algorithm

- Use Prim’s algorithm to design a minimum-cost communication network connecting all the computers represented by the following graph

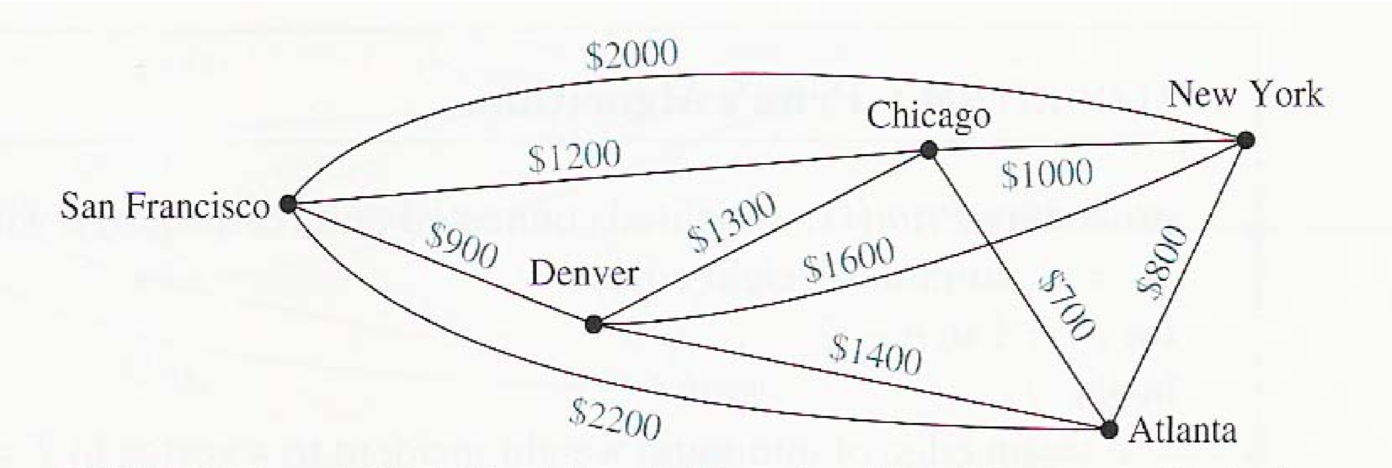


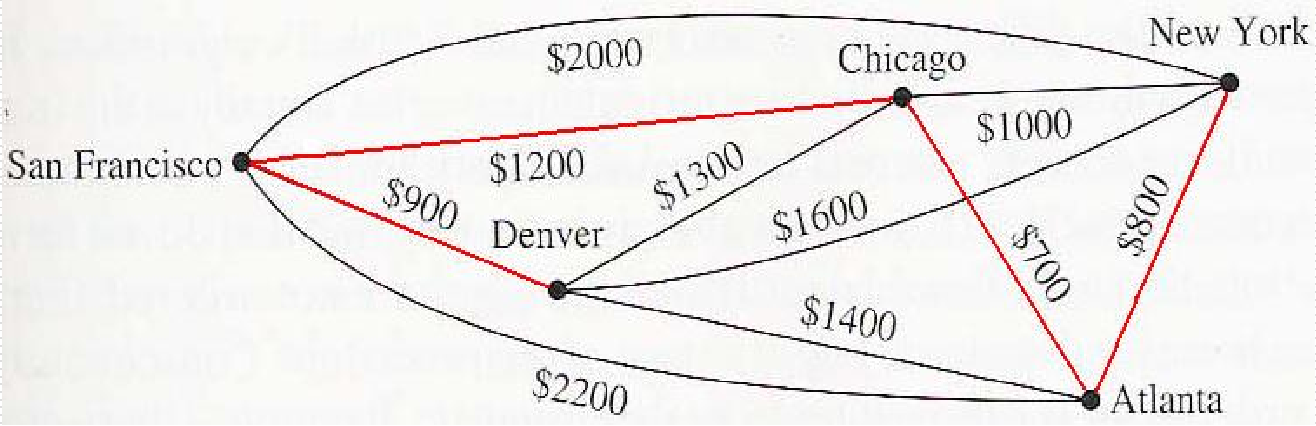
FIGURE 1 **A Weighted Graph Showing Monthly Lease Costs for Lines in a Computer Network.**

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 689

Example Using Prim's Algorithm

- Choosing and initial edge of minimum weight.
- Successively adding edges of minimum weight that are incident to a vertex in a tree and do not form a simple circuit.

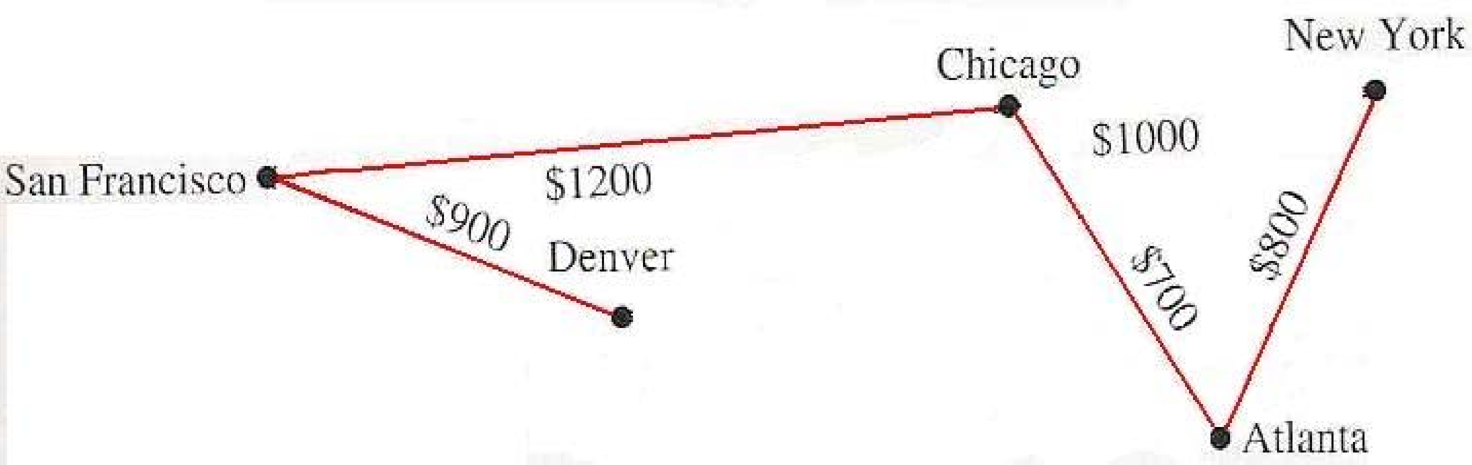
Example Using Prim’s Algorithm



Choice	Edge	Cost
1	{Chicago, Atlanta}	$700
2	{Atlanta, New York}	$800
3	{Chicago, San Francisco}	$1200
4	San Francisco, Denver}	$900
	Total:	$3600

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 691

Example Using Prim's Algorithm

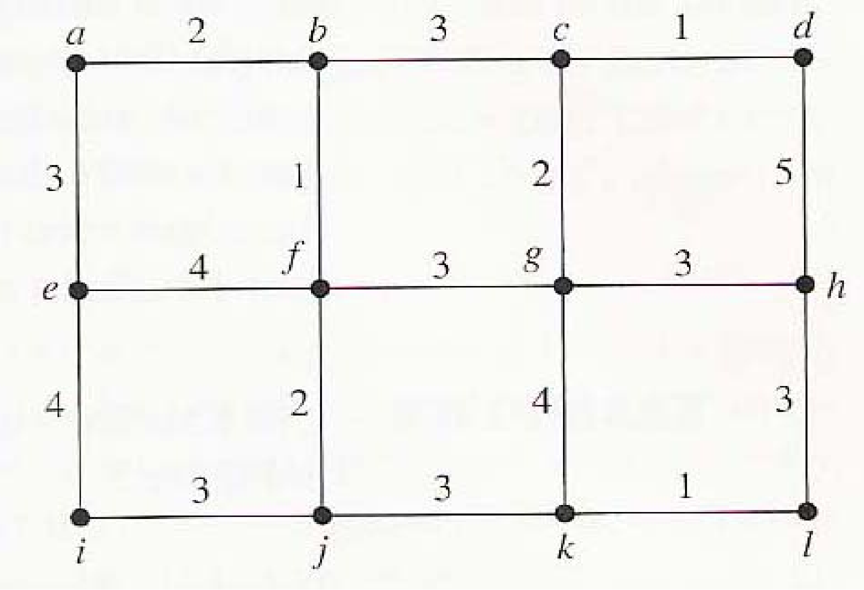


Choice	Edge	Cost
1	{Chicago, Atlanta}	$700
2	{Atlanta, New York}	$800
3	{Chicago, San Francisco}	$1200
4	San Francisco, Denver}	$900
	Total:	$3600

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 691

Example Using Prim’s Algorithm

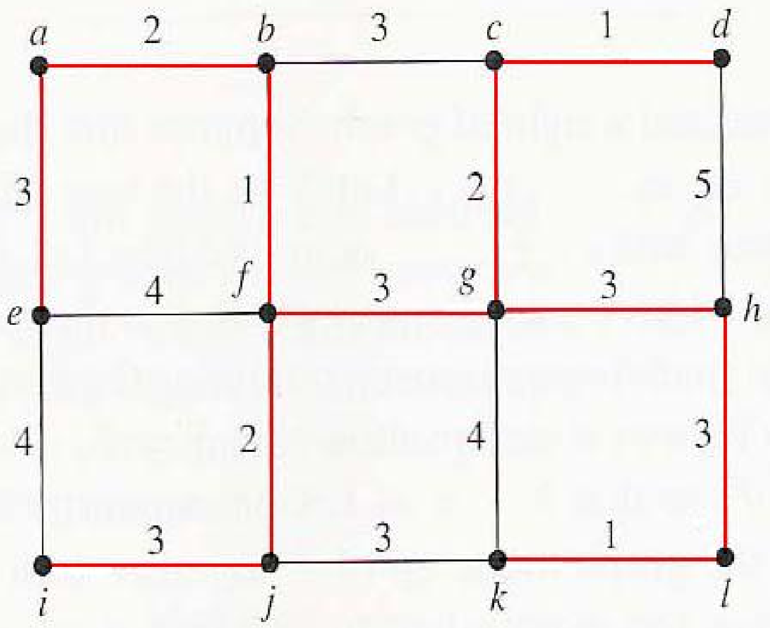
- Use Prim’s algorithm to find a minimum spanning tree in the following graph.



“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 691

Example Using Prim's Algorithm

- Use Prim's algorithm to find a minimum spanning tree in the following graph.

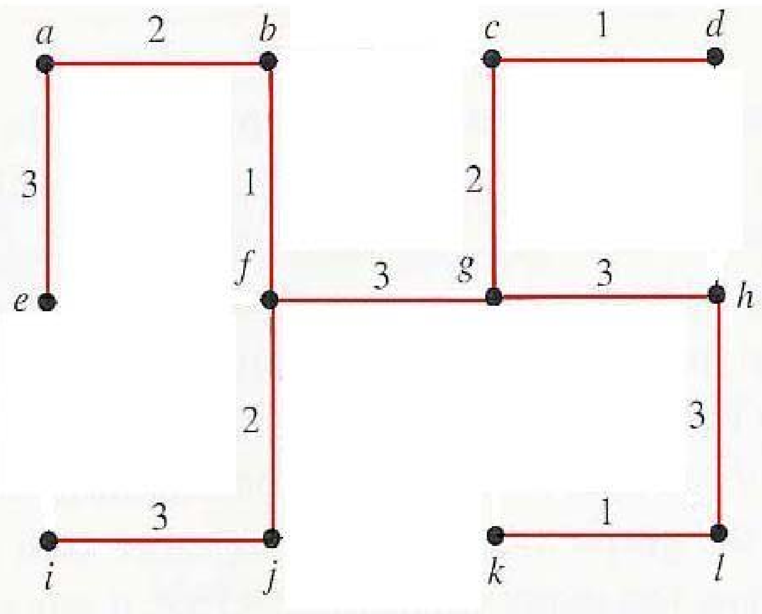


Choice	Edge	Weight
1	$\{b, f\}$	1
2	$\{a, b\}$	2
3	$\{f, j\}$	2
4	$\{a, e\}$	3
5	$\{i, j\}$	3
6	$\{f, g\}$	3
7	$\{c, g\}$	2
8	$\{c, d\}$	1
9	$\{g, h\}$	3
10	$\{h, l\}$	3
11	$\{k, l\}$	1
	Total:	24

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 691

Example Using Prim's Algorithm

- Use Prim's algorithm to find a minimum spanning tree in the following graph.



Choice	Edge	Weight
1	{*b*, *f*}	1
2	{*a*, *b*}	2
3	{*f*, *j*}	2
4	{*a*, *e*}	3
5	{*i*, *j*}	3
6	{*f*, *g*}	3
7	{*c*, *g*}	2
8	{*c*, *d*}	1
9	{*g*, *h*}	3
10	{*h*, *l*}	3
11	{*k*, *l*}	1
	Total:	24

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 691

Kruskal’s Algorithm

- This algorithm finds a minimum spanning tree in a connected weighted graph with n > 1 vertices.

Kruskal’s Algorithm

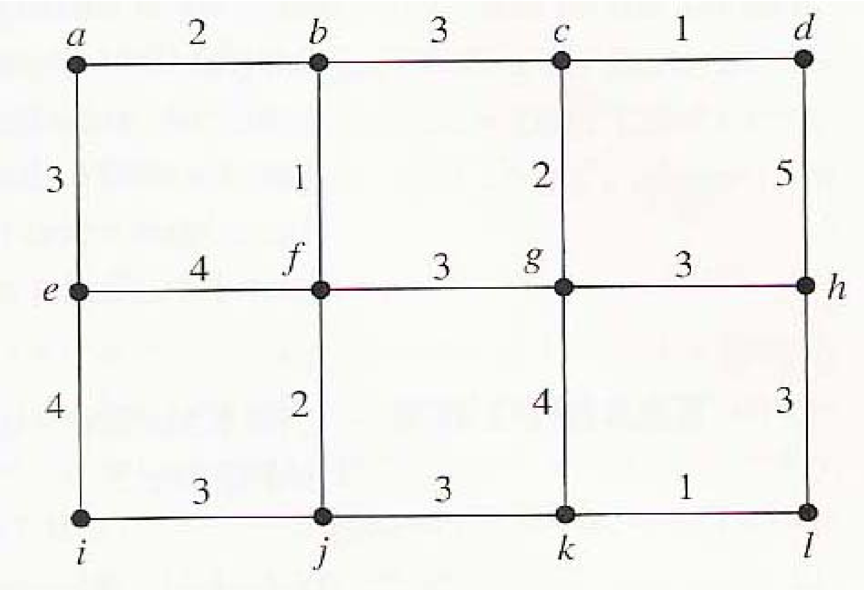
```
Step 1: Find an edge of least weight and call this $e_1$.
    Set $k = 1$.
Step 2: While $k < n$

        if there exits an edge e such that
        $\{e\} \cup \{e_1,. e_2, \ldots, e_k\}$ does not contain a circuit

          * let $e_{k+1}$ be such an edge of least weight;
            replace k by $k + 1$;
    else output $e_1, e_2, \ldots, e_k$ and stop
    end while
```

Example Using Kruskal’ Algorithm

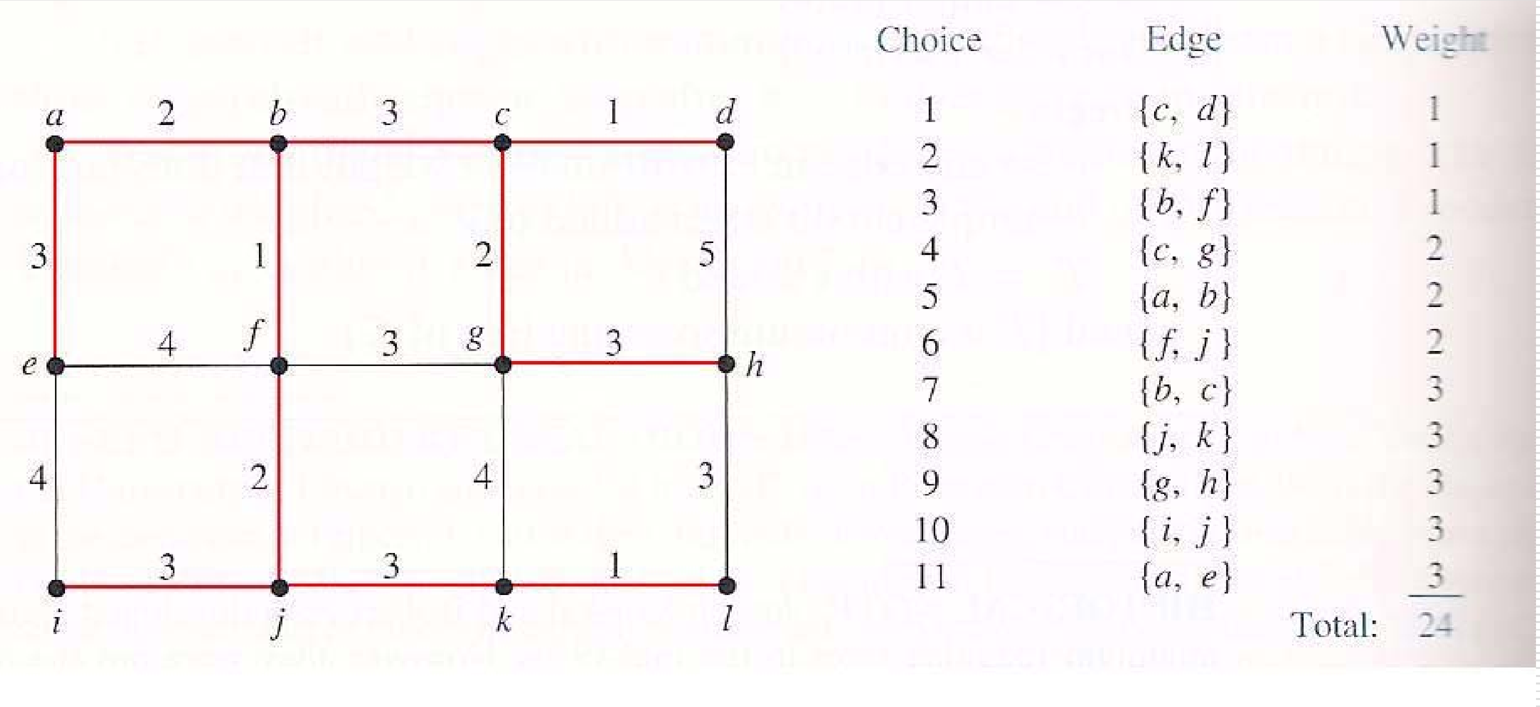
- ☐ Use Kruskal’s algorithm to find a minimum spanning tree in the following weighted graph.



“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 691

Example Using Kruskal’ Algorithm

- Use Kruskal’s algorithm to find a minimum spanning tree in the following weighted graph.

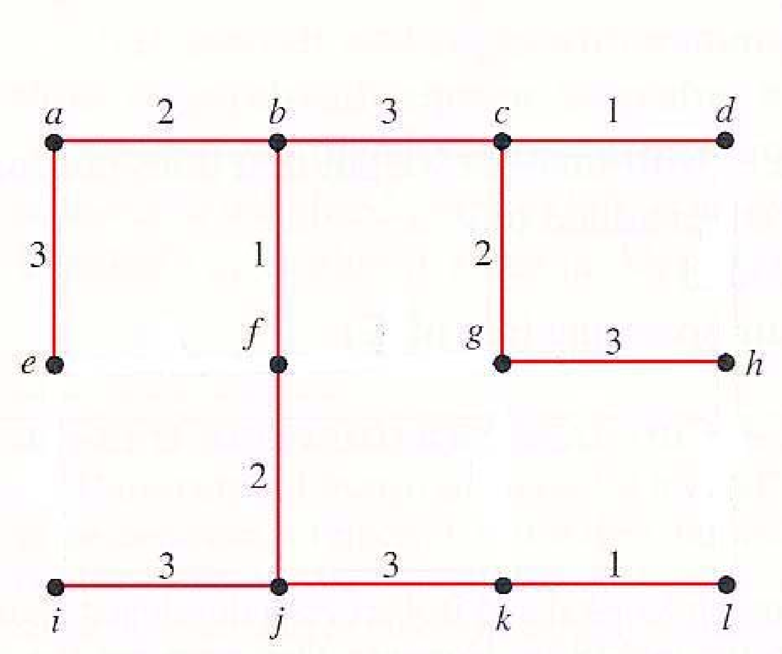


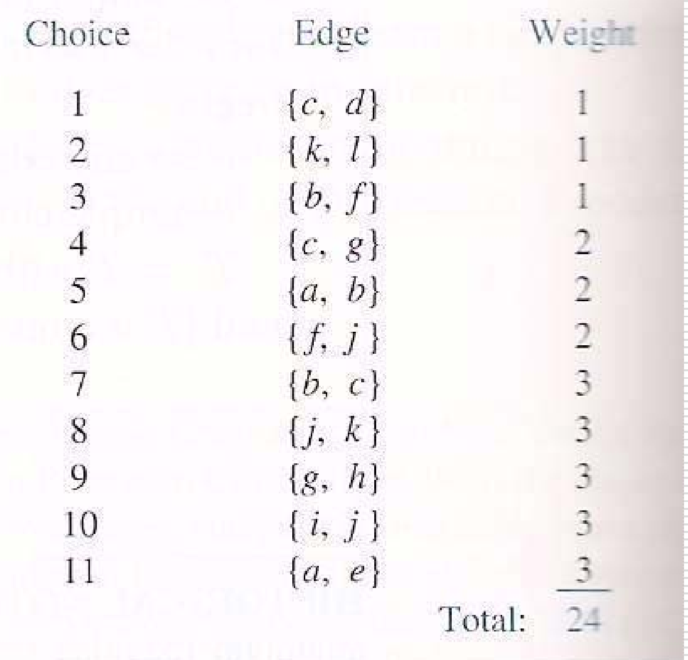
Choice	Edge	Weight
1	{*c*, *d*}	1
2	{*k*, *l*}	1
3	{*b*, *f*}	1
4	{*c*, *g*}	2
5	{*a*, *b*}	2
6	{*f*, *j*}	2
7	{*b*, *c*}	3
8	{*j*, *k*}	3
9	{*g*, *h*}	3
10	{*i*, *j*}	3
11	{*a*, *e*}	3
	Total:	24

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 692

Example Using Kruskal' Algorithm

- Use Kruskal's algorithm to find a minimum spanning tree in the following weighted graph.





Choice	Edge	Weight
1	{*c*, *d*}	1
2	{*k*, *l*}	1
3	{*b*, *f*}	1
4	{*c*, *g*}	2
5	{*a*, *b*}	2
6	{*f*, *j*}	2
7	{*b*, *c*}	3
8	{*j*, *k*}	3
9	{*g*, *h*}	3
10	{*i*, *j*}	3
11	{*a*, *e*}	3
	Total:	24

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 692

Digraphs

- A digraph is a pair (V, E) of sets, V nonempty and each element of E an ordered pair of distinct elements of V.
- The elements of V are called **vertices** and the elements of E are called **arcs**.

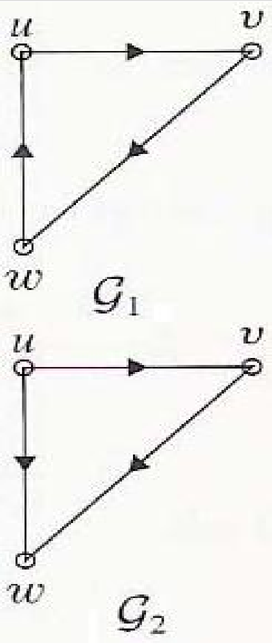
Digraphs

- The same terms can be used for graphs and digraphs.
- The exception: In a digraph we use term arc instead of edge.
- An arc is an ordered pair (u, v) or (v, u).
- An edge is an unordered pair of vertices {u, v}.

Digraphs

- The vertices of a graph have degrees, a vertex of a digraph has an indegree and outdigree.
- Indegree is the number of arcs directed into a vertex.
- Outdegree is the number of arcs directed away from the vertex.

Examples: Digraphs

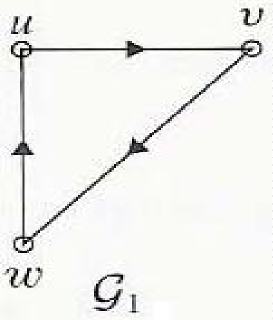


- G_1: u has outdegree 1,
  v has outdegree 1,
  w has outdegree 1
- G_2: u has outdegree 2,
  v has outdegree 1,
  w has outdegree 0

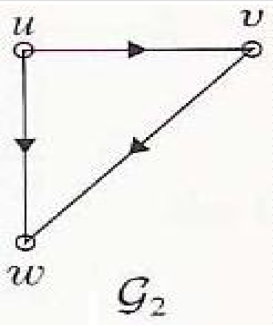
- G_1 and G_2 are not isomorphic.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 344

Examples: Digraphs



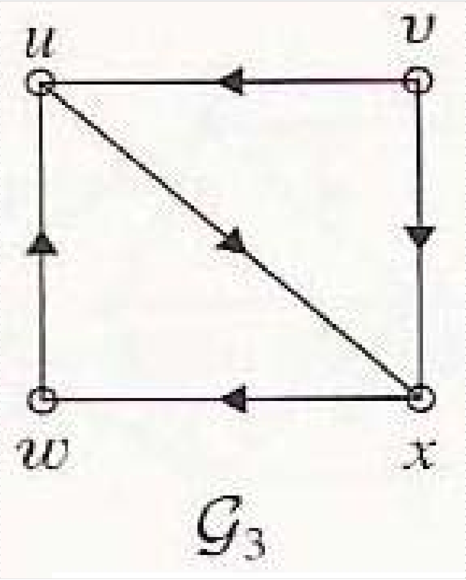
- G_1 is Eulerian because uvwu is an Eulerian circuit and a Hamiltonian cycle.



- G_2 has neither an Eulerian nor a Hamiltonian cycle, but it has a Hamiltonian path uvw.

"Discrete Mathematics with Graph Theory." Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 344

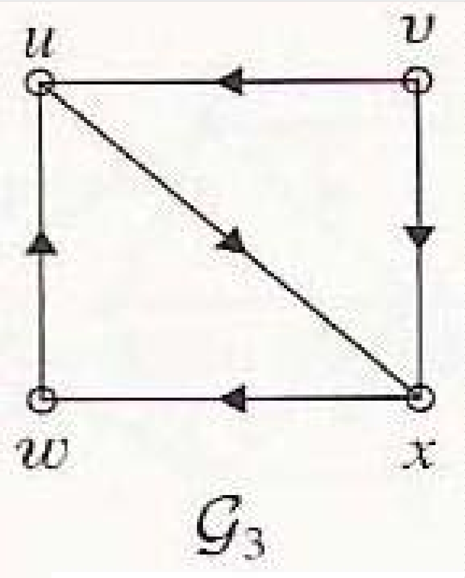
Examples: Digraphs



- G_3 has vertices u and x with indegree 2 and outdegree 1.
- Vertex v has indegree 0 and outdegree 2 and vertex w has indegree 1 and outdegree 1.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 344

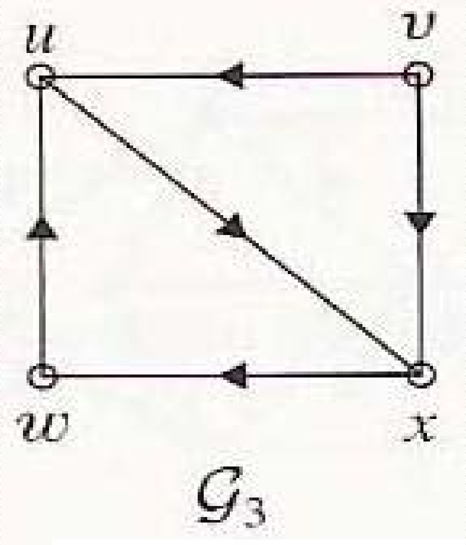
Examples: Digraphs



- The indegree sequence is 2, 2, 1, 0, and the outdegree sequence is 2, 1, 1, 1.
- The sum of the indegrees of the vertices equals the sum of the outdegrees of the vertices is the number of arcs.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 344

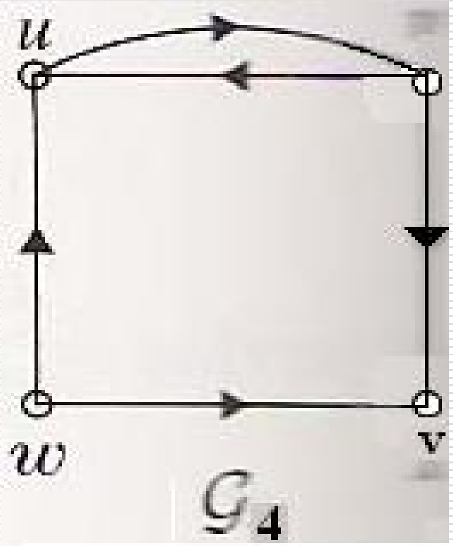
Examples: Digraphs



- G_3 is not Hamiltonian because vertex v has indegree 0.
- There is no way of reaching v on a walk respecting orientation edges, no Hamiltonian cycle can exist.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 344

Examples: Digraphs



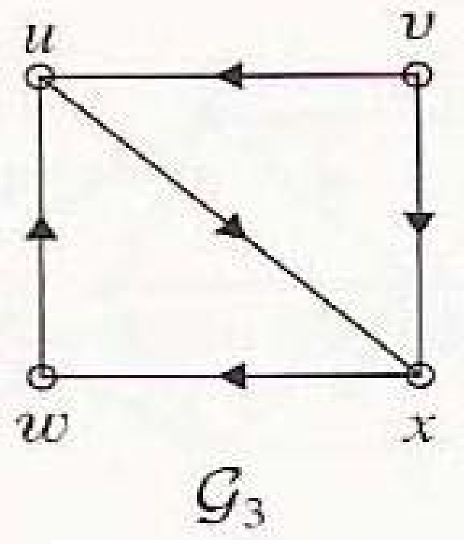
- ☐ G_4 is not Hamiltonian because vertex x has outdegree 0, so no walk respecting orientations can leave x.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 344

Digraphs

- A digraph is called strongly connected if and only if there is a walk from any vertex to any other vertex that respects the orientation of each arc.
- A digraph is Eulerian if and only if it is strongly connected and, for every vertex, the indegree equals the outdegree.

Examples: Digraphs

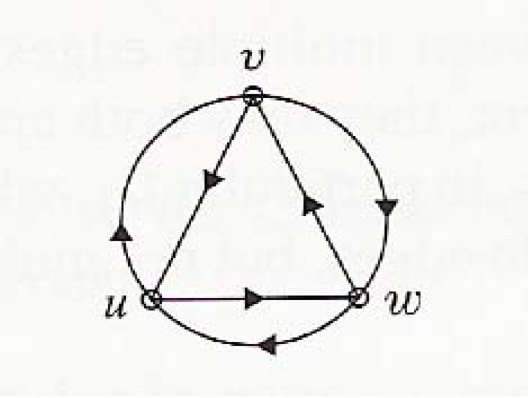


- G_3 is not Eulerian. It is not strongly connected (there is no way to reach v).
- The indegrees and outdegrees of three vertices (u, v, and x) are not the same.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 346

Examples: Digraphs

- This digraph is Eulerian.



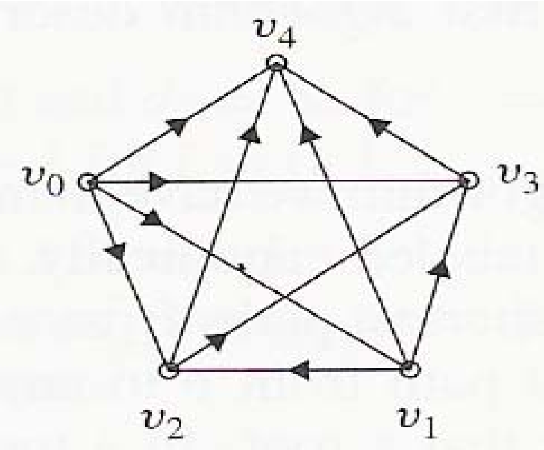
- It is strongly connected (there is a circuit uvwu that permits travel in the direction of arrows between two vertices

- The indegree and outdegree of every vertex are 2 (an Euler circuit uwvuvwu).

"Discrete Mathematics with Graph Theory." Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 346

Acyclic Digraphs

- A directed graph is a acyclic if it contains no directed cycles.

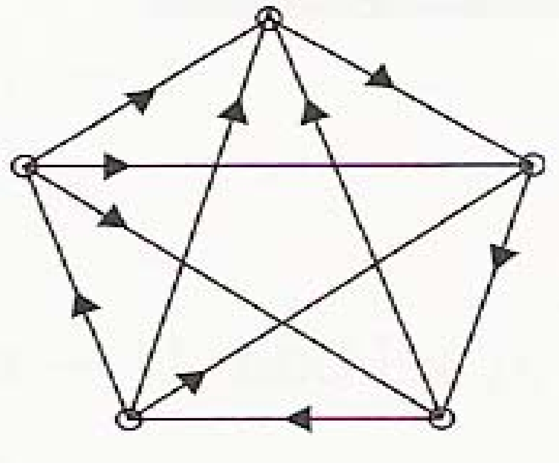


- This digraph is acyclic.
- There are no cycles.
- There is never an arc on which to return to the first vertex.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 393

Acyclic Digraphs

- A directed graph is a acyclic if it contains no directed cycles.



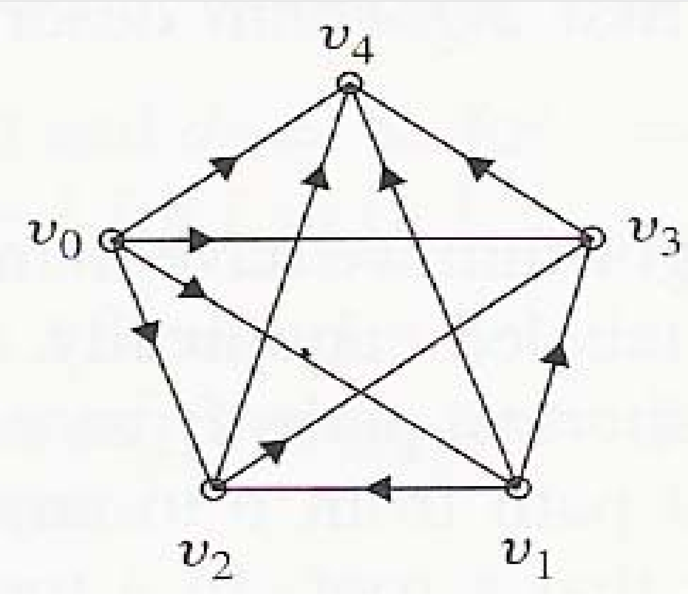
- This digraph is not acyclic.
- There are cycles.

“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 393

A Canonical Ordering

- A labeling v_0, v_1, ..., v_{n-1} of the vertices of a digraph is called canonical if the only arcs have the form v_iv_j with $i < j$.
- A canonical labeling of vertices is also called a canonical ordering.
- A digraph has a canonical ordering of vertices if and only if it is acyclic.

A Canonical Ordering



- A digraph has a canonical ordering of vertices if and only if it is acyclic.
- A digraph is acyclic if and only if it has a canonical labeling of vertices

"Discrete Mathematics with Graph Theory." Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 393

Strongly connected Orientation

- To orient or to assign an orientation to an edge in a graph is to assign a direction to that edge.

- To orient or assign an orientation to a graph is to orient every edge in the graph.

- A graph has a strongly connected orientation if it is possible to orient it in such a way that the resulting digraph is strongly connected.

Depth-First Search

- Depth-first search is a simple and efficient procedure used as the for a number of important computer algorithms in graphs.
- We can build a spanning trees for a connected simple graph using a depth-first search.

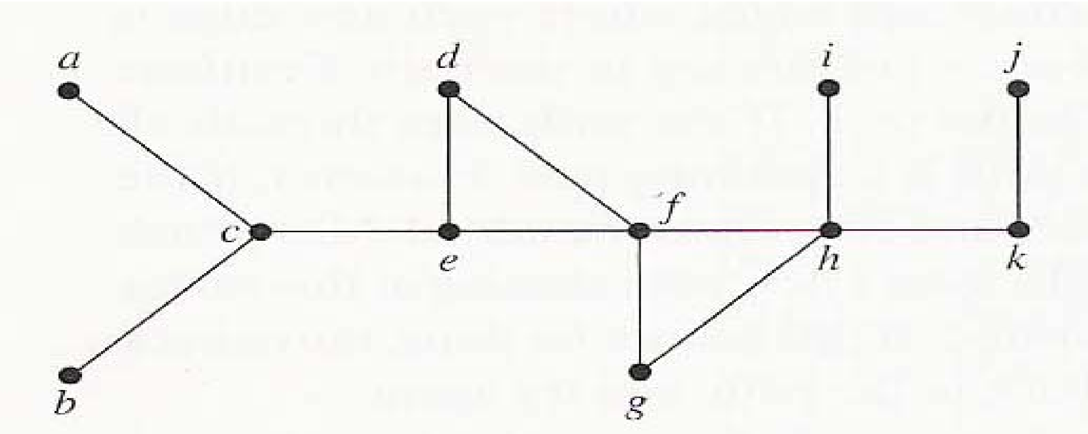
Depth-First Search Algorithm

- Let G be a graph with n vertices.

```
Step 1. Choose any vertex and label it 1. Set k = 1.
Step 2. While there are unlabeled vertices
            if there exists an unlabeld vertex adjacent to k,
                assign to it the smallest unused label l from
                the set {1, 2, ..., n} and set k = l
            else if k = 1 stop;
                else backtrack to the vertex l from which k
            was labeled and set k = l.
Step 3. end while.
```

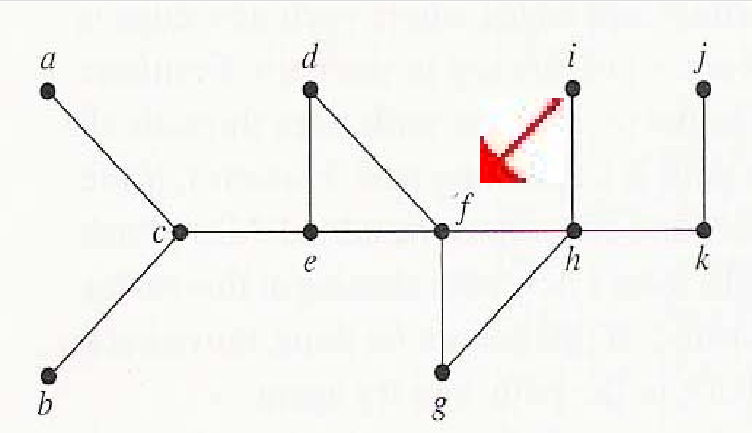
Example: Depth-First Search

- Use a depth-first search to find a spanning tree for the graph G.



“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 678

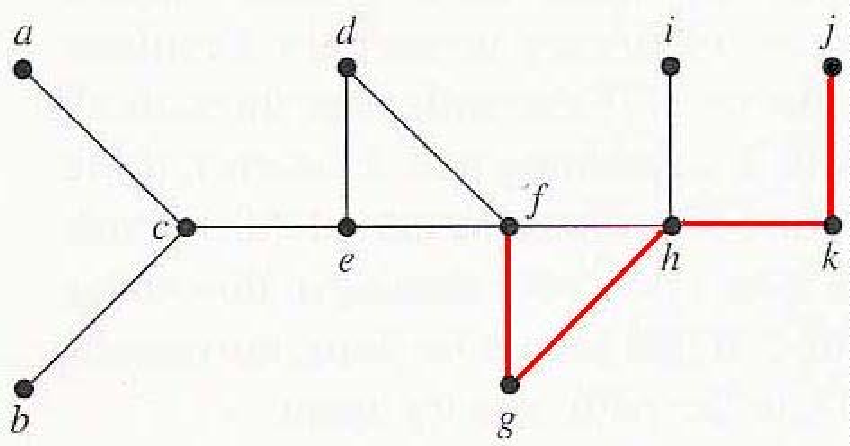
Example: Depth-First Search

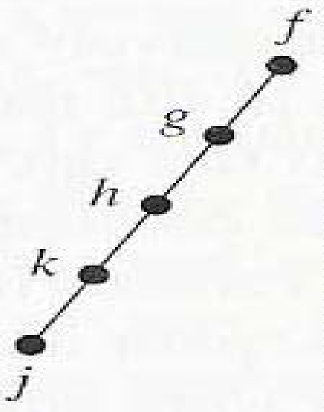


- Arbitrary start with vertex f.
- A path is built by successively adding edges incident with vertices not already in the path, as long as possible

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 678

Example: Depth-First Search

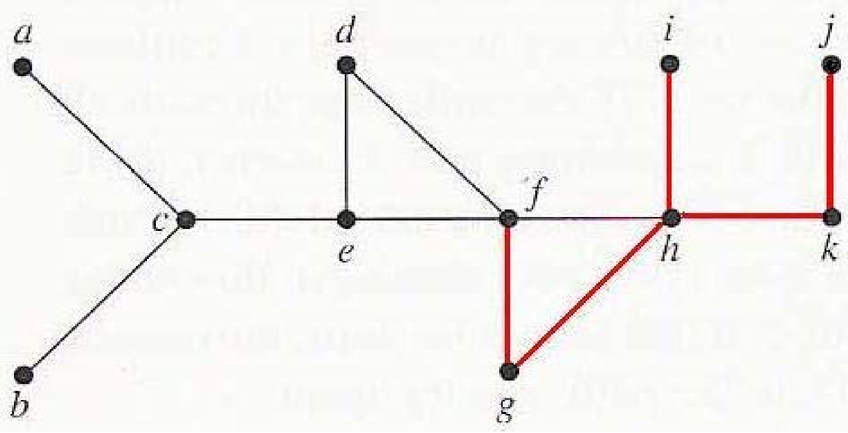


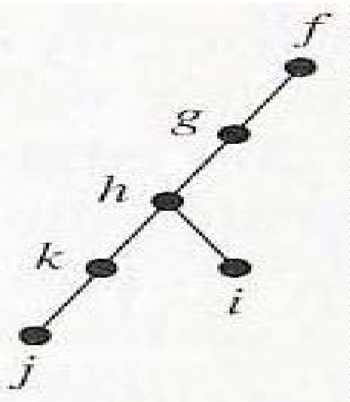


- From f create a path f, g, h, j
- (other path could have been built).

"Discrete Mathematics and its Applications." Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 678

Example: Depth-First Search

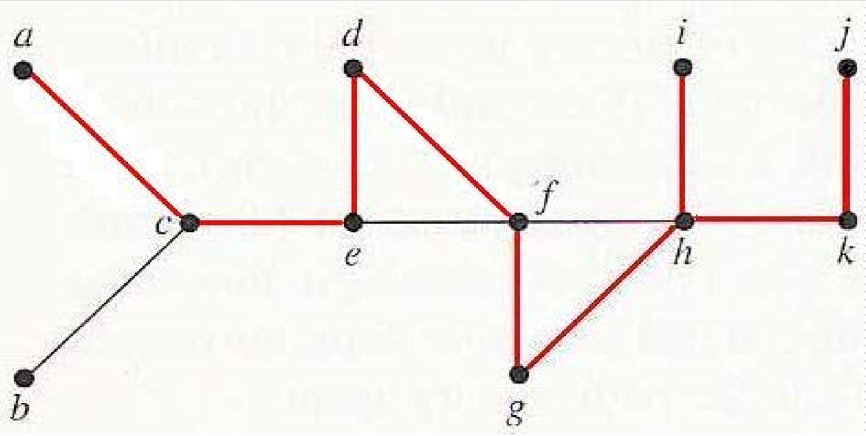


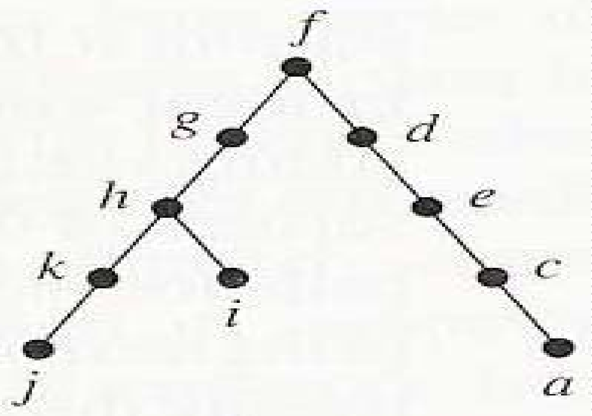


- Next, backtrack to k. There is no path beginning at k containing vertices not already visited.
- So, backtrack to h.
- Form the path h, i.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 678

Example: Depth-First Search

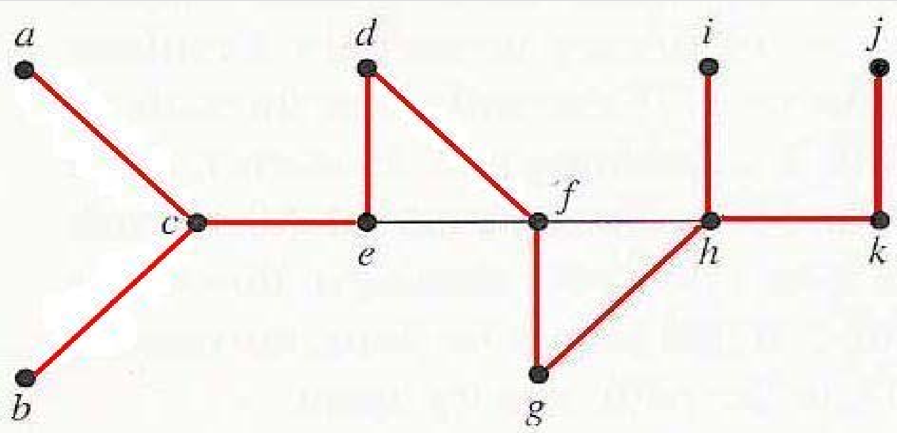


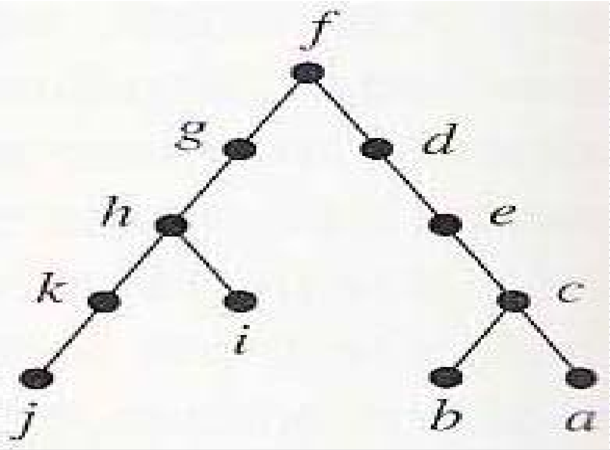


- Then, backtrack to h, and then to f.
- From f build the path f, d, e, c, a.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 678

Example: Depth-First Search



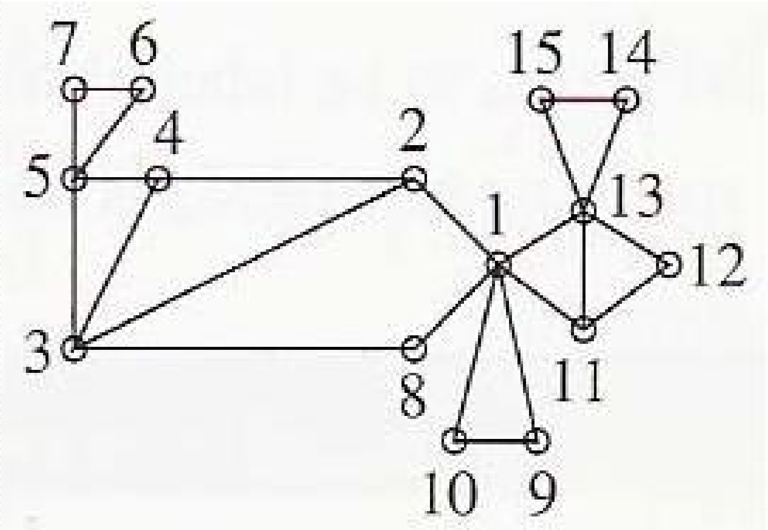


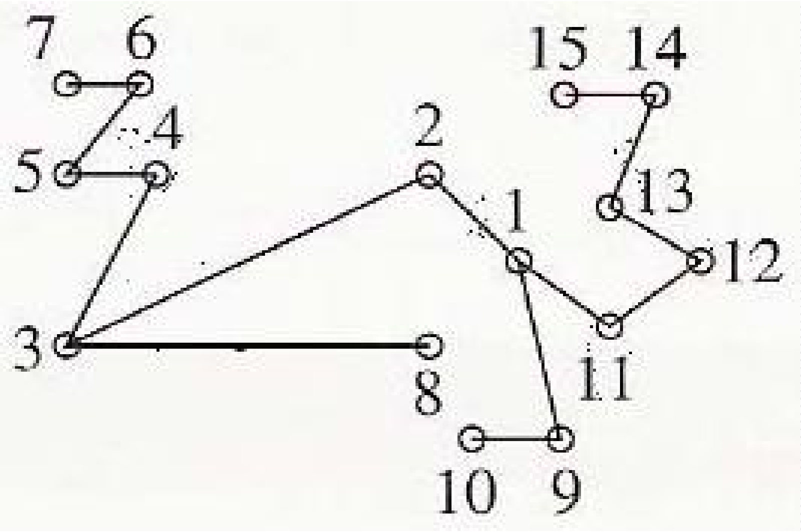
- Then, backtrack to c, and form the path c, b.
- The result is the spanning graph.

“Discrete Mathematics and its Applications.” Fifth Edition, by Kenneth H. Rosen. Mc Graw Hill, 2003. pag 678

Example: Depth-First Search

- Use a depth-first search to find a spanning tree for the graph G.





“Discrete Mathematics with Graph Theory.” Fifth Edition, by E. G. Goodaire ane M. Parmenter Prentce Hall, 2006. pag 401

Topics covered

- Trees and their properties.
- Spanning trees and minimum spanning trees algorithms.
- Depth-First Search.

Reference

- “Discrete Mathematics with Graph Theory”, Third Edition, E. Goodaire and Michael Parmenter, Pearson Prentice Hall, 2006. pp 370-410.

Reference

- “Discrete Mathematics and Its Applications”, Fifth Edition, Kenneth H. Rosen, McGraw-Hill, 2003. pp 631-694.