Chapter 4 Graphs and Trees

4.1 Introduction

4.2 What is a Graph

4.3 Definitions
- 4.3.1 The Definition of a Graph
- 4.3.2 Connected and Disconnected Graphs
- 4.3.3 Complete graph

4.4 The Handshaking Lemma

4.5 Matrix Representations of Graphs
- 4.5.1 Matrices and Undirected Graphs
- 4.5.2 Matrices and Directed Graphs

4.6 Applications of Graphs
- 4.6.1 Social Sciences
- 4.6.2 Chemistry
- 4.6.3 Circulation diagram

4.7 Paths and Circuits
- 4.7.1 Definitions
- 4.7.2 Theorem 4.1 (Counting Walks of Length N)
- 4.7.3 The Shortest Path Algorithm

4.8 Eulerian Graphs
- 4.8.1 Definition
- 4.8.2 Theorem 4.2
- 4.8.3 Königsberg bridges problem

4.9 Hamiltonian Graphs
- 4.9.1 Definition
- 4.9.2 Theorem 4.3 (DIRAC'S THEOREM)
- 4.9.3 Theorem 4.4 (ORE'S THEOREM)

4.10 Trees
- 4.10.1 Definition
- 4.10.2 Properties of Tree
- 4.10.3 Alternative definitions of the tree
- 4.10.4 Spanning Trees

4.11 Minimum Spanning Tree
- 4.11.1 Definition
- 4.11.2 Greedy Algorithm

4.12 References

4.1 Introduction

Since many situations and structures give rise to graphs, graph theory becomes an important mathematical tool in a wide variety of subjects, ranging from operations research, computing science and linguistics to chemistry and genetics.

Recently, there has been considerable interest in tree structures arising in the computer science and artificial intelligence. We often organize data in a computer memory store or the flow of information through a system in tree structure form. Indeed, many computer operating systems are designed to be tree structures.

4.2 What is a Graph

Let us consider Figures 4.1 and 4.2 which depict, respectively, part of an electrical network and part of a road map.

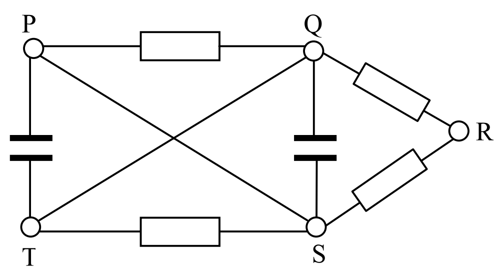


Figure 4.1

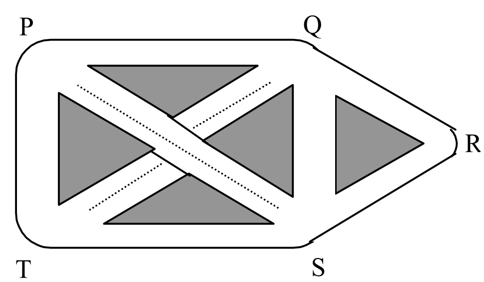


Figure 4.2

It is clear that either of them can be represented diagrammatically by means of points and lines in Figure 4.3. The points P, Q, R, S and T are called **vertices** and the lines are called **edges**; the whole diagram is called a **graph**.

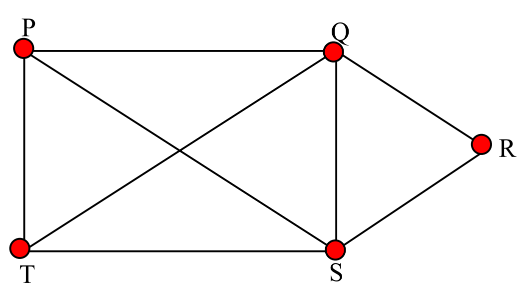


Figure 4.3

4.3 Definitions

4.3.1 The Definition of a Graph

An **undirected graph** G consists of a non-empty set of elements, called **vertices**, and a set of **edges**, where each edge is associated with a list of **unordered** pairs of either one or two vertices called its **endpoints**. The correspondence from edges to end-points is called the **edge-endpoint function**. The set of vertices of the graph G is called the **vertex-set** of G, denoted by **V(G)**, and the set of edges is called the **edge-set** of G, denoted by **E(G).**

Example 4.1

Figure 4.4 represents the simple undirected graph G whose

vertex-set V(G) = {u, v, w, z}

edge-set E(G) = $\{e_1, e_2, e_3, e_4\}$

edge-endpoint function

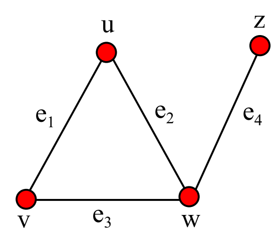


Figure 4.4

Edge	e_1	e_2	e_3	e_4
Endpoints	{u, v}	{u, w}	{v, w}	{w, z}

and G = {V(G), E(G)}
= $\{\{u, v, w, z\}, \{e_1, e_2, e_3, e_4\}\}$

A **directed graph** G consists of vertices, and a set of edges, where each edge is associated with a list of **ordered** pair endpoints.

Example 4.2

Figure 4.5 represents the directed graph G whose

vertex-set V(G) = {u, v, w, z}

edge-set E(G) = $\{e_1, e_2, e_3, e_4\}$

and edge-endpoint function;

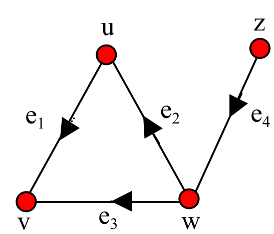


Figure 4.5

Edge	e_1	e_2	e_3	e_4
Endpoints	(u, v)	(w, u)	(w, v)	(z, w)

Two or more edges joining the same pair of vertices are called **multiple edges** and an edge joining a vertex to itself is called a **loop**. A graph with no loops or multiple edges is called a **simple graph**.

The **degree** of a vertex is the number of edges meeting at a given vertex, and is denoted by **deg** v. Each loop contributes 2 to the degree of the corresponding vertex. The **total degree** of G is the sum of the degrees of all the vertices of G.

Example 4.3
Figure 4.6 illustrates these definitions.

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

E(G) = $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

deg u = 6, deg v = 5, deg w = 2, deg z = 1

total degree = 6 + 5 + 2 + 1 = 14

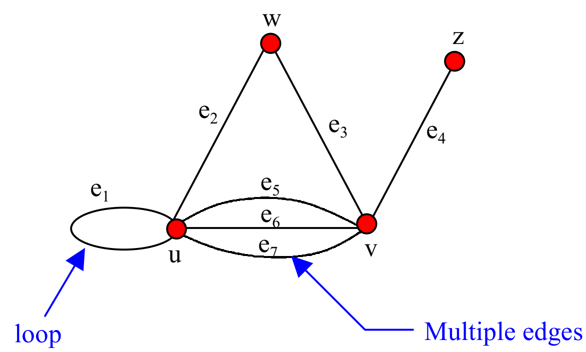


Figure 4.6

edge-endpoint function:

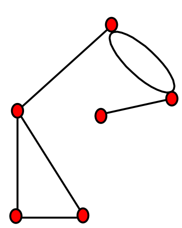
Edge	e_1	e_2	e_3	e_4	e_5	e_6	e_7
Edgepoints	{u, u}	{u, w}	{v, w}	{v, z}	{u, v}	{u, v}	{u, v}

Two vertices v and w of a graph G are said to be **adjacent** if there is an edge joining them; the vertices v and w are then said to be **incident** to such an edge. Similarly, two distinct edges of G are **adjacent** if they have at least one vertex in common.

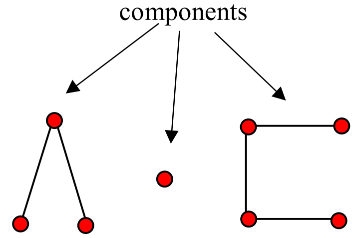
4.3.2 Connected and Disconnected Graphs

A graph G is **connected** if there is a path in G between any given pair of vertices, and **disconnected** otherwise. Every disconnected graph can be split up into a number of connected subgraphs, called **components**.

Example 4.4



Connected Graph



Disconnected Graph

Figure 4.7

4.3.3 Complete graph

A complete graph on n vertices, denoted K_n, is a simple graph with n vertices v_1, v_2, ……v_n whose set of edges contains exactly one edge for each pair of distinct vertices.

Example 4.5

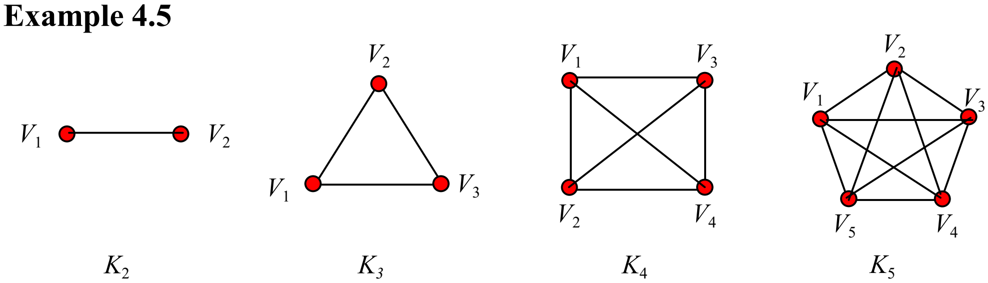


Figure 4.8

Furthermore, the graph K_n is regular of degree n – 1, and has $\frac{1}{2}n(n-1)$ edges.

4.4 The Handshaking Lemma

In any graph, the sum of all the vertex-degrees is equal to twice the number of edges.

As the consequences of the Handshaking Lemma, the sum of all the vertex-degrees is an **even** number and the number of vertices of **odd** degree is **even**.

In Figure 4.6,

the number of edges $= 7$
the sum of all the vertex-degrees $= 6 + 5 + 2 + 1 = 14$ (even)
$= 2 \times 7$
= twice the number of edges

the number of odd degree = 2 (even)

4.5 Matrix Representations of Graphs

A given graph can be specified and stored in matrix format. The adjacency matrix and the incidence matrix are often used in practice.

4.5.1 Matrices and Undirected Graphs

The **Adjacency Matrix A(G)** is the n × n matrix in which the entry in row i and column j is the number of edges joining the vertices i and j.

Example 4.6

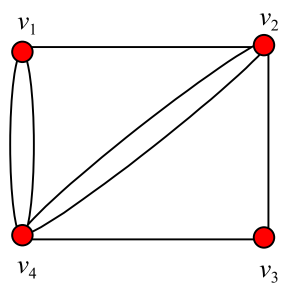


Figure 4.9

$$A(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix} \end{array}$$

The **Incidence Matrix I(G)** is the n × m matrix in which the entry in row i and column j is 1 if vertex i is incident with edges j, and 0 otherwise.

Example 4.7

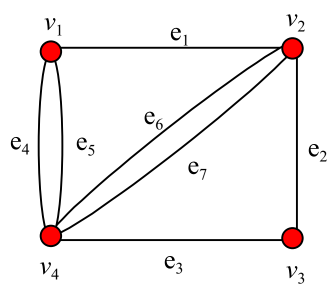


Figure 4.10

$$I(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

4.5.2 Matrices and Directed Graphs

Example 4.8

$$A(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

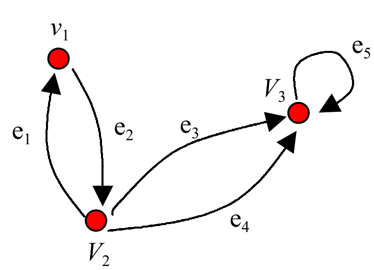


Figure 4.11

$$I(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} -1 & +1 & 0 & 0 & 0 \\ +1 & -1 & +1 & +1 & 0 \\ 0 & 0 & -1 & -1 & 2 \end{pmatrix} \end{array}$$

4.6 Applications of Graphs

4.6.1 Social Sciences

Graphs have been used extensively to represent interpersonal (international) relationships. The vertices correspond to individuals in a group (nations), and the edges join pairs of individuals who are related in some way; such as likes, hates, agrees with, avoids, communicates, etc.(allied, maintain diplomatic relations, agree on a particular strategy, etc)

Consider the following **signed** graph (Figure 4.12), which shows the working relationships with four employees. The graph with either + or – associated with each edge, indicating a positive relationship (likes) or a negative one (dislikes). There are no strong feelings about each other for no connection.

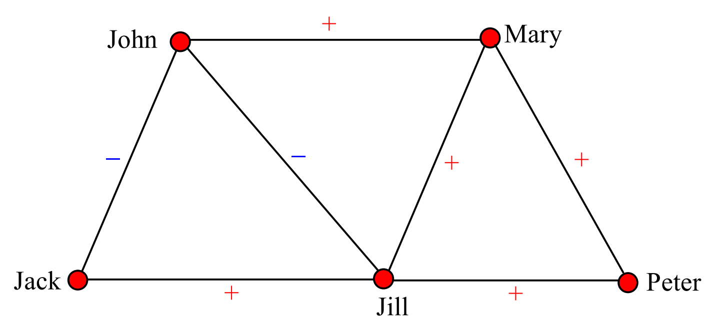


Figure 4.12

Now consider the following diagrams, which illustrate some of the situations that can occur when three people work together.

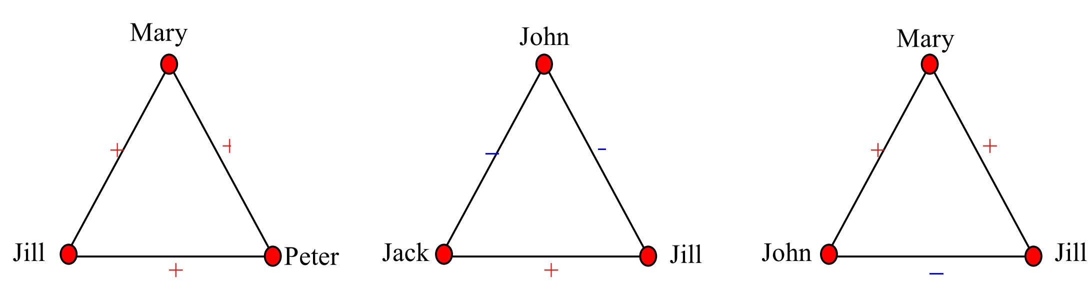
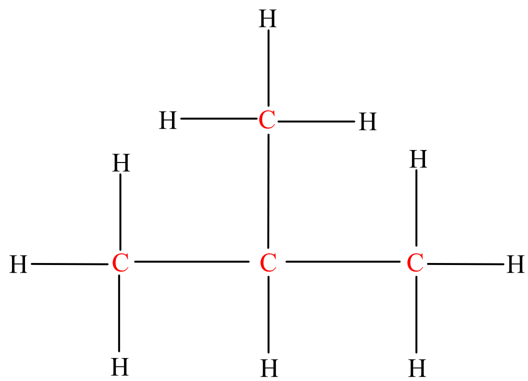


Figure 4.13

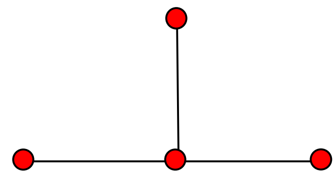
In the first case, all three get on well. In the second case, Jack and Jill get on well and both dislike John; the result is that John works on his own. In the third case, Mary would like to work with both John and Jill, but Jack and Jill do not wish to work together; in this case, no suitable working arrangement can be found and there is tension.

4.6.2 Chemistry

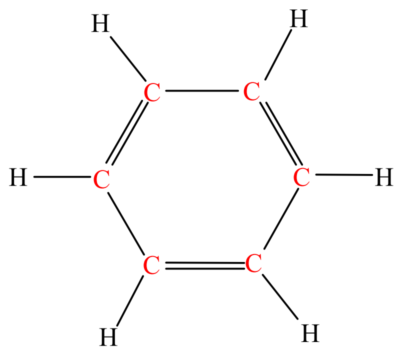
Chemical molecule can be represented as a graph whose vertices correspond to the atoms and whose edges correspond to the chemical bonds connecting them.



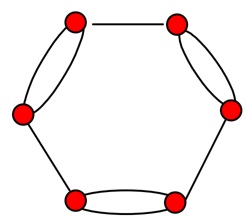
2-methylpropne



Carbon-graph



Benzene



Carbon-graph

Figure 4.14

4.6.3 Circulation diagram

Graphs are used to analyze the movements of people in large buildings. In particular, they have been used in the designing of airports, and in planning the layout of supermarkets.

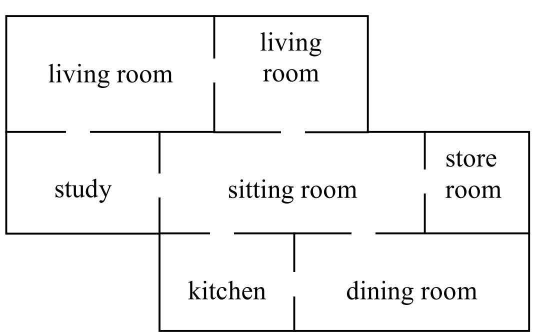


Figure 4.15 Architectural Floor Plan

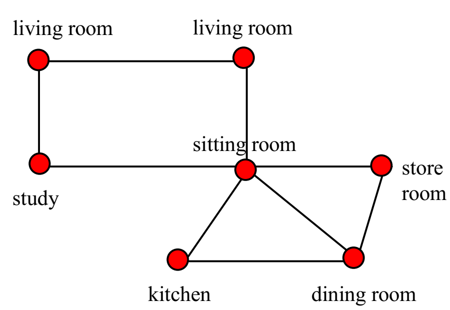


Figure 4.16 Circulation diagram

4.7 Paths and Circuits

4.7.1 Definitions

A **walk of length** k between v_1 and v_8 in a graph G is a succession of k edges of G of the form $v_1 e_1 v_2$, $v_2 e_2 v_3$, $v_3 e_3 v_4$, … ,$v_7 e_7 v_8$.

We denote this walk by $v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_6 e_6 v_7 e_7 v_8$.

A **closed walk** is a walk that starts and ends at the same vertex.

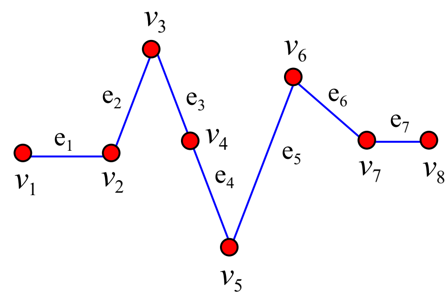


Figure 4.17

For example, in the following graph

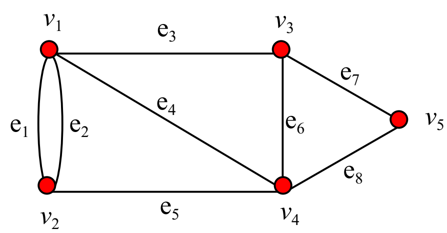


Figure 4.18

$v_1 e_4 v_4 e_5 v_2 e_1 v_1 e_3 v_3 e_7 v_5$ is a walk of length 5 between v_1 and v_5.

If all the edges (but not necessarily all the vertices) of a walk are different, then the walk is called a **path**. If, in addition, all the vertices are different, then the trail is called **simple path**. A closed path is called a **circuit**. A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last.

Example 4.19 In Figure 4.19, $v_1\ v_2\ v_4\ v_1\ v_3\ v_5$ is a path, $v_1\ v_2\ v_4\ v_3\ v_5$ is a simple path, $v_1\ v_2\ v_4\ v_3\ v_5\ v_4\ v_1$ is a circuit and $v_1\ v_2\ v_4\ v_5\ v_3\ v_1$ is a simple circuit.

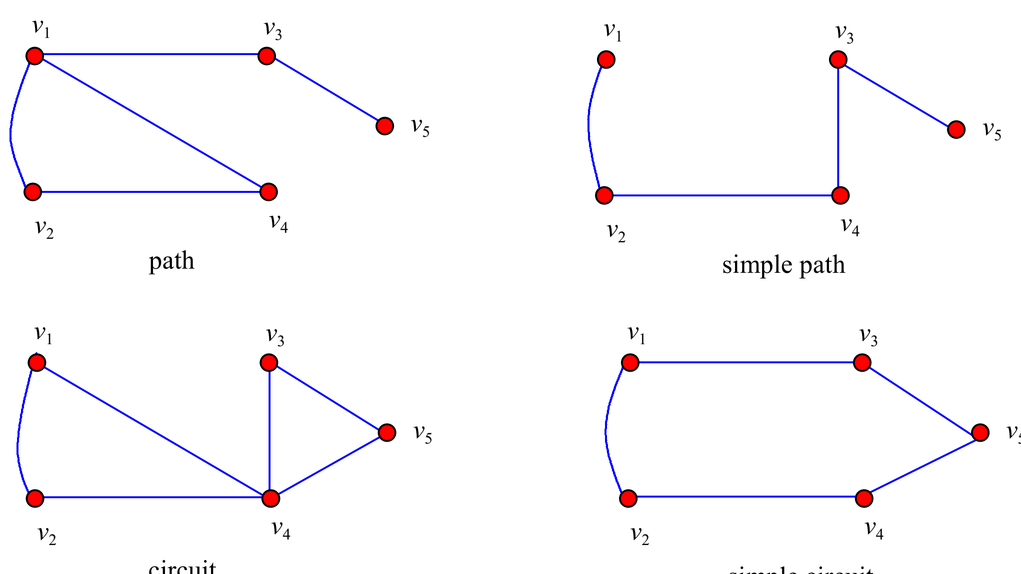


Figure 4.19

For ease of reference, these definitions are summarized in the following table:

	Repeated Edge	**Repeated Vertex**	**Starts and Ends at Same Point**
Walk	Allowed	Allowed	Allowed
Path	No	Allowed	Allowed
Simple path	No	No	No
Closed walk	Allowed	Allowed	Yes
Circuit	No	Allowed	Yes
Simple circuit	No	First and last only	Yes

4.7.2 Counting Walks of Length N

Theorem 4.1

If G is a graph with vertices v_1, v_2, …… v_m and A(G) is the adjacency matrix of G, then for each positive integer n,

the *ij*th entry of A^n = the number of walks of length n form v_i to v_j.

Example 4.10
Consider the following graph G.

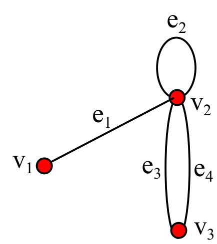


Figure 4.20

The adjacency matrix A(G) is

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

Then

$$A^2 = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 6 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

Therefore,

one walk of length 2 connecting v_1 to v_1;
($v_1 e_1 v_2 e_1 v_1$)
one walk of length 2 connecting v_1 to v_2;
($v_1 e_1 v_2 e_2 v_2$)
two walks of length 2 connecting v_1 to v_3;
($v_1 e_1 v_2 e_3 v_3$, $v_1 e_1 v_2 e_4 v_3$)
six walks of length 2 connecting v_2 to v_2;
($v_2 e_1 v_1 e_1 v_2$, $v_2 e_2 v_2 e_2 v_2$, $v_2 e_3 v_3 e_3 v_2$, $v_2 e_3 v_3 e_4 v_2$, $v_2 e_4 v_3 e_4 v_2$, $v_2 e_4 v_3 e_3 v_2$)
two walks of length 2 connecting v_2 to v_3;
($v_2 e_2 v_2 e_3 v_3$, $v_2 e_2 v_2 e_4 v_3$)
four walks of length 2 connecting v_3 to v_3.
($v_3 e_3 v_2 e_3 v_3$, $v_3 e_3 v_2 e_4 v_3$, $v_3 e_4 v_2 e_4 v_3$, $v_3 e_4 v_2 e_3 v_3$)

4.7.3 The Shortest Path Algorithm

The objective of this algorithm is to find the shortest path from vertex S to vertex T in a given network. We demonstrate the steps of the algorithm by finding the shortest distance from S to T in the following network:

Example 4.11

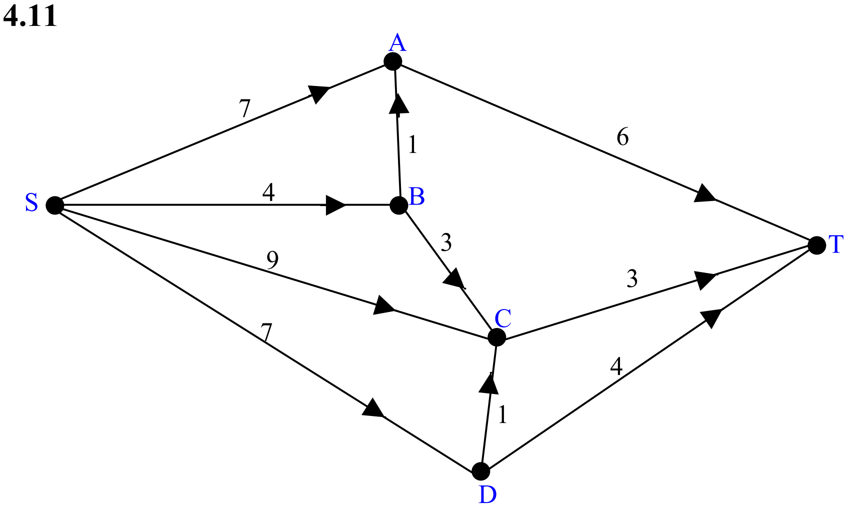


Figure 4.21

Solution

Iteration *n*	**Solved Nodes Directly Connected to Unsolved Nodes**	**Closest Connected Unsolved Node**	**Total Distance Involved**	***n*th Nearest Node**	**Minimum Distance**	**Last Connection**
1	S	B	4	B	4	SB
2	B S	A A, D	4 + 1 = 5 7	A	5	BA
3	A B S	T C D	5 + 6 = 11 4 + 3 = 7 7	C D	7 7	BC SD
4	A C D	T T T	5 + 6 = 11 7 + 3 = 10 7 + 4 = 11	T	10	CT

Final Stage:

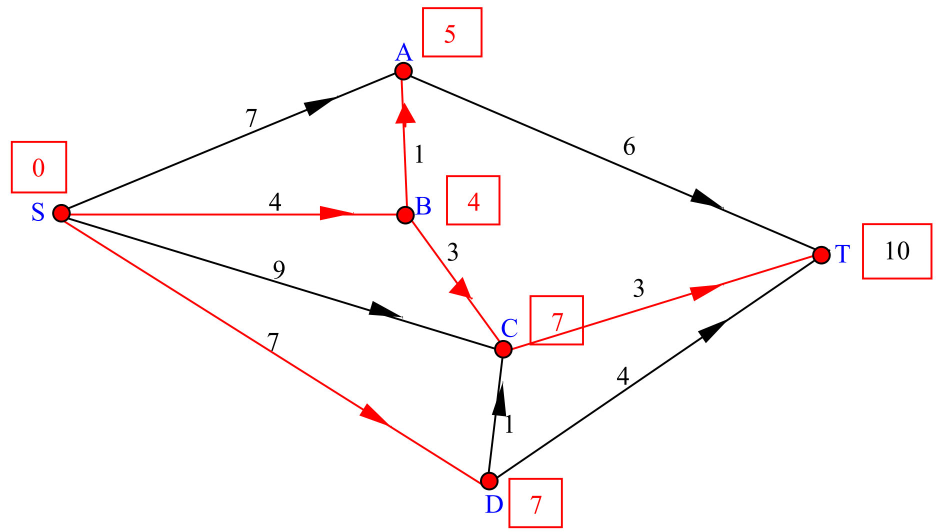


Figure 4.22

Conclusion
Therefore, the shortest path from S to T is SBCT with path length 10.

4.8 Eulerian Graphs

4.8.1 Definition

A connected graph G is Eulerian if there is a closed path which includes every edge of G; such path is called an Eulerian path.

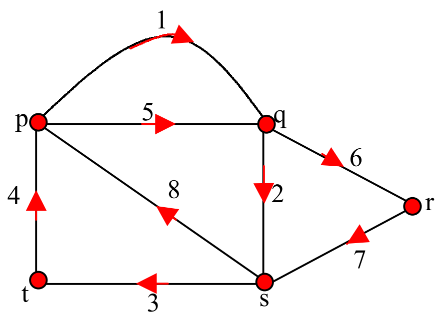


Figure 4.23 Eulerian graph

4.8.2 Theorem 4.2

Let G be a connected graph. Then G is Eulerian if and only if every vertex of G has even degree.

For example, since K_4 graph has odd degree vertices (deg 3), K_3 graph is not Eulerian. However all the vertices of K_5 are even (deg 4). Therefore by theorem 4.2, it is Eulerian.

Example 4.12
Consider the following four graphs

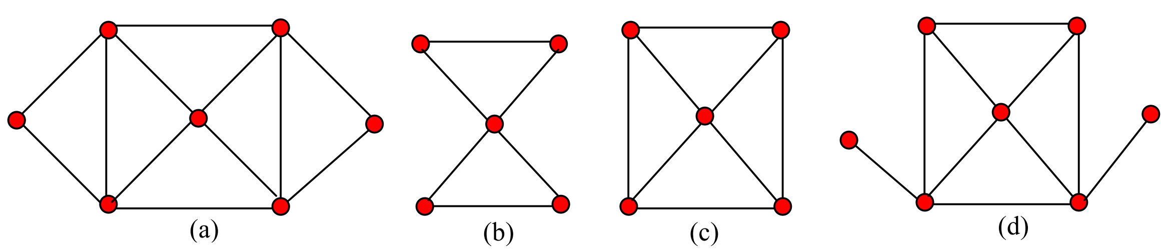


Figure 4.24

Which graph(s) is/are Eulerian?

Answer: Graph (a) and (b) are Eulerian.

4.8.3 Königsberg bridges problem

The four parts (A, B, C and D) of the city Königsberg were interconnected by seven bridges (p, q, r, s, t, u and v) as shown in the following diagram. Is it possible to find a route crossing each bridge exactly once ?

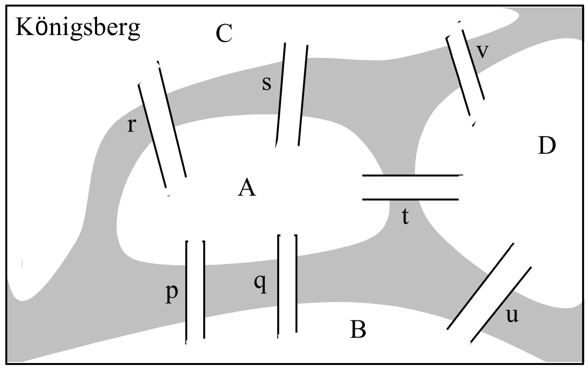


Figure 4.25 Königsberg map

We can express the Königsberg bridges problem in terms of a graph by taking the four land areas as vertices and the seven bridges as edges joining the corresponding pairs of vertices.

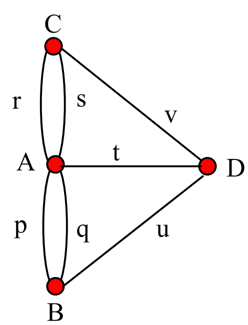


Figure 4.26

By theorem 4.2, it is not a Eulerian graph. It follows that there is no route of the desired kind crossing the seven bridges of Königsberg.

4.9 Hamiltonian Graphs

4.9.1 Definition

A connected graph G is **Hamiltonian** if there is a cycle which includes every vertex of G; such a cycle is called a **Hamiltonian** cycle.

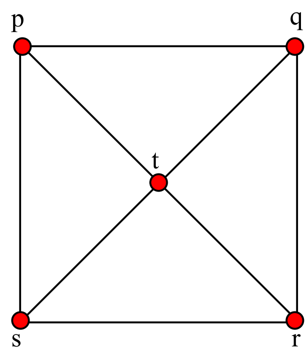


Figure 4.29 Hamiltonian Graph

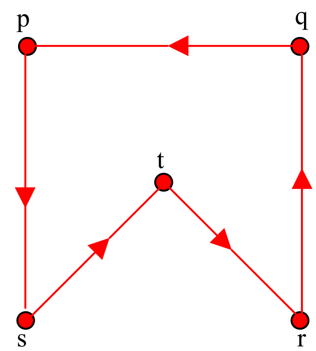


Figure 4.30 Hamiltonian Cycle

4.9.2 Theorem 4.3 (DIRAC'S THEOREM)

Let G be a simple graph with n vertices, where $n \geq 3$. If deg $v \geq \frac{1}{2}n$ for each vertex v, then G is Hamiltonian.

Example 4.14

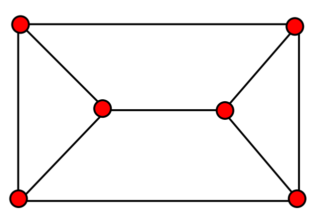


Figure 4.31

For the above graph, $n = 6$ and deg $v = 3$ for each vertex v, so this graph is Hamitonian by Dirac's theorem.

4.9.3 Theorem 4.4 (ORE'S THEOREM)

Let G be a simple graph with n vertices, where $n \geq 3$.

$$\deg v + \deg w \geq n,$$

for each pair of non-adjacent vertices v and w, then G is Hamiltonian.

Example 4.15

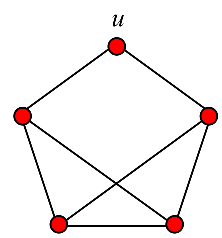


Figure 4.32

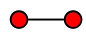
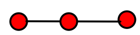
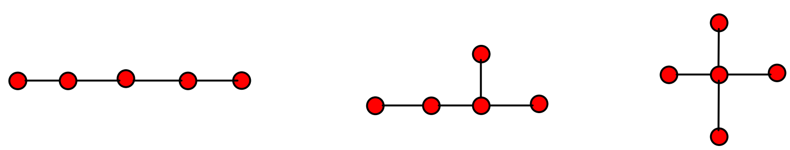
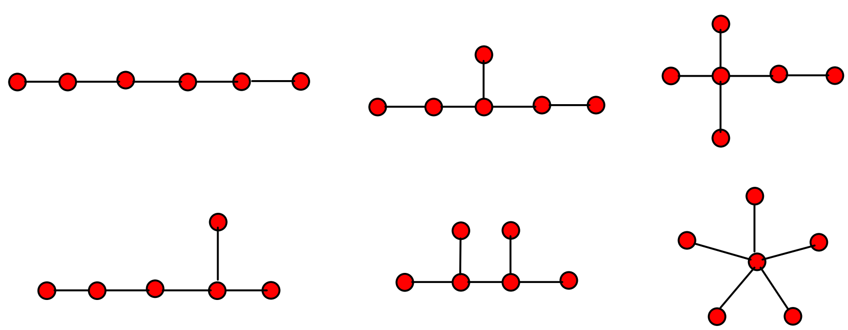
For figure 4.32, n = 5 but deg u = 2, so Dirac's theorem does not apply. However, $\deg v + \deg w \geq 5$ for all pairs of non-adjacent vertices v and w, so this graph is Hamiltonian by Ore's theorem.

4.10 Trees

4.10.1 Definition

A tree is a connected graph which contains no cycles.

Example 4.16

# of vertices	Tree Structure
1	
2	
3	
4	
5	
6	

4.10.2 Properties of Tree

1 Every tree with n vertices has exactly n – 1 edges.
2 Any two vertices in a tree are connected by exactly one path.
3 Each edge of a tree is a bridge.

4.10.3 Alternative definitions of the tree

- T is connected and has n – 1 edges.
- T has n – 1 edges and contains no cycles.
- T is connected and each edge is a bridge.
- Any two vertices of T are connected by exactly one path.
- T contains on cycles, but the addition of any new edges creates exactly one cycle.

4.10.4 Spanning Trees

Definition:
Let G be a connected graph. A **spanning tree** in G is a subgraph of G that includes all the vertices of G and is also a tree. The edges of the tree are called **branches**.

Example 4.17

A graph G

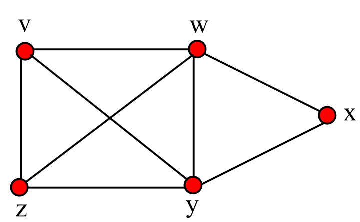


Figure 4.33

Spanning Trees

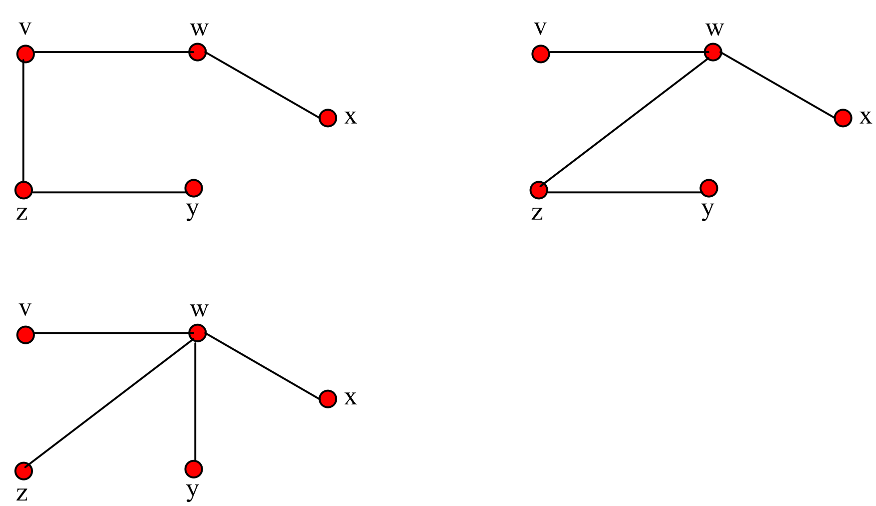


Figure 4.34

4.11 Minimum Spanning Tree

4.11.1 Definition

Let T be a spanning tree of minimum total weight in a connected weighted graph G. Then T is a **minimum spanning tree** of G.

4.11.2 Greedy Algorithm

Example 4.18

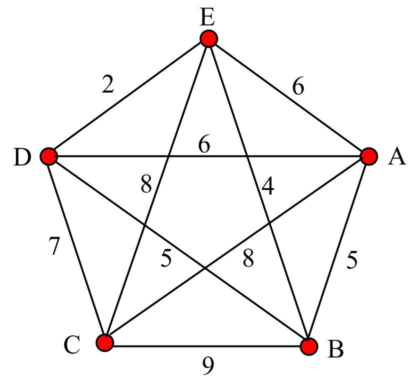


Figure 4.35

Iteration 1 (Choose an edge of minimum weight)

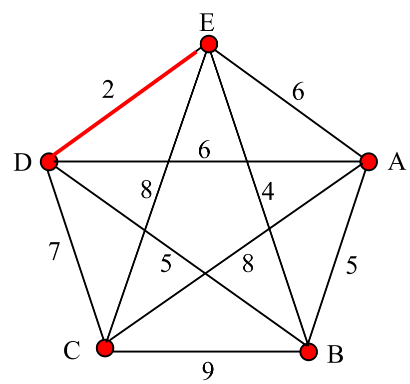


Figure 4.36a

Iteration 2 (Select the vertex from unconnected set {A,B,C} that is closest to connected set {D,E})

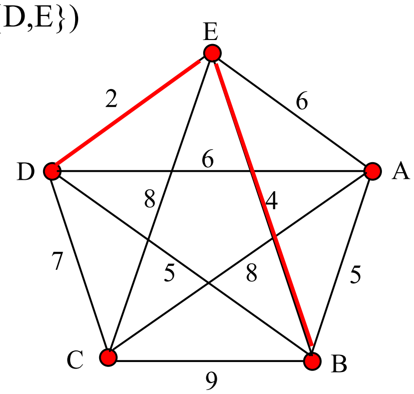


Figure 4.36b

Iteration 3 (Select the vertex from unconnected set {A,C} that is closest to connected set {B,D,E,})

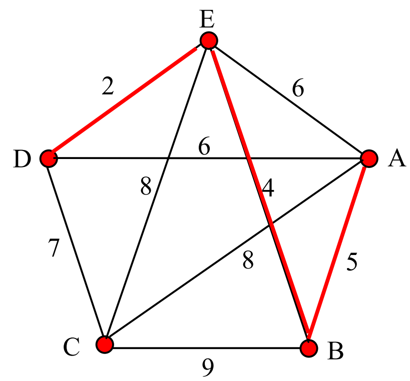


Figure 4.36c

Iteration 4 (Select the vertex from unconnected set {C} that is closest to connected set {A,B,D,E})

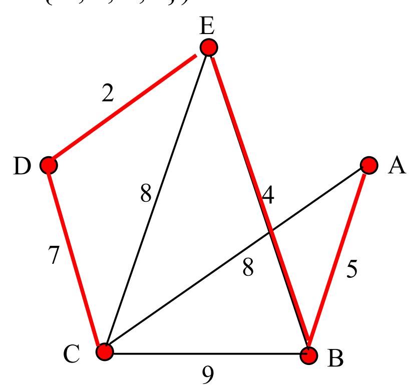


Figure 4.36d

Therefore, the minimum spanning tree is with weight 18.

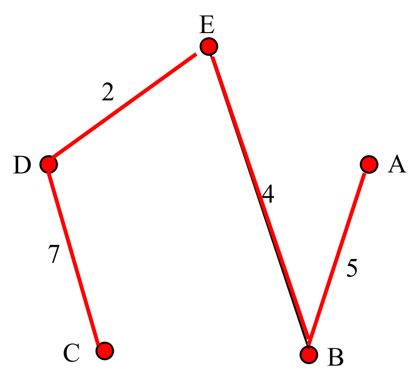


Figure 4.36e

4.12 References

4.12.1 Discrete Mathematics with Applications, second edition, Susanna S. Epp. PWS.
4.12.2 http://www.utm.edu/departments/math/graph/
4.12.3 http://www.gega.net/personal/bishay/math.htm