Ceng 124
Discrete Structures

2018-2019 Spring Semester

by Sibel T. Özyer, Spring 2019

Topics

- 10.1 Graphs and Graph Models
- 10.2 Graph Terminology and Special Types of Graphs

Definition of Graph

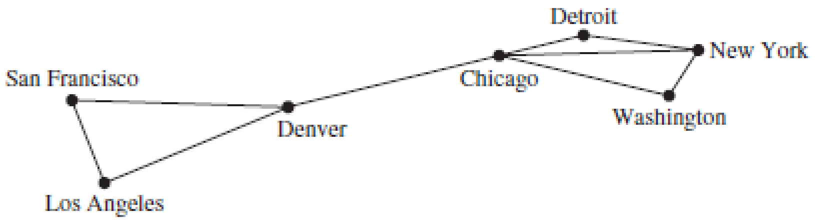
A *graph* $G = (V, E)$ consists of V, a nonempty set of *vertices* (or *nodes*) and E, a set of *edges*. Each edge has either one or two vertices associated with it, called its *endpoints*. An edge is said to *connect* its endpoints.

Categories of Graphs

- There are 5 main categories of graphs:
  - Simple graph
  - Multigraph
  - Pseudograph
  - Directed graph
  - Directed multigraph

Simple Graph

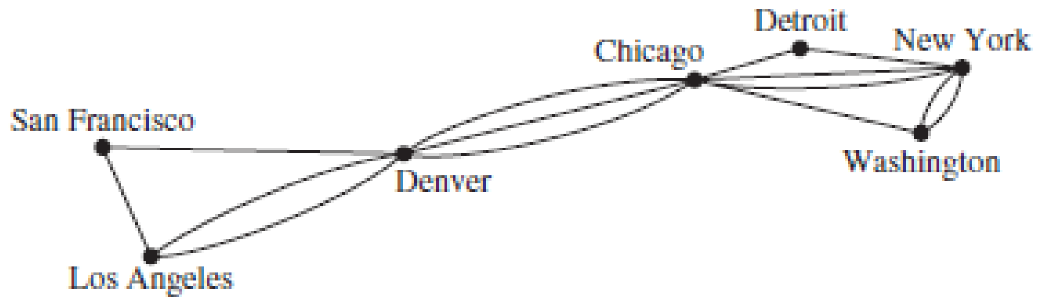
Suppose that a network is made up of data centers and communication links between computers



A computer network with multiple lines

Multigraphs

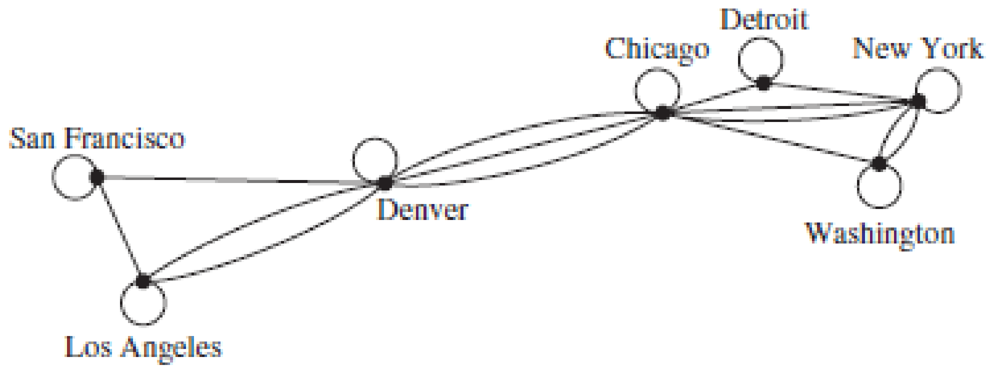
- A computer network may contain multiple links between data centers, as shown in Figure.



Computer Network with Multiple Links between Data Centers.

Pseudograph

Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are sometimes called **pseudographs.**



A Computer Network with Diagnostic Links.

Graph Types

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

- Social Networks
- Communication Networks
- Information Networks
- Software Design Applications
- Transportation Networks
- Biological Networks

10.2 Graph Terminology and Special Types of Graphs

- We introduce some of the basic vocabulary of graph theory in this section.

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.

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 $\deg(v)$.

Example

- What are the degrees and what are the neighborhoods of the vertices in the graphs *G* and *H* displayed in Figure 1?

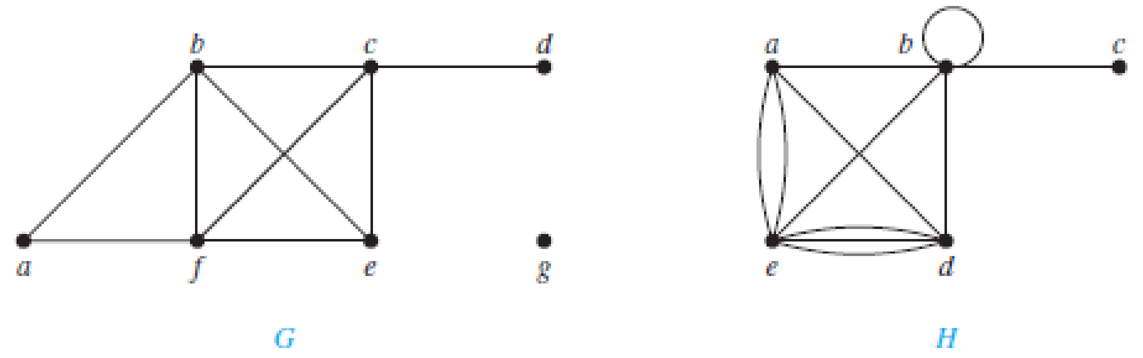


Figure 1: The Undirected Graphs *G* and *H*.

Solution

- In G, 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 $N(g) = \emptyset$.
- In H, 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 $N(e) = \{a, b, d\}$.

In-degree and Out-degree

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

- Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in Figure 2.

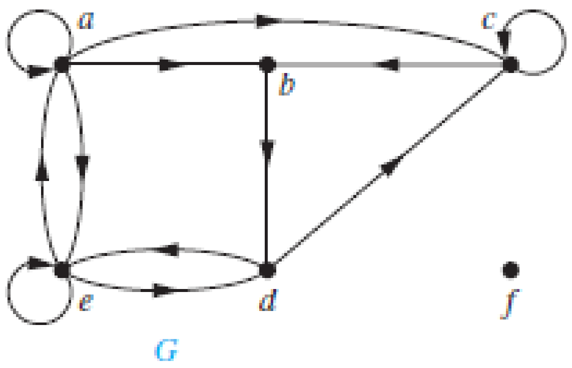


Figure 2: The Directed Graph G.

Solution

- The in-degrees in G 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 $\deg^+(f) = 0$.

Sum of the out degrees and in degrees

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

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

Some Special Simple Graphs

- **Complete Graphs** A **complete graph on *n* vertices**, denoted by *Kn*, is a simple graph that contains exactly one edge between each pair of distinct vertices. The graphs *Kn*, for *n* = 1, 2, 3, 4, 5, 6, are displayed in Figure 3.

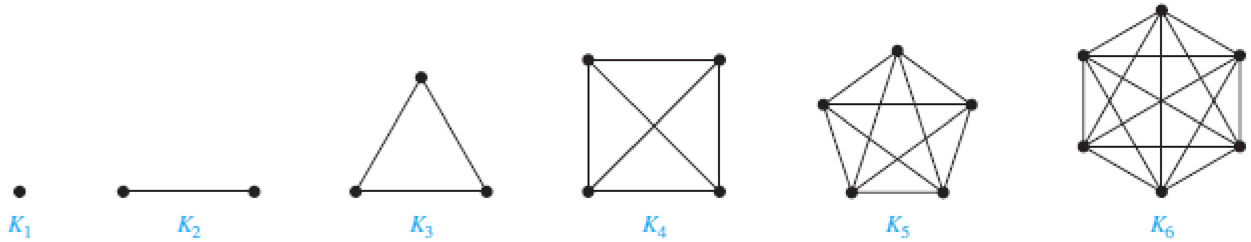


Figure 3: The Graphs *Kn* for $1 \le n \le 6$.

Some Special Simple Graphs (cont.)

- **Cycles** A **cycle** C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$. The cycles C_3, C_4, C_5, and C_6 are displayed in Figure 4.

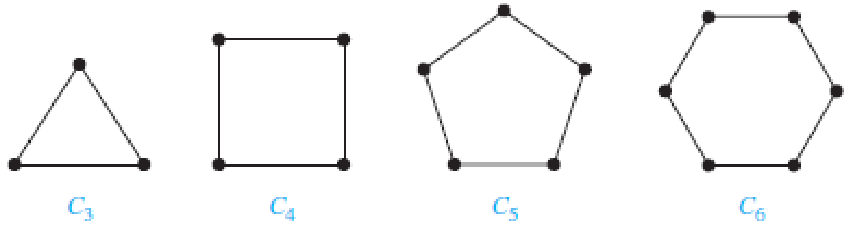


Figure 4: The Cycles C_3, C_4, C_5, and C_6.

Some Special Simple Graphs (cont.)

- **Wheels** We obtain a **wheel *Wn*** when we add an additional vertex to a cycle *Cn*, for $n \geq 3$, and connect this new vertex to each of the *n* vertices in *Cn*, by new edges. The wheels *W*3, *W*4, *W*5, and *W*6 are displayed in Figure 5.

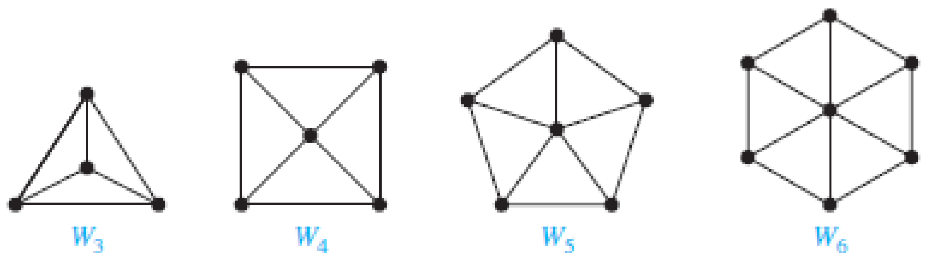


Figure 5: The Wheels *C*3, *C*4, *C*5, and *C*6.

Some Special Simple Graphs (cont.)

- *n*-Cubes An ***n*-dimensional hypercube**, or ***n*-cube**, denoted by***Qn***, is a graph that has vertices representing the 2*n* bit strings of length *n*. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position. We display *Q*1, *Q*2, and *Q*3 in Figure 6.

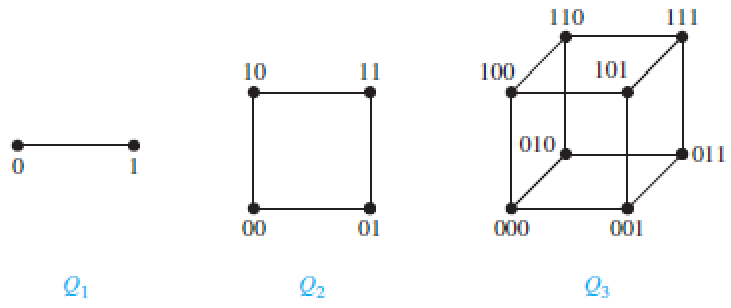


Figure 6: The *n*-cube *Qn*, *n* = 1, 2, 3.

Bipartite Graphs

- Sometimes a graph has the property that its vertex set can be divided into two disjoint subsets such that each edge connects a vertex in one of these subsets to a vertex in the other subset.
- For example, consider the graph representing marriages between men and women in a village, where each person is represented by a vertex and a marriage is represented by an edge.

Definition

A simple graph G is called *bipartite* if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2). When this condition holds, we call the pair (V_1, V_2) a *bipartition* of the vertex set V of G.

Examples

- $C6$ is bipartite, as shown in Figure 7, 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 $V2$.

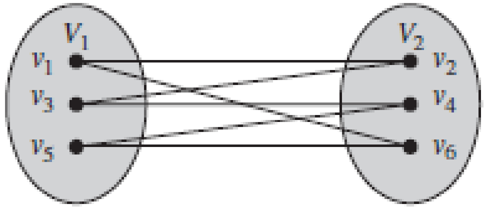


Figure 7: Showing That *C6* is Bipartite.

- Are the graphs *G* and *H* displayed in Figure 8 bipartite?

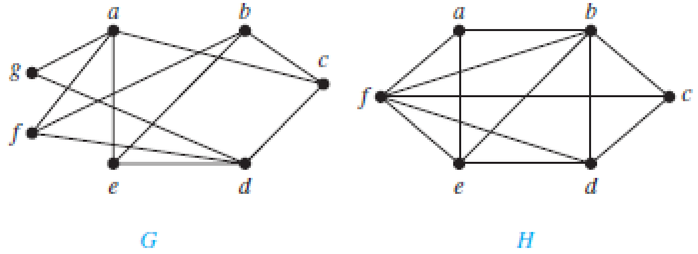


Figure 8: The Undirected Graphs *G* and *H*.

Solution

- Graph G is bipartite because its vertex set is the union of two disjoint sets, $\{a, b, d\}$ and $\{c, e, f, g\}$, and each edge connects a vertex in one of these subsets to a vertex in the other subset. (Note that for G to be bipartite it is not necessary that every vertex in $\{a, b, d\}$ be adjacent to every vertex in $\{c, e, f, g\}$. For instance, b and g are not adjacent)
- Graph H is not bipartite because its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset. (The reader should verify this by considering the vertices a, b, and f)

Theorem

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

Complete Bipartite Graphs

- A **complete bipartite graph** $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.
- The complete bipartite graphs $K_{2,3}$, $K_{3,3}$, $K_{3,5}$, and $K_{2,6}$ are displayed in Figure 9.

Some Complete Bipartite Graphs

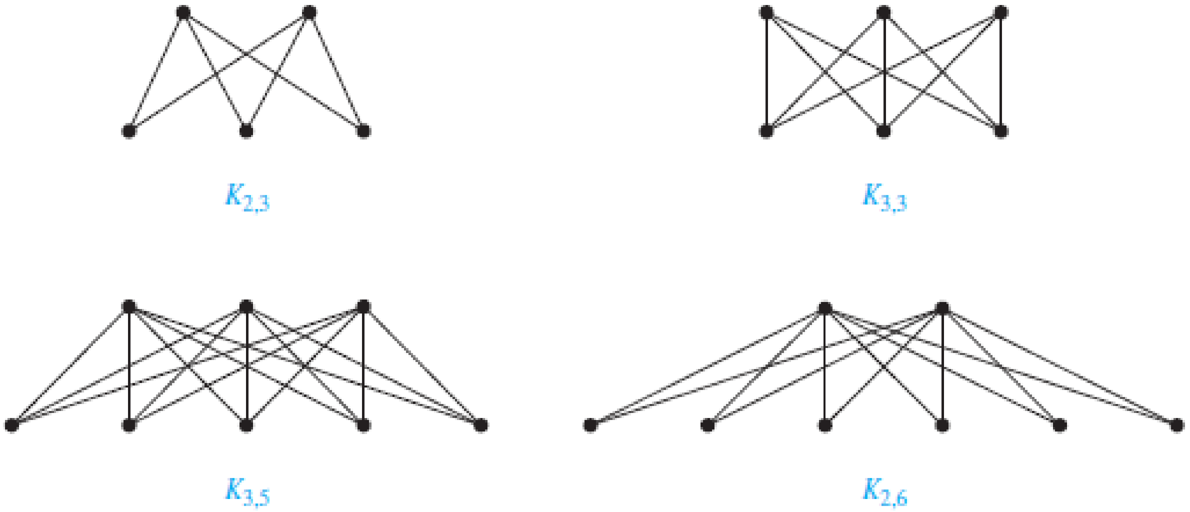


Figure 9: Some Complete Bipartite Graphs.

Bipartite Graphs and Matchings

- Bipartite graphs can be used to model many types of applications that involve matching the elements of one set to elements of another, as Example illustrates.

Modelling Job Assignments

These graphs are bipartite, where the bipartition is *(E, J)* where *E* is the set of employees and *J* is the set of jobs.

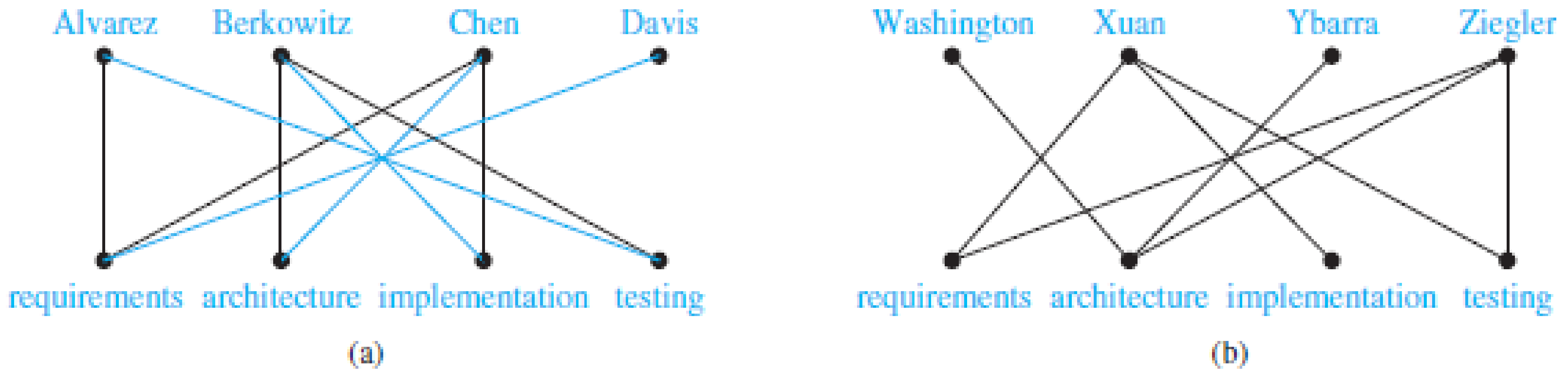


Figure 10: Modeling the Jobs for Which Employees Have Been Trained.

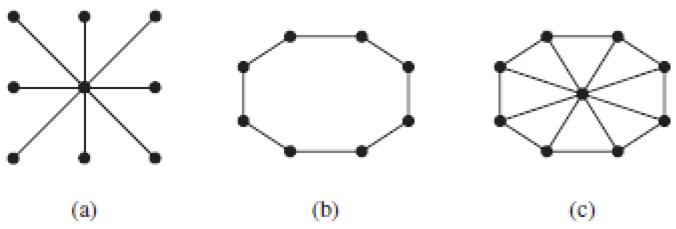
Suppose that Alvarez has been trained to do requirements and testing - Figure 10 (a)
Washington has been trained to do architecture - Figure 10 (b)

Matching

- Assignment of jobs to employees can be thought of as finding a matching in the graph model, where a **matching** *M* in a simple graph $G = (V, E)$ is a subset of the set *E* of edges of the graph such that no two edges are incident with the same vertex.
- In other words, a matching is a subset of edges such that if $\{s, t\}$ and $\{u, v\}$ are distinct edges of the matching, then *s*, *t* , *u*, and *v* are distinct. A vertex that is the endpoint of an edge of a **matching** *M* is said to be **matched** in *M*; otherwise it is said to be **unmatched.**
- Example: Marriages on an Island. A bipartite graph $G = (V1, V2)$ where *V*1 is the set of men and *V*2 is the set of women. A matching in this graph consists of a set of edges which are husband-wife pairs.

Some Applications of Special Types of Graphs

Local Area Networks



Figuere 11: Star, Ring, and Hybrid Topologies for Local Area Networks.

Some Applications of Special Types of Graphs (cont.)

- Interconnection Networks for Parallel Computation
- Parallel Processing:

Figurte 12: A Linear Array for Six Processors.

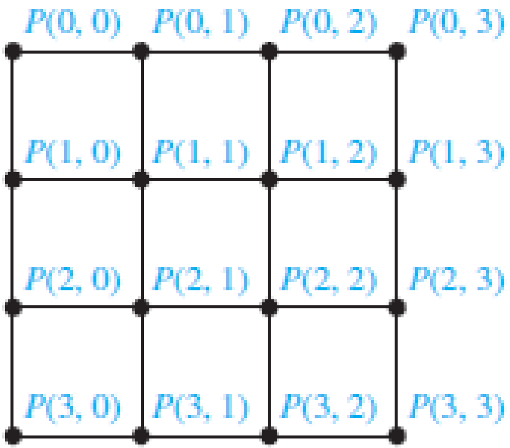


Figure 13: A Mesh Network for 16 Processors.