Chapter 1: Graphs and Its Basic Terminology

Definition 1: A *graph* G = (V ,E) consists of V , a nonempty set of vertices (or nodes) and E, a set of edges. Each edge has either one or two vertices associated with it, called its *endpoints*. An edge is said to *connect* its endpoints.

A graph with an infinite vertex set or an infinite number of edges is called **an infinite graph**, and in comparison, a graph with a finite vertex set and a finite edge set is called a **finite graph**.

A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a **simple graph**.

Graphs that may have **multiple edges** connecting the same vertices are called **multigraphs**.

Definition 2: A directed graph (or digraph) (V ,E) consists of a nonempty set of vertices V and a set of directed edges (or arcs) E. Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u, v) is said to start at u and end at v.

Mixed graph: A graph with both directed and undirected edges is called a **mixed graph**.

Loop: an edge connecting a vertex with itself.

Pseudograph: an undirected graph that may contain multiple edges and loops

Definition: The **intersection graph** of a collection of sets *A*1, *A*2, . . . , *An* is the graph that has a vertex for each of these sets and has an edge connecting the vertices representing two sets if these sets have a nonempty intersection.

TABLE 1 **Graph Terminology.**

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Application/ Graph Models

Example 1: A major publishing company has ten editors {1,2,3,4,5,6,7,8,9,10} in the scientific, technical and computing areas. These ten editors have a standard meeting time during the first Friday of every month and have divided themselves into seven committees to meet later in the day to discuss specific topics of interest to the company. These committees and their members are:

C1=advertising={1,2,3}

C2=securing and review={1,3,4,5}

C3=contacting new potential authors={2,5,6,7}

C4=finances={4,7,8,9}

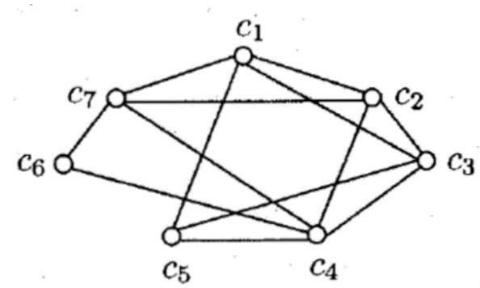
C5=copies and editions={2,6,7}

C6=competing textbooks={8,9,10}

C7=textbook representative={1,3,9,10}

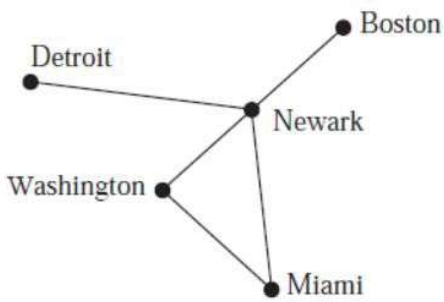
Draw the graph models of this problem.

Solution:

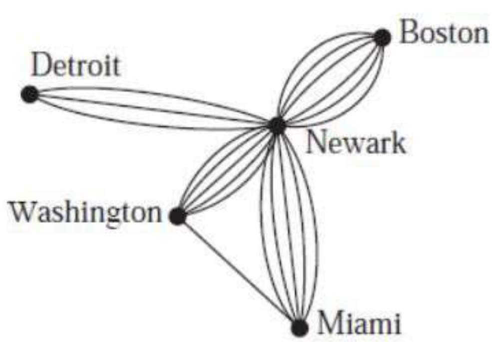


Example 2: Draw graph models, stating the type of graph used, to represent airline routes where every day there are four flights from Boston to Newark, two flights from Newark to Boston, three flights from Newark to Miami, two flights from Miami to Newark, one flight from Newark to Detroit, two flights from Detroit to Newark, three flights from Newark to Washington, two flights from Washington to Newark, and one flight from Washington to Miami, with

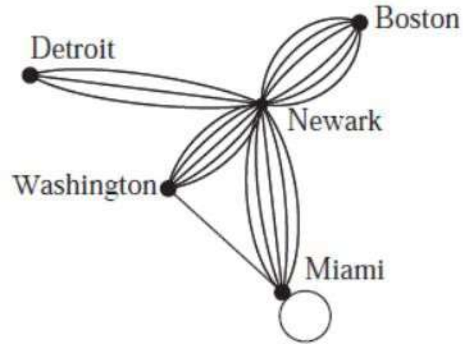
a) an edge between vertices representing cities that have a flight between them (in either direction).
b) an edge between vertices representing cities for each flight that operates between them (in either direction).
c) an edge between vertices representing cities for each flight that operates between them (in either direction), plus a loop for a special sightseeing trip that takes off and lands in Miami.
d) an edge from a vertex representing a city where a flight starts to the vertex representing the city where it ends.
e) an edge for each flight from a vertex representing a city where the flight begins to the vertex representing the city where the flight ends.
Solution:
a)



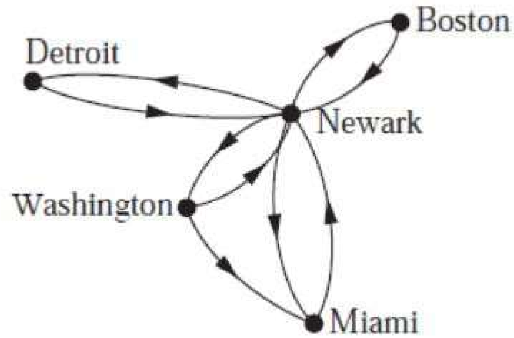
b)



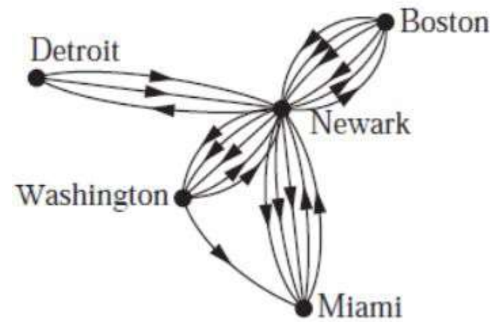
c)



d)



e)



Example 3: Construct the intersection graph of these collections of sets.

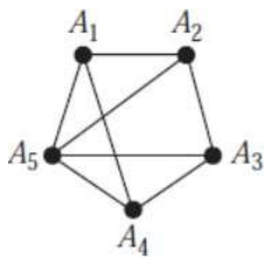
a) $A1 = \{0, 2, 4, 6, 8\}$, $A2 = \{0, 1, 2, 3, 4\}$, $A3 = \{1, 3, 5, 7, 9\}$, $A4 = \{5, 6, 7, 8, 9\}$, $A5 = \{0, 1, 8, 9\}$

b) $A1 = \{\ldots, -4, -3, -2, -1, 0\}$,
$A2 = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$,
$A3 = \{\ldots, -6, -4, -2, 0, 2, 4, 6, \ldots\}$,
$A4 = \{\ldots, -5, -3, -1, 1, 3, 5, \ldots\}$,
$A5 = \{\ldots, -6, -3, 0, 3, 6, \ldots\}$

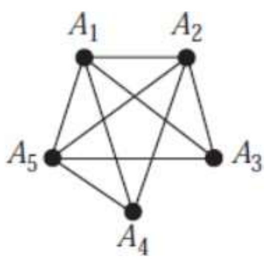
c) $A1 = \{x \mid x < 0\}$,
$A2 = \{x \mid -1 < x < 0\}$,
$A3 = \{x \mid 0 < x < 1\}$,
$A4 = \{x \mid -1 < x < 1\}$,
$A5 = \{x \mid x > -1\}$,
$A6 = \mathbf{R}$

Solution:

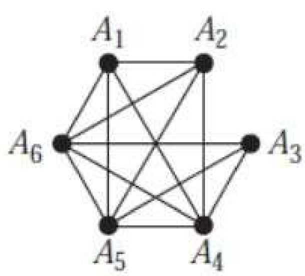
a)



b)



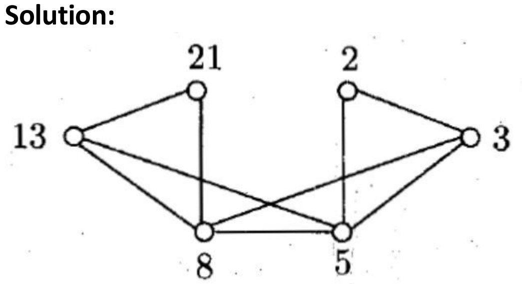
c)



Example 4: Consider the set S={2,3,5,8,13,21} of six specific Fibonacci number. Draw the graph model such that there is an edge if the sum or difference between any two distinct number belongs to S. **OR**

Consider the set S={2,3,5,8,13,21} of six specific Fibonacci number. Draw the graph model with vertex set V={2,3,5,8,13,21} and edge set E={{2,3}, {2,5}, {3,5}, {3,8},{5,8},{5,13},{8,13},{8,21},{13,21}}

Solution:



Exercise 1: Consider the set S={2,3,4,7,11,13}. Draw the graph model such that there is an edge if the sum or difference between any two distinct number belongs to S.

Exercise 2: Consider the set S={-6,-3,0,3,6}. Draw the graph model such that there is an edge if the sum or difference between any two distinct number belongs to S.

Basic Terminology

Definition 1: The number of vertices (edges) in a graph G is called the **order** (**size**) of the graph G.

Definition 2: Two vertices u and v in an undirected graph G are called ***adjacent*** (or neighbors) in G if u and v are endpoints of an edge e of G. Such an edge e is called ***incident*** with the vertices u and v and e is said to connect u and v.

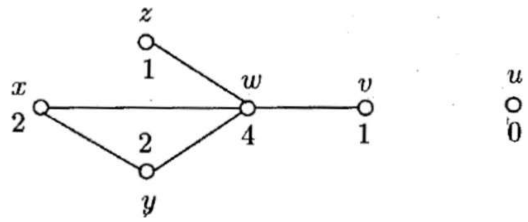
Definition 3: The set of all neighbors of a vertex v of G = (V ,E), denoted by N(v), is called the **neighborhood** of v. If A is a subset of V , we denote by N(A) the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A)=\bigcup_{v\in A} N(v)$

Definition 4: The ***degree of a vertex in an undirected graph*** is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by deg(v).

Definition 5: A vertex of degree zero is called **isolated**. A vertex is **pendant** if and only if it has degree one. The **minimum (maximum) degree** of a graph G is the minimum (maximum) degree among the vertices of G and is denoted as δ(G) (Δ(G)). So, if the G is a graph of order n and v is any vertex of G, then

$$0 \le \delta(G) \le \deg v \le \Delta(G) \le n-1.$$

If δ(G) =Δ(G), then graph is **regular graph**.

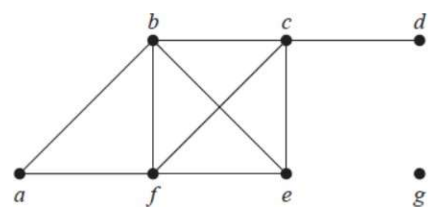


This graph has order 6 and size 5. The degree of each vertex is labeled on the vertex. The minimum degree of G $\delta(G)=1$ and maximum degree $\Delta(G)=4$. If we add the degrees of the vertices of G, we obtain 0+1+1+2+2+4=10, that is exactly equal to the twice of size of G. This refer to the FIRST THEOREM OF GRAPH THEORY.

Theorem: THE HANDSHAKING THEOREM
Let G = (V ,E) be an undirected graph with e edges. Then

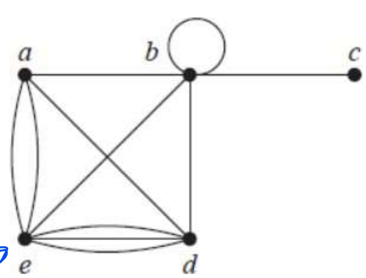
$$2e = \sum_{v\in V} \deg(v)$$

EXAMPLE: What are the degrees and what are the neighborhoods of the vertices in the given graphs?



deg(a) = 2, deg(b) = deg(c) = deg(f) = 4, deg(d) = 1, deg(e) = 3, and deg(g) = 0.
The neighborhoods of these vertices are: N(a) = {b, f }, N(b) = {a, c, e, f }, N(c) = {b, d, e, f }, N(d) = {c}, N(e) = {b, c, f }, N(f) = {a, b, c, e}, and N(g) = ∅.

edge→



node

deg(a) = 4, deg(b) = deg(e) = 6, deg(c) = 1, and deg(d) = 5.

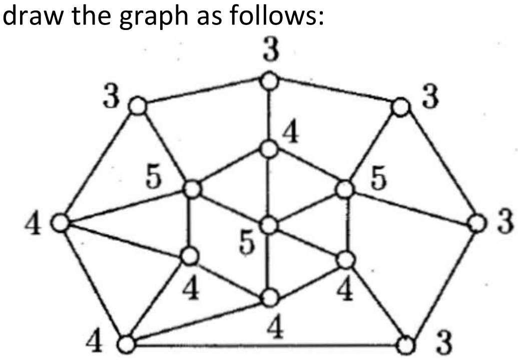
The neighborhoods of these vertices are: N(a) = {b, d, e}, N(b) = {a, b, c, d, e}, N(c) = {b}, N(d) = {a, b, e}, and N(e) = {a, b, d}.

EXAMPLE: How many edges are there in a graph with 10 vertices each of degree six?
Solution: Because the sum of the degrees of the vertices is 6 * 10 = 60, it follows that 2e = 60, where e is the number of edges. Therefore, e = 30.

Example: A graph G has order 14 and size 27. The degree of of each vertex of G is 3, 4 or 5. There are 6 vertices of degree 4. How many vertices of degree 3 and 5.
Solution: Let x be the number of vertices of degree 3. Since the order of G is 14, so the number of vertices of degree 5 are:
14-x-6=8-x
By applying the First Theorem we get

$$\begin{aligned} 3\cdot x + 4\cdot 6 + 5\cdot(8-x) &= 2\cdot 27 \\ 3x + 24 + 40 - 5x &= 54 \\ -2x &= -10 \\ x &= 5 \end{aligned}$$

Hence, the vertices of degree 3 are 5 and vertices of degree 5 are 8-5=3. We can draw the graph as follows:

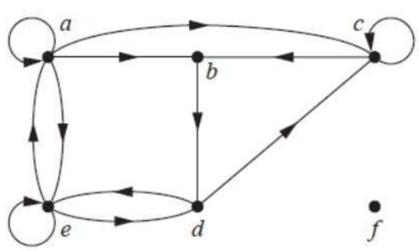


A vertex of even degree is called **even vertex** and a vertex of odd degree is called **odd vertex.** It follows the following
CORROLARY 1: Every undirected graph has an even number of odd vertices.
DEFINITION 6: When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to v and v is said to be adjacent from u. The vertex u is called the initial node

initial vertex of (u, v), and v is called the terminal or end vertex of (u, v). The initial vertex and terminal vertex of a loop are the same.

DEFINITION 7: In a graph with directed edges the in-degree of a vertex v, denoted by deg—(v), is the number of edges with v as their terminal vertex. The out-degree of v, denoted by deg+ (v), is the number of edges with v as their initial vertex. (Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

EXAMPLE 4: Find the in-degree and out-degree of each vertex in the given graph with directed edges.



Solution: The in-degrees are: deg—(a) = 2, deg—(b) = 2, deg—(c) = 3, deg—(d) = 2,deg—(e) = 3, and deg—(f) = 0.

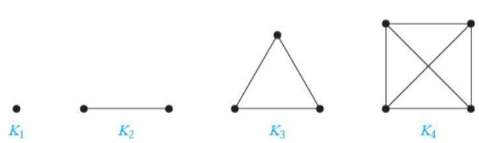
The out-degrees are: deg+(a) = 4, deg+(b) = 1, deg+(c) = 2,deg+(d) = 2, deg+(e) = 3, and deg+(f) = 0.

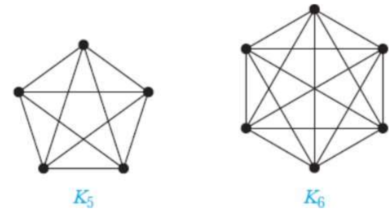
THEOREM 3: Let G = (V ,E) be a graph with directed edges. Then

$$\sum_{v\in V}\deg^-(v) = \sum_{v\in V}\deg^+(v) = |E|.$$

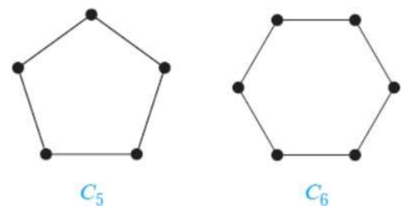
Definition: A **complete graph** on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices. The graphs K_n, for n = 1, 2, 3, 4, 5, 6, are given below.

The size of K_n is $\frac{n(n-1)}{2}$

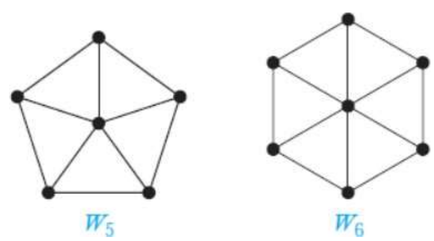




Definition: A **cycle** C_n, n ≥ 3, consists of n vertices v1, v2, . . . , vn and edges {v1, v2}, {v2, v3}, . . . , {vn−1, vn}, and {vn, v1}. The cycles C5, and C6 are shown below. The size of Cn is n.

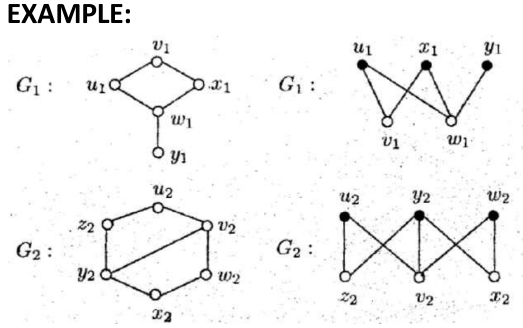


Definition: A **wheel graph** Wn is obtain by adding an additional vertex to a cycle Cn, for n ≥ 3, and connect this new vertex to each of the n vertices in Cn, by new edges. The wheels W5 and W6 are shown below. The order of Wn is n+1 and size is 2n.

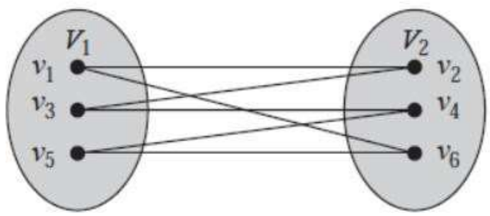


DEFINITION: A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V1 and V2 such that every edge in the graph connects a vertex in V1 and a vertex in V2.

EXAMPLE:

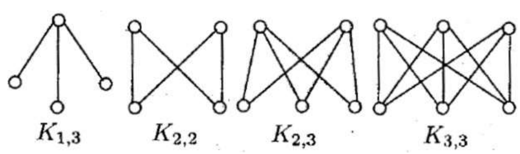


C6 is bipartite, as shown below, because its vertex set can be partitioned into the two sets V1 = {v1, v3, v5} and V2 = {v2, v4, v6}, and every edge of C6 connects a vertex in V1 and a vertex in V2.



EXAMPLE 10 K3 is not bipartite. To verify this, note that if we divide the vertex set of K3 into two disjoint sets, one of the two sets must contain two vertices. If the graph were bipartite, these two vertices could not be connected by an edge, but in K3 each vertex is connected to every other vertex by an edge.

Definition: A complete bipartite graph Km,n is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively such that every vertex of one subset is adjacent to every vertex of second subset. The order and size of $K_{m,n}$ is m+n and mn, respectively. The complete bipartite graphs $K_{2,3}$, $K_{3,3}$ are displayed below.



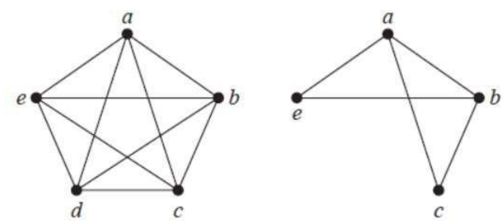
THEOREM 4: A simple graph is bipartite if and only if it is possible to assign one of two different colours to each vertex of the graph so that no two adjacent vertices are assigned the same colour.

THEOREM: A nontrivial graph G is a bipartite graph if and only if G contain NO odd cycle.

DEFINITION 7: A subgraph of a graph G = (V ,E) is a graph H = (W, F), where W ⊆ V and F ⊆ E. A subgraph H of G is a proper subgraph of G if G≠H.

DEFINITION: A subgraph F of a graph G is called an **induced subgraph** if whenever u and v are vertices of F and uv is an edge of G, then uv is an edge of F as well.

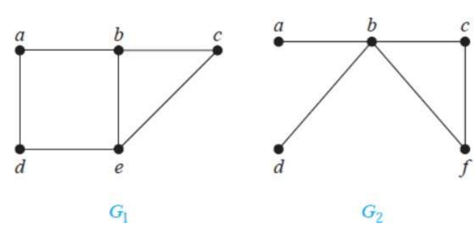
EXAMPLE 18 The graph shown below is a subgraph of K5. If we add the edge connecting c and e to graph, we obtain the subgraph induced by W = {a, b, c, e}.



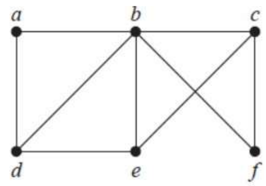
Definition: A proper subgraph of a graph G is obtained by removing edges and vertices from G. If we remove an edge e from G, then resulting graph is denoted by G-e. If we add a new edge e by connecting two previously non-incident vertices to the graph G, then resulting graph is denoted by G+e. If we remove a vertex v and all incident edges to it from G, then resulting graph is denoted by G-v.

DEFINITION 9: The union of two simple graphs G1 = (V1,E1) and G2 = (V2,E2) is the simple graph with vertex set V1 ∪ V2 and edge set E1 ∪ E2. The union of G1 and G2 is denoted by G1 ∪ G2.

EXAMPLE 19 Find the union of the graphs G1 and G2.



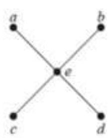
Solution: The vertex set of the union G1 ∪ G2 is the union of the two vertex sets, namely, {a, b, c, d, e, f }. The edge set of the union is the union of the two edge sets , namely, {ab,bc,ce,ed,da,be,ef,bf,bd}. The union is displayed in below:



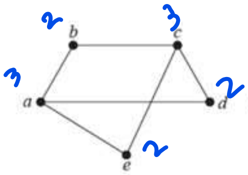
$G_1 \cup G_2$

Definition: The **degree sequence** of a graph is the sequence of the degrees of the vertices of the graph in nonincreasing order. A sequence d1, d2, . . . , dn is called **graphic** if it is the degree sequence of a simple graph.

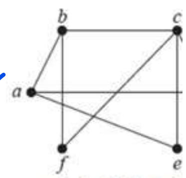
Example: Find the degree sequences for each of the graphs



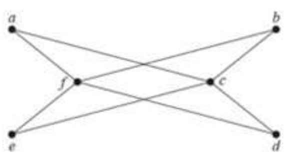
4,1,1,1,1



3,3,2,2,2



4,3,3,2,2,2



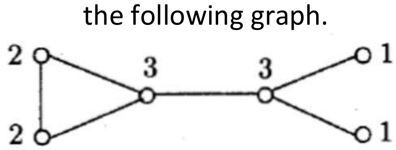
4,4,2,2,2,2

Example: Which of the following sequences are graphical?

1) S1: 3,3,2,2,1,1
2) S2: 6,5,5,4,3,3,3,2,2

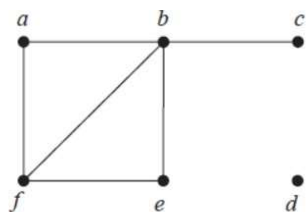
Solution:

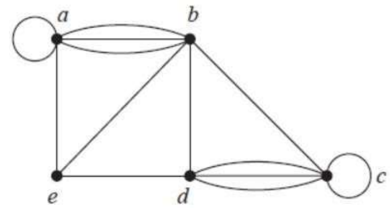
1) The sequence S1 is graphic. Indeed, it is a degree sequence of the following graph.



2) Since S2 has the number of odd degree vertices are odd, Therefore, S2 is not graphic

Exercise 1: Find order, size and the degree of each vertex in the given undirected graph. Identify all isolated and pendant vertices.





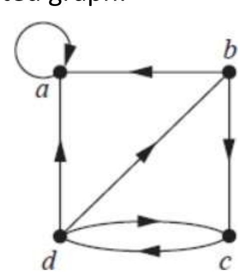
Exercise 2: The degree of each vertex of a certain graph of order 12 and size 31 is either 4 or 6. How many vertices of degree 4 and 6 are there.

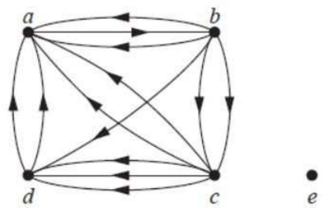
Exercise 3: Give an example of a graph G of order 6 and size 10 such that δ(G)=3 and Δ(G)=4.

Exercise 4: The degree of every vertex of a graph G of order 25 and size 62 is 3,4,5 or 6. There are two vertices of degree 4 and 11 vertices of degree 6. How many vertices of G have degree 5?

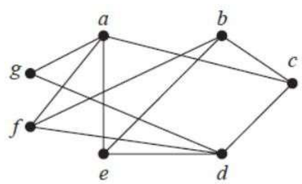
Exercise 5: Let G be a bipartite graph of order 22 with partite sets U and W, where |U|=12. Suppose that every vertex in U has degree 3; while every vertex of W has degree 2 or 4. How many vertices of G have degree 2?

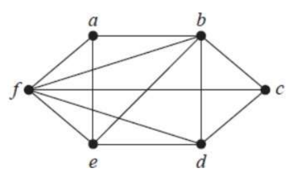
Exercise 6: Find the order, size, in-degree and out-degree of each vertex for the given directed graph.

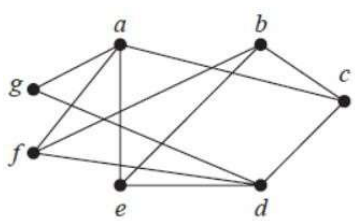




Exercise 7: Are given below graphs bipartite? If graph is bipartite then give the bipartition of vertices otherwise give reason.







Exercise 8: Which of the following sequences are graphical?

S1: 7,6,4,4,3,3,3

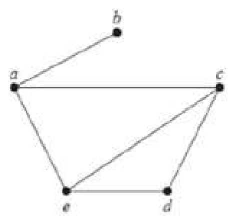
S2: 3,3,3,1

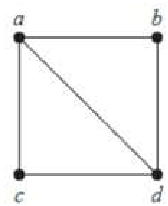
Chapter 2: Representation of Graphs, Isomorphism and connectivity

Adjacency lists

One way to represent a graph without multiple edges is to list all the edges of this graph. Another way to represent a graph with no multiple edges is to use **adjacency lists**, which specify the vertices that are adjacent to each vertex of the graph.

EXAMPLE 1: Use adjacency lists to describe the simple graph given below.



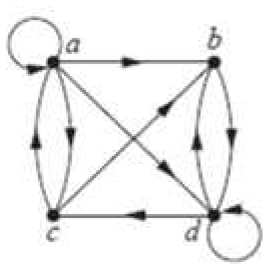


An Adjacency List for a Simple Graph.

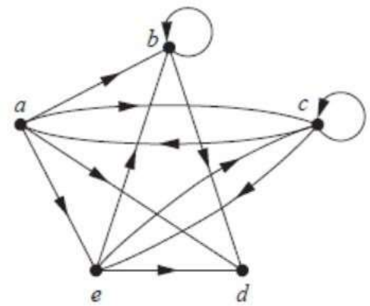
Vertex	Adjacent Vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

Vertex	Adjacent Vertices
a	b, c, d
b	a, d
c	a, d
d	a, b, c

EXAMPLE 2: Use an adjacency list to represent the given directed graph.



Vertex	Terminal Vertices
a	a, b, c, d
b	d
c	a, b
d	b, c, d



An Adjacency List for a Directed Graph.

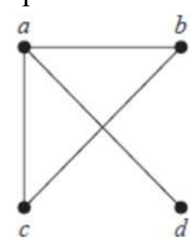
Initial Vertex	Terminal Vertices
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

Adjacency Matrices

Let G be a graph of order n with vertex set $V=\{v_1,v_2,v_3, \ldots, v_n\}$. The adjacency matrix of G is the n×n matrix $A=[a_{ij}]$ defined as:

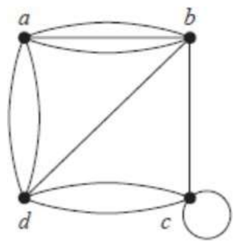
$$a_{ij} = \begin{cases} 1 & \text{if } v_iv_j \in E(G) \\ 0 & \text{otherwise;} \end{cases}$$

EXAMPLE 3: Use an adjacency matrix to represent the graph below.



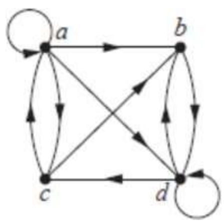
$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

EXAMPLE 4: Use an adjacency matrix to represent the given pseudograph.



$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 3 & 0 & 2 \\ b & 3 & 0 & 1 & 1 \\ c & 0 & 1 & 1 & 2 \\ d & 2 & 1 & 2 & 0 \end{array}$$

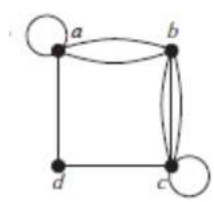
EXAMPLE 5: Represent the graph with an adjacency matrix



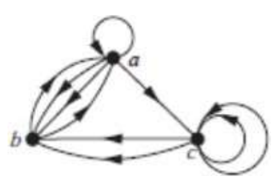
$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ b & 0 & 0 & 0 & 1 \\ c & 1 & 1 & 0 & 0 \\ d & 0 & 1 & 1 & 1 \end{array}$$

EXAMPLE 6: Draw an undirected graph represented by the given adjacency matrix.

$$\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$$

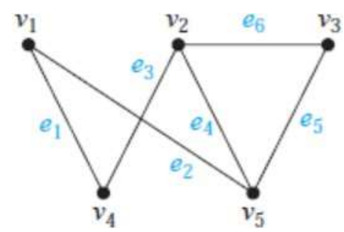


<u>**Incidence Matrices**</u>

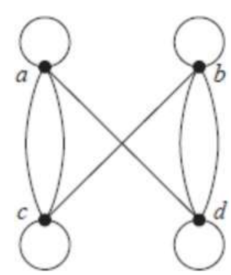
Let G = (V ,E) be an undirected graph. Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the n × m matrix $M = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$

EXAMPLE 7: Find an incidence matrix of the given graph.



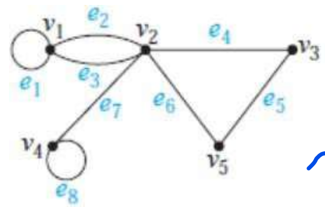
$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_3 & 0 & 0 & 0 & 0 & 1 & 1 \\ v_4 & 1 & 0 & 1 & 0 & 0 & 0 \\ v_5 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$



$$\begin{array}{c|cccccccccc} a & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ b & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ c & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ d & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}$$

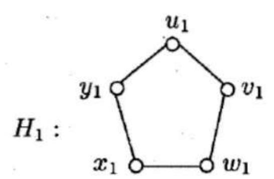
EXAMPLE 8: Draw the graph of the following incidence matrix.

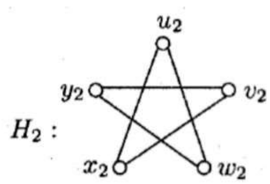
$$\begin{array}{cc} & \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

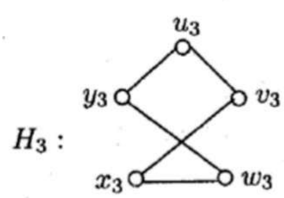


Isomorphism of Graphs

Two graphs G and H are equal if V(G)=V(H) and E(G)=E(H). We have called two graphs G and H are "Isomorphic" if they the same structure and have written G $\cong$ H if the vertices of G and H can be labeled (or relabeled) to produced two equal graphs. For Example: There are three graphs of order 5 and size 5.

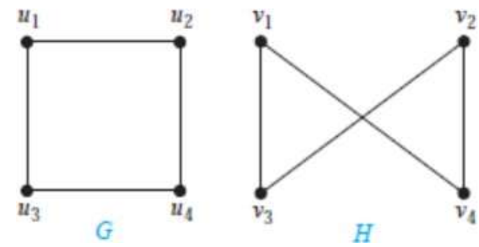






Formally, the definition of isomorphism as: The simple graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ are isomorphic if there exists a bijective function *f* from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if *f(a)* and *f(b)* are adjacent in G_2, for all a and b in V_1. Such a function *f* is called an isomorphism. Two simple graphs that are not isomorphic are called non-isomorphic.

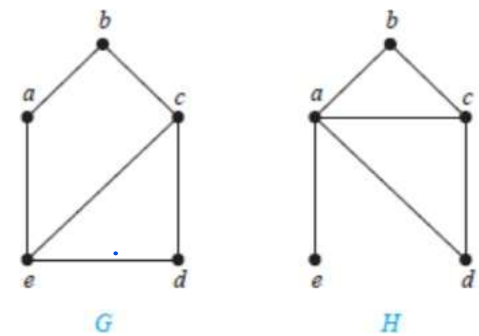
EXAMPLE 9: Show that the graphs G = (V ,E) and H = (W, F), displayed below, are isomorphic.



Solution: These graphs are isomorphic. Let f is isomorphism defined as:

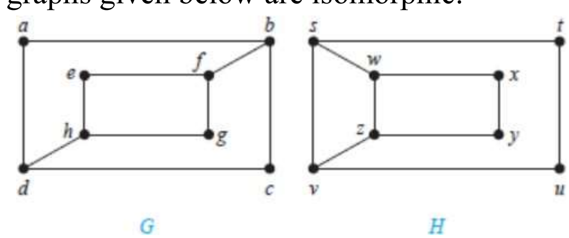
$f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$.

EXAMPLE 10: Show that the graphs displayed below are not isomorphic.



Solution: Both G and H have five vertices and six edges. However, H has a vertex of degree one, namely, e, whereas G has no vertices of degree one. It follows that G and H are not isomorphic.

EXAMPLE 11: Determine whether the graphs given below are isomorphic.



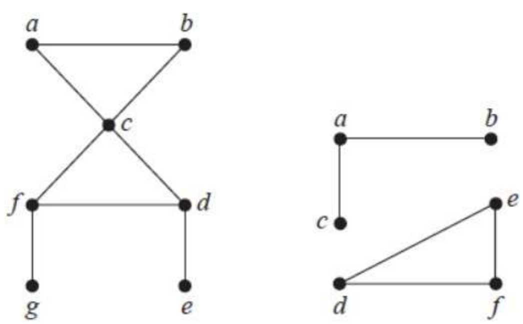
Solution: The graphs G and H are not isomorphic because any vertex of degree 2 in graph G is adjacent with two vertices of degree 3 but any vertex of degree 2 in graph H is adjacent with one vertex degree 3 and one vertex of degree 2.

Connectivity

Definition: A **path** is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph. A *path* of length greater than zero that begins and ends at the same vertex is called a **circuit or cycle**. *A path or circuit* is called ***simple*** if it does not contain the same edge more than once.

Definition: An undirected graph is called **connected** if there is a path between every pair of distinct vertices of the graph. An undirected graph that is not connected is called disconnected. We say that we disconnect a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Example 12: Which of the following graphs are connected and disconnected?



G_1 G_2

Solution: The graph G1 is connected, because for every pair of distinct vertices there is a path between them.

However, the graph G2 is disconnected, because there is no path in G2 between vertices a and d.

Theorem 1: There is a simple path between every pair of distinct vertices of a connected undirected graph.

Definition: A **connected component** of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.

Definition: The removal of a vertex and all of its incident edges produces a subgraph with more connected components. Such vertices are called **cut vertices. OR**

A vertex v in a connected graph G is a cut-vertex of G if G-v is disconnected.

Definition: The minimum number of vertices that can be removed to disconnect a graph is known as **vertex connectivity** and it is denoted by *k(G)*.

Definition: The removal of an edge produces a graph with more connected components than in the original graph is called a **cut edge or bridge.**

Definition: A set of edges E′ is called an **edge cut** of G if the subgraph G − E′ is disconnected.

Definition: The **edge connectivity** of a graph G, denoted by λ(G), is the minimum number of edges in an edge cut of G. **or**

The minimum number of edges that can be removed to disconnect a graph is known as **edge connectivity.**

Theorem 2: For every graph G

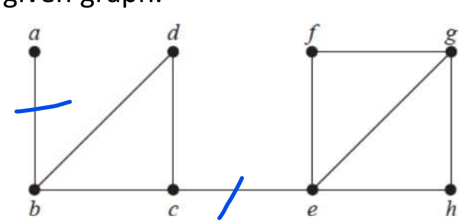
$$\kappa(G) \le \lambda(G) \le \min_{v \in V} \deg(v).$$

Theorem 3: Let v be a vertex incident with a bridge in a connected graph G. Then v is cut vertex of G if and only if deg(v)≥2.

Corollary: Let G be a connected graph of order 3 or more. If G contains a bridge, then G contain a cut vertex.

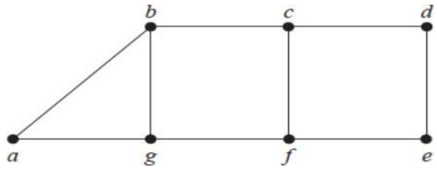
Corollary: Every nontrivial connected graph contains at least two vertices that are not cut vertices.

Example 13: Find the cut vertices, cut edges, vertex connectivity and edge connectivity of given graph.

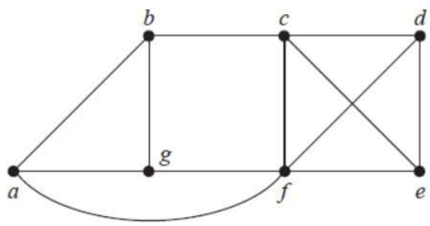


cut vertices: **b,c,e**

cut edges: **ab,ce**
vertex connectivity k(G): **1**
edge connectivity λ(G): **1**



cut vertices: **No cut vertex**
cut edges: **No cut edge**
vertex connectivity k(G): **2**
edge connectivity λ(G): **2**

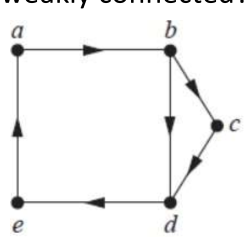


cut vertices: **No cut vertex**
cut edges: **No cut edge**
vertex connectivity k(G): **2**
edge connectivity λ(G): **3**

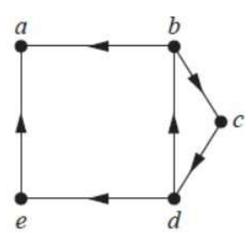
Definition: A directed graph is **strongly connected** if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

Definition: A directed graph is **weakly connected** if there is a path between every two vertices in the underlying undirected graph.

Example 13: Are the directed graphs G and H given below strongly connected? Are they weakly connected?



G



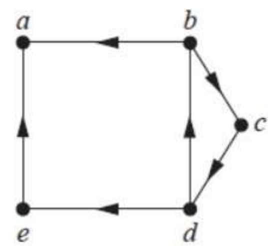
H

Solution: G is strongly connected because there is a path between any two vertices in this directed graph. Hence, G is also weakly connected.

The graph H is not strongly connected. There is no directed path from a to b in this graph. However, H is weakly connected, because there is a path between any two vertices in the underlying undirected graph of H.

Definition: The subgraphs of a directed graph G that are strongly connected but not contained in larger strongly connected subgraphs, that is, the maximal strongly connected subgraphs, are called the **strongly connected components** or **strong components** of G.

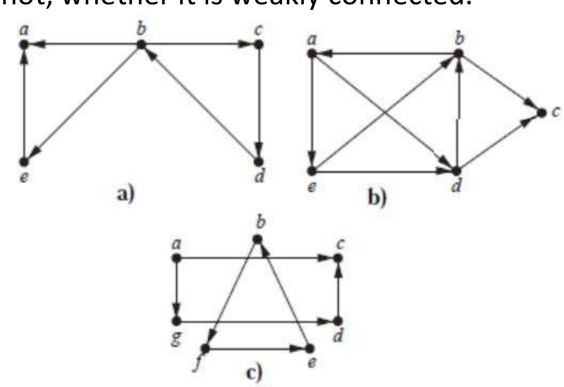
Example 11: Find the strongly connected components of the given graph H.



H

Solution: The graph H has three strongly connected components, consisting of the vertex a; the vertex e; and the subgraph consisting of the vertices b, c, and d and edges (b, c), (c, d), and (d, b).

Exercise 14. Determine whether each of these graphs is strongly connected and if not, whether it is weakly connected.



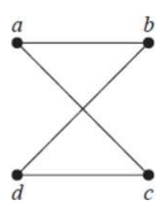
Solution: a) The graph is not strongly connected. There is no directed path from a to b in this graph. However, the graph is weakly connected, because there is a path between any two vertices in the underlying undirected graph.

b) The graph is not strongly connected. There is no directed path from c to b in this graph. However, the graph is weakly connected, because there is a path between any two vertices in the underlying undirected graph.

c) The graph is not strongly connected. There is no directed path from a to b in this graph. However, the graph is also not weakly connected, because there is no path from a to b vertices in the underlying undirected graph.

Theorem 3: Let G be a graph with adjacency matrix A with respect to the ordering v1, v2, . . . , vn of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from vi to vj , where r is a positive integer, equals the (i, j)th entry of A^r .

Example 15: How many paths of length four are there from a to d in the given simple graph.



Solution:

Adjacency matrix
$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Hence, the number of paths of length four from a to d is the (1, 4)th entry of A^4.

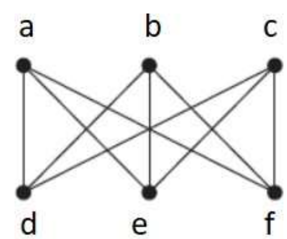
$$A_2 = A.A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$
$$A^4 = A^2.A^2 = \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$
$$= \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

there are exactly eight paths of length four from a to d.

Example 15: Find the number of paths of length n between two adjacent vertices in $K_{3,3}$ if n is

a) 2 b) 3 c) 4 d) 5

Solution:



$K_{3,3}$

Adjacency matrix

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array}\begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

a)

$$A^2 = A.A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 3 & 3 & 0 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 \end{bmatrix}$$

there are exactly zero paths of length 2 between two adjacent vertices in $K_{3,3}$.

b)

$$A^3 = A^2.A = \begin{bmatrix} 3 & 3 & 3 & 0 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 3 & 3 \end{bmatrix} . \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 9 & 9 & 9 \\ 0 & 0 & 0 & 9 & 9 & 9 \\ 0 & 0 & 0 & 9 & 9 & 9 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \end{bmatrix}$$

there are exactly 9 paths of length 3 between two adjacent vertices in $K_{3,3}$.

c)

$$A^4 = A^3.A = \begin{bmatrix} 0 & 0 & 0 & 9 & 9 & 9 \\ 0 & 0 & 0 & 9 & 9 & 9 \\ 0 & 0 & 0 & 9 & 9 & 9 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \\ 9 & 9 & 9 & 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 27 & 27 & 27 & 0 & 0 & 0 \\ 27 & 27 & 27 & 0 & 0 & 0 \\ 27 & 27 & 27 & 0 & 0 & 0 \\ 0 & 0 & 0 & 27 & 27 & 27 \\ 0 & 0 & 0 & 27 & 27 & 27 \\ 0 & 0 & 0 & 27 & 27 & 27 \end{bmatrix}$$

there are exactly zero paths of length 4 between two adjacent vertices in $K_{3,3}$.

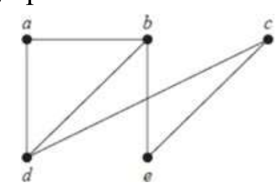
d)

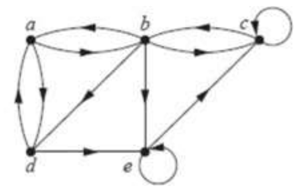
$$A^5 = A^4.A = \begin{bmatrix} 27 & 27 & 27 & 0 & 0 & 0 \\ 27 & 27 & 27 & 0 & 0 & 0 \\ 27 & 27 & 27 & 0 & 0 & 0 \\ 0 & 0 & 0 & 27 & 27 & 27 \\ 0 & 0 & 0 & 27 & 27 & 27 \\ 0 & 0 & 0 & 27 & 27 & 27 \end{bmatrix} . \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 81 & 81 & 81 \\ 0 & 0 & 0 & 81 & 81 & 81 \\ 0 & 0 & 0 & 81 & 81 & 81 \\ 81 & 81 & 81 & 0 & 0 & 0 \\ 81 & 81 & 81 & 0 & 0 & 0 \\ 81 & 81 & 81 & 0 & 0 & 0 \end{bmatrix}$$

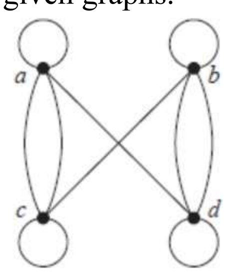
there are exactly 81 paths of length 5 between two adjacent vertices in $K_{3,3}$.

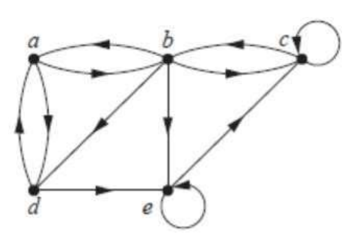
Exercise 1: Use an adjacency list to represent the given graphs.





Exercise 2: Find an adjacency matrix of the given graphs.





Exercise 3: Represent each of these graphs with an adjacency matrix.

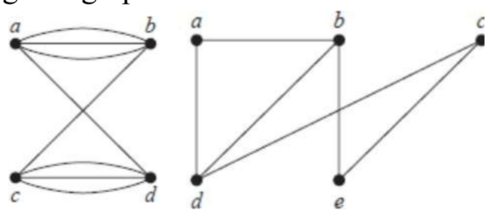
a) K_4 b) $K_{1,4}$ c) $K_{2,3}$ d) C_4

e) W_4

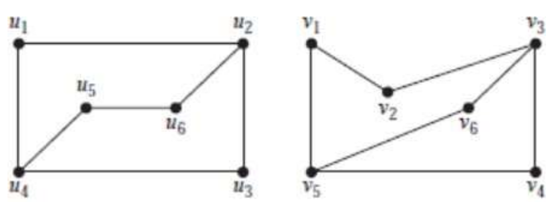
Exercise 4: Draw graph represented by the given adjacency matrix.

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 3 & 0 & 4 \\ 1 & 2 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

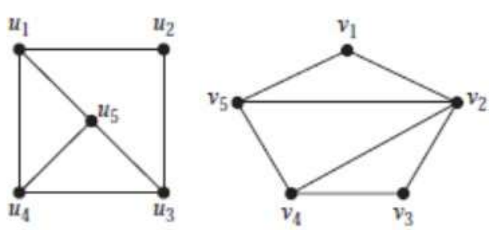
Exercise 5: Find an incidence matrix of the given graph.



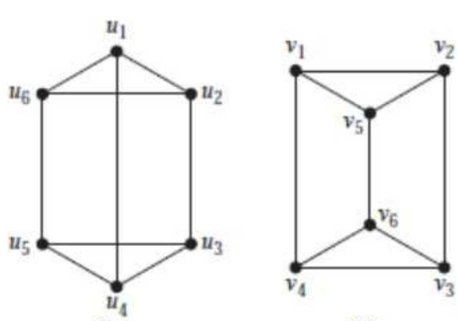
Exercise 6: Determine whether the given pair of graphs is isomorphic. Exhibit an isomorphism or provide a rigorous argument that none exists.



G H

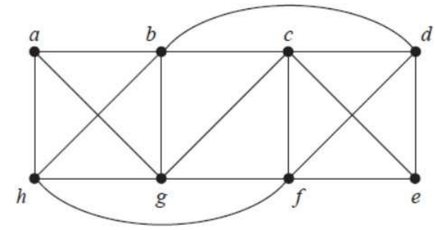


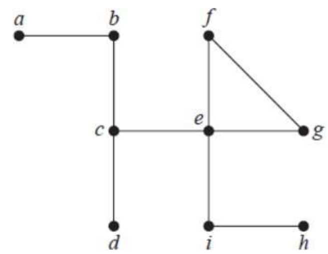
G H

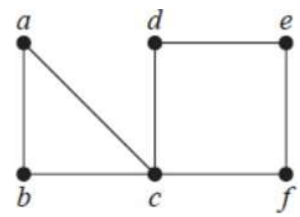


G H

Exercise 7: Find the cut vertices, cut edges, vertex connectivity and edge connectivity.







Exercise 8. Find the number of paths of length n between two different vertices in K_4 if n is

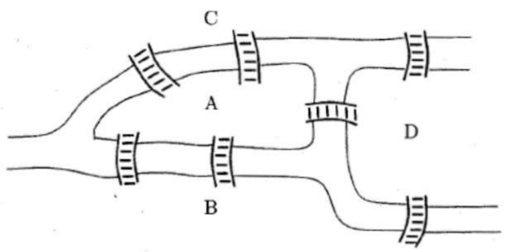
a) 2 b) 3 c) 4 d) 5

Exercise 9: Find the number of paths of length n between two nonadjacent vertices in $K_{3,3}$ if n is

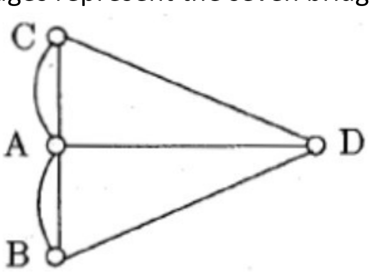
a) 2 b) 3 c) 4 d) 5

Chapter 3:
TRAVERSABILITY AND PALNARITY

The city of Konigsberg (located in Northern Germany) would play an interesting role in Euler's life and in the history of graph theory. The River Pregel flowed through Konigsberg, separating it into four landed areas. Seven bridges were built over the river that allowed the citizens of Konigsberg to travel between these land areas. A map of Konigsberg, showing the four land areas (labelled A, B, C, D), the location of the river and the bridges at that time are given below:



The story goes that the citizens of Konigsberg enjoyed going for walks near the river. Some citizens wondered whether it was possible to go for a walk in Konigsberg and pass over each bridge exactly once. This became known as the **Konigsberg Bridge Problem.** This problem remains unsolved for some time and become well-known throughout the region. This problem came to attention of Euler. Euler observed that this problem can be represented by graph, where vertices represent four land areas and edges represent the seven bridges.

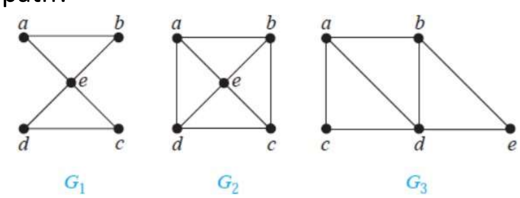


Definition 1: An **Euler circuit** in a graph G is a simple circuit containing every edge of G. An **Euler path** in G is a simple path containing every edge of G.

Theorem 1: A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has even degree.

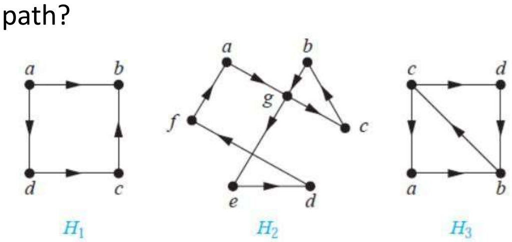
Theorem 2: A connected multigraph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree.

EXAMPLE 1: Which of the following undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?



Solution: The graph G1 has an Euler circuit, for example, a, e, c, d, e, b, a. Neither of the graphs G2 or G3 has an Euler circuit. However, G3 has an Euler path, namely, a, c, d, e, b, d, a, b. G2 does not have an Euler path.

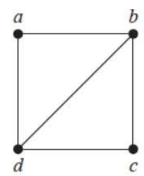
EXAMPLE 2: Which of the following undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?



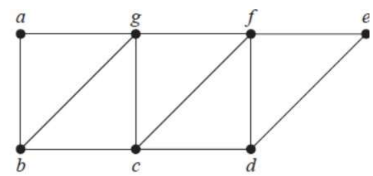
Solution: The graph H2 has an Euler circuit, for example, a, g, c, b, g, e, d, f, a. Neither H1 nor H3 has an Euler circuit. H3 has an Euler path, namely, c, a, b, c, d, b, but H1 does not.

12/12

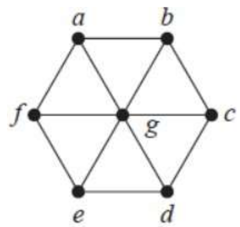
Example 4: Which of the following graphs have an Euler path?



G_1



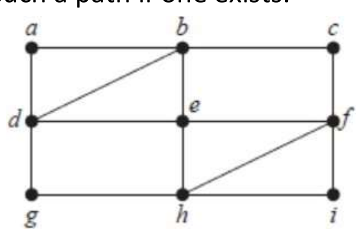
G_2

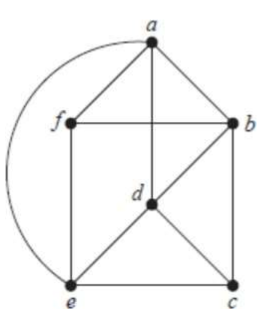


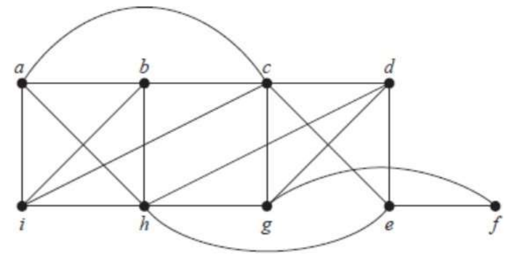
G_3

Solution: G1 contains exactly two vertices of odd degree, namely, b and d. Hence, it has an Euler path: d, a, b, c, d, b. Similarly, G2 has exactly two vertices of odd degree, namely, b and d. So it has an Euler path: b, a, g, f, e, d, c, g, b, c, f, d. G3 has no Euler path because it has six vertices of odd degree.

Example: Determine whether the given graph has an Euler circuit. Construct such a circuit when one exists. If no Euler circuit exists, determine whether the graph has an Euler path and construct such a path if one exists.





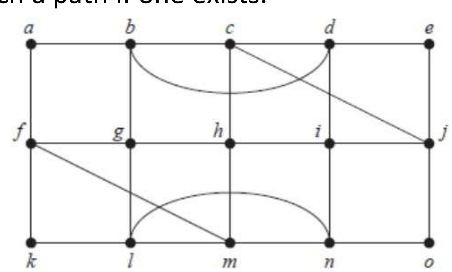


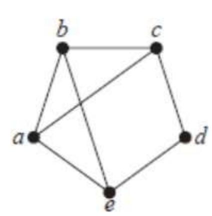
Solution: Since all the vertices in the graph has even degree, therefore the graph has an Euler circuit: a,b,c,f,I,h,f,e,h,g,d,e,b,d,a

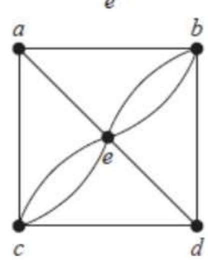
Since, exactly two vertices have odd degree, therefore, the graph has an Euler path: f,a,b,f,e,a,d,b,c,e,d,c.

Since all the vertices in the graph have even degree, therefore the graph has an Euler circuit: a, i, h, g, d, e, f, g, c, e, h, d, c, a, b, i, c, b, h, a.

Exercise 1: Determine whether the given graph has an Euler circuit. Construct such a circuit when one exists. If no Euler circuit exists, determine whether the graph has an Euler path and construct such a path if one exists.





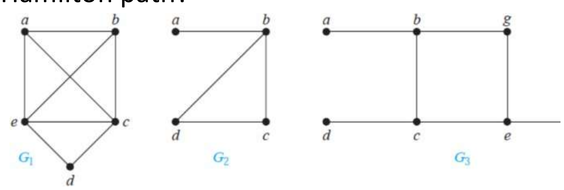


Exercise 2. For which values of *n* do these graphs have an Euler circuit?

a) K_n **b)** C_n **c)** W_n

Definition 2: A simple path in a graph G that passes through every vertex exactly once is called **a Hamilton path**, and a simple circuit in a graph G that passes through every vertex exactly once is called a **Hamilton circuit**.

Example: Which of the following simple graphs have a Hamilton circuit or, if not, a Hamilton path?

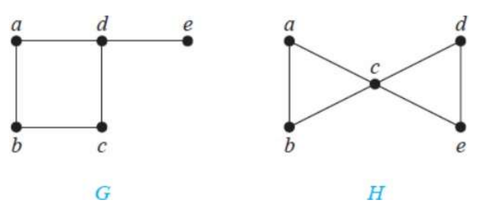


Soultion: G1 has a Hamilton circuit: a, b, c, d, e, a.

There is no Hamilton circuit in G2 because any circuit containing every vertex must contain the edge {a, b} twice, but G2 does have a Hamilton path: a, b, c, d.

G3 has neither a Hamilton circuit nor a Hamilton path, because any path containing all vertices must contain one of the edges {a, b}, {e, f }, and {c, d} more than once.

Example: Show that neither graph given below has a Hamilton circuit.



Soultion: There is no Hamilton circuit in G because G has a vertex of degree one, namely, e.

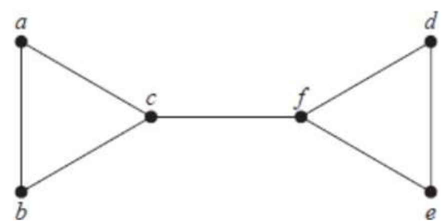
It is now easy to see that no Hamilton circuit can exist in H.

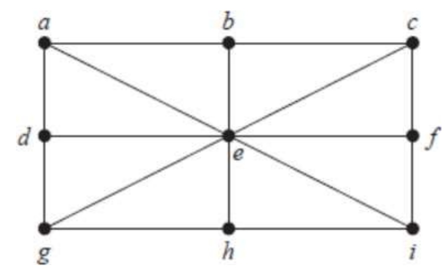
Theorem 3: ***DIRAC'S THEOREM*** If G is a simple graph with n vertices with n ≥ 3 such that the degree of every vertex in G is at least n/2, then G has a Hamilton circuit.

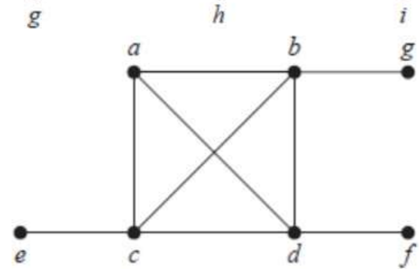
Theorem 4: ***ORE'S THEOREM*** If G is a simple graph with n vertices with n ≥ 3 such that deg(u) + deg(v) ≥ n for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.

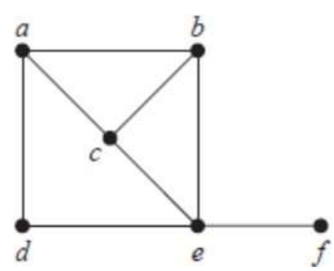
Theorem 5: Let u and v be nonadjacent vertices in a graph G of order n such that deg(u) + deg(v) ≥ n. Then G+uv is Hamilton if and only if G is Hamilton.

Exercises: Determine whether the given graph has a Hamilton circuit. If it does, find such a circuit. If it does not, give an argument to show why no such circuit exists.







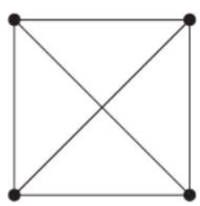


Planar Graphs

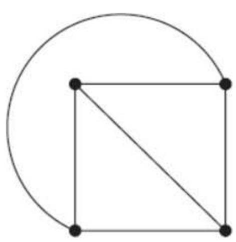
Definition: A graph is called ***planar*** if it can be drawn in the plane without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a ***planar representation*** of the graph. A graph that is not planar is called **nonplanar.** A graph G is called a **plane graph** if it is drawn in the plane so that so two edges of G cross.

Example: Every cycle, path, star is a planar.

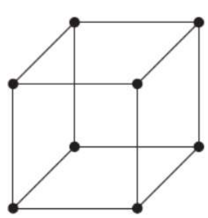
EXAMPLE: Is K_4 planar?



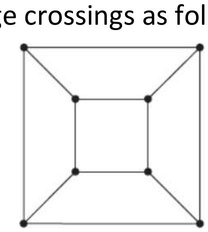
K_4 is planar because it can be drawn without edge crossings as follow.

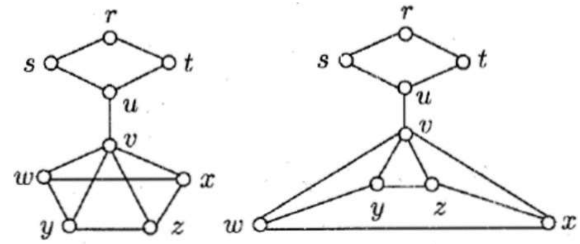


EXAMPLE 2: Is Q_3 planar?



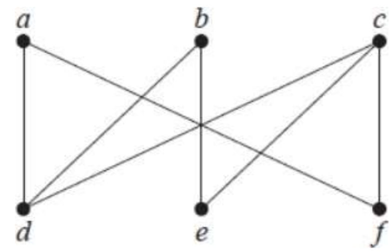
Q_3 is planar because it can be drawn without edge crossings as follow.

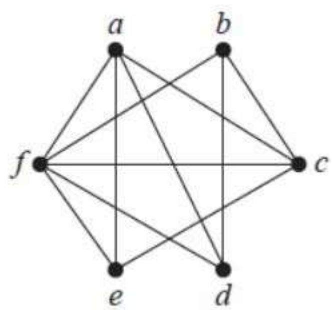




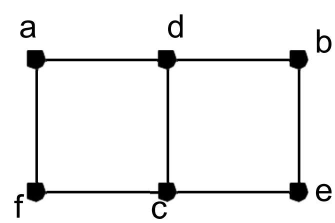
A planar graph and a plane graph

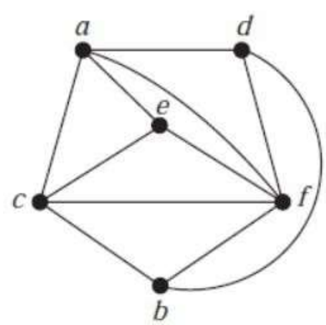
EXAMPLE: Determine whether the given graph is planar. If so, draw it so that no edges cross.



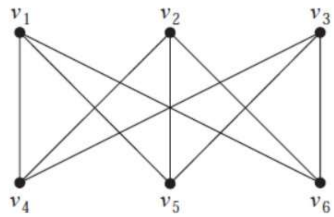


Solution:





EXAMPLE: Is $K_{3,3}$ planar?

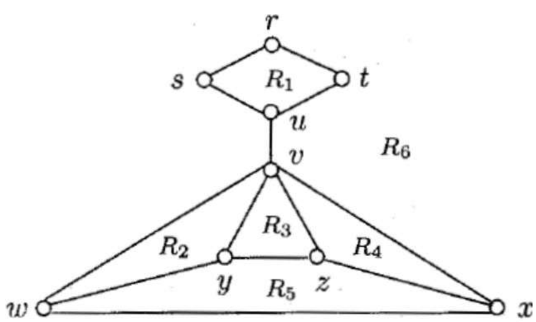


Solution:

$K_{3,3}$ is not planar because it cannot be drawn without edge crossings.

Definition: A plane graph divides the plane into connected pieces called **regions**. The unbounded region is called **exterior region**. The subgraph of a plane graph whose vertices and edges are incident with a given region R is called the **boundary** of R.

Example:



Theorem: EULER'S FORMULA/IDENTITY

Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

EXAMPLE: Suppose that a connected planar simple graph has 20 vertices, each of degree 3. Into how many regions does a representation of this planar graph split the plane?

Solution: This graph has 20 vertices, each of degree 3, so v = 20 and $2.e = 3 \cdot 20 = 60$, so we have e = 30. Euler's formula is $r = e - v + 2 = 30 - 20 + 2 = 12$.

EXAMPLE: Suppose that a connected planar graph has six vertices, each of degree four. Into how many regions is the plane divided by a planar representation of this graph?

Solution: This graph has 6 vertices, each of degree 4, so v = 6 and $2.e = 4 \cdot 6 = 24$, so we have e = 12. Euler's formula is $r = e - v + 2 = 12 - 6 + 2 = 8$.

EXAMPLE: Suppose that a connected planar graph has 30 edges. If a planar representation of this graph divides the plane into 20 regions, how many vertices does this graph have?

Solution: This graph has 30 edges, so e = 30 and r = 20. Euler's formula is $r = e - v + 2$.

$20 = 30 - v + 2$. This implies that v=12.

COROLLARY: If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then $e \leq 3v - 6$.

COROLLARY: If G graph with e edges and v vertices, where $v \geq 3$, such that $e > 3v - 6$, then G is nonplanar.

COROLLARY: Every planar graph contains a vertex of degree 5 or less.

COROLLARY: If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$.

EXAMPLE: Show that $K_{3,3}$ is nonplanar.

Solution: Because $K_{3,3}$ has no circuits of length three. $K_{3,3}$ has six vertices and nine edges. Because e = 9 and 2v − 4 = 8, by using Corollary ($e \leq 2v - 4$) shows that $K_{3,3}$ is nonplanar.

COROLLARY: Show that K_5 is nonplanar.

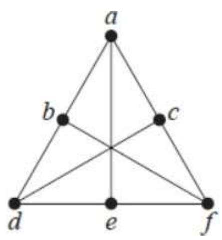
Solution: The graph K_5 has five vertices and 10 edges. However, the inequality $e \leq 3v - 6$ is not satisfied for this graph because e = 10 and 3v − 6 = 9. Therefore, K_5 is not planar.

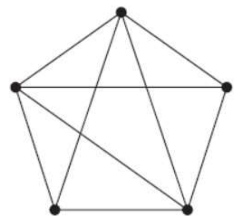
Definition: A graph G is a **maximal planar** if G is planar but the addition of an edge between any two nonadjacent vertices of G results in a nonplanar graph.

Theorem (Kuratowski's Theorem):

A graph G is a planar if and only if G does not contain $K_{3,3}$ or K_5.

Exercise 1: Determine whether the given graph is planar. If so, draw it so that no edges cross.





Exercise 2: Which of these nonplanar graphs have the property that the removal of any vertex and all edges incident with that vertex produces a planar graph?

a) K_5 b) K_6 c) $K_{3,3}$ d) $K_{3,4}$

Exercise 2: Suppose that a connected planar graph has eight vertices, each of degree three. Into how many regions is the plane divided by a planar representation of this graph?

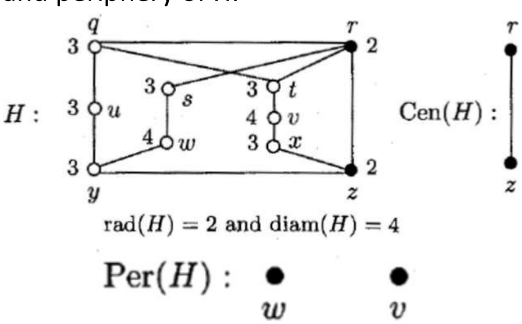
Chapter 4: Distance in Graph

Definition: For two vertices u and v in a graph G, the **distance** d(u,v) from u to v is the length of shortest path u-v path in G. The distance has the following four properties for any connected graph.

1. $d(u,v) \geq 0$ for all $u, v \in V(G)$.
2. $d(u,v) = 0$ if and only if $u = v$.
3. $d(u,v) = d(v,u)$ for all $u, v \in V(G)$
4. $d(u,w) \leq d(u,v) + d(v,w)$ for all $u, v, w \in V(G)$

Definition: For a vertex v in a connected graph G, the **eccentricity e(v)** of v is the distance between v and a vertex farthest from v in G. The minimum eccentricity among the vertices of G is its **radius rad(G)** and the maximum eccentricity is its **diameter diam(G).** If e(v)=rad(G), then v is **central vertex.** The subgraph induced by the central vertices is the **centre Cen(G)** of G. If every vertex of G is a central vertex, then Cen(G)=G and G is called **self-centred.** A vertex v in a connected graph G is called **peripheral vertex** if e(v)=diam(G). The subgraph induced by the peripheral vertices is the **periphery Per(G)** of G.

Example: Determine the eccentricity of each vertex, radius, diameter, centre of H and periphery of H.



The vertices w and v are the periphery of H.

From this, the following Theorems holds:

Theorem: For every nontrivial connected graph G,

$$\mathrm{rad}(G) \leq \mathrm{diam}(G) \leq 2\,\mathrm{rad}(G).$$

Theorem: For every two adjacent vertices u and v in a connected graph,

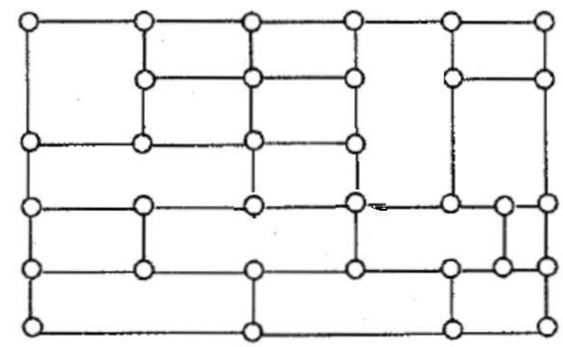
$$|e(u) - e(v)| \leq 1.$$

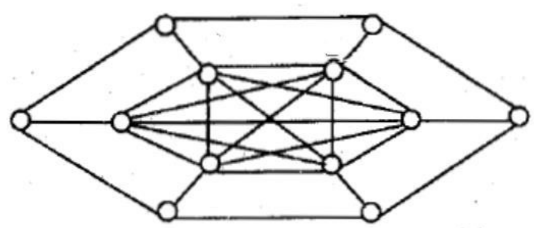
Theorem: Let u and v be adjacent vertices u and v in a connected graph G. Then

$$|d(u,x) - d(v,x)| \leq 1$$

for every vertex x of G.

Exercise 1: Determine the eccentricity of each vertex, radius, diameter, centre of G and periphery of G.





Definition: Let G be a connected graph. For an ordered set W={w_1,w_2,…, w_k} of vertices of G and a vertex v of G, the **locating code** of v with respect to W is the k-vector

$$c_W(v) = (d(v,w_1), d(v,w_2), \cdots, d(v,w_k)).$$

The set W is a **locating/resolving set** for G if the vertices have different codes. The minimum number of vertices in the locating set is called locating **number loc(G)/metric dimension md(G).**

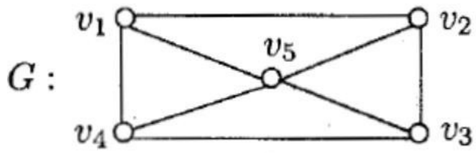
Theorem: A connected graph G of order n has the locating number (metric dimension) 1 if and only if G is a path.

Theorem: A connected graph G of order n≥2 has the locating number (metric dimension) n-1 if and only if G is a complete graph.

Theorem: A connected graph G of order n, then

$$1 \leq \operatorname{loc}(G) \leq n-1.$$

Example: Determine metric dimension/locating number of the graph G.



Proof: Let W={v_1,v_2}, the vertices representation of vertices of G with respect of W are:

$c_W(v_1)=(0,1),\quad c_W(v_2)=(1,0),\quad c_W(v_3)=(2,1),$
$c_W(v_4)=(1,2),\quad c_W(v_5)=(1,1).$

Since, all the vertex representation are distinct and W has minimum number of vertices. Therefore, the metric dimension/location number of G is 2.

Channel Assignment

Radio waves, that are electromagnetic waves propagated by antennas, have different frequencies. When a radio receiver is tuned to a particular frequency, a specific signal can be accessed. In the United States, it is the responsibility of the Federal Communications Commission (FCC) to decide which frequencies are used for which purposes. It is also the FCC that licenses specific frequencies to radio stations as well as call letters for the stations. AM (amplitude modulated) radio is in a band of 550 kHz (kilohertz) to 1700 kHz which means that AM radio broadcasts in a frequency band of 550,000 to 1,700,000 cycles per second.

The FM radio frequency band, which, as we said, begins at 88.0 MHz and ends at 108.0 MHz, is divided into 100 chanels, each having a width of 0.2 MHz (or 200 kHz). The frequency that is identified with and FM radio station is the midpoint of its 200 kHz channel.

In genral then, FM radio stations located within a certain proximity of one another must be assigned distinct channel. The nearer two stations are to each other, the greater the difference must be in their asigned channels. The task of efficiently allocating channels to transmitters is called the **Channel Assignment Problem**

The use of graph theory to study the Channel Assignment Problem dates back to atleast 1970. In 1980 William Hale provided a model of the Channel Assignment Problem. Most often the Channel Assignment Problem has been modeled as a graph coloring problem, where (1) the transmitters are the vertices of a graph, (2) two vertices (transmitters) are adjacent if they are sufficiency close to each other, (3) the colors of the vertices are the channels assigned to the transmitters, and (4) some sort of minimum seperation rule is stipulated, that is, for every pair of colors, there is a minimum allowable distance between two distinct vertices assigned these colors.
We consider one of these models that was inspired by the Channel Assignment Problem. For a connected graph G of order n and an integer k with 1 ≤ k ≤ diam(G), a radio k-coloring of a G is a function c: V(G)→N for which

$$d(u,v)+|c(u)-c(v)| \geq 1+k$$

for very two distinct vertices u and v of G. For k=2, a radio 2-coloring then requires that

$$d(u,v)+|c(u)-c(v)| \geq 3$$

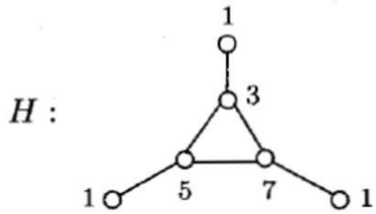
For every two distinct vertices u and v of G. This says that

(1) Colors assigned to adjacent vertices must differ by at least 2,

(2) The colors assigned to vertices whose distance is 2 must differ and

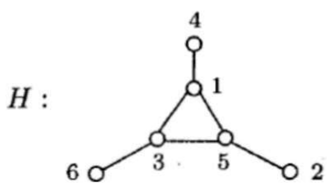
(3) There is no restriction on colors assigned to vertices whose distance is 3 or more.

A radio 2-coloring of a graph H is shown below:



The **value** $rc_k(c)$ of a radio k-coloring c of a connected graph G is the maximum color assigned to a vertex of G, while the **radio k-chromatic number** $rc_k(c)$ of G is min{ $rc_k(c)$} over all the radio k-coloring c of G. A radio k-coloring c of G is a **minimum radio k-coloring** if $rc_k(c)= rc_k(G)$.

A minimum radio 2-coloring of a graph H is shown below:



A minimum radio 2-coloring of a graph

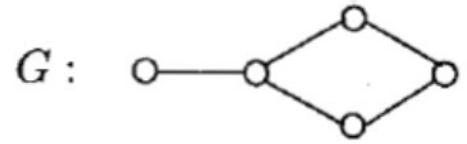
Radio 2-coloring have also been studied and are also referred to as **labelings at distance 2** and **L(2,1)-labelings**. For connected graphs of diameter d, a radio d-coloring of G is also called a **radio labeling**. A **radio labeling** of a connected graph G is a function c: V(G)→N with property that

$$d(u,v) + |c(u) - c(v)| \geq 1 + \mathrm{diam}(G)$$

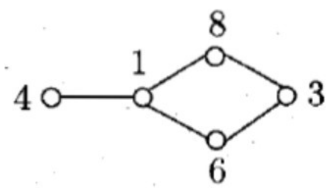
for very two distinct vertices u and v of G. Since d(u,v)≤ diam(G) for every two vertices u and v of G, it follows that no two vertices are labeled the same in a radio labeling.

For a radio labeling c of a connected graph G, the values rn(c) of c is the maximum label assigned to a vertex of G, while the **radio number** rn(G) of a graph G is the minimum values of a radio labeling of G. A radio labeling c with rn(c)=rn(G) is a **minimum radio labeling**.

Example: Determine rn(G) for the given graph G.



Solution: Since the diam(G)=3, it follows that in any radio labeling of G, the labels of every two adjacent vertices must differ by at least 3 and the labels of every two vertices whose distance is 2 must be differ by at least 2. The labels of two vertices can be differ by 1 only if their distance is 3. Thus the labeling of the graph G is given below:

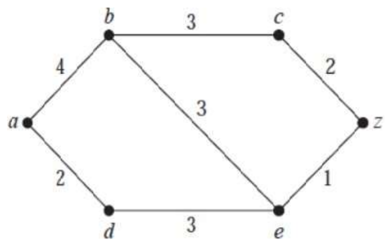


Consequently, rn(G) ≤ 8. On the other hand, rn(G)≠7, for assume, to contrary, that there is a radio labeling c of G with rn(c)=rn(G)=7. Since exactly two of the integers 2,3,4,5,6 are is not used in this labeling, either three consecutive integers in {1,2, 3…,7} are labels for the vertices of G or two pairs of consecutive integers are labels both of which are impossible since u and v are the only two vertices of G whose distance is 3. Therefore, rn(G)=8 and the labeling given above is a minimum radio labeling.

Definition: Graphs that have a number assigned to each edge are called **weighted graphs**.

Definition: The **length** of a path in a weighted graph be the sum of the weights of the edges of this path.

EXAMPLE 1: What is the length of a shortest path between *a* and *z* in the given weighted graph?



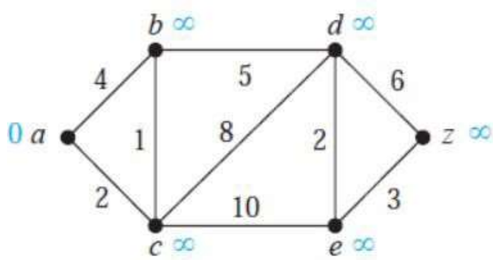
Solution: The shortest path from a to z is:

$a \xrightarrow{2} d \xrightarrow{3} e \xrightarrow{1} z$

The length of shortest path from a to z is: 2+3+1=6.

Theorem 1: Dijkstra's algorithm finds the length of a shortest path between two vertices in a connected simple undirected weighted graph.

EXAMPLE 2 Use Dijkstra's algorithm to find the length of a shortest path between the vertices *a* and *z* in the given weighted graph.



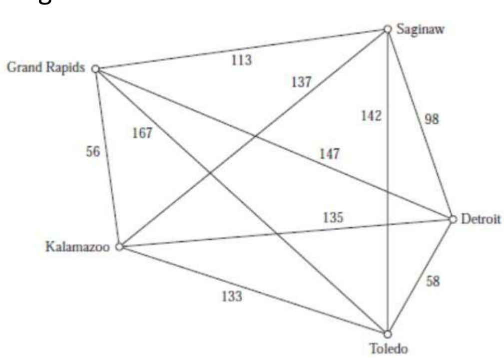
Solution:

The shortest path from a to z is:

$a \xrightarrow{2} c \xrightarrow{1} b \xrightarrow{5} d \xrightarrow{2} e \xrightarrow{3} z$

The length of shortest path from a to z is: 2+1+5+2+3=13.

Example: Solve the traveling salesperson problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



Solution:

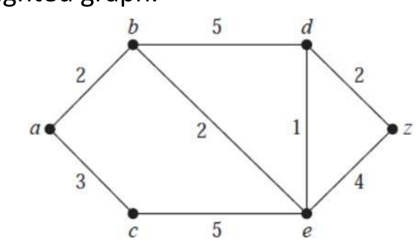
Route	Total Distance (miles)
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

Hamilton circuit with minimum total weight is

Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit =458.

Definition: A practical approach to the traveling salesperson problem when there are many vertices to visit is to use an **approximation algorithm.**

EXAMPLE : Find the length of a shortest path between *a* and *z* in the given weighted graph.

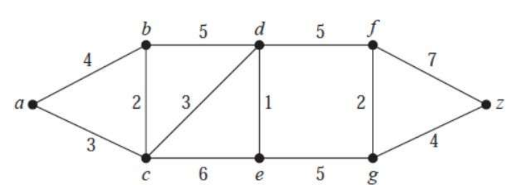


Solution:

The shortest path from a to z is:

$a \xrightarrow{2} b \xrightarrow{2} e \xrightarrow{1} d \xrightarrow{2} z$ The length of shortest path from a to z is: 2+2+1+2=7.

EXAMPLE: Find the length of a shortest path between *a* and *z* in the given weighted graph.



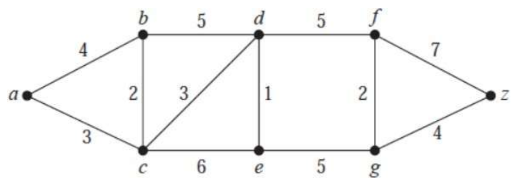
Solution:

The shortest path from a to z is:

$a \xrightarrow{3} c \xrightarrow{3} d \xrightarrow{1} e \xrightarrow{5} g \xrightarrow{4} z$

The length of shortest path from a to z is: 3+3+1+5+4=16.

EXAMPLE: Find the length of a shortest path between these pairs of vertices in the given weighted graph.



a) a and d

b) a and f

c) c and f

d) b and z

Solution:

a) The shortest path from a to d is:

$a \xrightarrow{3} c \xrightarrow{3} d$ The length of shortest path from a to d is: 3+3=1.

b) The shortest path from a to f is:

$a \xrightarrow{3} c \xrightarrow{3} d \xrightarrow{5} f$ The length of shortest path from a to f is: 3+3+5=11.

c) The shortest path from c to f is:

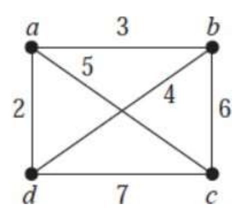
$c \xrightarrow{3} d \xrightarrow{5} f$

The length of shortest path from c to f is: 3+5=8.

d) The shortest path from b to z is:

$b \xrightarrow{5} d \xrightarrow{1} e \xrightarrow{5} g \xrightarrow{4} z$ The length of shortest path from b to z is: 5+1+5+4=15.

EXAMPLE: Solve the traveling salesperson problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



Solution:

$a \xrightarrow{3} b \xrightarrow{6} c \xrightarrow{7} d \xrightarrow{2} a$.

Total weight of Hamilton circuit= 18.

$a \xrightarrow{3} b \xrightarrow{4} d \xrightarrow{7} c \xrightarrow{5} a$,

total weight of Hamilton circuit = 19.

$a \xrightarrow{5} c \xrightarrow{6} b \xrightarrow{4} d \xrightarrow{2} a$,

total weight of Hamilton circuit = 17.

$a \xrightarrow{5} c \xrightarrow{7} d \xrightarrow{4} b \xrightarrow{3} a$

Total weight of Hamilton circuit = 19

$a \xrightarrow{2} d \xrightarrow{4} b \xrightarrow{6} c \xrightarrow{5} a$

Total weight of Hamilton circuit = 17

$a \xrightarrow{2} d \xrightarrow{7} c \xrightarrow{6} b \xrightarrow{3} a$

Total weight of Hamilton circuit = 18

Minimum total weight of Hamilton circuit is:

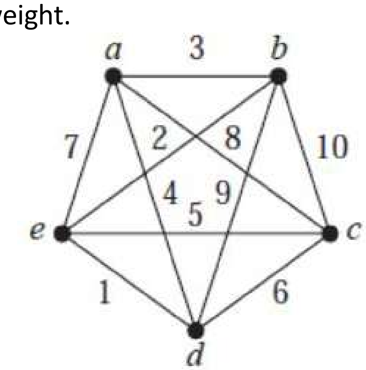
$a \xrightarrow{5} c \xrightarrow{6} b \xrightarrow{4} d \xrightarrow{2} a$,

total weight of Hamilton circuit = 17.

$a \xrightarrow{2} d \xrightarrow{4} b \xrightarrow{6} c \xrightarrow{5} a$

Total weight of Hamilton circuit = 17.

EXAMPLE: Solve the traveling salesperson problem for this graph by finding the total weight of all Hamilton circuits and determining a circuit with minimum total weight.



Solution:

$a \xrightarrow{3} b \xrightarrow{10} c \xrightarrow{6} d \xrightarrow{1} e \xrightarrow{7} a$

Total weight=3+10+6+1+7=27

$a \xrightarrow{3} b \xrightarrow{10} c \xrightarrow{5} e \xrightarrow{1} d \xrightarrow{4} a$

Total weight=3+10+5+1+4=23

$a \xrightarrow{3} b \xrightarrow{9} d \xrightarrow{6} c \xrightarrow{5} e \xrightarrow{7} a$

Total weight=3+9+6+5+7=30

$a \xrightarrow{3} b \xrightarrow{9} d \xrightarrow{1} e \xrightarrow{5} c \xrightarrow{8} a$

Total weight=3+9+1+5+8=26

$a \xrightarrow{3} b \xrightarrow{2} e \xrightarrow{5} c \xrightarrow{6} d \xrightarrow{4} a$
Total weight=3+2+5+6+4=20
$a \xrightarrow{3} b \xrightarrow{2} e \xrightarrow{1} d \xrightarrow{6} c \xrightarrow{8} a$
Total weight=3+2+1+6+8=20

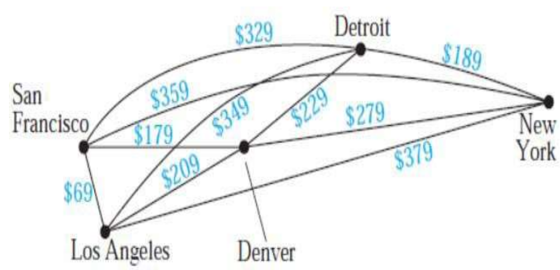
$a \xrightarrow{8} c \xrightarrow{10} b \xrightarrow{9} d \xrightarrow{1} e \xrightarrow{7} a$
Total weight=8+10+9+1+7=35
$a \xrightarrow{8} c \xrightarrow{10} b \xrightarrow{2} e \xrightarrow{1} d \xrightarrow{4} a$
Total weight=8+10+2+1+4=25
$a \xrightarrow{8} c \xrightarrow{6} d \xrightarrow{9} b \xrightarrow{2} e \xrightarrow{7} a$
Total weight=8+6+9+2+7=32
$a \xrightarrow{8} c \xrightarrow{6} d \xrightarrow{1} e \xrightarrow{2} b \xrightarrow{3} a$
Total weight=8+6+1+2+3=20
$a \xrightarrow{8} c \xrightarrow{5} e \xrightarrow{2} b \xrightarrow{9} d \xrightarrow{4} a$
Total weight=8+5+2+9+4=28
$a \xrightarrow{8} c \xrightarrow{5} e \xrightarrow{1} d \xrightarrow{9} b \xrightarrow{7} a$
Total weight=8+5+1+9+7=30
$a \xrightarrow{4} d \xrightarrow{9} b \xrightarrow{10} c \xrightarrow{5} e \xrightarrow{7} a$
Total weight=4+9+10+5+7=35
$a \xrightarrow{4} d \xrightarrow{9} b \xrightarrow{2} e \xrightarrow{5} c \xrightarrow{8} a$
Total weight=4+9+2+5+8=28
$a \xrightarrow{4} d \xrightarrow{6} c \xrightarrow{10} b \xrightarrow{2} e \xrightarrow{7} a$
Total weight=4+6+10+2+7=29
$a \xrightarrow{4} d \xrightarrow{6} c \xrightarrow{5} e \xrightarrow{2} b \xrightarrow{3} a$
Total weight=4+6+5+2+3=20
$a \xrightarrow{4} d \xrightarrow{1} e \xrightarrow{2} b \xrightarrow{10} c \xrightarrow{8} a$
Total weight=4+1+2+10+8=25
$a \xrightarrow{4} d \xrightarrow{1} e \xrightarrow{5} c \xrightarrow{10} b \xrightarrow{3} a$
Total weight=4+1+5+10+3=23

$a \xrightarrow{7} e \xrightarrow{2} b \xrightarrow{10} c \xrightarrow{6} d \xrightarrow{4} a$
Total weight=7+2+10+6+4=29
$a \xrightarrow{7} e \xrightarrow{2} b \xrightarrow{9} d \xrightarrow{6} c \xrightarrow{8} a$
Total weight=7+2+9+6+8=32
$a \xrightarrow{7} e \xrightarrow{5} c \xrightarrow{10} b \xrightarrow{9} d \xrightarrow{4} a$
Total weight=7+5+10+9+4=35
$a \xrightarrow{7} e \xrightarrow{5} c \xrightarrow{6} d \xrightarrow{9} b \xrightarrow{7} a$
Total weight=7+5+6+9+7=34
$a \xrightarrow{7} e \xrightarrow{1} d \xrightarrow{9} b \xrightarrow{10} c \xrightarrow{8} a$
Total weight=7+1+9+10+8=35
$a \xrightarrow{7} e \xrightarrow{1} d \xrightarrow{6} c \xrightarrow{10} b \xrightarrow{3} a$
Total weight=7+1+6+10+3=27.
The Hamilton circuits with minimum total weight are:

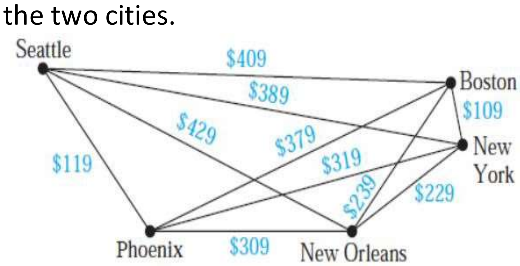
$a \xrightarrow{3} b \xrightarrow{2} e \xrightarrow{5} c \xrightarrow{6} d \xrightarrow{4} a$
Total weight=3+2+5+6+4=20
$a \xrightarrow{3} b \xrightarrow{2} e \xrightarrow{1} d \xrightarrow{6} c \xrightarrow{8} a$
Total weight=3+2+1+6+8=20
$a \xrightarrow{8} c \xrightarrow{6} d \xrightarrow{1} e \xrightarrow{2} b \xrightarrow{3} a$
Total weight=8+6+1+2+3=20
$a \xrightarrow{4} d \xrightarrow{6} c \xrightarrow{5} e \xrightarrow{2} b \xrightarrow{3} a$
Total weight=4+6+5+2+3=20.

Exercise: Determine the locating number and minimum radio labeling of complete graph K_n for n≥ 3.
Exercise: Determine the locating number and minimum radio labeling of complete bipartite graph $K_{m,n}$ for n,m=2,3.
Exercise: Determine the locating number and minimum radio labeling of path graph P_n for n,=2,3,4,5.
Exercise: Determine the locating number and minimum radio labeling of cycle graph C_n for n,=3,4,5.

Exercise: Find a route with the least total airfare that visits each of the cities in this graph, where the weight on an edge is the least price available for a flight between the two cities.



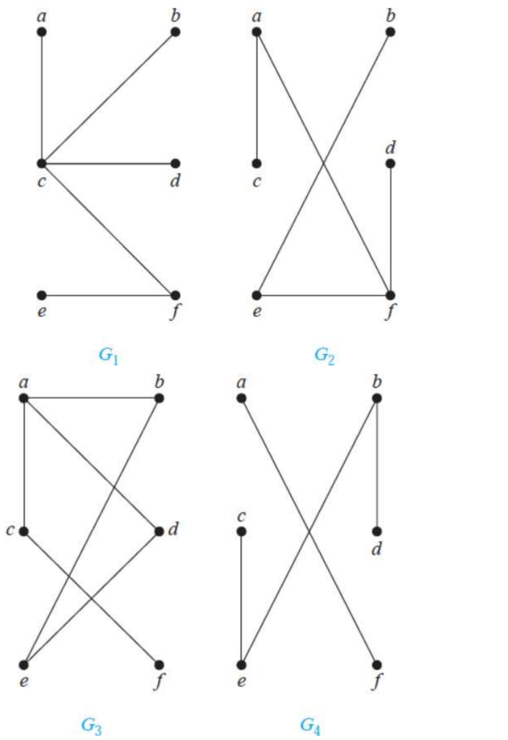
Exercise: Find a route with the least total airfare that visits each of the cities in this graph, where the weight on an edge is the least price available for a flight between the two cities.



Chapter 5: Trees

Definition 1: A **tree** is a connected undirected graph with no simple circuits.

EXAMPLE 1: Which of the following graphs are trees?



Solution: G1 and G2 are trees, because both are connected graphs with no simple circuits. G3 is not a tree because e, b, a, d, e is a simple circuit in this graph. Finally, G4 is not a tree because it is not connected.

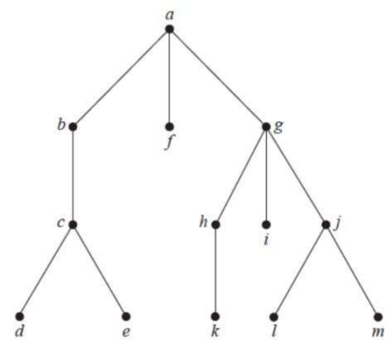
Definition: A disconnected graph is called *forest* if each of its each of their connected components is a tree.

THEOREM 1: An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

Definition: 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.

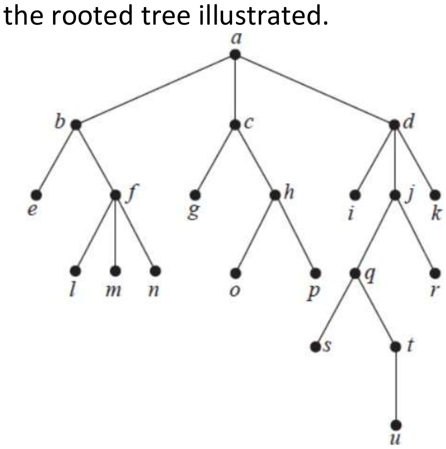
Suppose that T is a rooted tree. If v is a vertex in T other than the root, the **parent** of v is the unique vertex u such that there is a directed edge from u to v. When u is the parent of v, v is called a **child** of u. Vertices with the same parent are called **siblings**. The **ancestors** of a 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 vertices that have v as an ancestor. A vertex of a rooted 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.

EXAMPLE 2: In the given rooted tree, find the parent of c, the children of g, the siblings of h, all ancestors of e, all descendants of b, all internal vertices, and all leaves.



Solution: The parent of c = b.
The children of g = h, i, and j .
The siblings of h = i and j .
The ancestors of e = c, b, and a.
The descendants of b = c, d, and e.
The internal vertices = a, b, c, g, h, and j .
The leaves are d, e, f , i, k, l, and m.

Example: Answer these questions about the rooted tree illustrated.

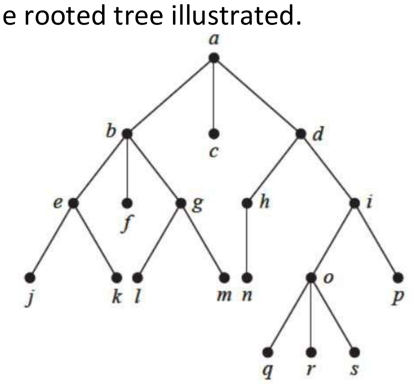


i. Which vertex is the root?
ii. Which vertices are internal?
iii. Which vertices are leaves?
iv. Which vertices are children of j?
v. Which vertex is the parent of h?
vi. Which vertices are siblings of o?
vii. Which vertices are ancestors of m?
viii. Which vertices are descendants of b?

Solution:

i. a
ii. a,b,c,d,f,h,j,q,t
iii. e,g,i,k,l,m,n,o,p,r,s,u
iv. q,r
v. c
vi. p
vii. f,b,a
viii. e,f,l,m,n

Example : Answer these questions about the rooted tree illustrated.



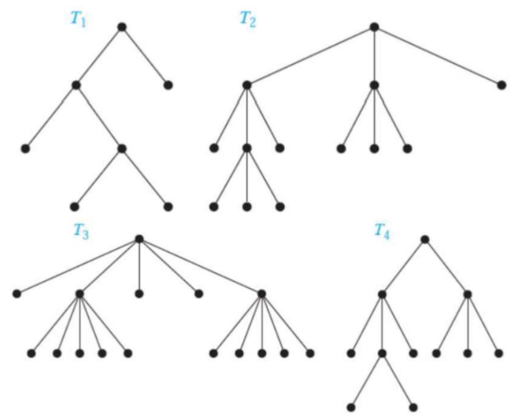
i. Which vertex is the root?
ii. Which vertices are internal?
iii. Which vertices are leaves?
iv. Which vertices are children of j?
v. Which vertex is the parent of h?
vi. Which vertices are siblings of o?
vii. Which vertices are ancestors of m?
viii. Which vertices are descendants of b?

Solution:

i. a
ii. a,b,d,e,g,h,i,o
iii. c,f,j,k,l,m,n,p,q,r,s
iv. No children
v. d
vi. p
vii. g,b,a
viii. e,f, g,j,k,l,m

DEFINITION 3: A rooted tree is called an **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 3: Are the rooted trees in given graphs full m-ary trees for some positive integer m?



Solution: T1 is a full binary tree because each of its internal vertices has two children.

T2 is a full 3-ary tree because each of its internal vertices has three children.

In T3 each internal vertex has five children, so T3 is a full 5-ary tree.

T4 is not a full m-ary tree for any m because some of its internal vertices have two children and others have three children.

THEOREM 2: A tree with n vertices has n − 1 edges.

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

THEOREM 4: A full m-ary tree with

i. n vertices has i = (n − 1)/m internal vertices and l = [(m − 1)n + 1]/m leaves,
ii. i internal vertices has n = mi + 1 vertices and l = (m − 1)i + 1 leaves,
iii. l leaves has n = (ml − 1)/(m − 1) vertices and i = (l − 1)/(m − 1) internal vertices.

EXAMPLE: Suppose that someone starts a chain letter. Each person who receives the letter is asked to send it on to four 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 one 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, m=4. Because 100 people did not send out the letter, the number of leaves in this rooted tree is l = 100. As we know that

i = (l − 1)/(m − 1)=(100-1)/(4-1)=99/3=33,

so 33 people sent out the letter.

Example: How many edges does a tree with 10,000 vertices have?

Solution: e=n-1=10,000-1=9999.

Example: How many vertices does a full 5-ary tree with 100 internal vertices have?

Solution: m=5, i=100, n=?

We know that

n = mi + 1=5*100+1=501.

Example: How many edges does a full binary tree with 1000 internal vertices have?

Solution: m=2, i=1000, e=?

We know that

n = mi + 1=2*1000+1=2001.

Also we know that e=n-1=2001-1=2000.

Example: How many leaves does a full 3-ary tree with 100 vertices have?

Solution: m=3, n=100, l=?

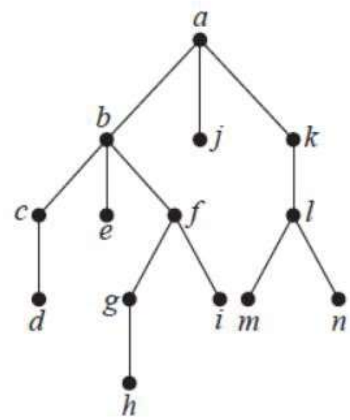
We know that

l = [(m − 1)n + 1]/m

=[(3-1)100+1]/3=67.

Definition: 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.

EXAMPLE 10: Find the level of each vertex in the given rooted. What is the height of this tree?



Solution: The root a is at level 0.

Vertices at Level 1: b, j , and k.

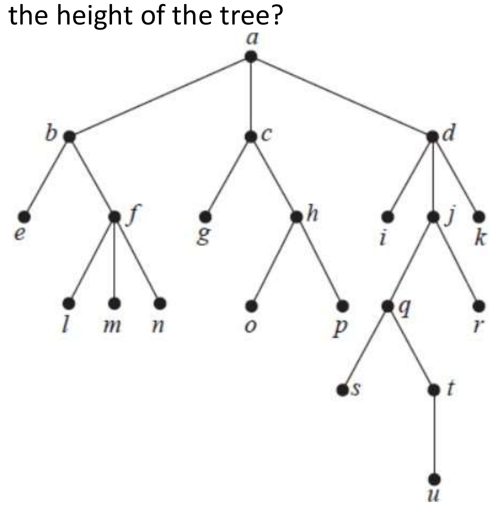
Vertices at Level 2: c, e, f , and l

Vertices at Level 3: d, g, i, m, and n

Vertices at Level 4: h

The height is 4.

Example: What is the level of each vertex of the rooted tree given graph? Also what is the height of the tree?



Solution: The root a is at level 0.

Vertices at Level 1: b, c , and d.

Vertices at Level 2: e,f,g,h,i,j, and k

Vertices at Level 3: l, m,n,o,p,q and r

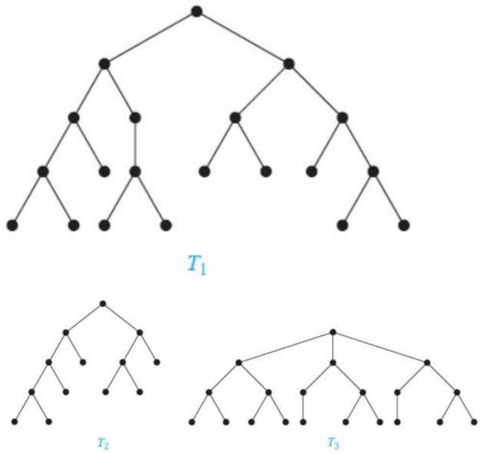
Vertices at Level 4: s,t

Vertices at Level 5: u.

The height is 5.

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

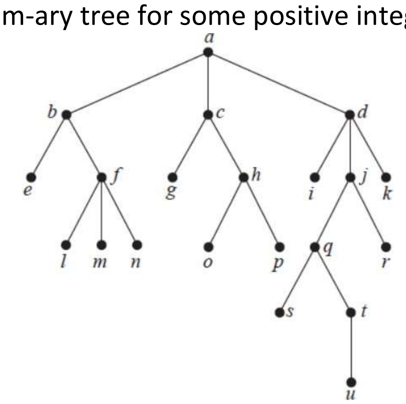
EXAMPLE: Which of the rooted trees shown below are balanced?

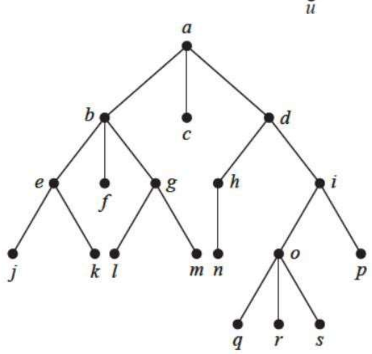


Solution: T1 is balanced, because all its leaves are at levels 3 and 4.
However, T2 is not balanced, because it has leaves at levels 2, 3, and 4.
Finally, T3 is balanced, because all its leaves are at level 3.

THEOREM 5: There are at most m^h leaves in an m-ary tree of height h.

Example: Is the given rooted trees a full m-ary tree for some positive integer m?



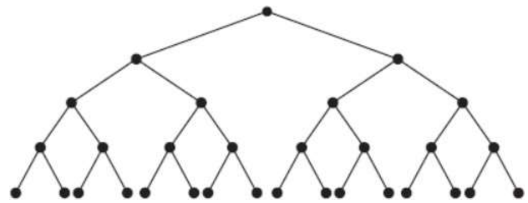


Solution: No

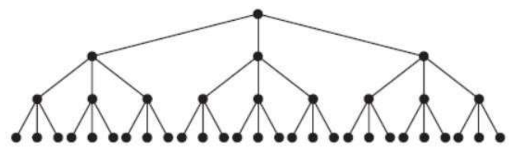
A **complete m-ary tree** is a fullm-ary tree in which every leaf is at the same level.

Example: Construct a complete binary tree of height 4 and a complete 3-ary tree of height 3.

Solution: Complete binary tree of height 4:



complete 3-ary tree of height 3.



Exercise: Suppose 1000 people enter a chess tournament. Use a rooted tree model of the tournament to determine how many games must be played to determine a champion, if a player is eliminated after one loss and games are played until only one entrant has not lost. (Assume there are no ties.)

Exercise: A chain letter starts when a person sends a letter to five others. Each person who receives the letter either sends it to five other people who have never received it or does not send it to anyone. Suppose that 10,000 people send out the letter before the chain ends and that no one receives more than one letter. How many people receive the letter, and how many do not send it out?

Exercise: How many vertices and how many leaves does a complete m-ary tree of height h have?

Applications of Trees

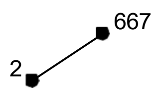
Binary search tree: a binary tree in which the vertices are labeled with items so that a label of a vertex is greater than the labels of all vertices in the left subtree of this vertex and is less than the labels of all vertices in the right subtree of this vertex

EXAMPLE: Form a binary search tree for the numbers 667, 2, 150, 900, 870, 6, 5555, 203.

Solution:

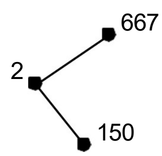
<u>Step 1</u>

<u>Step 2</u>



2 <667

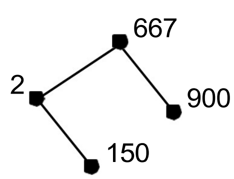
<u>Step 3</u>



150 <667

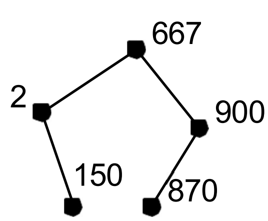
150>2

<u>Step 4</u>



900>667

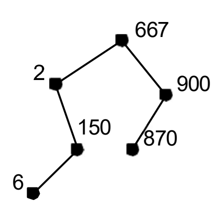
<u>Step 5</u>



870>667

870<900

<u>Step 6</u>

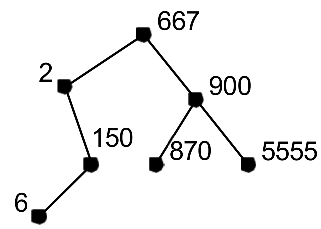


6<667

6>2

6<150

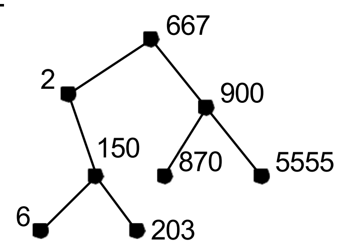
<u>Step 7</u>



5555>667

5555>900

<u>Step 8</u>



203<667

203>2

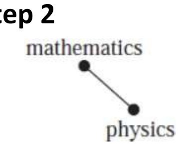
203>150

EXAMPLE: Form a binary search tree for the word's mathematics, physics, geography, zoology, meteorology, geology, psychology, and chemistry (using alphabetical order).

Solution:

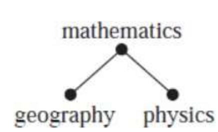
Step 1

Step 2



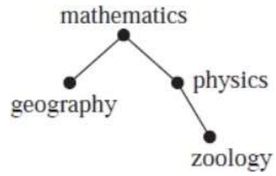
physics > mathematics

Step 3



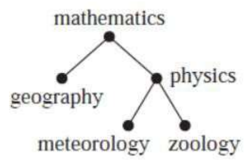
geography < mathematics

Step 4



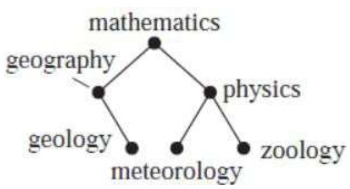
zoology > mathematics
zoology > physics

Step 5



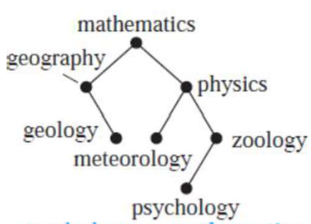
meteorology > mathematics
meteorology < physics

Step 6



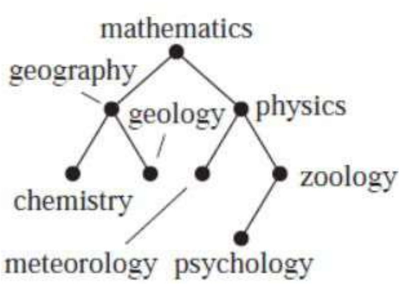
geology < mathematics
geology > geography

Step 7



psychology > mathematics
psychology > physics
psychology < zoology

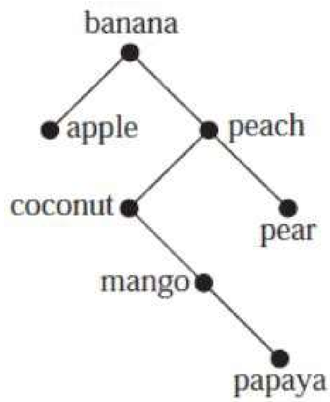
Step 8



chemistry < mathematics
chemistry < geography

EXAMPLE: Build a binary search tree for the words banana, peach, apple, pear, coconut, mango, and papaya using alphabetical order.

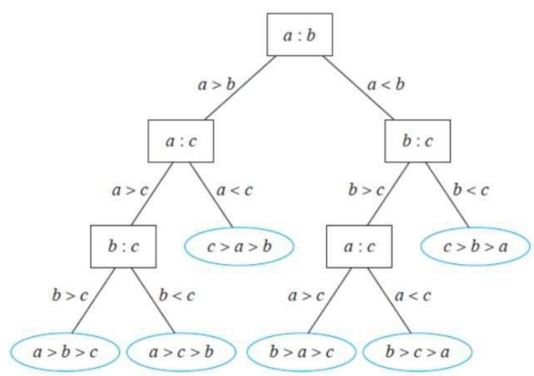
Solution:



Definition: A rooted tree in which each internal vertex corresponds to a decision, with a subtree at these vertices for each possible outcome of the decision, is called a **decision tree**.

EXAMPLE 4: Find a decision tree that orders the elements of the list a, b, c.

Solution:



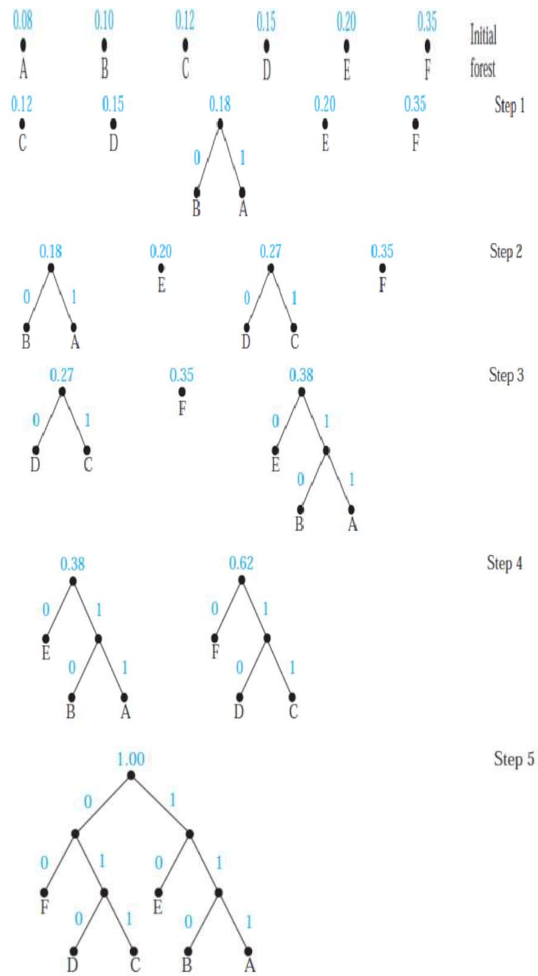
Huffman coding: a procedure for constructing an optimal binary code for a set of symbols, given the frequencies of these symbols

Game tree: a rooted tree where vertices represent the possible positions of a game as it progresses and edges represent legal moves between these positions

EXAMPLE 5: Use Huffman coding to encode the following symbols with the frequencies listed: A: 0.08, B: 0.10, C:

0.12, D: 0.15, E: 0.20, F: 0.35. What is the average number of bits used to encode a character?

Solution:



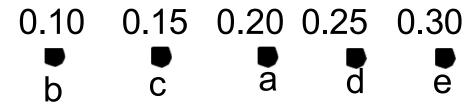
The above figure displays the steps used to encode these symbols. The encoding produced encodes A by 111, B by 110, C by 011, D by 010, E by 10, and F by 00. The average number of bits used to encode a symbol using this encoding is

3 * 0.08 + 3 * 0.10 + 3 * 0.12 + 3 * 0.15 + 2 * 0.20 + 2 * 0.35 = 2.45.

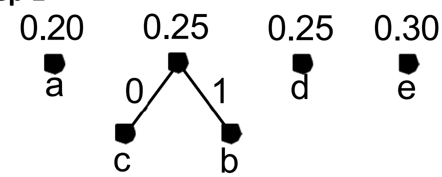
Example: Use Huffman coding to encode these symbols with given frequencies: a: 0.20, b: 0.10, c: 0.15, d: 0.25, e: 0.30. What is the average number of bits required to encode a character?

Solution:

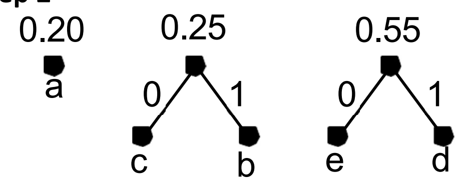
Initial forest



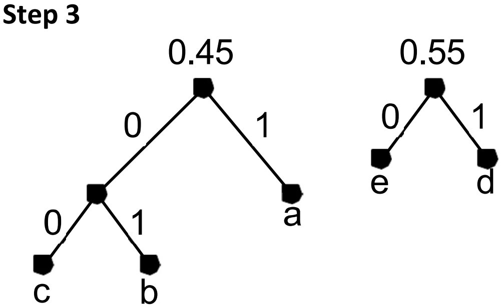
Step 1



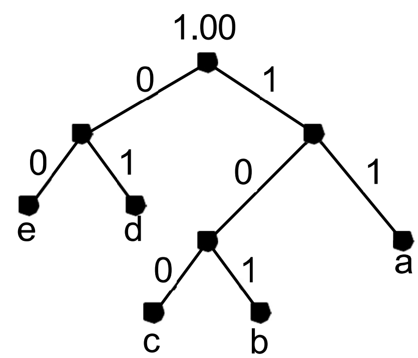
Step 2



Step 3



Step 4



The above figure displays the steps used to encode these symbols. The encoding produced encodes a by 11, b by 101, c by 100, d by 01, and e by 00. The average number of bits used to encode a symbol using this encoding is

2 * 0.20 + 3 * 0.10 + 3 * 0.15 + 2 * 0.25 + 2 * 0.30= 2.25.

Exercise: Build a binary search tree for the word's oenology, phrenology, campanology, ornithology, ichthyology, limnology, alchemy, and astrology using alphabetical order.

Exercise: How many comparisons are needed to locate or to add each of these words in the search tree for Exercise 1(the words banana, peach, apple, pear, coconut, mango, and papaya), starting fresh each time?

a) pear **b)** banana **c)** kumquat **d)** orange

Solution: Hint

a) 3 **b)** 1 **c)** 4 **d)** 5

Exercise:. How many comparisons are needed to locate or to add each of the words in the search tree for Exercise 2(the words oenology, phrenology, campanology, ornithology, ichthyology, limnology, alchemy, and astrology), starting fresh each time?

a) palmistry **b)** etymology **c)** paleontology **d)** glaciology

Exercise: Use Huffman coding to encode these symbols with given frequencies: A: 0.10, B: 0.25, C: 0.05, D: 0.15, E: 0.30, F: 0.07, G: 0.08. What is the average number of bits required to encode a symbol?

Chapter 6: Tree Traversal and its Algorithms

Universal Address Systems

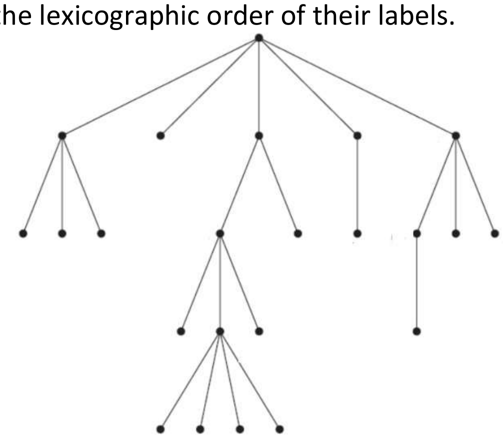
Procedures for traversing all vertices of an ordered rooted tree rely on the orderings of children. To produce this ordering, we must first label all the vertices. We do this recursively:

1. Label the root with the integer 0. Then label its *k* children (at level 1) from left to right with 1*,* 2*,* 3*, . . . , k*.

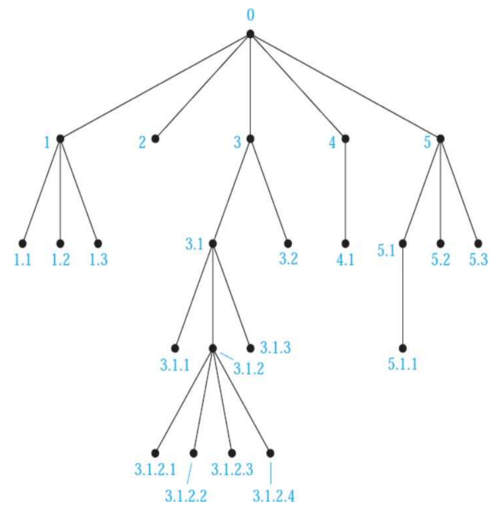
2. For each vertex *v* at level *n* with label *A*, label its *kv* children, as they are drawn from left to right, with *A*.1*,A*.2*, . . . , A.kv*.

Following this procedure, a vertex *v* at level *n*, for *n* ≥ 1, is labeled *x*1*.x*2 ....*xn*, where the unique path from the root to *v* goes through the *x*1st vertex at level 1, the *x*2nd vertex at level 2, and so on. This labeling is called the **universal address system** of the ordered rooted tree.

Example: Construct the universal address system for the given ordered rooted tree. Then use this to order its vertices using the lexicographic order of their labels.



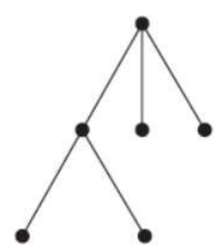
Solution:

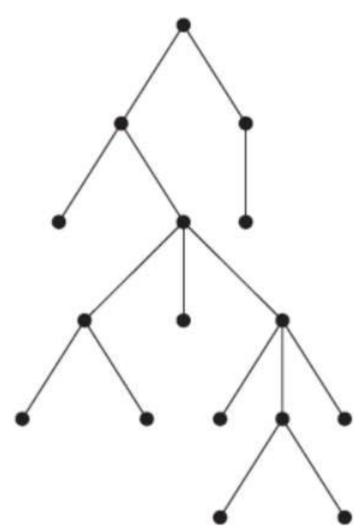


The lexicographic ordering of the labelings is

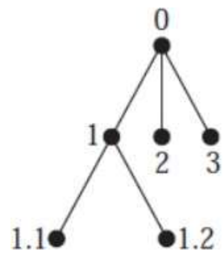
0 < 1 < 1.1 < 1.2 < 1.3 < 2 < 3 < 3.1 < 3.1.1 < 3.1.2 < 3.1.2.1 < 3.1.2.2< 3.1.2.3 < 3.1.2.4 < 3.1.3 < 3.2 < 4 < 4.1 < 5 < 5.1 < 5.1.1 < 5.2 < 5.3

Example : Construct the universal address system for the given ordered rooted tree. Then use this to order its vertices using the lexicographic order of their labels.



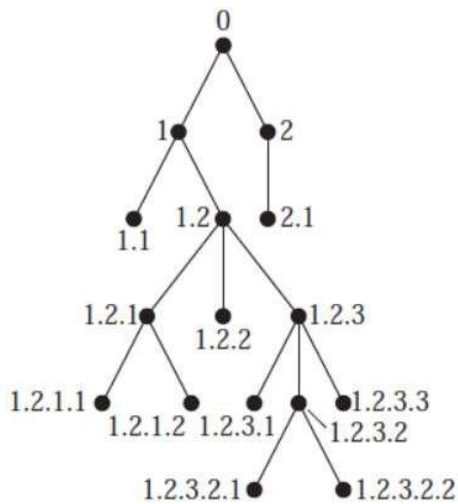


Solution:



The lexicographic order of their labels:

$0 < 1 < 1.1 < 1.2 < 2 < 3$



The lexicographic order of their labels:

$0 < 1 < 1.1 < 1.2 < 1.2.1 < 1.2.1.1 < 1.2.1.2 < 1.2.2 < 1.2.3 < 1.2.3.1 < 1.2.3.2 < 1.2.3.2.1 < 1.2.3.2.2 < 1.2.3.3 < 2 < 2.1$

EXAMPLE: Can the leaves of an ordered rooted tree have the following list of universal addresses? If so, construct such an ordered rooted tree.

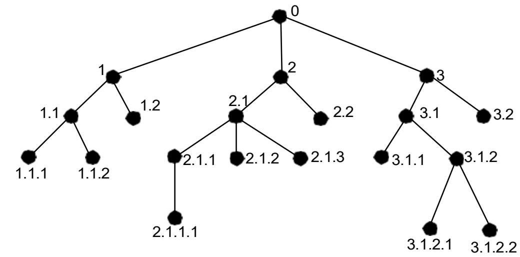
a) 1.1.1, 1.1.2, 1.2, 2.1.1.1, 2.1.2, 2.1.3, 2.2, 3.1.1, 3.1.2.1, 3.1.2.2, 3.2

b) 1.1, 1.2.1, 1.2.2, 1.2.3, 2.1, 2.2.1, 2.3.1, 2.3.2, 2.4.2.1, 2.4.2.2, 3.1, 3.2.1, 3.2.2

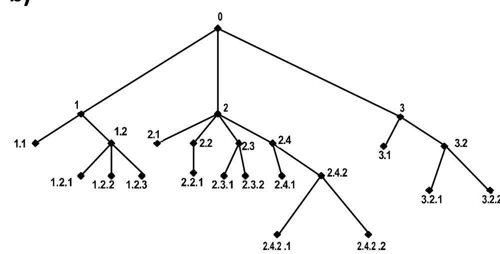
c) 1.1, 1.2.1, 1.2.2, 1.2.2.1, 1.3, 1.4, 2, 3.1, 3.2, 4.1.1.1

Solution:

a)



b)

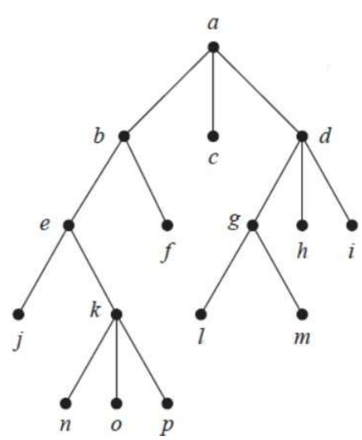


(c) In the given list of universal addresses 1.2.2 is not leaf because it has a child 1.2.2.1.

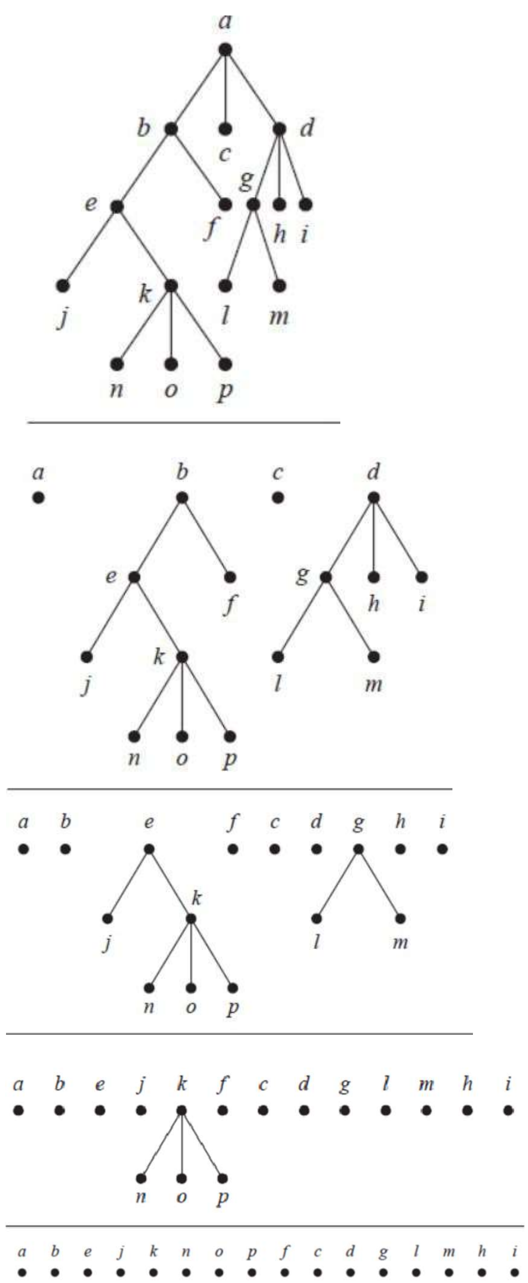
Definition: Procedures for systematically visiting every vertex of an ordered rooted tree are called **traversal algorithms**.

DEFINITION 1 Let T be an ordered rooted tree with root r. If T consists only of r, then r is the *preorder traversal* of T . Otherwise, suppose that T1, T2, . . . , Tn are the subtrees at r from left to right in T . The *preorder traversal* begins by visiting r. It continues by traversing T1 in preorder, then T2 in preorder, and so on, until Tn is traversed in preorder.

EXAMPLE: In which order does a preorder traversal visit the vertices in the ordered rooted tree T shown in

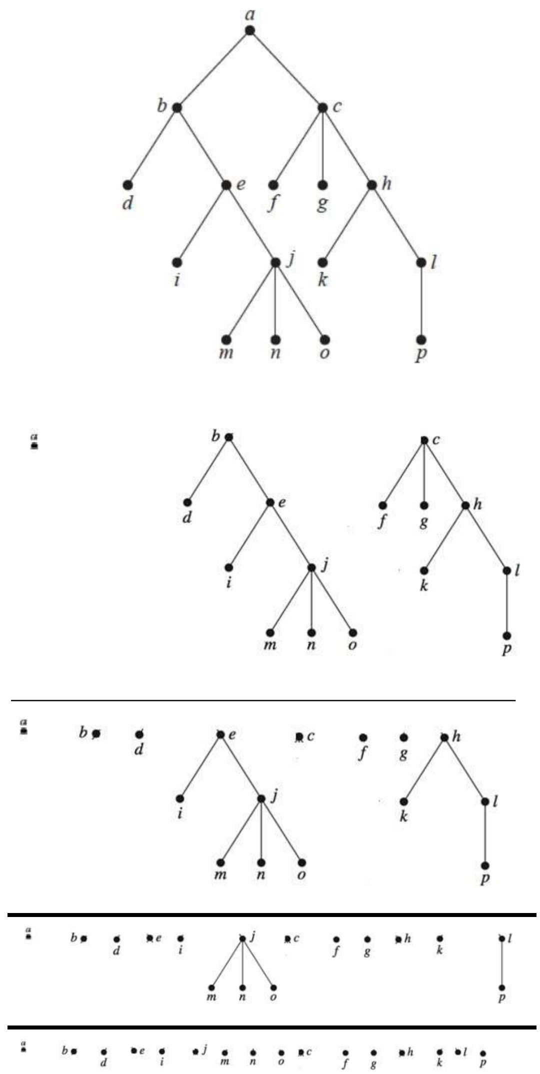


Solution:



Consequently, the preorder traversal is a, b, e, j , k, n, o, p, f , c, d, g, l, m, h, i.

Example: Determine the order in which a preorder traversal visits the vertices of the given ordered rooted tree.

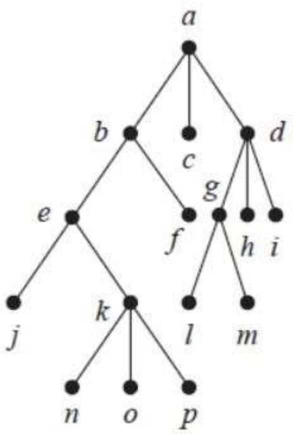


ALGORITHM 1 Preorder Traversal.

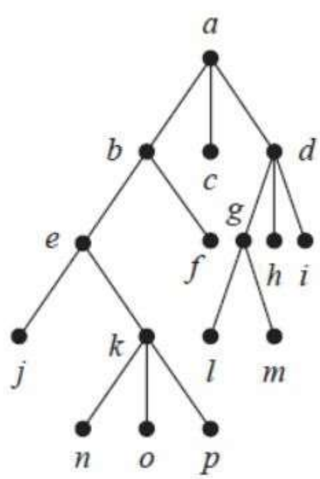
procedure *preorder*(T: ordered rooted tree)
$r :=$ root of T
list r
for each child c of r from left to right
  $T(c) :=$ subtree with c as its root
  preorder($T(c)$)

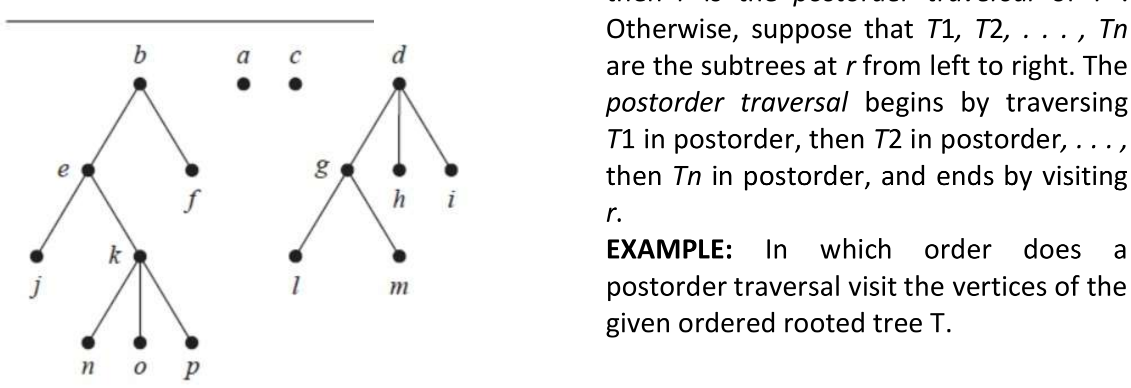
DEFINITION: Let T be an ordered rooted tree with root r. If T consists only of r, then r is the inorder traversal of T . Otherwise, suppose that T1, T2, . . . , Tn are the subtrees at r from left to right. The inorder traversal begins by traversing T1 in inorder, then visiting r. It continues by traversing T2 in inorder, then T3 in inorder, . . . , and finally Tn in inorder.

EXAMPLE: In which order does an inorder traversal visit the vertices of the given ordered rooted tree T?



Solution:





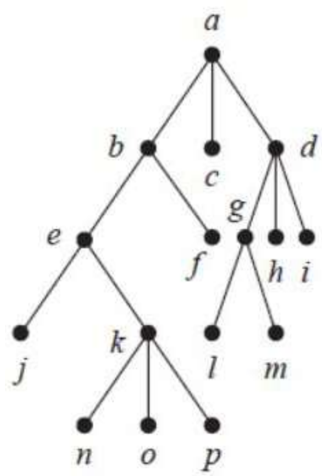
Consequently, the inorder listing of the ordered rooted tree is *j* , *e*, *n*, *k*, *o*, *p*, *b*, *f* , *a*, *c*, *l*, *g*, *m*, *d*, *h*, *i*.

ALGORITHM 2 Inorder Traversal.

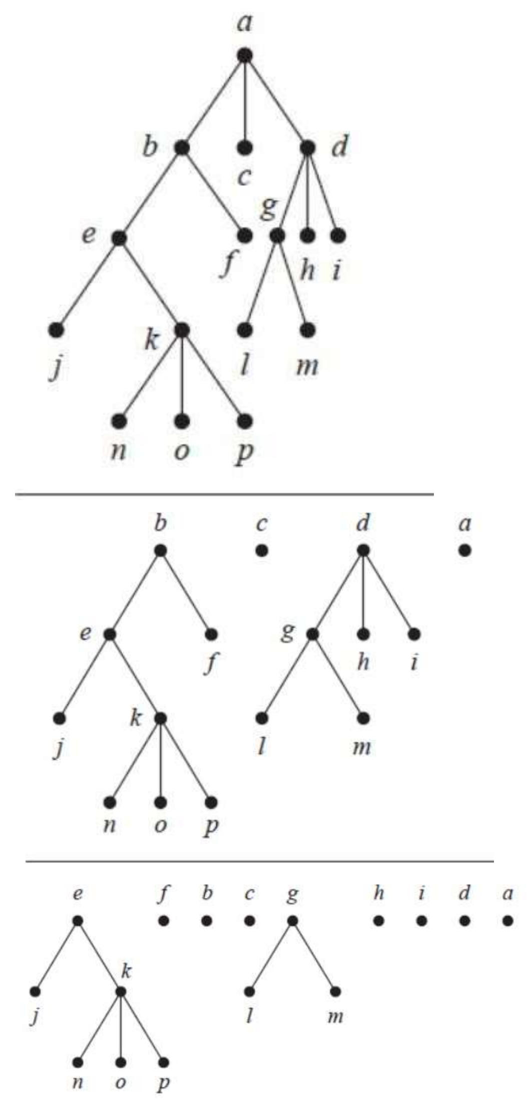
```
procedure inorder(T: ordered rooted tree)
r := root of T
if r is a leaf then list r
else
   l := first child of r from left to right
   T(l) := subtree with l as its root
   inorder(T(l))
   list r
   for each child c of r except for l from left to right
      T(c) := subtree with c as its root
      inorder(T(c))
```

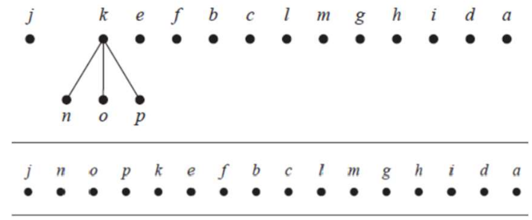
DEFINITION 3 Let *T* be an ordered rooted tree with root *r*. If *T* consists only of *r*, then *r* is the *postorder traversal* of *T* . Otherwise, suppose that *T*1*, T*2*, . . . , Tn* are the subtrees at *r* from left to right. The *postorder traversal* begins by traversing *T*1 in postorder, then *T*2 in postorder*, . . . ,* then *Tn* in postorder, and ends by visiting *r*.

EXAMPLE: In which order does a postorder traversal visit the vertices of the given ordered rooted tree T.



Solution:





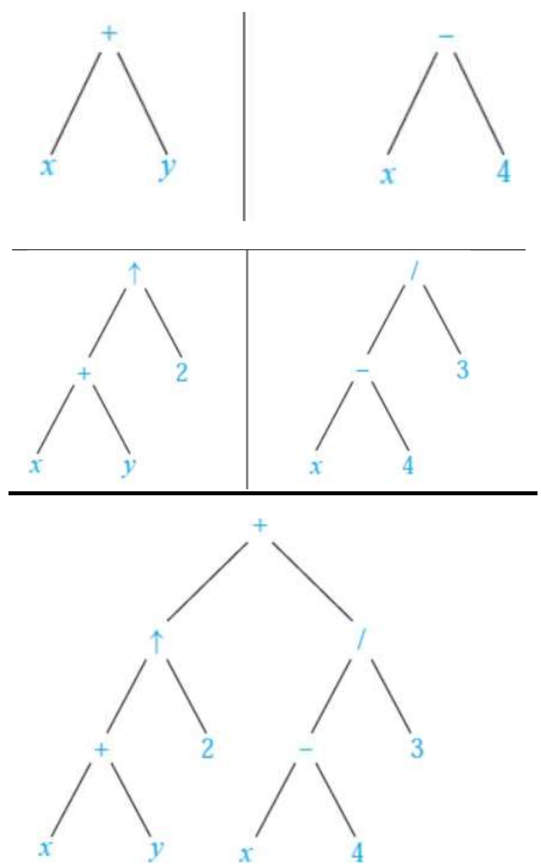
Therefore, the postorder traversal of *T* is *j* , *n*, *o*, *p*, *k*, *e*, *f* , *b*, *c*, *l*, *m*, *g*, *h*, *i*, *d*, *a*.

ALGORITHM 3 Postorder Traversal.

procedure *postorder*(T: ordered rooted tree)
r := root of T
for each child c of r from left to right
    $T(c)$:= subtree with c as its root
    postorder($T(c)$)
list r

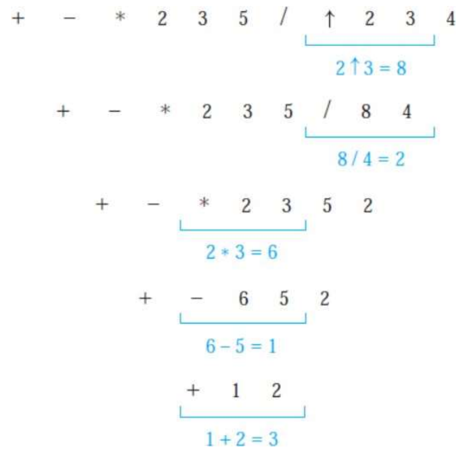
EXAMPLE: What is the ordered rooted tree that represents the expression ((x + y)↑2) + ((x − 4)/3)?

Solution:



EXAMPLE 7 What is the value of the prefix expression $+ - * \ 2\ 3\ 5 / \uparrow\ 2\ 3\ 4$?

Solution:



The value of this expression is 3.

Example: What is the value of each of these prefix expressions?

a) $- * 2 / 8\ 4\ 3$

b) $\uparrow - * 3\ 3 * 4\ 2\ 5$

c) $+ - \uparrow 3\ 2 \uparrow 2\ 3 / 6 - 4\ 2$

d) $* + 3 + 3 \uparrow 3 + 3\ 3\ 3$

Solution: a)

$$- * 2\ \underbrace{/\ 8\ 4}_{8/4=2}\ 3$$

$$- \underbrace{*\ 2\ 2}_{2*2=4}\ 3$$

$$\underbrace{-\ 4\ 3}_{4-3=1}$$

b)

$$\uparrow - \underbrace{*\ 3\ 3}_{3*3=9}\ \underbrace{*\ 4\ 2}_{4*2=8}\ 5$$

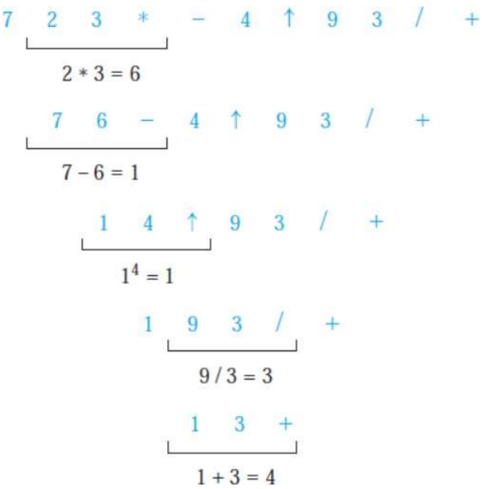
$$\uparrow \underbrace{-\ 9\ 8}_{9-8=1}\ 5$$

$$\underbrace{\uparrow\ 1\ 5}_{1^5=1}$$

Answer:

a) 1 b) 1 c) 4 d) 2205

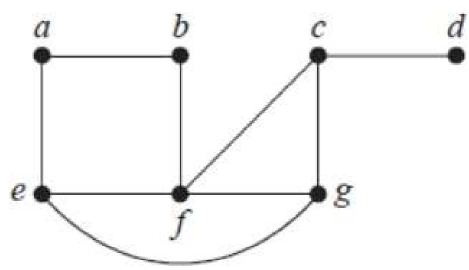
EXAMPLE: What is the value of the postfix expression $7\ 2\ 3 * - 4 \uparrow 9\ 3 / +$?



Spanning Trees

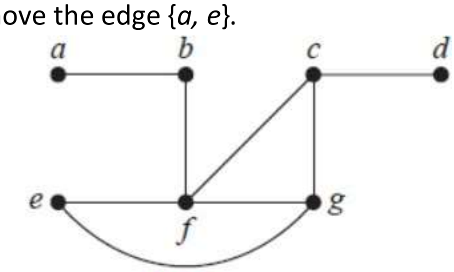
DEFINITION 1 Let *G* be a simple graph. A *spanning tree* of *G* is a subgraph of *G* that is a tree containing every vertex of *G*.

EXAMPLE: Find a spanning tree of the simple graph *G* shown in Figure 2.



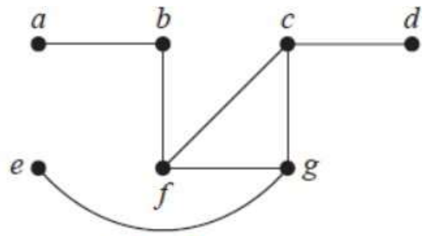
***Solution*:**

The graph *G* is connected, but it is not a tree because it contains simple circuits. Remove the edge {*a, e*}.



Edge removed: {*a, e*}

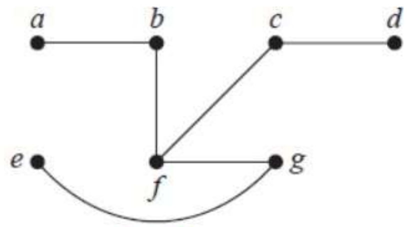
The graph *G* is connected, but it is not a tree because it contains simple circuits. Remove the edge {*f, e*}.



Edge removed: {e, f}

The graph *G* is connected, but it is not a tree because it contains simple circuits.

Remove the edge {*c, g*}.



Edge removed: {c, g}

The graph *G* is connected, and it is a tree because it does not contain simple circuits.

THEOREM: A simple graph is connected if and only if it has a spanning tree.

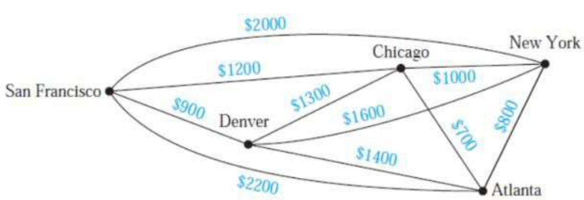
Minimum Spanning Trees

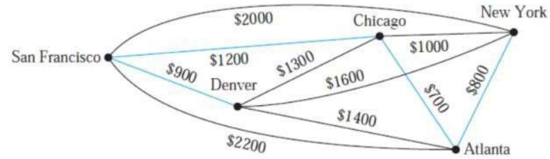
DEFINITION: A *minimum spanning tree* in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.

ALGORITHM 1 Prim's Algorithm.

procedure *Prim*(G: weighted connected undirected graph with n vertices)
T := a minimum-weight edge
for $i := 1$ **to** $n - 2$
  e := an edge of minimum weight incident to a vertex in T and not forming a simple circuit in T if added to T
  $T := T$ with e added
return T {T is a minimum spanning tree of G}

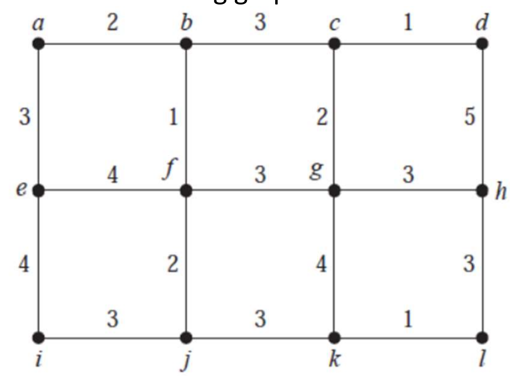
EXAMPLE: Use Prim's algorithm to design a minimum-cost communications network connecting all the computers represented by the given graph.





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

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

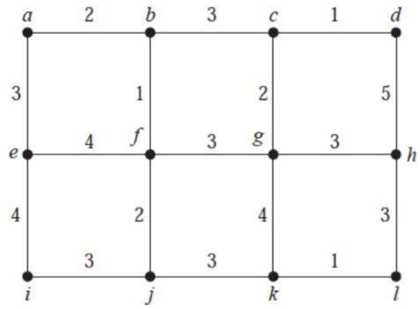


Choice	Edge	Weight
1	{b, f}	1
2	{a, b}	2
3	{f, j}	2
4	{a, e}	3
5	{i, j}	3
6	{f, g}	3
7	{c, g}	2
8	{c, d}	1
9	{g, h}	3
10	{h, l}	3
11	{k, l}	1
	Total:	24

ALGORITHM 2 Kruskal's Algorithm.

procedure *Kruskal*(G: weighted connected undirected graph with n vertices)
T := empty graph
for $i := 1$ **to** $n - 1$
  e := any edge in G with smallest weight that does not form a simple circuit when added to T
  $T := T$ with e added
return T {T is a minimum spanning tree of G}

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



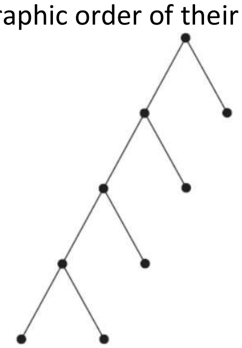
Choice	Edge	Weight
1	$\{c, d\}$	1
2	$\{k, l\}$	1
3	$\{b, f\}$	1
4	$\{c, g\}$	2
5	$\{a, b\}$	2
6	$\{f, j\}$	2
7	$\{b, c\}$	3
8	$\{j, k\}$	3
9	$\{g, h\}$	3
10	$\{i, j\}$	3
11	$\{a, e\}$	3
	Total:	24

Prim's algorithm: a procedure for producing a minimum spanning tree in a weighted graph that successively adds edges with minimal weight among all edge's incident to a vertex already in the tree so that no edge produces a simple circuit when it is added

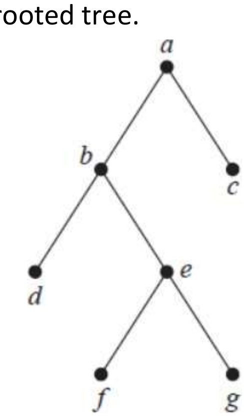
Kruskal's algorithm: a procedure for producing a minimum spanning tree in a weighted graph that successively adds edges of least weight that are not already in the tree such that no edge produces a simple circuit when it is added

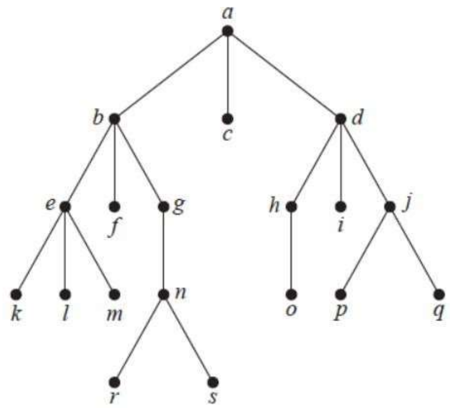
Huffman coding: a procedure for constructing an optimal binary code for a set of symbols, given the frequencies of these symbols

Exercise: Construct the universal address system for the given ordered rooted tree. Then use this to order its vertices using the lexicographic order of their labels.



Exercise: Determine the order in which a preorder, inorder, postorder traversal visits the vertices of the given ordered rooted tree.





Exercise: Suppose that the address of the vertex v in the ordered rooted tree T is 3.4.5.2.4.

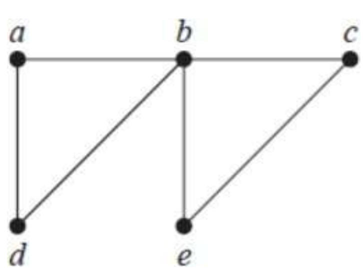
a) At what level is v?
b) What is the address of the parent of v?
c) What is the least number of siblings v can have?
d) What is the smallest possible number of vertices in Tif v has this address?
e) Find the other addresses that must occur.

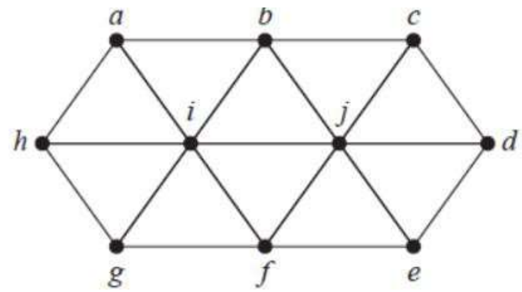
Exercise: Construct the ordered rooted tree whose preorder traversal is a, b, f, c, g, h, i, d, e, j, k, l, where a has four children, c has three children, j has two children, b and e have one child each, and all other vertices are leaves.

Exercise: What is the value of each of these postfix expressions?

a) $5\ 2\ 1 - - 3\ 1\ 4 + + *$
b) $9\ 3\ /\ 5 + 7\ 2 - *$
c) $3\ 2 * 2 \uparrow 5\ 3 - 8\ 4\ / * -$

Exercise: Find a spanning tree for the graph shown by removing edges in simple circuits.





Exercise: Find a spanning tree for each of these graphs.

a) $K5$ **b)** $K4,4$ **c)** $K1,6$ **d)** $Q3$ **e)** $C5$ **f)** $W5$

Exercises: use Prim's algorithm to find a minimum spanning tree for the given weighted graph.