Graphs

Module 5

MAT1007: Discrete Mathematics

Learning Object

- Graphs and Graph Models .
- Graph Terminology and Special Types of Graphs.
- Representing Graphs and Graph Isomorphism,
- Connectivity, Euler and Hamilton Paths,
- Shortest-Path Problems,
- Planar Graphs
- Graph Colouring.

Applications (examples)

- Distinguish between two chemical compounds with the same molecular formula but different structures
- Solve shortest path problems between cities
- Scheduling exams and assign channels to television stations
- Computer networks
- Social Networks
- Communication Networks
- Information Networks
- Computer networks
- Biological Networks

➢ *What is a graph?*

We begin by considering Figs. 1.1 and 1.2, which depict part of a road map and part of an electrical network. Either of these situations can be represented diagrammatically by means of points and lines, as in Fig. 1.3.

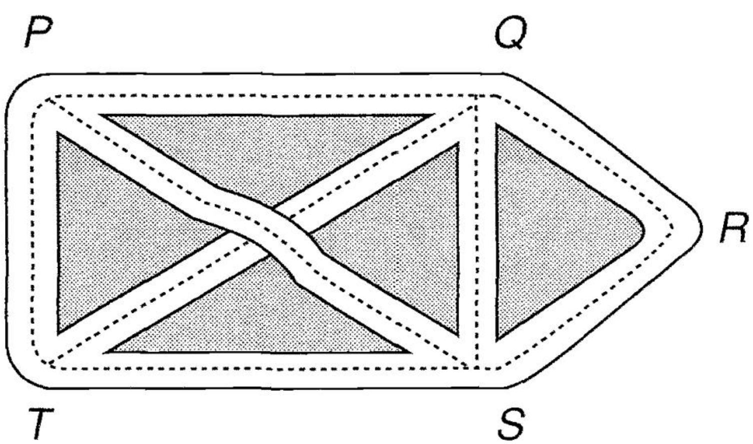


Fig. 1.1

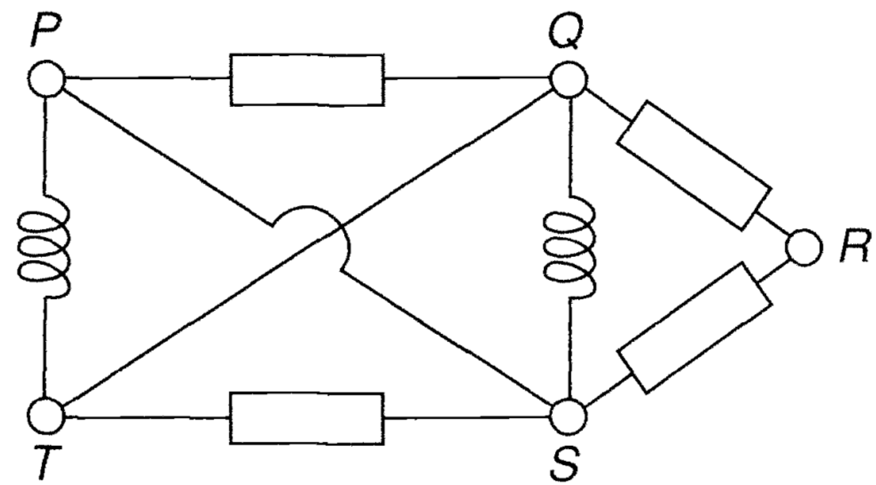


Fig. 1.2

The points P, g, R, S and T are called **vertices**, the lines are called **edges**, and the whole diagram is called a **graph**. Note that the intersection of the lines PS and QT is not a vertex, since it does not correspond to a cross-roads or to the meeting of two wires. The **degree** of a vertex is the number of edges with that vertex as an end-point; it corresponds in Fig. 1.1 to the number of roads at an intersection. For example, the degree of the vertex Q is 4.

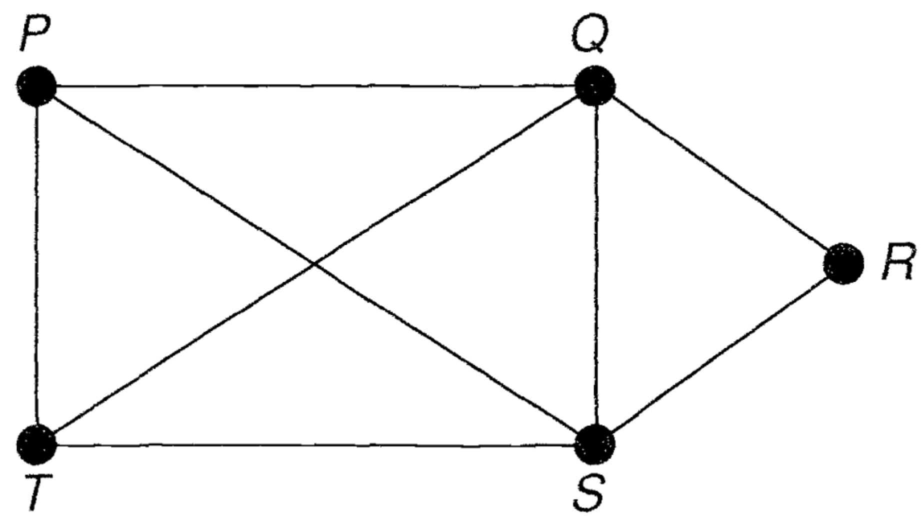


Fig. 1.3

A generalization of the simple concept of a set of dots (**vortices**), links, edges or arcs.

Representation:

Graph G =(V, E) consists set of vertices denoted by V, or by V(G) and set of edges E, or E(G)

Graphs

Definition:

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

Note:

- The set of vertices V of a graph G may be infinite.
- A graph with infinite vertex set is called infinite graph.
- A graph with finite vertex set is called finite graph.
- In mathematics, an **unordered pair** or **pair set** is a set of the form $\{a, b\}$, i.e. a set having two elements a and b with no particular relation between them. In contrast, an ordered pair (a, b) has a as its first element and b as its second element.

In other words we can define graph as

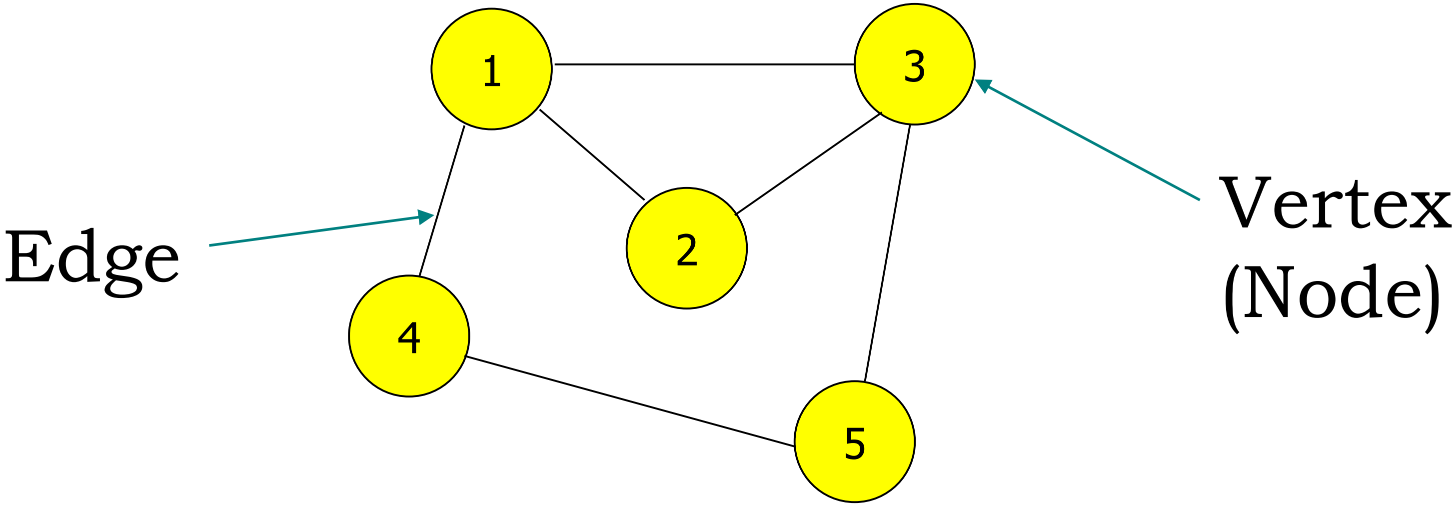
- A graph G = (V,E,ϕ) consists of V a set of vertices, E a set of edges and a function $\phi: E \rightarrow \{ \{u, v\} | \, u, v \in V\}$
- A directed graph G = (V,E,ϕ) consists of V a set of vortices, E a set of edges and a function

$$\phi: E \rightarrow \{(u, v): u, v \in V\}$$

That is if $\phi(e) = (u, v)$, the e is directed from u to v, where e-is a edge.

Examples

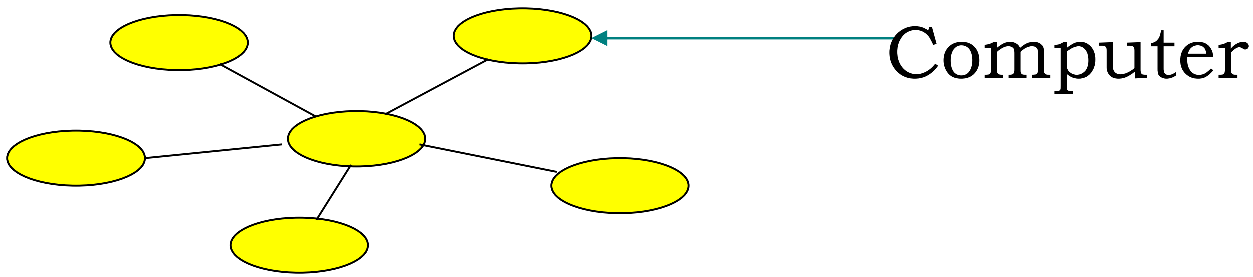
For example,



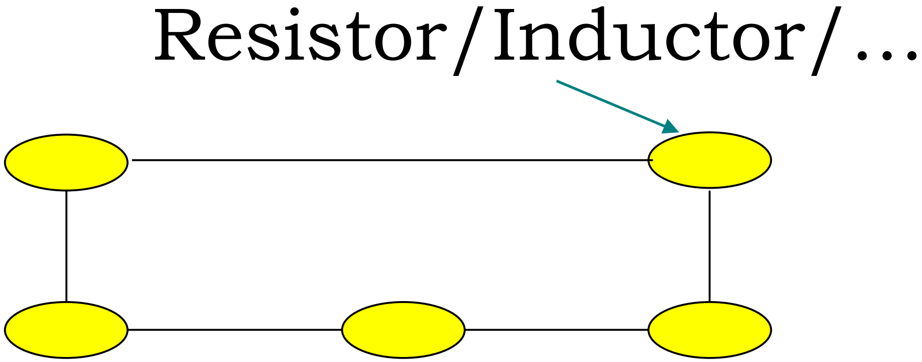
- V = {1, 2, 3, 4, 5}
- E = { (1,2), (1,3), (1,4), (2,3), (3,5), (4,5) }

Some Application of Graphs

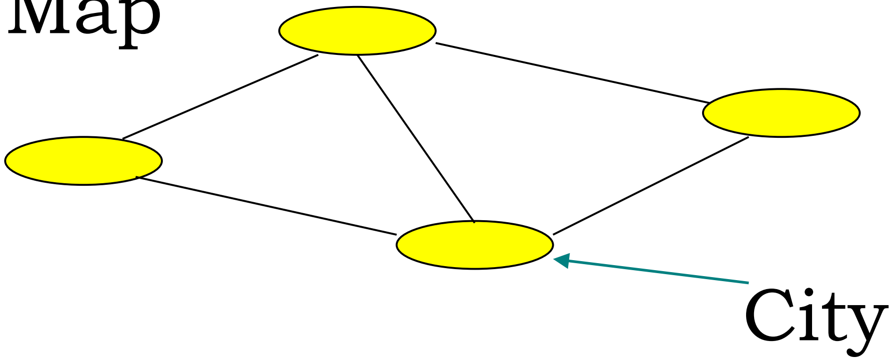
- Computer Networks



- Electrical Circuits



- Road Map

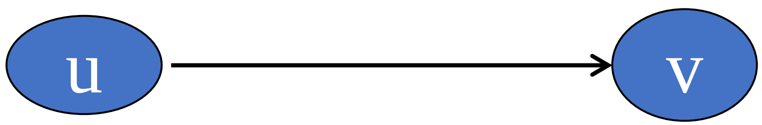


Definitions – Edge Type

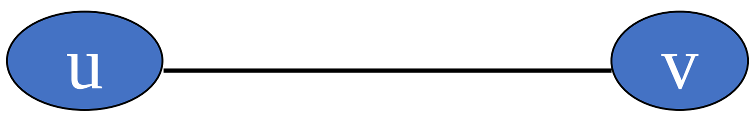
A graph having **no edges** is called a **Null Graph.**

A graph with **only one vertex** is called a **Trivial Graph.**

Directed: Ordered pair of vertices. Represented as (u, v) directed from vertex u to v.



Undirected: Unordered pair of vertices. Represented as {u, v}. Disregards any sense of direction and treats both end vertices interchangeably.



Definitions – Graph Type

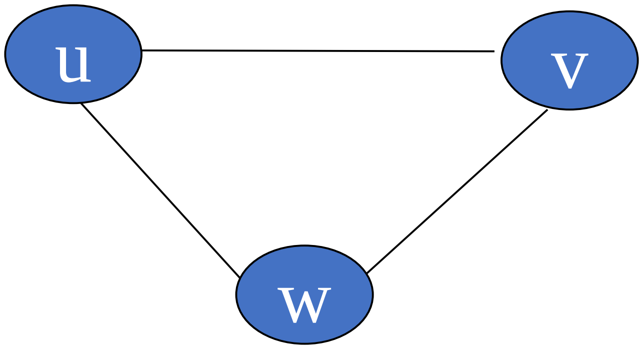
Simple (Undirected) Graph:

Consists of V, a nonempty set of vertices, and E, a set of unordered pairs of distinct elements of V called edges (undirected) **or in other words**

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**.

Representation Example: G(V, E),

V = {u, v, w}, E = {{u, v}, {v, w}, {u, w}}



We can also define the simple graph (undirected) as below

Graphs with no loops and no multiple (parallel) edges, such as the graph in Fig. 1.4, are called **simple graphs.**

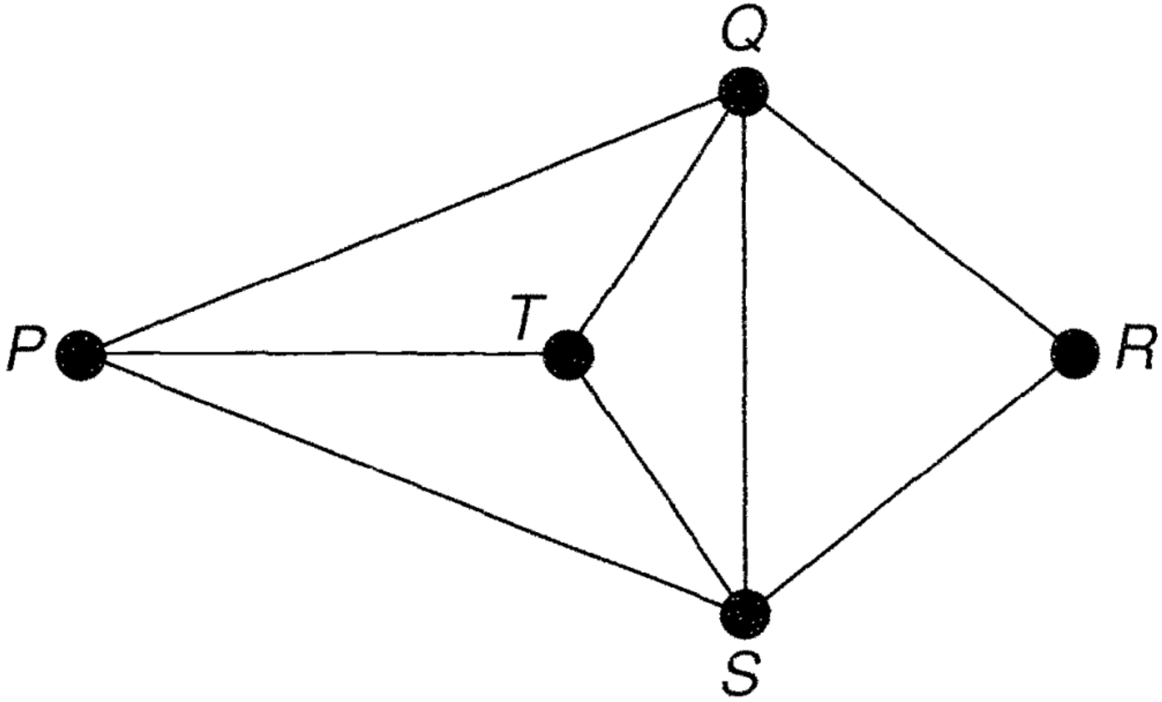
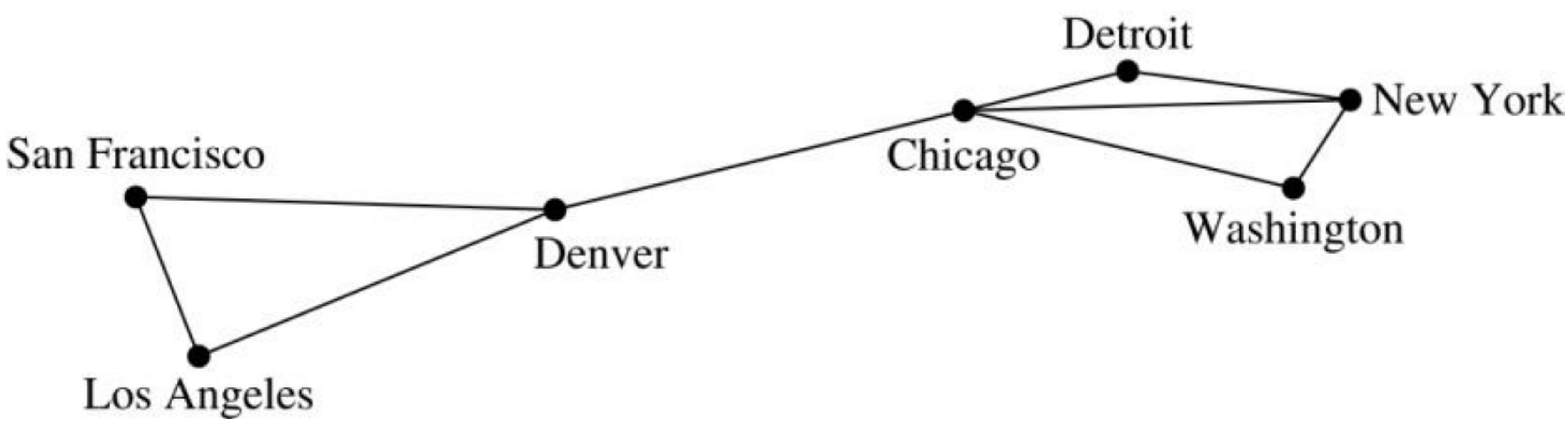


Fig. 1.4

When we draw a graph, we generally try to draw edges so that they do not cross. However, this is not necessary because any depiction using points to represent vertices and any form of connection between vertices can be used. Indeed, there are some graphs that cannot be drawn in the plane without edges crossing (see Section 10.7). The key point is that the way we draw a graph is arbitrary, as long as the correct connections between vertices are

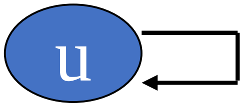


Definitions – Edge Type

- **Loop**:

A loop is an edge whose endpoints are equal i.e., an edge joining a vertex to it self is called a loop. Represented as {u, u} = {u}.

That is an edge e is called a loop if $\phi(e)$ is a singleton set.



- **Multiple Edges:**

Two or more edges joining the same pair of vertices.

Definitions – Graph Type

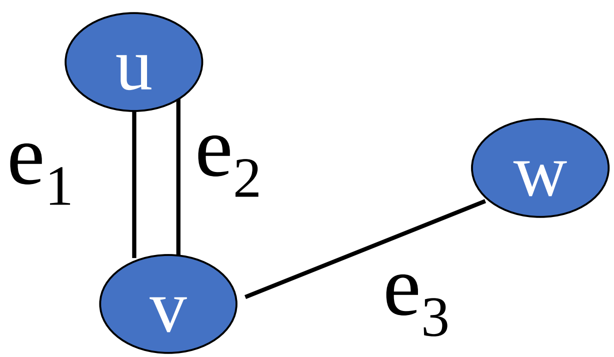
Multigraph:

G(V,E), consists of set of vertices V, set of Edges E and a function ϕ from E to $\{\{u, v\} | u, v \in V, u \neq v\}$. The edges e_1 and e_2 are called multiple or parallel edges if $\phi(e_1) = \phi(e_2)$.

Note: The edges e_i and e_j are called ***a multiple*** or ***parallel edges*** if they are incident to the same pair of vertices.

Representation Example:

V = {u, v, w}, E = {e_1, e_2, e_3}



Graph Models: Computer Networks (*continued*)

A computer network may contain multiple links between data centers, as shown in Figure 2. To model such networks we need graphs that have more than one edge connecting the same pair of vertices. Graphs that may have **multiple edges** connecting the same vertices are called **multigraphs**. When there are m different edges associated to the same unordered pair of vertices $\{u, v\}$, we also say that $\{u, v\}$ is an edge of multiplicity m. That is, we can think of this set of edges as m different copies of an edge $\{u, v\}$.

.

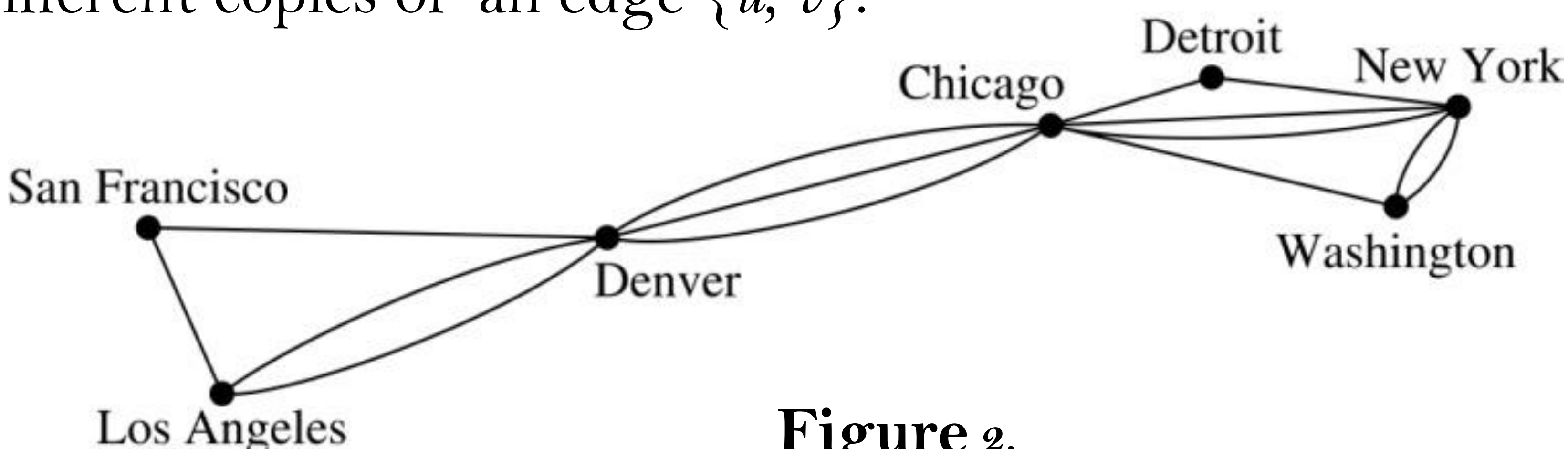


Figure 2.

Definitions – Graph Type

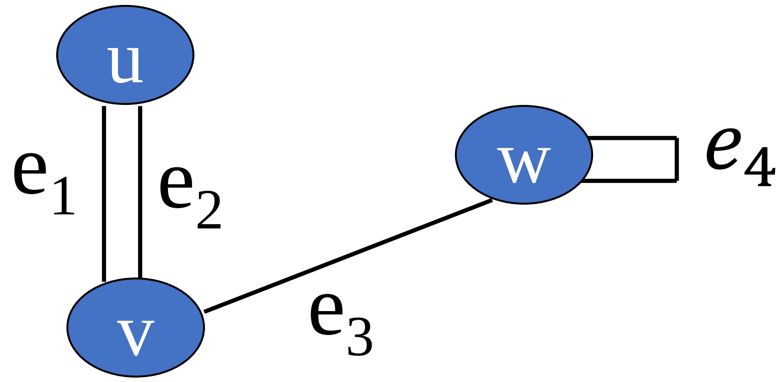
Pseudograph:

G(V,E), consists of set of vertices V, set of Edges E and a function F from E to $\{\{u, v\} | u, v \in V\}$. Loops allowed in such a graph.

Representation Example: V = {u, v, w}, E = {e_1, e_2, e_3, e_4}

Or in other words

A pseudograph is a non-simple graph in which both graph loops and multiple edges are permitted



Sometimes a communications link connects **a data center with itself, perhaps a feedback loop for diagnostic purposes.** Such a network is illustrated in Figure 3. To model this network we need to include edges that connect a vertex to itself. Such edges are called **loops**, and sometimes we may even have more than one loop at a vertex. Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are sometimes called **pseudographs**

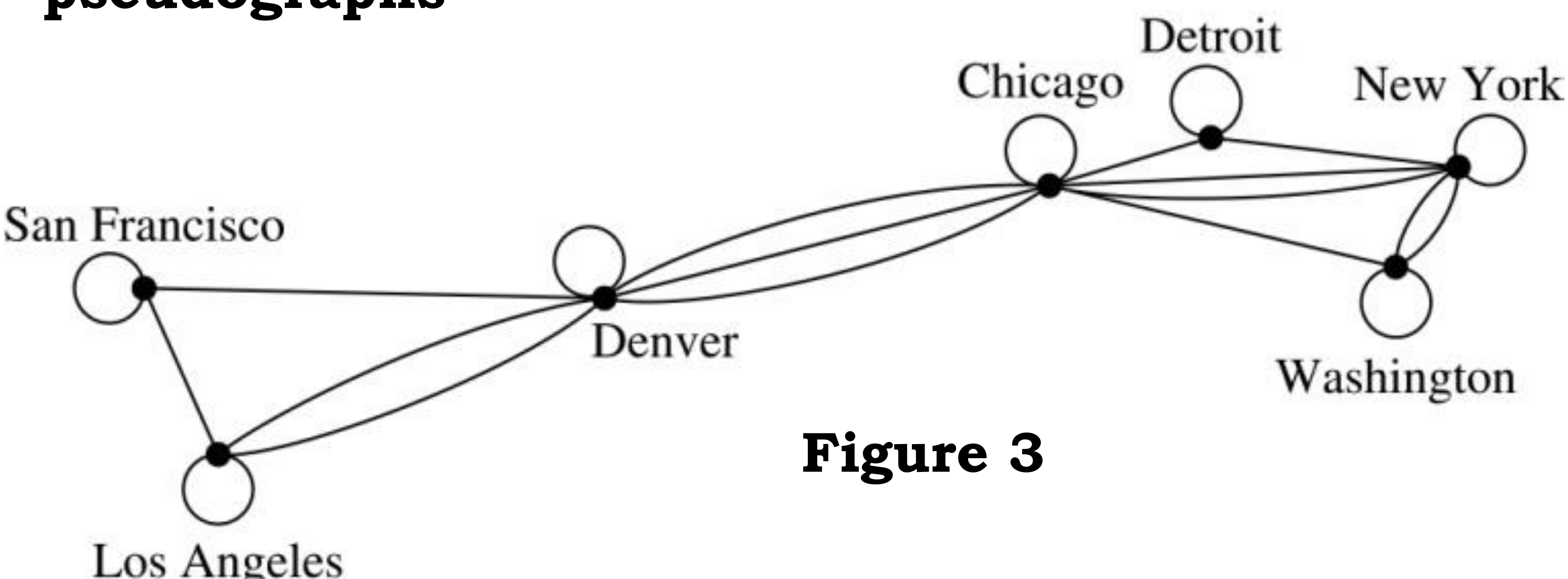


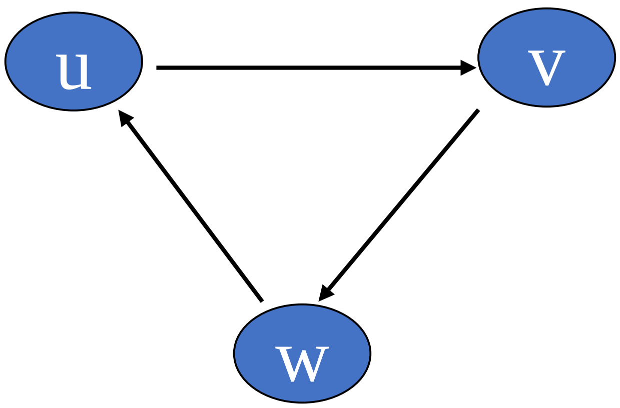
Figure 3

Definitions – Graph Type

Directed Graph: G(V, E), set of vertices V, and set of Edges E, that are ordered pair of elements of V (directed edges or **each edge has a direction**). The directed edge associated with the ordered pair (u, v) is said to *start* at u and *end* at v.

Representation Example:

G(V, E), V = {u, v, w}, E = {(u, v), (v, w), (w, u)}



So far the graphs we have introduced are **undirected graphs**. Their edges are also said to be **undirected**. However, to construct a graph model, we may find it necessary to assign directions to the edges of a graph. For example, in a computer network, some links may operate in only one direction (such links are called single duplex lines). This may be the case if there is a large amount of traffic sent to some data centers, with little or no traffic going in the opposite direction. Such a network is shown in Figure 4.

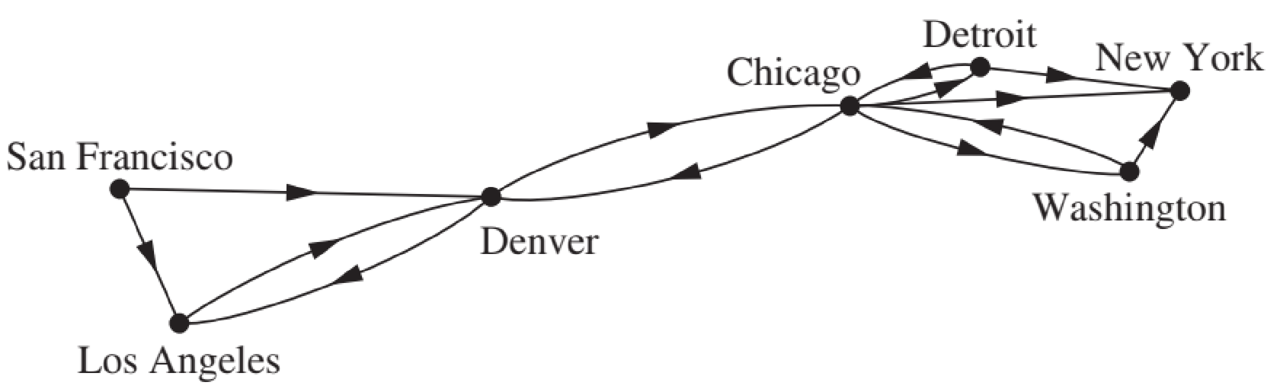


FIGURE 4 **A Communications Network with One-Way Communications Links.**

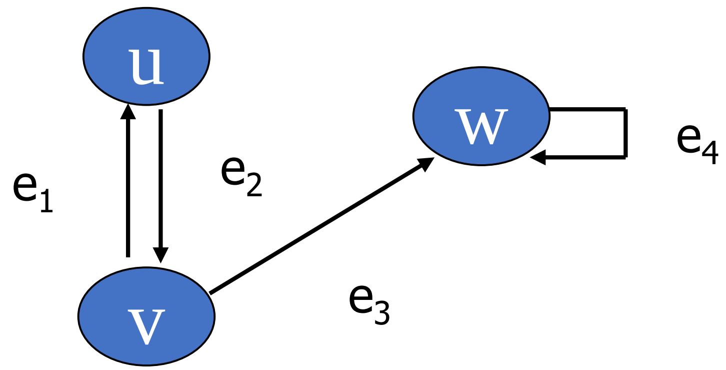
Definitions – Graph Type

Directed Multigraph:

G(V,E), consists of set of vertices V, set of Edges E and a function f from E to $\{\{u, v\} \mid u, v \in V\}$. The edges e_1 and e_2 are multiple edges if $f(e_1) = f(e_2)$.

Representation Example:

V = {u, v, w}, E = {e_1, e_2, e_3, e_4}



When a directed graph has no loops and has no multiple directed edges, it is called a **simple directed graph**. Because a simple directed graph has at most one edge associated to each ordered pair of vertices (u, v), we call (u, v) an edge if there is an edge associated to it in the graph.

In some computer networks, multiple communication links between two data centers may be present, as illustrated in Figure 5. Directed graphs that may have **multiple directed edges** from a vertex to a second (possibly the same) vertex are used to model such networks. We called such graphs **directed multigraphs**. When there are m directed edges, each associated to an ordered pair of vertices (u, v), we say that (u, v) is an edge of **multiplicity** m.

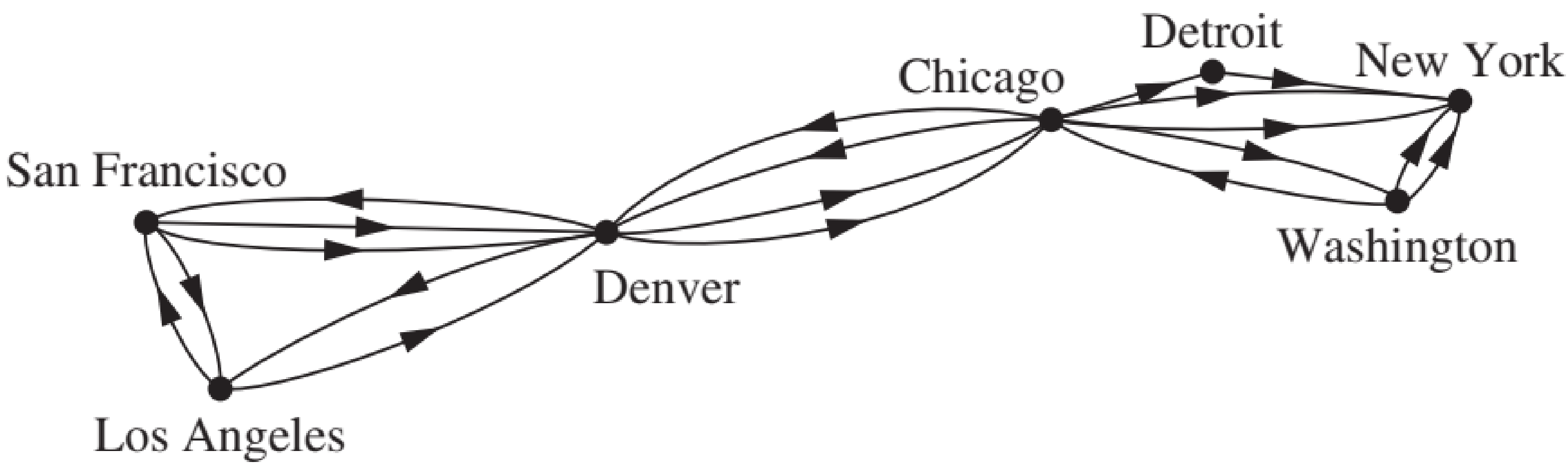


FIGURE 5 **A Computer Network with Multiple One-Way Links.**

For some models we may need a graph where some edges are undirected, while others are directed. A graph with both directed and undirected edges is called a **mixed graph**. For example, a mixed graph might be used to model a computer network containing links that operate in both directions and other links that operate only in one direction.

This terminology for the various types of graphs is summarized in Table 1. We will sometimes use the term **graph** as a general term to describe graphs with directed or undirected edges (or both), with or without loops, and with or without multiple edges. At other times, when the context is clear, we will use the term graph to refer only to undirected graphs.

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

Graph Models: Computer Networks

Now suppose that a network is made up of data centers and communication links between computers. We can represent the location of each data center by a point and each communications link by a line segment, as shown in Figure 1.

This computer network can be modeled using a graph in which the vertices of the graph represent the data centers and the edges represent communication links. In general, we visualize graphs by using points to represent vertices and line segments, possibly curved, to represent edges, where the endpoints of a line segment representing an edge are the points representing the endpoints of the edge. depicted.

Graph Terminology: Summary

- To understand the structure of a graph and to build a graph model, we ask three key questions :
- Are the edges of the graph undirected or directed (or both)?
- If the edges are undirected, are multiple edges present that connect the same pair of vertices? If the edges are directed, are multiple directed edges present?
- Are loops present?

Other Applications of Graphs

- We will illustrate how graph theory can be used in models of:
  - Social networks
  - Communications networks
  - Information networks
  - Software design
  - Transportation networks
  - Biological networks
- It’s a challenge to find a subject to which graph theory has not yet been applied. Can you find an area without applications of graph theory?

Graph Models: Social Networks

- Graphs can be used to model social structures based on different kinds of relationships between people or groups.
- In a *social network*, vertices represent individuals or organizations and edges represent relationships between them.
- Useful graph models of social networks include:
  - *friendship graphs* - undirected graphs where two people are connected if they are friends (in the real world, on Facebook, or in a particular virtual world, and so on.)
  - *collaboration graphs* - undirected graphs where two people are connected if they collaborate in a specific way
  - *influence graphs* - directed graphs where there is an edge from one person to another if the first person can influence the second person

Graph Models: Social Networks (*continued*)

Acquaintanceship and Friendship Graphs We can use a simple graph to represent whether two people know each other, that is, whether they are acquainted, or whether they are friends (either in the real world in the virtual world via a social networking site such as Facebook).
Each person in a particular group of people is represented by a vertex. An undirected edge is used to connect two people when these people know each other, when we are concerned only with acquaintanceship, or whether they are friends. No multiple edges and usually no loops are used. (If we want to include the notion of self-knowledge, we would include loops.) A small acquaintanceship graph is shown in Figure 6.

Graph Models: Social Networks (*continued*)

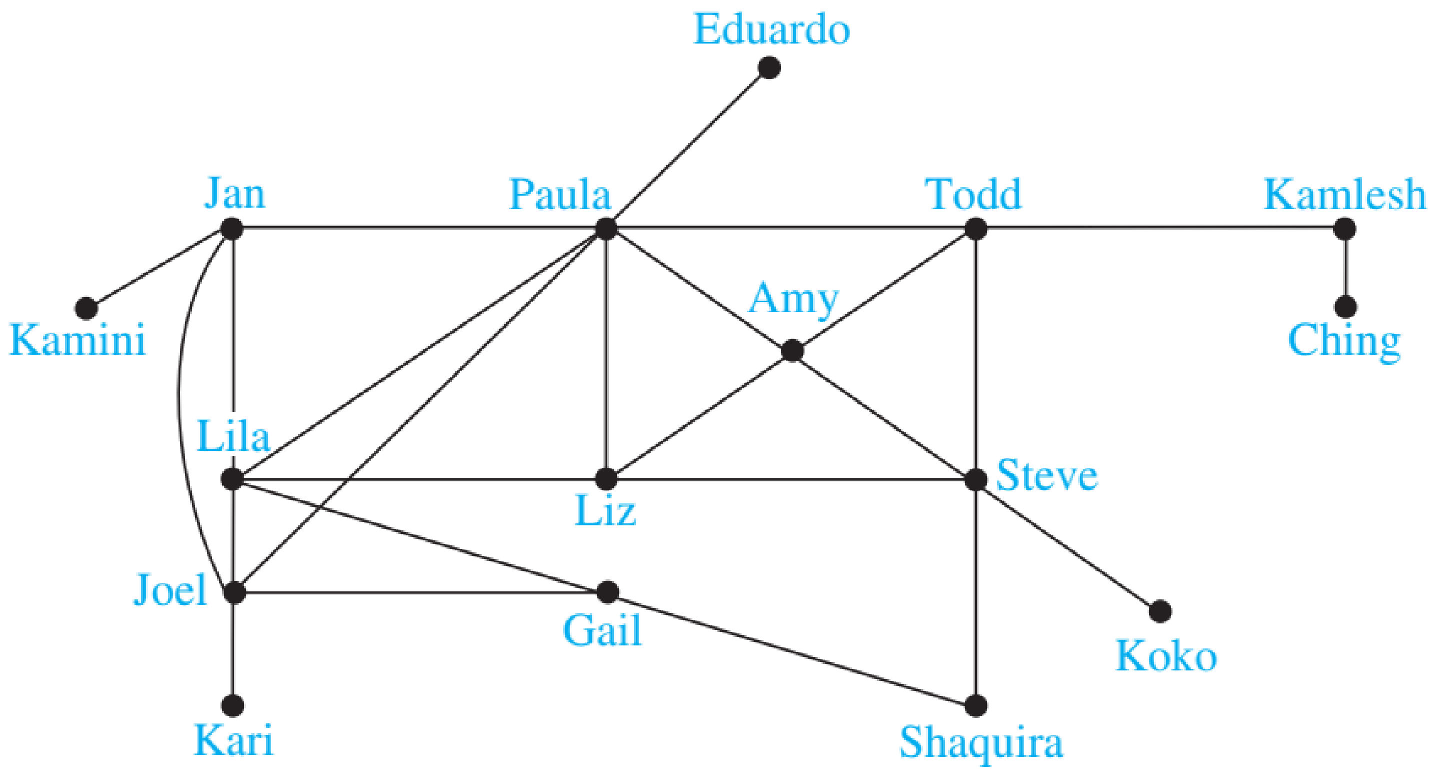


FIGURE 6 **An Acquaintanceship Graph.**

Example: A friendship graph where two people are connected if they are Facebook friends.

Influence Graphs In studies of group behavior it is observed that certain people can influence the thinking of others. A directed graph called an **influence graph** can be used to model this behavior. Each person of the group is represented by a vertex. There is a directed edge from vertex a to vertex b when the person represented by vertex a can influence the person represented by vertex b. This graph does not contain loops and it does not contain multiple directed edges.

An example of an influence graph for members of a group is shown in Figure 7. In the group modeled by this influence graph, Deborah cannot be influenced, but she can influence Brian, Fred, and Linda. Also, Yvonne and Brian can influence each other.

Graph Models: Social Networks (*continued*)

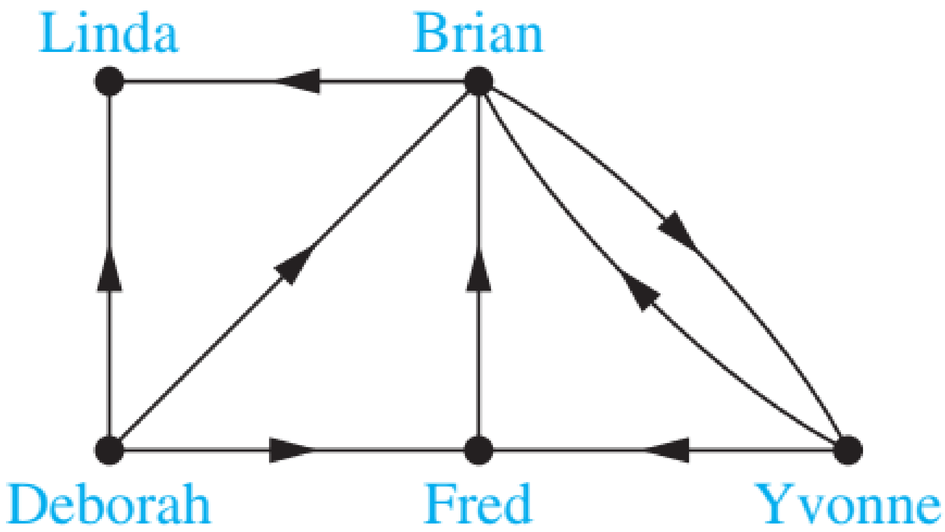


FIGURE 7 **An Influence Graph.**

For the examples 03-14, I suggest you to refer Textbook.