CHRISTIAN EMINENT COLLEGE, INDORE

(Academy of Management, Professional Education and Research)

An Autonomous Institution Accredited with 'A' Grade by NAAC

E-Content

on

B.SC.-III YEAR

DISCRETE MATHS- GRAPH

Prof. Sumit Sharma

UNIT-III

- Contents:
- Graph
- Types of graphs
- Euler and Hemiltonian graphs
- Shortest path
- Dijktra’s Algorithm for shortest paths.

Graphs: Definition

G = (V, E)

Edges (arcs)

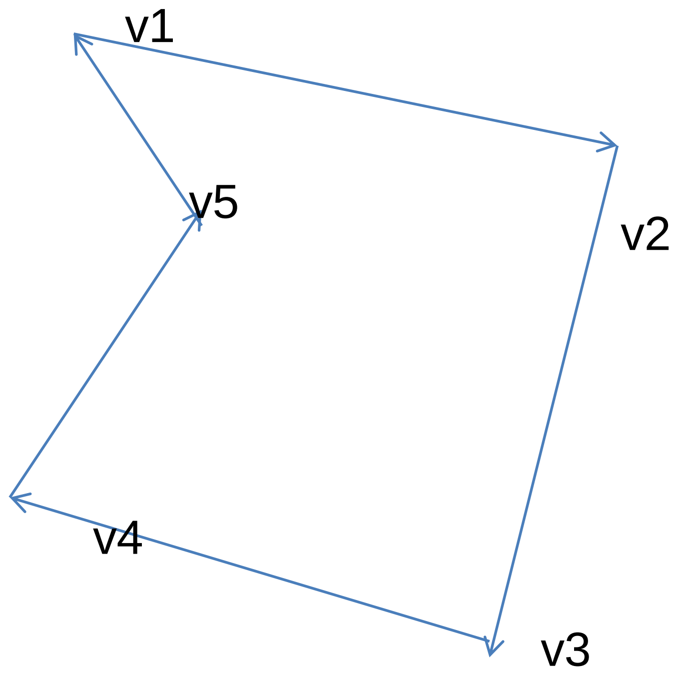
Vertices (nodes)

Edges are a subset of V × V

{(v,v') : v and v' are in V}

We also write v→ v' instead of (v.v')

Simple graph



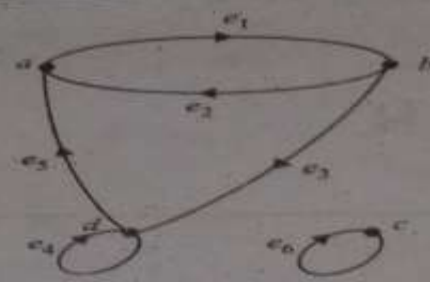


Fig. 1

Fig. 2

Parallel Edges in a Graph

Let $G = (V, E)$ be a graph. Then all edges having the same pair of end vertices are called **parallel edges.**

For example, the edges e_1 and e_2 in Fig. 1 are parallel edges.

Undirected Graph [Vikram (Vth Sem.), 2011; Sagar (Vth Sem.), 2013]

Definition. An **undirected graph** G is defined abstractly as an ordered pair (V, E) where V is a non-empty set and E is a multisets of two elements from V.

An undirected graph can be represented geometrically as a set of marked points V with a set of lines E between the points.

Example. $G = (\{a, b, c, d\}, \{\{a, b\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, c\}\})$ is an undirected graph as shown in Fig. 2 above.

From now onwards, when it is clear from the context, we shall use the term **graph** to mean either a directed graph, or an undirected graph, or both.

Simple Graph [Rewa 2001; Sagar 2001; Sagar (Vth Sem.), 2013; Jiwaji 2001; Jiwaji (Vth Sem.), 2012]

Definition. A graph $G = (V, E)$ that has neither self-loop nor parallel edges is called a **simple graph.**

Example. Let $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{e_1, e_2, e_3, e_4\}$, where

$e_1 = (v_1, v_2)$, $e_2 = (v_2, v_3)$, $e_3 = (v_1, v_3)$ and $e_4 = (v_3, v_4)$.

Then $G = (V, E)$ is a simple graph as shown in Fig. 3 below :

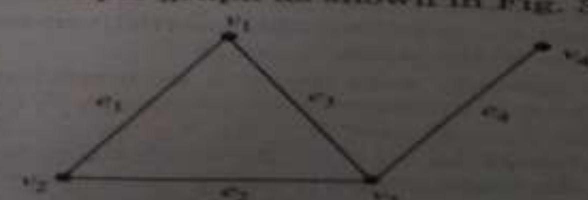


Fig. 3

Note that, in drawing a graph, it is immaterial whether the lines are drawn straight or curved, long or short; what is important is the incidence betw... edges and vertices.

The three graphs drawn in Fig. 4 (a), (b) and (c) are the same, because incidence between edges and vertices are the same.

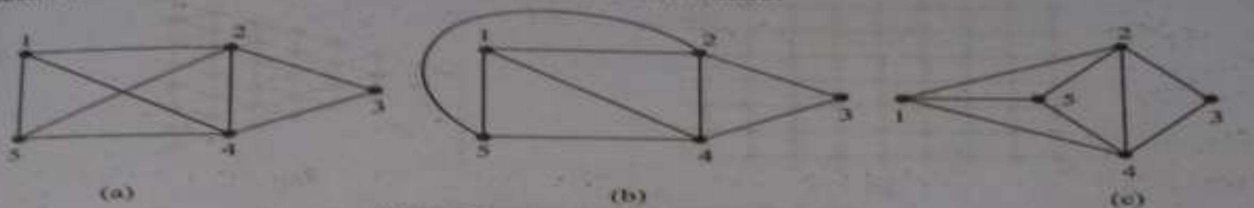


Fig. 4. Same graph drawn differently

Sometimes, in a graph two edges may seem to intersect at a point that does not represent a vertex. Such edges should be thought of as being in different planes and thus having no common point.

In Fig. 5, a graph is shown in which edges e_5 and e_6 have no common point.

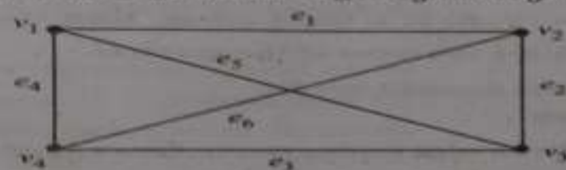


Fig. 5. Edges e_5 and e_6 have no common point

Finite and Infinite Graphs

[Sagar (Vth Sem), 2013; Vikram (Vth Sem.), 2011]

Definition. A graph with a finite number of vertices as well as a finite number of edges is called a **finite graph;** otherwise, it is an **infinite graph.**

We can picture a finite graph by representing each vertex with a point, and each edge $\{v_i, v_j\}$ with a line between points v_i and v_j.

In the adjoining Fig. 6 (a) and (b) both picture the same finite graph $(v_1, v_2), (v_1, v_5), (v_2, v_3), (v_2, v_4), (v_3, v_5)$. If $\{v_i, v_j\}$ is an edge, we say that vertices v_i and v_j are adjacent. In the adjoining Fig. 6, v_1 is adjacent to v_2 but not to v_3.

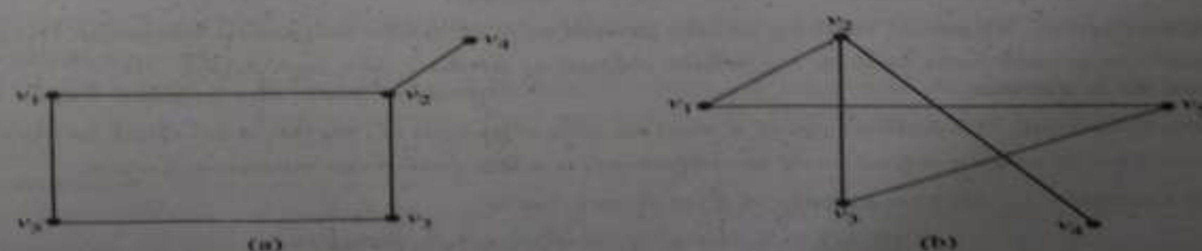


Fig 6. Representations of the same finite graph

Portions of two infinite graphs are shown in Fig. 7 (a) and (b).

(a) (b)

Fig 7. Portions of two infinite graphs

In this chapter we shall confine ourselves to the study of finite graphs, and unless otherwise stated the term "graph" will always mean a finite graph.

Order of a Group

Definition. If $G = (V, E)$ is a finite group, then the number of vertices is denoted by $|V|$ and is called the order of the graph G.

The number of edge is denoted by $|E|$.

Incidence and Adjacency

Definition. Let e_k be an edge joining two vertices v_i and v_j of a graph $G = (V, E)$. Then the edge e_k is said to be **incident** on each of its end vertices v_i and v_j.

Example. In the graph of Fig. 3, edge e_2 is incident on vertices v_2 and v_3.

Definition. Two vertices in a graph $G = (V, E)$ are said to be **adjacent** if there exists an edge joining the vertices.

Example. In graph of Fig. 3, vertices v_1 and v_3 are adjacent while vertices v_1 and v_4 are not adjacent.

Definition. Two non-parallel edges of a graph $G = (V, E)$ are said to be **adjacent** if they are incident on a common vertex.

Example. Edges e_1 and e_2 in the graph of Fig. 3 are adjacent because they are incident on a common vertex v_2 while edges e_1 and e_4 are not adjacent.

Degree of a Vertex [Indore 2001; Jiwaji 2001; Sagar 2001]

Definition. The **degree** of a vertex v in a graph G, written as $d(v)$, is equal to the number of edges which are incident on v with self-loop counted twice.

Example 1. In the graph of Fig. 3, we have

$$d(v_1) = 2,\ d(v_2) = 2,\ d(v_3) = 3,\ d(v_4) = 1.$$

Example 2. *Find the degree of each vertex of the following graph :* [Jabalpur 2005]

Fig. 10

We now derive three simple but important results on degree of vertices.

Theorem 1. *The sum of the degrees of all vertices in a graph G is equal to twice the number of edges in G.* [Indore (Vth Sem.), 2011, 2012]

Proof. Let $G = (V, E)$ be a graph. Then the number of edges in G is $|E|$. Since each edge in G is incident on two vertices, it contributes 2 to the sum of the degrees of the graph. Thus the sum of degrees of all the vertices in G is given by

$$\sum_{v \in V} d(v) = 2|E|.$$

This result is also known as **handshaking lemma.**

To verify the above theorem, consider the graph in Fig. 11.

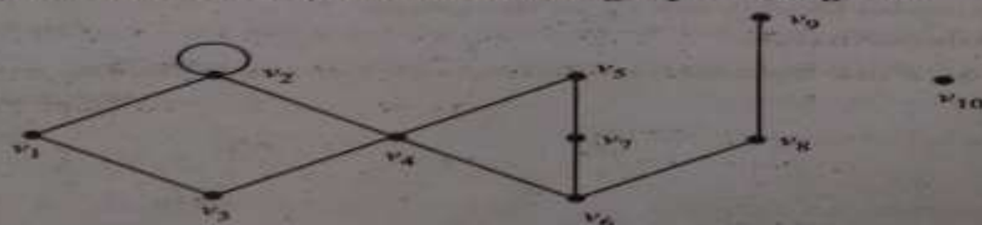


Fig. 11

In this graph, we have

$$d(v_1) = 2,\ d(v_2) = 4,\ d(v_3) = 2,\ d(v_4) = 4,\ d(v_5) = 2,$$
$$d(v_6) = 3,\ d(v_7) = 2,\ d(v_8) = 2,\ d(v_9) = 1,\ d(v_{10}) = 0.$$

Therefore, $\sum_{i=1}^{10} d(v_i) = 22 = 2 \times 11 =$ twice the number of edges.

Even and Odd Vertices

Definition. A vertex is said to be **even or odd vertex** according as its degree is an even or an odd number. For example, in the graph of Fig. 11, vertices $v_1, v_2, v_3, v_4, v_5, v_7, v_8$ and v_{10} are even vertices whereas v_6 and v_9 are odd vertices.

Theorem 2. *The vertices of odd degree (odd vertices) in a graph is always even.*

[Vikram (Vth Sem.), 2011; Indore 2008]

Proof. Let $G = (V, E)$ be a graph. Let V_e and V_0 denote the set of even and odd vertices in G. Then $V_e \subseteq V$ and $V_0 \subseteq V$ such that

$$V = V_e \cup V_0 \text{ and } V_e \cap V_0 = \phi.$$

Hence $$\sum_{v \in V} d(v) = \sum_{v \in V_e} d(v) + \sum_{v \in V_0} d(v)$$

$$\Rightarrow \quad 2|E| = 2k + \sum_{v \in V_0} d(v)$$

$$\left[\because \sum_{v \in V_e} d(v) = \text{an even number} = 2k, \text{ say}\right]$$

$$\Rightarrow \quad \sum_{v \in V_0} d(v) = 2(|E| - k) = \text{ an even number.}$$

But each term $d(v)$ in the L.H.S. is odd, the total number of terms in the sum must be even (to make the sum an even number). Thus, the number of vertices of odd degree must be even; *i.e.*,

$$|V_0| = \text{an even number.}$$

In-Degree and Out-Degree of a Vertex

Definitions. In a directed graph, the number of edges incident into a vertex v_i is called **in-degree** of the vertex v_i and is denoted by $d^-(v_i)$.

The number of edges incident out of a vertex v_i is called out-degree of the vertex v_i and is denoted by $d^+(v_i)$.

Thus for a vertex v_i, we have

In-degree of $v_i = d^-(v_i) =$ number of vertices in $\{w \mid (w, v_i) \in E\}$

Out-degree of $v_i = d^+(v_i) =$ number of vertices in $\{w \mid (v_i, w) \in E\}$.

For example, in the directed graph of Fig. 12.

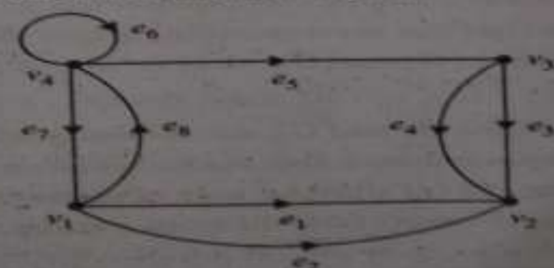


Fig. 12

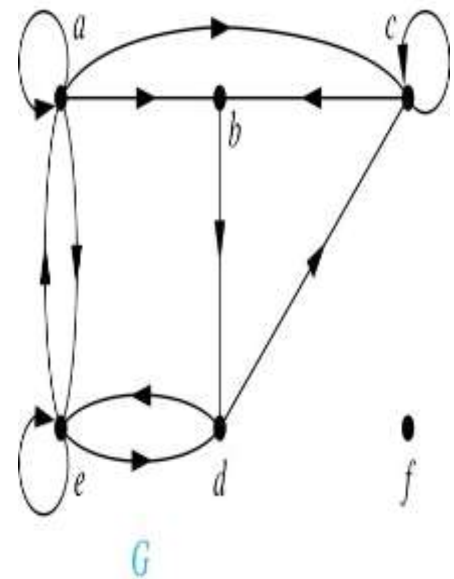
$d^-(v_1) = 1$, $\quad d^+(v_1) = 3$

$d^-(v_2) = 4$, $\quad d^+(v_2) = 0$

$d^-(v_3) = 1$, $\quad d^+(v_3) = 2$

$d^-(v_4) = 2$, $\quad d^+(v_4) = 3$.

- Directed Graphs (*continued*)
- **Definition:** The *in-degree of a vertex v*, denoted *deg*−(*v*), is the number of edges which terminate at *v*. The *out-degree of v*, denoted *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 the vertex.
- **Example:** In the graph *G* we have



- deg−(*a*) = 2, deg−(*b*) = 2, deg−(*c*) = 3, deg−(*d*) = 2,
- deg−(*e*) = 3, deg−(*f*) = 0.
-
- deg+(*a*) = 4, deg+(*b*) = 1, deg+(*c*) = 2, deg+(*d*) = 2, deg+ (*e*) = 3, deg+(*f*) = 0.

- Directed Graphs (*continued*)
- **Theorem 3**: I a directed $G = (V, E)$ with IEI edges, sum of the in-degrees = sum of the out-degrees= IEI. In other words

$$\sum_{v_i \in V} d^-(v_i) = \sum_{v_i \in V} d^+(v_i) = |\mathrm{E}|.$$

$$\text{Proof} - \text{Since} \sum_{v_i \in V} d(v_i) = 2|\mathrm{E}|$$

$$\Longrightarrow \sum_{v_i \in V} d(v_i) = \sum_{v_i \in V} d^-(v_i) + \sum_{v_i \in V} d^+(v_i) = 2$$

$$\Longrightarrow |\mathrm{E}| = \frac{1}{2}\sum_{v_i \in V} d(v_i) = \frac{1}{2}\sum_{v_i \in V} d^-(v_i) + \frac{1}{2}\sum_{v_i \in V} d^+(v_i)$$

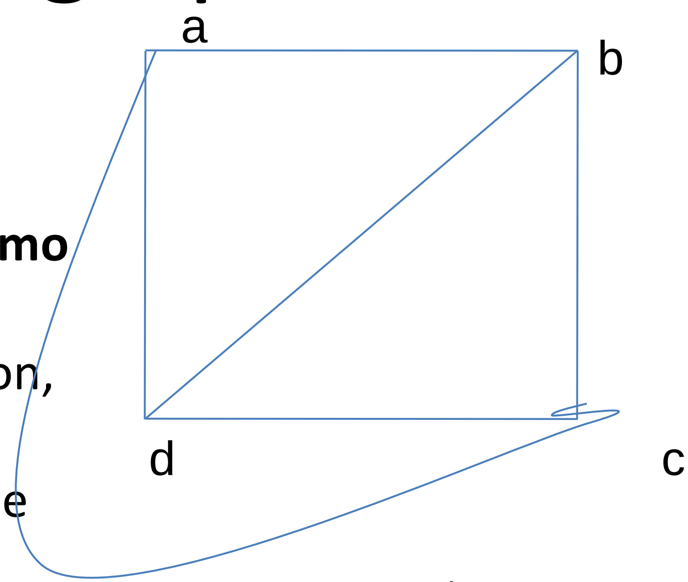
$$\Longrightarrow |\mathrm{E}| = \sum_{v_i \in V} d^-(v_i) = \sum_{v_i \in V} d^+(v_i)$$

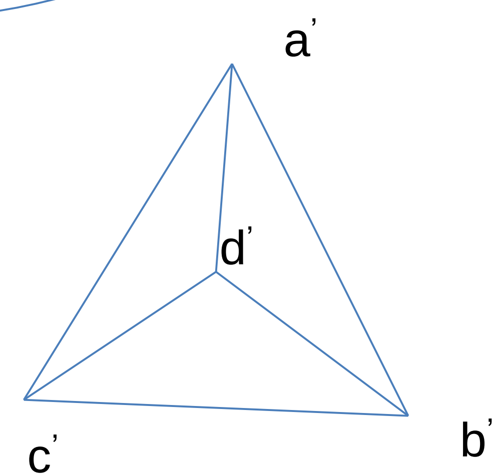
or

$$\sum_{v_i \in V} d^-(v_i) = \sum_{v_i \in V} d^+(v_i) = |\mathrm{E}|$$

Isomorphism of a graph

- **Definition 5.1.4** Suppose
- G_1=(V,E)and G_2=(W,F). G1→G1 and G2→G2 are **isomorphic** if there is a one-to-one f:V→W and such that {v1,v2}∈E if and only if {f(v_1),f(v_2)}∈F In addition, the repetition numbers of {v1,v2}{v1,v2} and {f(v1),f(v2)}{f(v1),f(v2)} are the same if multiple edges or loops are allowed. This bijection of f is called an **isomorphism**. When G1 → G1 and G2 → G2 are isomorphic, we write G1≅G2.
- Each pair of graphs in figure are isomorphic. For example, to show explicitly that G1≅G3, an isomorphism is
- f(v1)f(v2)f(v3)f(v4)=w3=w4=w2=w1.
- Clearly, if two graphs are isomorphic, their degree sequences are the same.





Example 1. ***Show that the graphs given below are isomorphic.***

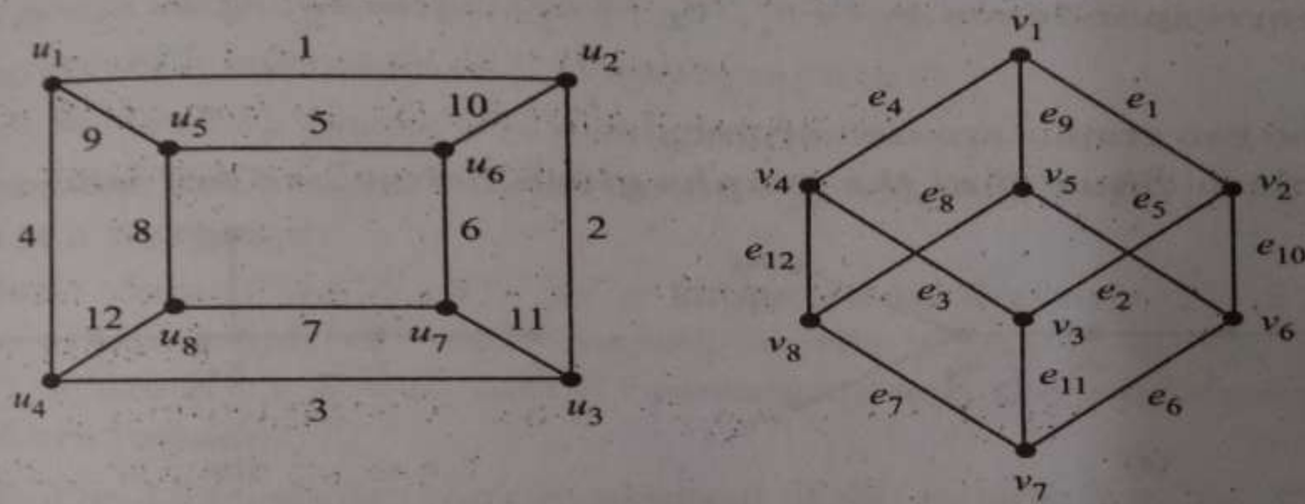


Fig. 14

Solution. We can establish the correspondence between the two graphs as follows :

Vertex correspondence : $u_1 \leftrightarrow v_1$, $u_2 \leftrightarrow v_2$, $u_3 \leftrightarrow v_3$, $u_4 \leftrightarrow v_4$, $u_5 \leftrightarrow v_5$, $u_6 \leftrightarrow v_6$, $u_7 \leftrightarrow v_7$, $u_8 \leftrightarrow v_8$.

Edge correspondence : $1 \leftrightarrow e_1$, $2 \leftrightarrow e_2$, $3 \leftrightarrow e_3$, $4 \leftrightarrow e_4$, $5 \leftrightarrow e_5$, $6 \leftrightarrow e_6$, $7 \leftrightarrow e_7$, $8 \leftrightarrow e_8$, $9 \leftrightarrow e_9$, $10 \leftrightarrow e_{10}$, $11 \leftrightarrow e_{11}$, $12 \leftrightarrow e_{12}$.

Thus the two graphs are isomorphic.

Example 2. *Show that the following two graphs (a) and (b) are isomorphic.*

(a) (b)

Fig. 15

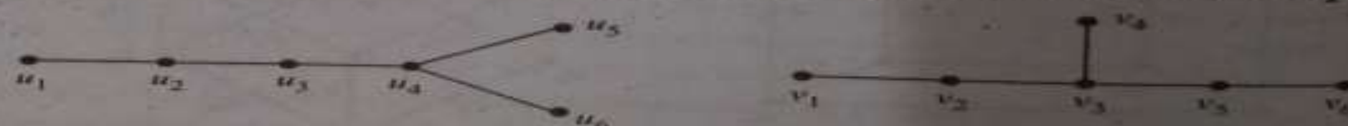
Solution. We define the correspondence between vertex sets and edge sets of two graphs (a) and (b) as given below :

Vertex Correspondence. $v_1 \leftrightarrow v_1^1$, $v_2 \leftrightarrow v_2^1$, $v_3 \leftrightarrow v_3^1$, $v_4 \leftrightarrow v_4^1$ and $v_5 \leftrightarrow v_5^1$.

Edge correspondence. $e_1 \leftrightarrow e_1^1$, $e_2 \leftrightarrow e_2^1$, $e_3 \leftrightarrow e_3^1$, $e_5 \leftrightarrow e_5^1$, $e_4 \leftrightarrow e_4^1$ and $e_6 \leftrightarrow e_6^1$.

Thus the two graphs are isomorphic.

Example 3. *Show that the graphs given below are not isomorphic.*



(a) (b)

Fig. 16

Solution. The two graphs in Fig. 14 are not isomorphic because [Sagar (Vth Sem.), 2018]

(i) If the graph in Fig. 16 (a) is isomorphic to the graph in Fig. 16 (b) then vertex u_4 in (a) must correspond to the vertex v_3 in (b), since $d(u_4) = 3 = d(v_3)$ and there are no other vertices of degree three.

(ii) There are two pendant vertices u_5 and u_6, adjacent to u_4 in (a) while there is only one pendant vertex v_4, adjacent to v_3 in (b).

§ 4.4. SUBGRAPHS AND COMPLEMENTS

Subgraph

Definition. Let $G = (V, E)$ be a graph. A graph $G' = (V', E')$ is said to be a subgraph of G if E' is a subset of E and V' is a subset of V such that each edge of G' has the same end vertices in G' as in G. [Jiwaji (Vth Sem.), 2012; Vikram 2001]

Subgraphs

Given a graph G = (V, E) and a graph G' = (V', E'), G is a **subgraph** of G' if:

- $V \subseteq V'$
- $E \subseteq E'$

Every element in the left set is an element in the right set

In other words, a **subgraph** of a graph G is any graph obtained from G by a deletion of edges and/or vertices from G. Note that when a vertex v is deleted, all edges terminating at v must also be deleted.

Example. The graph in Fig. 17 (b) is a subgraph of the graph in Fig. 17 (a).

Fig. 17. Graph (a) and one of its subgraph (b)

Spanning Subgraph

Definition. A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is said to be **spanning subgraph** if all the vertices of G are present in the subgraph G'.

The following observations can be made immediately :

1. Every graph is its own subgraph.
2. A subgraph of a subgraph of G is a subgraph of G.
3. A single vertex in a graph G is a subgraph of G.
4. A single edge in G, together with its end vertices, is also a subgraph of G.

Complement of a Subgraph

Definition. Let $G' = (V', E')$ be a subgraph of a graph $G = (V, E)$. The complement of subgraph G' with respect to the graph G is the subgraph $G'' = (V'', E'')$, where $E'' = E - E'$ and V'' containing only the vertices with which the edges in E'' are incident.

Example. Fig. 18 (c) shows the complement of the subgraph in Fig. 18 (b) with respect to the graph in Fig. 18 (a).

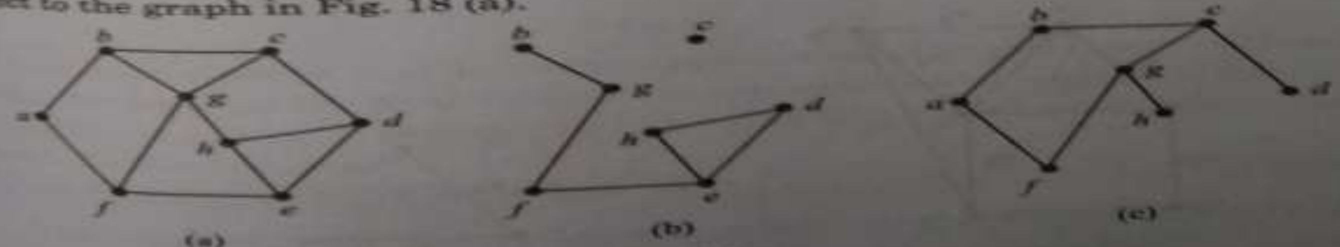
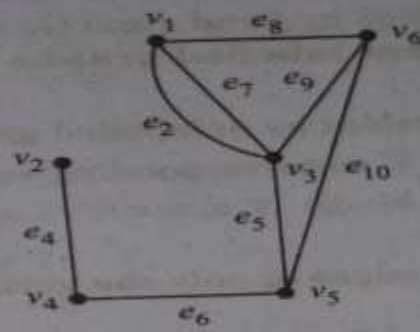


Fig. 18

Thus, the complement of a subgraph G' with respect to graph G is obtained from G by removing the edges of G'.



$G_1 \oplus G_2$

Ring sum of two graphs G_1 and G_2

Fig. 19

(d) Decomposition

Definition. A graph G is said to be **decomposed** into two subgraphs G_1 and G_2 if

$$G_1 \cup G_2 = G$$
$$G_1 \cap G_2 = \text{a null graph.}$$

For example, the graph G and its decomposed graphs are shown in Fig. 20 (a) and Fig. 20 (b), (c) below :

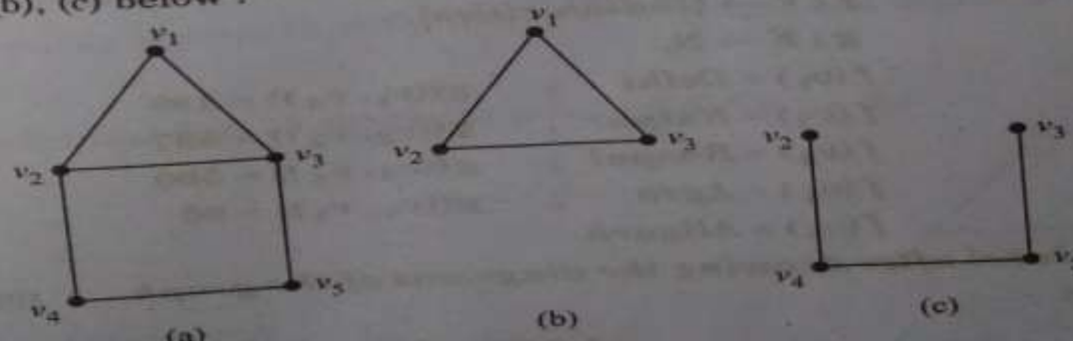


Fig. 20

It may be observe that in many applications of graph theory in computer science, the topology (V, E) of a graph is supplemented by additional information relating to either V or E or both. We can make this more precise by defining the concept of a labelled graph.

[Rewa 2001]

...LING OF A GRAPH

Graph Models Graphs are used in a wide variety of models. We will present a few graph models from diverse fields here. Others will be introduced in subsequent sections of this and later chapters. When we build a graph model, we need to make sure that we have correctly answered the three key questions we posed about the structure of a graph.

Links

Example 1 **Niche Overlap Graphs in Ecology** Graphs are used in many models involving the interaction of different species of animals. For instance, the competition between species in an ecosystem can be modeled using a **niche overlap graph**. Each species is represented by

Links

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

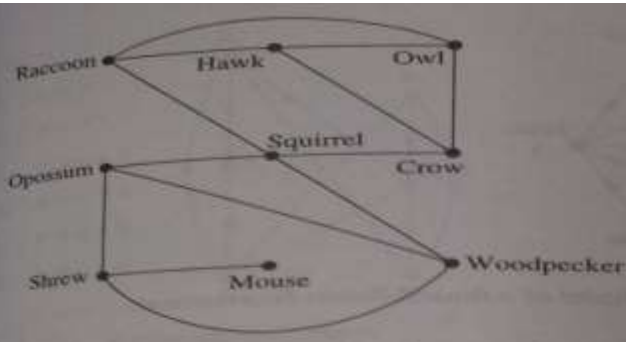


Figure 6 A Niche Overlap Graph

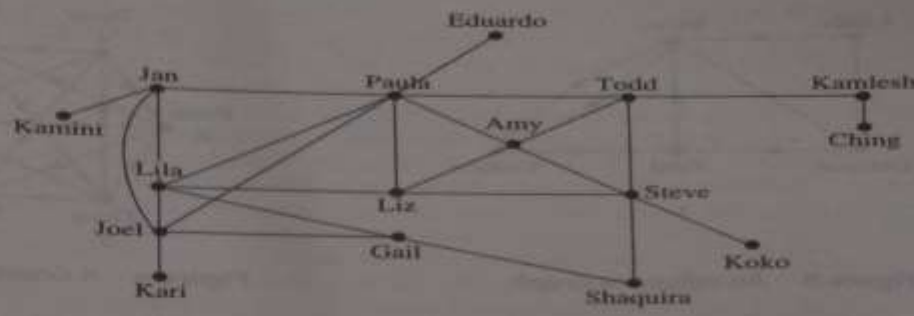


Figure 7 An Acquaintanceship Graph.

a vertex. An undirected edge connects two vertices if the two species represented by these vertices compete (that is, some of the food resources they use are the same). A niche overlap graph is a simple graph because no loops or multiple edges are needed in this model. The graph in Figure 6 models the ecosystem of a forest. We see from this graph that squirrels and raccoons compete but that crows and shrews do not. ◄

Example 2 **Acquaintanceship Graphs**

Links We can use graph models to represent various relationships between people. For example, we can use a simple graph to represent whether two people know each other, that is, whether they are acquainted. 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. 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 7. The acquaintanceship graph of all people in the world has more than six billion vertices and probably more than one trillion edges! We will discuss this graph further in Section 8.4. ◄

Example 3 **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 influences 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 8. In the group modeled by this influence graph, Deborah can influence Brian, Fred, and Linda, but no one can influence her. Also, Yvonne and Brian can influence each other. ◄

Example 4 **The Hollywood Graph** The **Hollywood graph** represents actors by vertices and connects two vertices when the actors represented by these vertices have worked together on a movie. This graph is a simple graph because its edges are undirected, it contains no multiple edges, and it contains no loops. According to the Internet Movie Database, in Janurary 2006 the Hollywood graph had 637,099 vertices representing actors who have appeared in 339,896 films, and had more than 20 million edges. We will discuss some aspects of the Hollywood graph later in Section 8.4. ◄

Example 8 **The Web Graph** The World Wide Web can be modeled as a directed graph where each Web page is represented by a vertex and where an edge starts at the Web page a and ends at the Web page b if there is a link on a pointing to b. Because new Web pages are created and others removed somewhere on the Web almost every second, the Web graph changes on an almost continual basis. Currently the Web graph has more than three billion vertices and 20 billion edges. Many people are studying the properties of the Web graph to better understand the nature of the Web. We will return to Web graphs in Section 8.4, and in Chapter 9 we will explain how the Web graph is used by the Web crawlers that search engines use to create indices of Web pages. ◀

Links

Example 9 **Precedence Graphs and Concurrent Processing** Computer programs can be executed more rapidly by executing certain statements concurrently. It is important not to execute a statement that requires results of statements not yet executed. The dependence of statements on previous statements can be represented by a directed graph. Each statement is represented by a vertex, and there is an edge from one vertex to a second vertex if the statement represented by the second vertex cannot be executed before the statement represented by the first vertex has been executed. This graph is called a **precedence graph.** A computer program and its graph are displayed in Figure 11. For instance, the graph shows that statement S_5 cannot be executed before statements S_1, S_2, and S_4 are executed. ◀

Example 10 **Roadmaps** Graphs can be used to model roadmaps. In such models, vertices represent intersections and edges represent roads. Undirected edges represent two-way roads and directed edges represent one-way roads. Multiple undirected edges represent multiple two-way roads connecting the same two intersections. Multiple directed edges represent multiple one-way roads that start at one intersection and end at a second intersection. Loops represent loop roads. Consequently, roadmaps depicting only two-way roads and no loop roads, and in which no two roads connect the same pair of intersections, can be represented using a simple undirected graph. Roadmaps depicting only one-way roads and no loop roads, and where no two roads start at the same intersection and end at the same intersection, can be modeled using simple directed graphs. Mixed graphs are needed to depict roadmaps that include both one-way and two-way roads. ◀

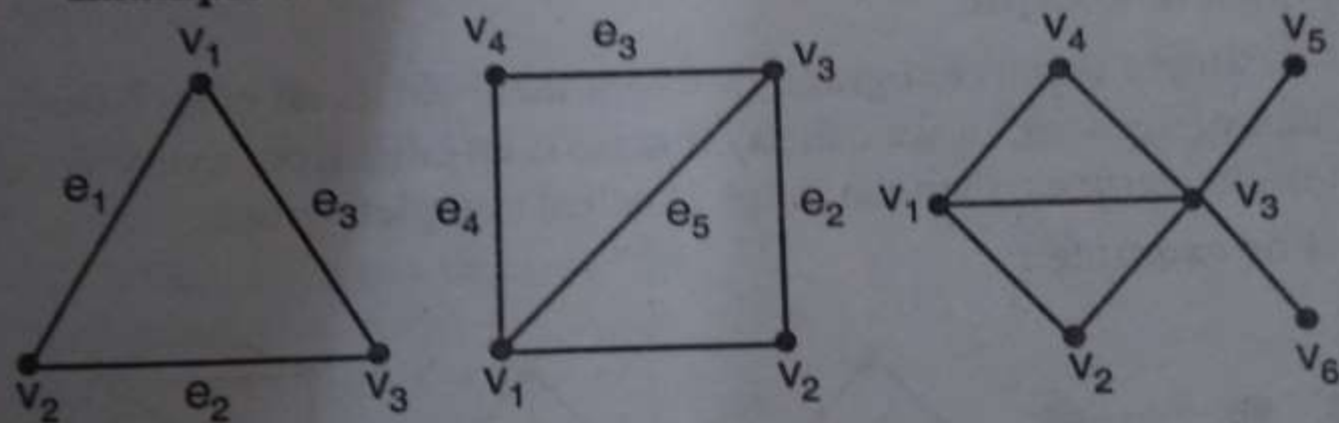
14.6. Types of the Graph

1. Simple Graph [Rewa 2007; Indore 2001]

A graph which has neither a self loop nor parallel edges is called a **simple graph.**

In other words, A graph which does not have any self loop and parallel edges is called a **simple graph.**

Example :



(Simple Graph)

2. Multi-Graph

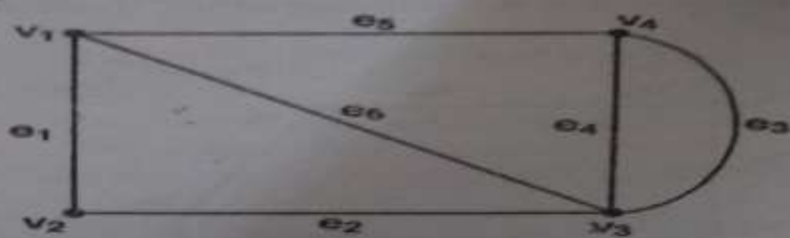
The graph consisting multiple edges (parallel edges) is called a **multi graph.**

OR

A multi graph $G = (V, E)$ consists of a set V of vertices, a set E of edges, and a function f from E to $\{(v_1, v_2) : v_1, v_2 \in V, v_1 \neq v_2\}$. The edges e_1 and e_2 are called multiple or paralle edges if

$$f(e_1) = f(e_2).$$

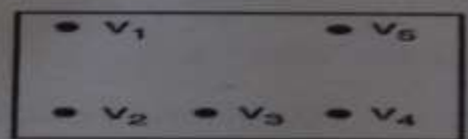
Example :



3. Null Graph

A graph in which edge set E is empty (*i.e.*, $E = \phi$), while vertex set V must not be empty *i.e.*, $V \neq \phi$ (otherwise there is no graph) is called **Null Graph.**

Example : Consider the graph $G = G(V, E)$ where $V = (v_1, v_2, v_3, v_4, v_5)$ and $E = \phi$



(Null Graph)

4. Complete Graph

A simple connected graph in which there exists an edge between every pair of vertices, or we can say if there is an edge from every vertex to rest of the vertices, then the graph is called complete graph.

For example :

Complete graph of one, two, three and four vertices.

Properties of the Complete Graph :

1. In a complete graph the degree of every vertex is one less than the number of vertices.

2. The total number of edges in a complete graph are always $\frac{n(n-1)}{2}$.

Where n is the number of vertices.

§ 4.7. COMPLETE AND BIPARTITE GRAPHS

[Vikram 2001; Jiwaji 2001; Rewa 2007]

Definitions :

Complete Graph

A graph $G = (V, E)$ is said to be **complete** if for all $v_i, v_j \in V$, we have $(v_i, v_j) \in E$.

The complete graph on n vertices is denoted by K_n. [Jiwaji 2001]

It can easily be seen that K_n has $\frac{n(n-1)}{2}$ edges and degree of each vertex is $(n-1)$.

The adjoining Fig. 23 shows the graphs $K_1, K_2, \ldots, K_6$.

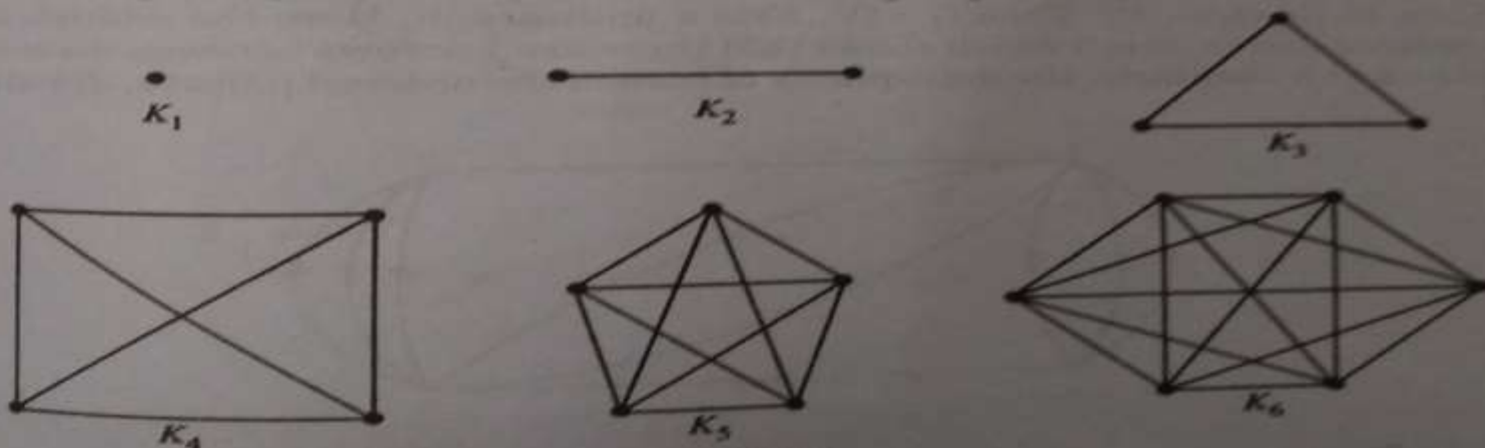


Fig. 23

[Jiwaji (Vth Sem.), 2012]

Regular Graph

A graph in which all vertices are of equal degree is called a regular graph. If the degree of each vertex is r, then the graph is called a regular graph of degree r.

Examples : (i) Every null graph is regular of degree zero.

(ii) The complete graph K_n is a regular graph of degree $n-1$.

Note. If a graph G has n vertices and is regular of degree r, then G has $\frac{1}{2}rn$ edges.

Bipartite Graph

A graph $G = (V, E)$ is said to be **bipartite** if there is a partition $\{V_1, V_2\}$ of V; *i.e.*, $V = V_1 \cup V_2$, $V_1 \neq \phi$, $V_2 \neq \phi$, $V_1 \cap V_2 = \phi$, such that no two vertices of V_1 are adjacent and no two vertices of V_2 are adjacent. In other words, each edge of G connects a vertex of V_1 to a vertex of V_2. A bipartite graph is said to be **complete** if for any pair $v_1 \in V_1$ and $v_2 \in V_2$, we have $(v_1, v_2) \in E$. If $|V_1| = m$ and $|V_2| = n$, then the complete graph (V, E) is written $K_{m,n}$.

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. In this graph, each edge connects a vertex in the subset of vertices representing males and a vertex in the subset of vertices representing females. This leads us to Definition 5.

Links

Definition 5 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.

In Example 9 we will show that C_6 is bipartite, and in Example 10 we will show that K_3 is not bipartite.

Example 9 C_6 is bipartite, as shown in Figure 7, because its vertex set can be partitioned into the two sets $V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$, and every edge of C_6 connects a vertex in V_1 and a vertex in V_2. ◀

Example 10 K_3 is not bipartite. To verify this, note that if we divide the vertex set of K_3 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 K_3 each vertex is connected to every other vertex by an edge. ◀

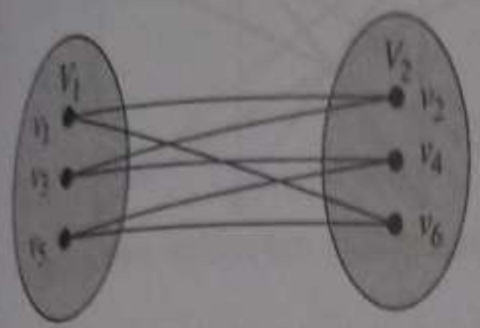


Figure 7 Showing That C_6 Is Bipartite.

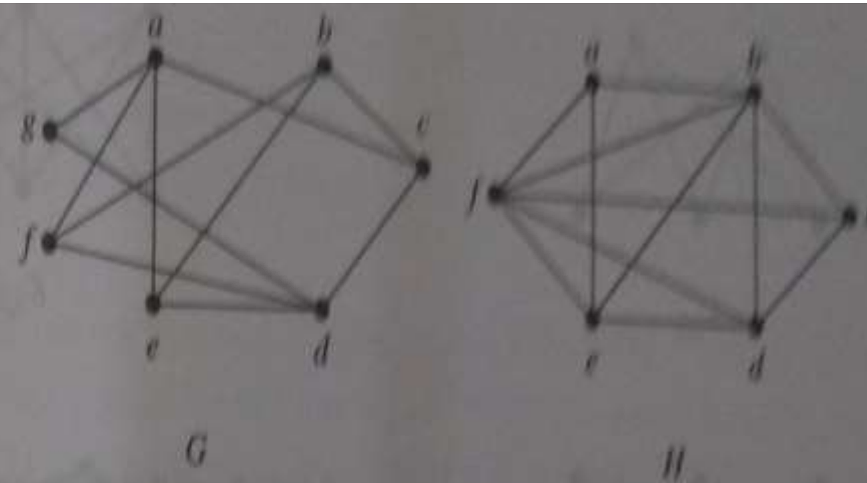


Figure 8 The Undirected Graphs G and H.

Example 11 *Are the graphs G and H displayed in Figure 8 bipartite?*

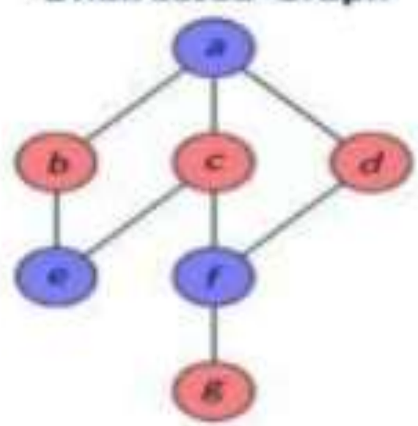
Solution Graph G is bipartite because its vertex set is the union of two disjoint sets, $\{a, b, d\}$ and $\{c, e, f, g\}$, and each edge connects a vertex in one of these subsets to a vertex in the other subset. (Note that for G to be bipartite it is not necessary that every vertex in $\{a, b, d\}$ 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.) ◀

Bipartite graphs coloring

Suppose there are two colors: blue and red. Color the first vertex blue. For each newly-discovered node, color it the opposite of the parent (i.e. red if parent is blue). If the child node has already been discovered, check if the colors are the same as the parent. If so, then the graph isn't bipartite. If the traversal completes without any conflicting colors, then the graph is bipartite.

Undirected Graph



Weighted Graph

[Jiwaji (Vth Sem.), 2012]

Definition. An ordered quadruple (V, E, f, g), or an ordered triple (V, E, f), or an ordered triple (V, E, g), where V is the set of vertices, E is the set of edges, f is a function whose domain is V, and g is a function whose domain is E is called a **weighted graph.** The function f is an assignment of weights to the vertices, and the function g is an assignment of weights to the edges. The weights can be numbers, symbols, or whatever quantities that we wish to assign to the vertices and edges. For example, in a graph that represents the outcomes of the matches in Wimbledon tennis tournament, 2004 we might wish to label each edge with the score and the date of the match between the players connected by the edge.

We present some more examples :

Example 1. Consider the problem of recognizing sentences consisting of an article, followed by at most three adjectives, followed by a noun, and then followed by a verb, as shown in the following :

1. The bus stops.
2. A little girl laughs.
3. The large fluffy (soft and downy) white clouds appear.
4. Mohan laughs.

From the weighted graph in Fig. 25 below it is clear that when we examine a sentence word by word by starting at vertex a, we can determine whether it is in this special form. If vertex g is reached, we conclude that the sentence is in the special form. In order to simplify the drawing of the graph, we use dotted arrows to indicate the discovery of words that are out of the normal order. In that case we reach vertex h, which signifies the detection of an "illegal" sentence.

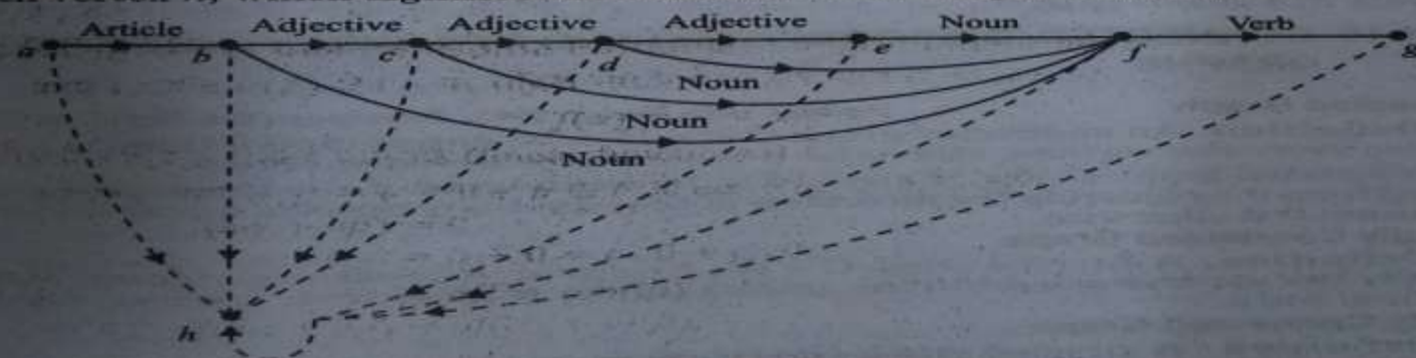
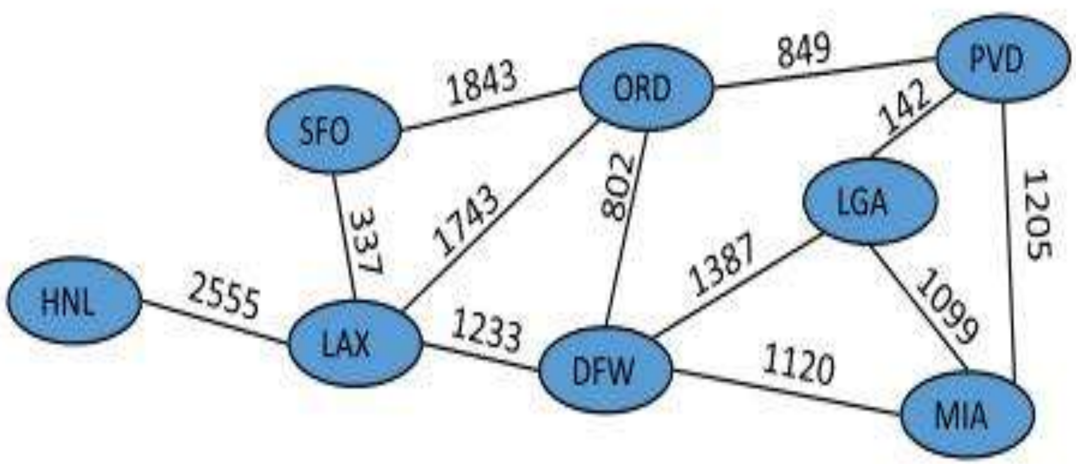


Fig. 25

Sentence	Article	1st Adjective	2nd Adjective	3rd Adjective	Noun	Verb	Classification of sentence
1.	The	—	—	—	bus	stops	legal
2.	A	little	—	—	girl	laughs	legal
3.	The	large	fluffy	white	clouds	appear	legal
4.	—	—	—	—	Mohan	laughs	illegal

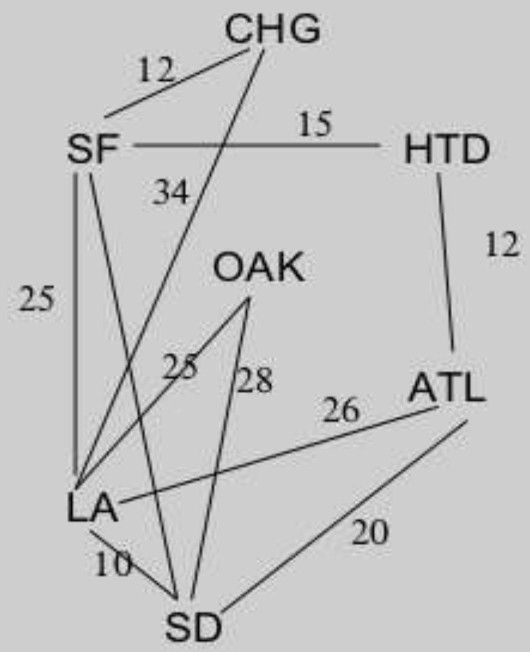
Weighted Graphs

- In a weighted graph, each edge has an associated numerical value, called the weight of the edge
- Edge weights may represent, distances, costs, etc.
- Example:
  - In a flight route graph, the weight of an edge represents the distance in miles between the endpoint airports



Weighted graphs

Example Consider the following graph, where nodes represent cities, and edges show if there is a direct flight between each pair of cities.



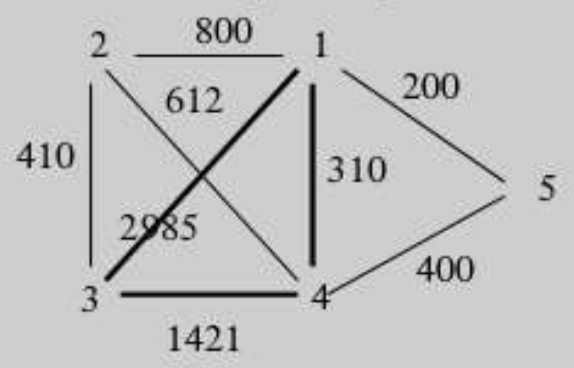
V = {SF, OAK, CHG, HTD, ATL, LA, SD}

E = {{SF, HTD}, {SF, CHG}, {SF, LA}, {SF, SD}, {SD, OAK}, {CHG, LA}, {LA, OAK}, {LA, ATL}, {LA, SD}, {ATL, HTD}, {SD, ATL}}

Problem formulation: find the "best" path between two vertices $v_1, v_2 \in V$ in graph G = (V, E). Depending on what the "best" path means, we have 2 types of problems:

- The **minimum spanning tree problem**, where the "best" path means the "lowest-cost" path.
- The **shortest path problem**, where the "best" path means the "shortest" path.

Note that here edge weights are not necessarily Euclidean distances. Example:



2985 > 1421 + 310, not the case here, however.

§ 4.9. WALK AND PATH

Walk

Definition. Let $G = (V, E)$ be a graph. Then by a **walk** we shall mean a finite sequence of edges of the form

(*) $\{(v_0, v_1), (v_1, v_2), (v_2, v_3), (v_3, v_4), \ldots, (v_{n-1}, v_n)\}$.

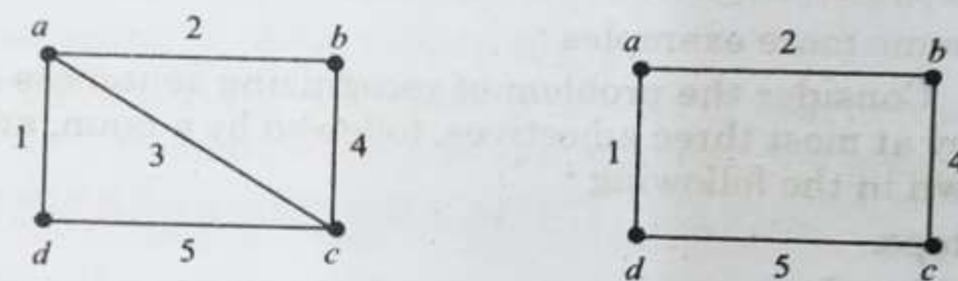
The **length** of a walk is the number of edges it contains.

Open and Closed Walks

Definition. The walk (*) is **closed** if $v_0 = v_n$; otherwise it is **open**.

A simple walk is a walk in which no edge appears more than once.

Example 1.



(a) (b)

Fig. 26

In the graph shown in Fig. 26 (a) and (b), walks 1, 2, 4, 5, 1, 3 and 1, 2, 4 both open walks from d to c, but only the graph 26 (b) is simple.

Path

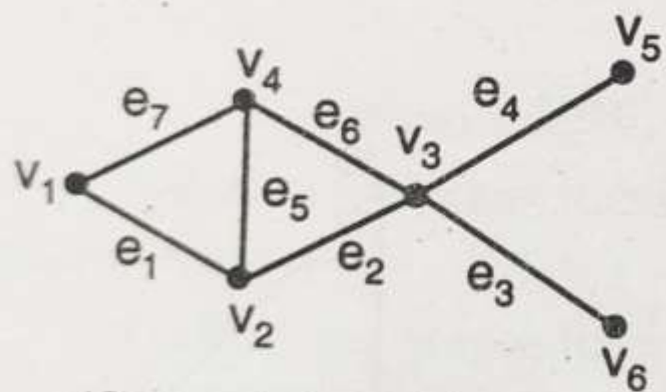
Definition. An open walk in which no vertex appears more than once is called a **path** (or a simple path or an elementary path).

Example 2. In Fig. 26 (a), the walk $d1\,a2\,b4c\,5d1\,a3c$ is not a path whereas the walk $d1a\,2b4c$ is path.

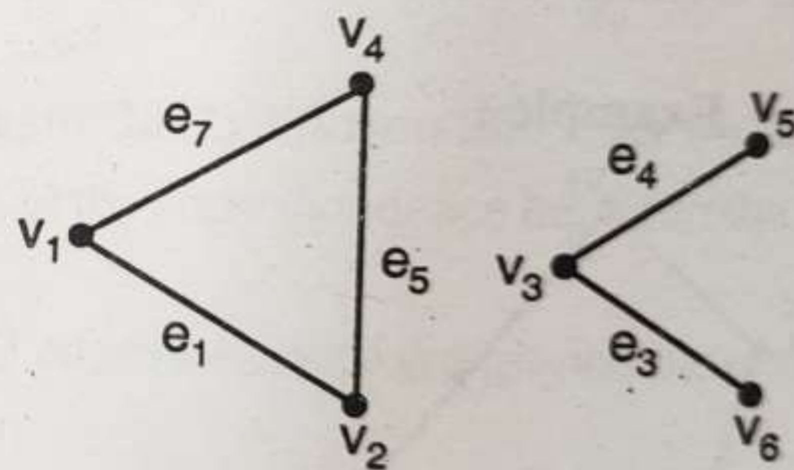
14.17. Connected and Disconnected Graphs

A graph is said to be **connected**, if there is at least one path betwe
every pair of vertices in G (or we can move from a vertex to anoth
vertex along edges). Otherwise G is **disconnected.**

Example :



(Connected graph)



(Disconnected graph)

Components or Maximal Connected subgraph

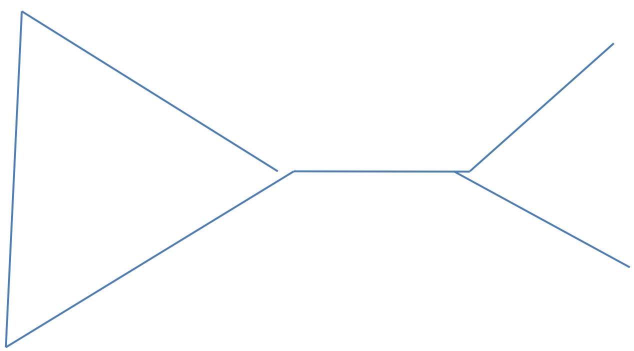
A subgraph S of graph G is said to be a component or Maximal Connected subgraph of G if the following holds

1. $x \in V(s) \Rightarrow V(s) = \{Y \in G : x \text{ and } y \text{ are connected in } G\}$.

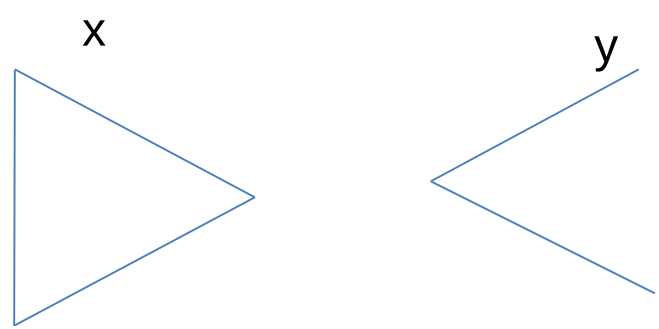
2. $S = < V(s) >$.

In other words each of the connected part of the disconnected graph is called components.

Example -



Connected graph



Components

where

Theorem 2. The number of vertices of odd degree in a graph is always even.

Proof : Let $G = G(V, E)$ be a graph such that

$$V = \{v_1, v_2, v_3, ..., v_n\} \text{ and } E = \{e_1, e_2, ..., e_m\}$$

with n vertices and m edges and we know that by **handshaking theorem.**

$$\sum_{i=1}^{n} d(v_i) = 2\,e = 2m. \qquad ...(1)$$

We can write the above equation such as

$$\sum_{i=1}^{n} d(v_i) = \sum_{i=even}^{n} d(v_i) + \sum_{i=odd}^{n} d(v_i)$$

$$\sum_{i=odd}^{n} d(v_i) = \sum_{i=1}^{n} d(v_i) - \sum_{i=even}^{n} d(v_i)$$

$= 2m$ – Even degree. [from (1)]

= Even degree – Even degree.

Where $2m$ is always even when m is an odd or even.

= Even degree.

[The subtraction of two even number is always even]

Hence the number of vertices of odd degree in a graph is always even. **Proved.**

Theorem 3. Show that the maximum number of edges in a simple graph with n vertices is $\frac{n(n-1)}{2}$.

Proof : Let $G = G(V, E)$ be a simple graph (neither parallel edges nor self loop) such that

$$V = \{v_1, v_2, \ldots, v_n\} \text{ and } E = \{e_1, e_2, \ldots, e_m\}$$

with n-vertices and e edges and we know that by handshaking theorem,

$$\sum_{i=1}^{n} d(v_i) = 2e$$

$$\Rightarrow \qquad d(v_1) + d(v_2) + \ldots + d(v_n) = 2e \qquad \ldots(1)$$

and we also know that the maximum degree of any vertex in a simple graph with n vertex is

$$= n - 1$$

Then from equation (1), we get

$$(n-1) + (n-1) + \ldots + n \text{ times} = 2e$$

$$\Rightarrow \qquad n \cdot (n-1) = 2e$$

$$\Rightarrow \qquad e = \frac{n(n-1)}{2}.$$

Hence Proved.

Theorem 4. Prove that a simple graph with a n-vertices must be connected if it has more than $\frac{(n-1)(n-2)}{2}$.

Proof : Consider a simple graph on n vertices. Choose $n-1$ vertices $v_1, v_2, \ldots, v_{n-1}$ of G. We have more than $\frac{(n-1)(n-2)}{2}$ number of edges only can be drawn between these vertices. Thus, if we have more than $\frac{(n-1)(n-2)}{2}$ edges atleast one edge should be drawn between the nth vertices v_n to some vertex v_i, $1 \le i \le n-1$ of G. Hence G must be connected.

Theorem 5. A simple graph with n-vertices and k-components can have at most $\frac{(n-k)(n-k+1)}{2}$ edges.

Proof : Let $G = (V, E)$ be a simple graph with n-vertices and k components, each of the components has $n_1, n_2, \ldots, n_k$ as number of vertices.

So that $\quad n_1 + n_2 + \ldots + n_k = n$

$$\Rightarrow \qquad \sum_{i=1}^{k} n_i = n \qquad \ldots(1)$$

We know that

$$\sum_{i=1}^{k} (n_i - 1) = n - k \qquad \left[\begin{array}{c} \because \sum_{i=1}^{k} 1 = k \\ \text{and from eqn. (1)} \end{array}\right]$$

Squaring both sides,

$$\left[\sum_{i=1}^{k} (n_i - 1)\right]^2 = (n-k)^2$$

$$\Rightarrow \quad \left[\sum_{i=1}^{k} (n_i - 1)\right]^2 = n^2 + k^2 - 2nk.$$

$$\Rightarrow \quad \sum_{i=1}^{k} (n_i - 1)^2 + 2 \text{ (some positive term)} = n^2 + k^2 - 2nk$$

$$\Rightarrow \quad \sum_{i=1}^{k} (n_i^2 + 1 - 2n_i) \le n^2 + k^2 - 2nk. \left[\begin{array}{l} \text{By the property of} \\ \text{summation} \end{array}\right]$$

$$\Rightarrow \quad \sum_{i=1}^{k} n_i^2 + \sum_{i=1}^{k} 1 - 2 \sum_{i=1}^{k} n_i \le n^2 + k^2 - 2nk$$

$$\Rightarrow \quad \sum_{i=1}^{k} n_i^2 + k - 2n \le n^2 + k^2 - 2nk \qquad \text{[from eqn. (1)]}$$

$$\Rightarrow \quad \sum_{i=1}^{k} n_i^2 \le n^2 + k^2 - 2nk - k + 2n. \qquad \ldots(2)$$

We know that the maximum number of edges in the i^{th} component of G which is a simple graph is $\frac{1}{2} n_i (n_i - 1)$.

Thus, total numbe of edges

$$= \frac{1}{2} \sum_{i=1}^{k} n_i (n_i - 1)$$

$$= \frac{1}{2} \left[\sum_{i=1}^{k} n_i^2 - n_i\right]$$

$$= \frac{1}{2} [n^2 + k^2 - 2nk - k + 2n - n] \qquad \text{[from eqns. (2) and (1)]}$$

$$= \frac{1}{2} [(n-k)^2 + (n-k)]$$

$$= \frac{(n-k)}{2} (n-k+1).$$

Proved.

Applications of Graphs

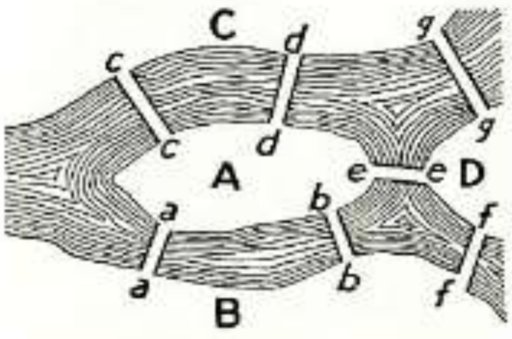
- Since they are powerful abstractions, graphs can be very important in modeling data. In fact,many problems can be reduced to known graph problems. Here we outline just some of the many applications of graphs.1.Social network graphs: to tweet or not to tweet. Graphs that represent who knows whom, who communicates with whom, who influences whom or other relationships in

- Social structures. An example is the twitter graph of who follows whom. These can be used to determine how information flows, how topics become hot, how communities develop, or even who might be a good match for who, or is that whom.

The Königsberg bridge problem

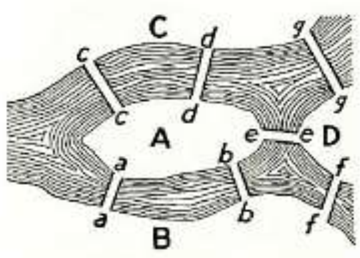
The town of Königsberg, Prussia (now a city in Russia called Kaliningrad) is built on the both banks of the river Preger as well as on an island in the river. At one time there were seven bridges linking one bank to the other as well as both banks to the island. The people in the town wondered if were possible to start at some point in the town, walk about the town crossing each bridge exactly once, and end up back at the starting point.

The Königsberg bridge problem



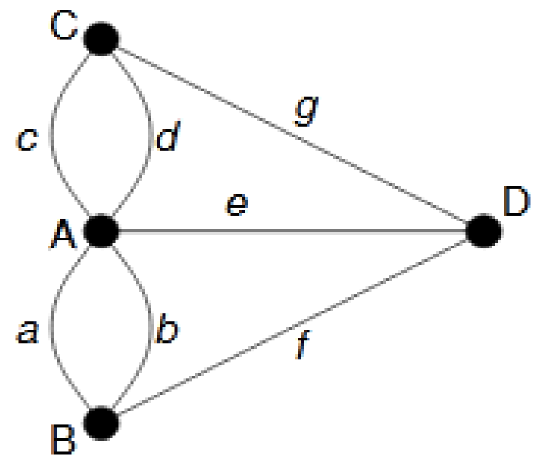
- A, B, C, and D label regions of land (A and D are islands).
- Bridges are labeled with a, b, c, etc.
- Try to find a route, starting anywhere you want, that crosses every bridge once and end up back where you started.

The Königsberg bridge problem



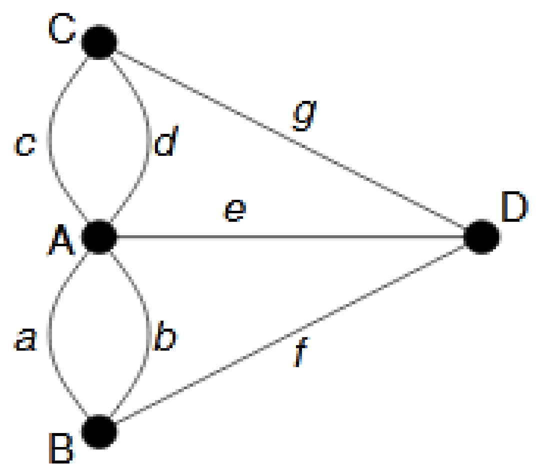
- A, B, C, and D label regions of land (A and D are islands).
- Bridges are labeled with a, b, c, etc.
- Try to find a route, starting anywhere you want, that crosses every bridge once and end up back where you started.

Königsberg bridge graph



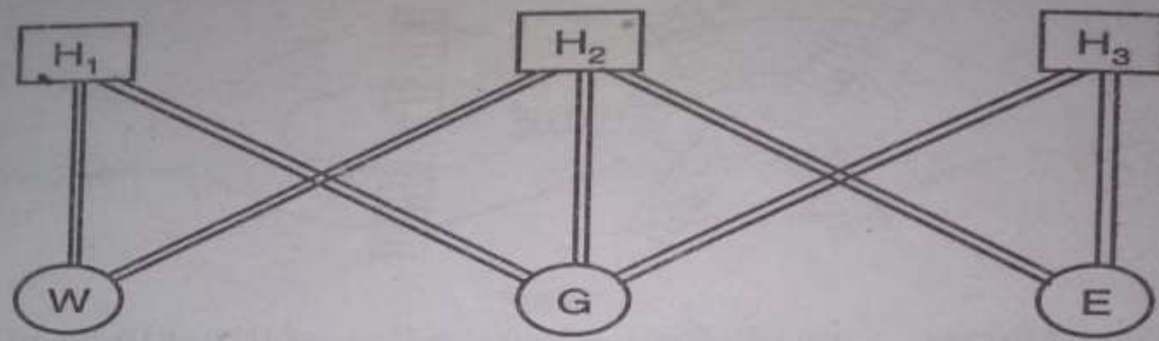
- Consider starting at A then visiting B, D, and C, before returning to A. This does not use each edge, but does visit each vertex.
- We could show this with the walk AaBfDgCcA.

Königsberg bridge graph



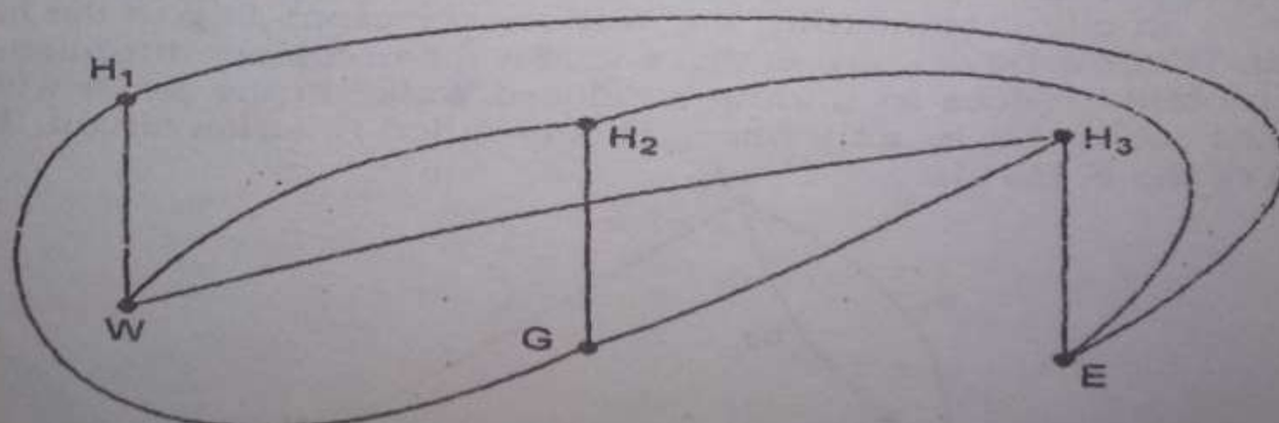
- Notice what we've established: if there is a vertex with an odd number of edges attached to it, we will be prevented from finding a route that uses all edges once and returns to the starting point.
- If every vertex has an even number of edges attached to it, then there is always an *entry-exit* pair so we can find a route.

conduits. The problem is, : Is it possible to make such connections without any crossovers of the conduits?



(Three utilities problem)

In the adjoining figure, we have shown that how this problem can be represented by a graph – the conduits are shown as edge while the houses and utility supply centres are vertices. This graph cannoe bd drawn in the plane without edges crossing over. Thus the answer to this problem is no.



(Graph of Three-utilities problem)

ithout edges ...

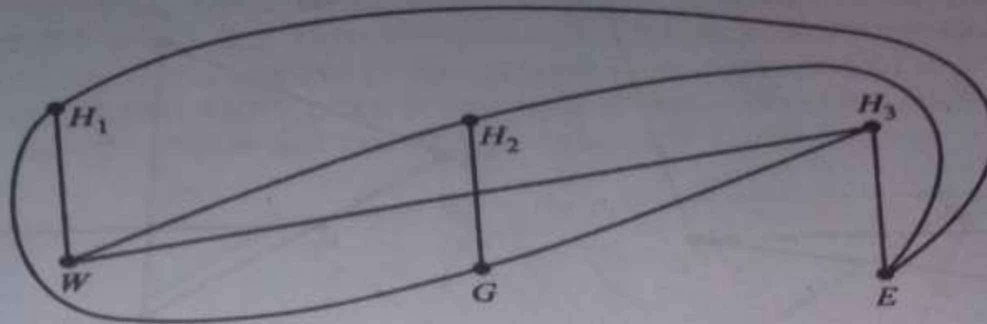


Fig. 33. Graph of Three-utilities Problem

3. Planarity Problem

Much research has been done on graphs which are **planar,** that is, which can be drawn in the Euclidean plane without intersecting edges. The graph in Fig. 34 (a) is planar, since it can be redraw on Fig. 34 (b).

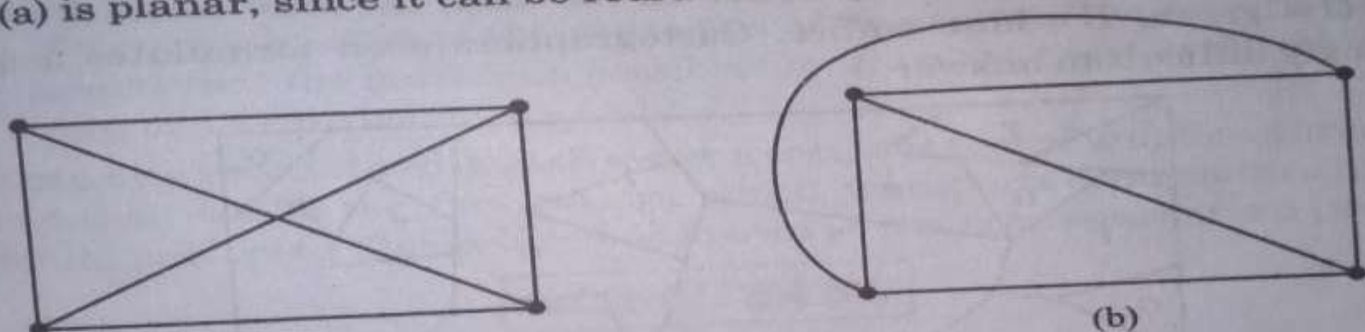


Fig. 34. Planar Graph

Two graphs are said to be **homeomorphic** if one can be obtained from the other by operations of the following kind :

(a) Replace an edge $\{v_i, v_j\}$ by a new vertex v_x and two edges $\{v_i, v_x\}$ and $\{v_i, v_j\}$, so that

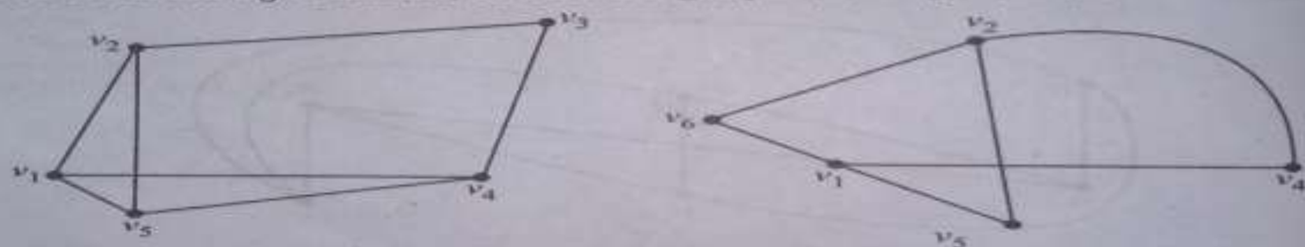
(a) (b)

(b) For a vertex v_k of degree 2, replace v_k and edges $\{v_i, v_k\}$ and $\{v_k, v_j\}$ by a single edge $\{v_i, v_j\}$, so that

(c) (d)

Fig. 35

The graphs in Fig. 36 are homeomorphic; the degree 2 vertex v_3 is deleted and the degree 2 vertex v_6 is added, to convert graph (a) to graph (b).



(a) (b)

Fig. 36. Homeomorphic Graphs

4. Four-Color Problems

In cartograph, a map is normally coloured in such a way that no countries with common boundary have the same color. In the map, the four colors R = red, Y = Yellow, G = green, B = blue suffice. Cartographers soon formulates a question surprisingly difficult to answer.

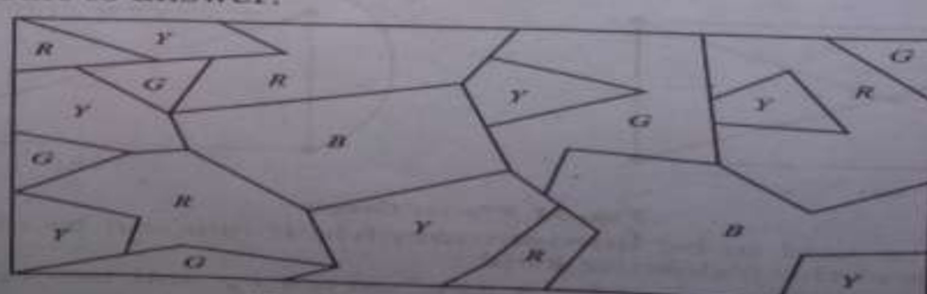
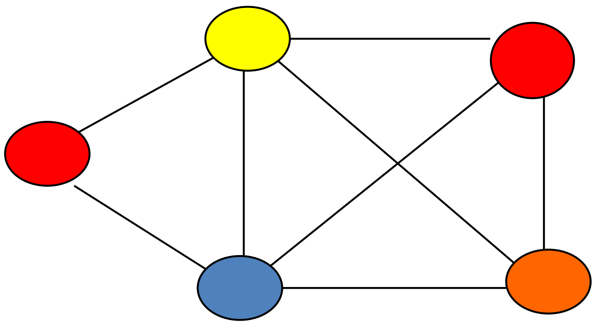
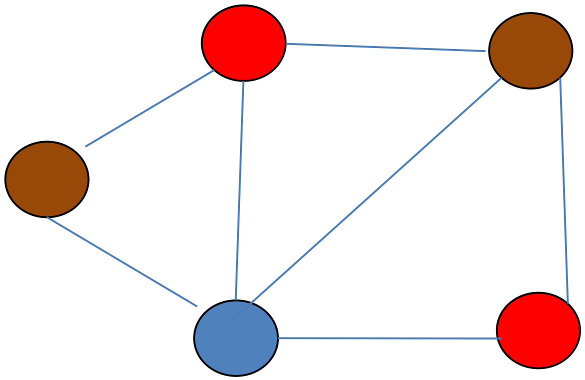


Fig. 37. Coloring of a Map

- Graph Coloring is an assignment of colors (or any distinct marks) to the vertices of a graph. Strictly speaking, a coloring is a proper coloring if no two adjacent vertices have the same color.



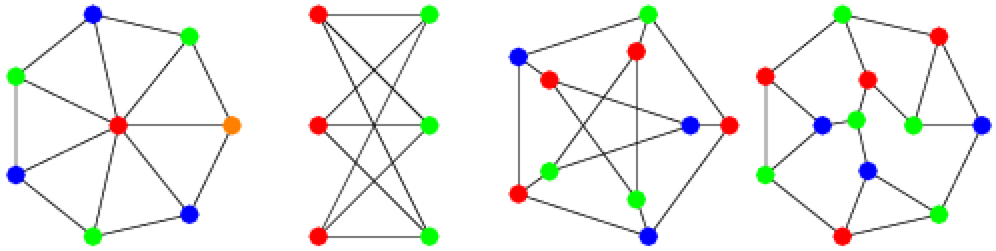
- **Definition**: A graph is planar if it can be drawn in a plane without edge-crossings.



- **The four color theorem:** For every planar graph, the chromatic number is ≤ 4.

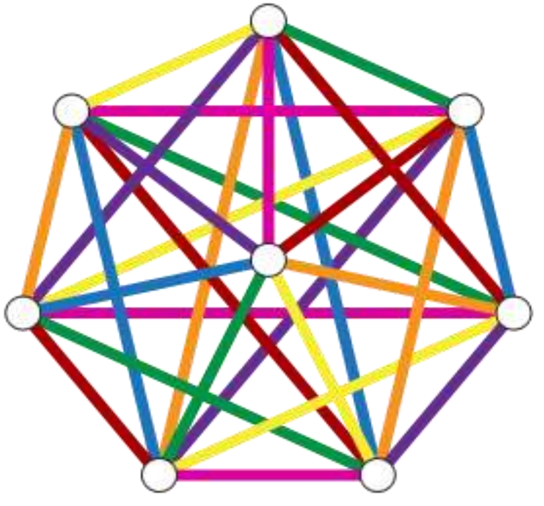
Vertex Coloring

- A **vertex coloring** is an assignment of labels or **colors** to each **vertex** of a **graph** such that no edge connects two identically colored **vertices**



Edge Coloring

- Similar to vertex coloring, except edges are color.
- Adjacent edges have different colors.



Applications of Graph Coloring

- Many problems can be formulated as a graph coloring problem including Time Tabling, Scheduling, Register Allocation, Channel Assignment.
- A lot of research has been done in this area so much is already known about the problem space.


India
States & Union Territories
Arabian Sea
Bay of Bengal

Explanation

- The standard approach to coloring a map is to use a single color for a state and never use the same color for two states.
- Two states whose common border is just one point can be colored, if we so choose, with the same color.

Travelling Salesperson Problem

Travelling salesman route will be plan in such a way that in a given N number of cities cost of travelling from one city to any other city what is the minimum round trip route that visit each city once and then return to the starting place. The goal is to find the shortest tour that visit each city in a given cities exactly ones and then return to the starting city. The only solution to the travelling salesman problem is to calculate and compare the length of all possible ordered combinations.

- Suppose a salesman wants to visit a certain number of cities allotted to him. He knows the distance of the journey between every pair of cities. His problem is to select a route the starts from his home city, passes through each city exactly once and return to his home city the shortest possible distance. If we represent the cities

- by vertices and road connecting two cities edges we get a weighted graph where, with every edge e_i a number w_i(weight) is associated.
- A physical interpretation of the above abstract is: consider a graph G as a map of n cities where w (i, j) is the distance between cities i and j. A salesman wants to have the tour of the cities which starts and ends at the same city includes visiting each of the remaining a cities once and only once.

6. Transportation Problem

In the adjoining Fig. 38 a weighted directed graph is shown whose vertex s is called a sourse and t a sink. The weights of the arrows are to be interpreted as capacities, the weight of an arrow is the maximum amount of some commodity that can be transported along that arrow in unit time. We can ask for the maximum amount that can be transported from s to t in unit time.

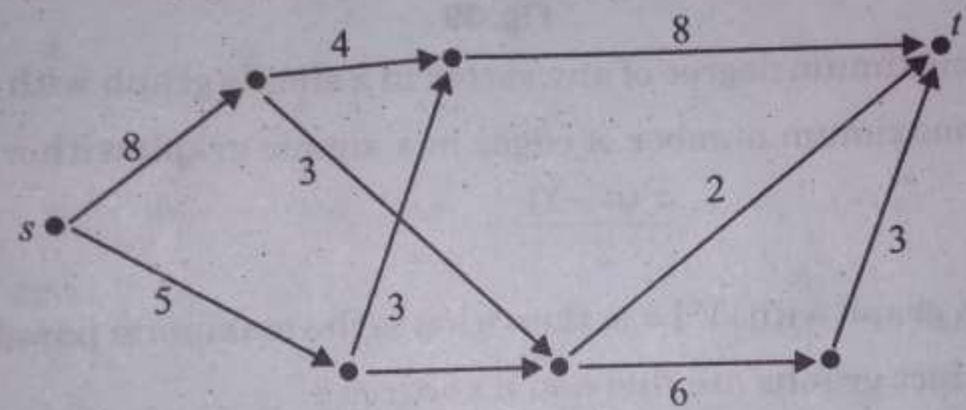


Fig. 38. Max-Flow = Min-Cut

A **cut** in this graph is a cut-set that leaves s and t in different components. The capacity of a cut is the sum of the capacities of its arrows. The max-flow min-cut theorem asserts that the maximum possible flow from s to t equals the minimum of the capacities of all cuts between s and t.

We can also associate with each arrow a cost, the cost of unit flow through that arrow and then ask for the flow pattern which transports commodities from s to t at minimum unit cost. This problem is known as the **transportation problem.**

Paths and Circuits, Shortest Paths, Eulerian Paths and Circuit

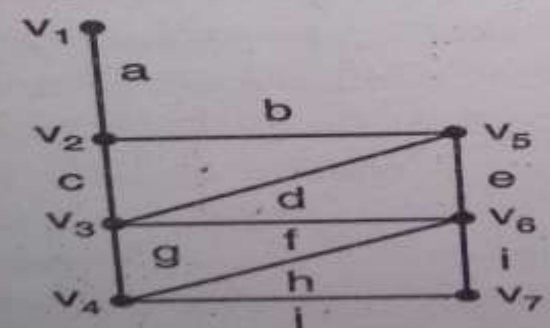
15.1. Introduction

In this chapter we will study about an important application of Graph theory, which is shortest path of a weighted graph. First we study some basic definations such as walk, path, circuit, then we study about shortest path.

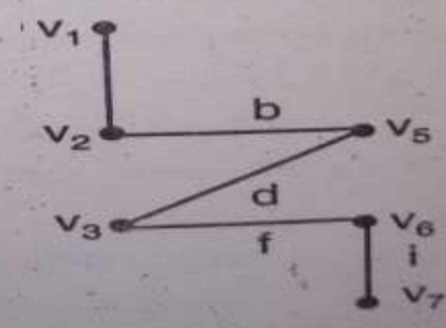
15.2. Walk

Walk in a graph G is a finite alternatively sequence of vertice and edges, beginning and ending with same or different vertices, such that no edge can appears more than once, however a vertex may be. Walk can also be referred as **edge chain.**

For example,



(i)



(ii)

Properties of Walk

1. Every walk of a graph G is subgraph of G.
2. A self loop may be a part of the walk.
3. A graph has more than once walk.

15.3. Types of Walk

There are two types of walk.

(1) Open Walk

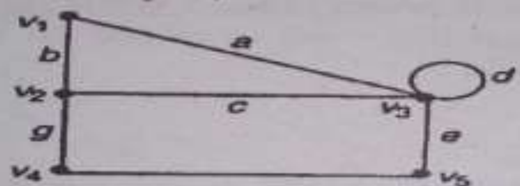
A walk in which starts and ends vertices are different (different terminal vertices) called open walk.

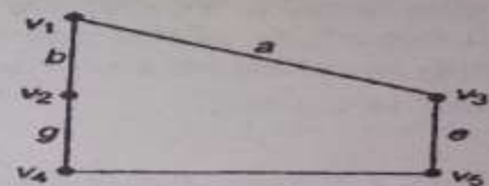
The above example is an open walk, because it starts with v_1 and ends on v_7.

(2) Closed Walk

A walk in which starts and ends vertices are same (same terminal vertices), called closed walk.

For example,



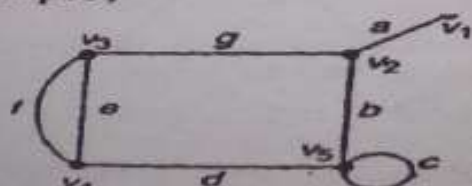


Note : A walk which is not closed is calle an open walk.

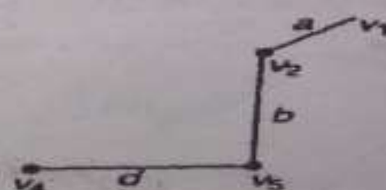
15.4. Path

An open walk in which no vertex appears more than once, is called a path. It can not intersect itself, and a self loop can not be a part of the path. The number of edges in a path is called **length of the path.** An edge which is not a self loop is a path of length one.

For example,



(Graph)



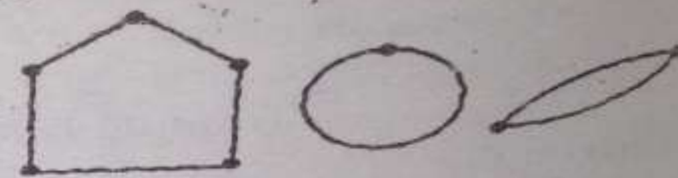
(Path)

Properties of Path

1. In a path terminal vertices are of degree one.
2. Rest of the intermediate vertices are of degree two only.
3. Length of the path is number of edges which contribute to form a path.

15.5. Circuit or Cycle

A closed walk in which no vertex appears more than once, is called a circuit. It is a non-intersecting walk, such that every vertex is of degree two. A self loop is also a circiut. A circuit is also referred as **circular path.**

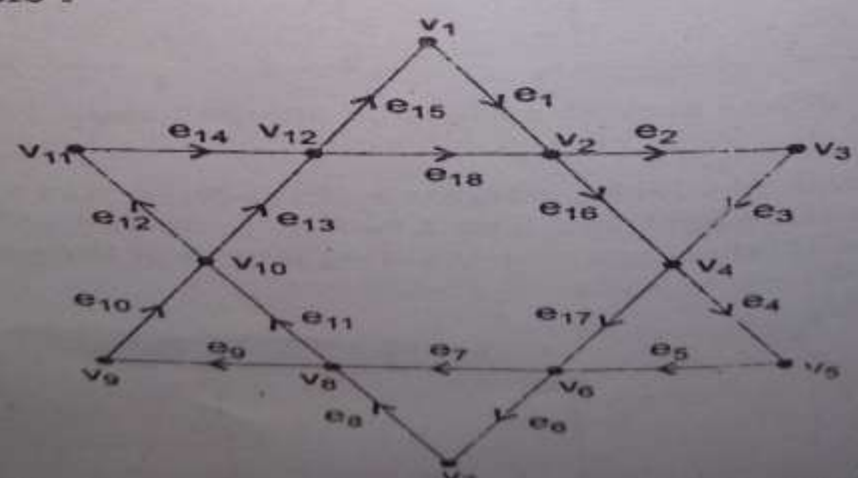


15.6. Eulerian Graphs

If we are moving on a graph, by **covering all the edges** of the graph and returning to the initial vertex, such a walk is called **Euler line** and graph is called **Euler graph** or **Eulerian graph.**

In other words, we can say that if some closed walk in a graph G contains all the edges of the graph then the walk is called **an Euler line** and the graph is called an **Euler graph.**

Example :



If we starting from the vertex v_1 and move such that

$v_1\, e_1\, v_2\, e_2\, v_3\, e_3\, v_4\, e_4\, v_5\, e_5\, v_6\, e_6\, v_7\, e_8\, v_8\, e_9\, v_9\, e_{10}\, v_{10}\, e_{12}\, v_{11}\, e_{14}\, v_{12}$
$e_{18}\, v_2\, e_{16}\, v_4\, e_{17}\, v_6\, e_7\, v_8\, e_{11}\, v_{10}\, e_{13}\, v_{12}\, e_{15}\, v_1.$

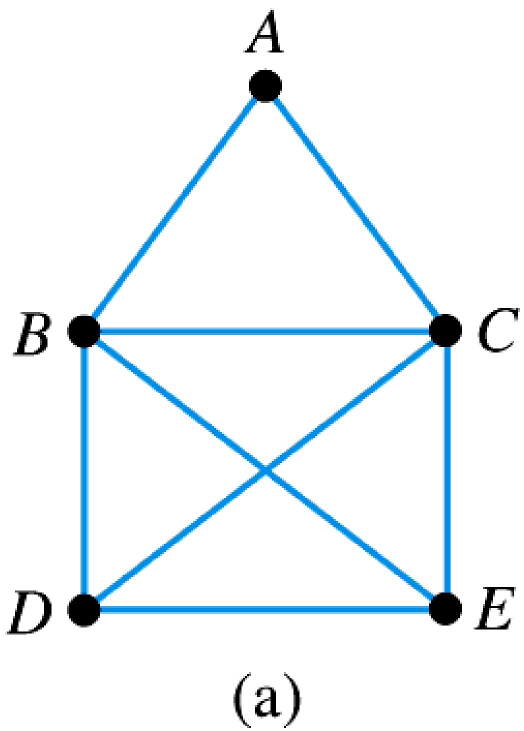
Clearly, this is a Euler graph which contains all the edges (without repetation).

Definitions

- An **Euler path** is a path that passes through each edge of a graph exactly one time.
- An **Euler circuit** is a circuit that passes through each edge of a graph exactly one time.
- The difference between an Euler path and an Euler circuit is that an Euler circuit must start and end at the same vertex.

Examples

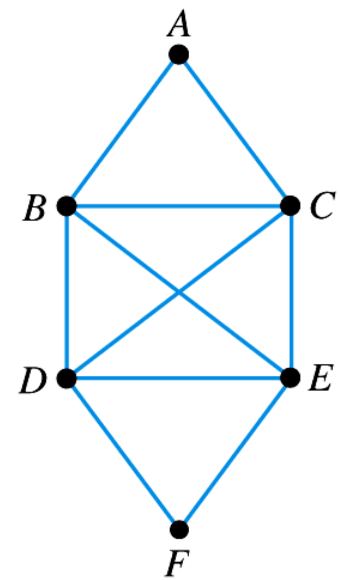
Euler path



(a)

An Euler path
$D, E, B, C, A, B, D, C, E$

Euler circuit



(b)

An Euler circuit
$D, E, B, C, A, B, D, C, E, F, D$

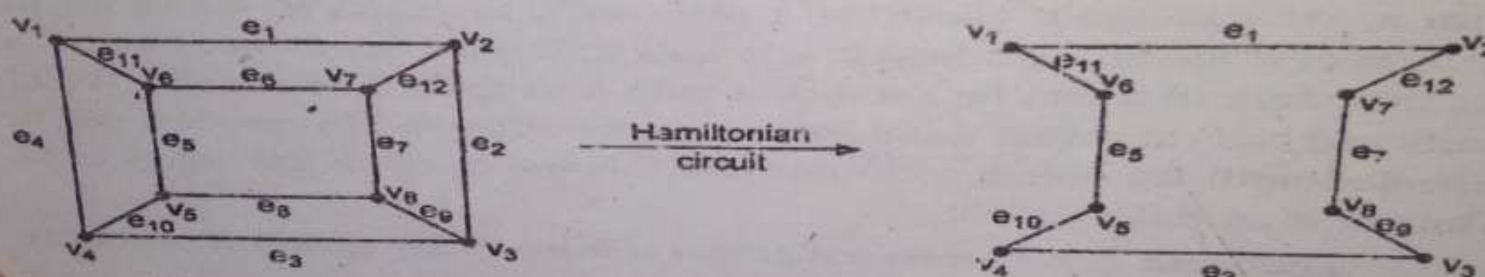
Properties of Euler Graph

1. Euler graph is always connected.
2. Euler graph does not contain any isolated or pendent vertices.
3. All the vertices of an Euler graph are of even degree.
4. Euler graph can be decomposed into circuit.

15.7. Hamiltonian Path and Circuit

If a closed walk contain every vertex of the graph G, such that the degree of every vertex is 2, then the walk is called **Hamiltonian circuit,** and if the walk is open then it is said to be **Hamiltonian path.**

Example :



Propeties of Hamiltonian Path and Circuit

1. Hamiltonian circuit have exactly the number of edges as the number of vertices.
2. Hamiltonian circuit is always in a connected graph, but every connected graph does not have a Hamiltonian circuit.
3. If we removen an edge from Hamiltonian circuit, we are left with a path and this path is hamiltonian path.

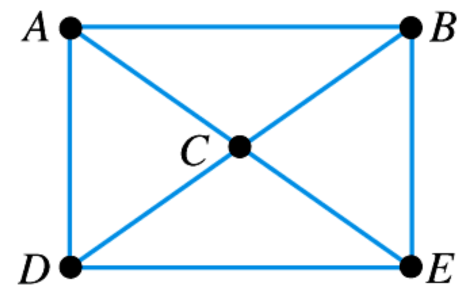
15.8. Some Theorems

Hamilton Paths and Hamilton Circuits

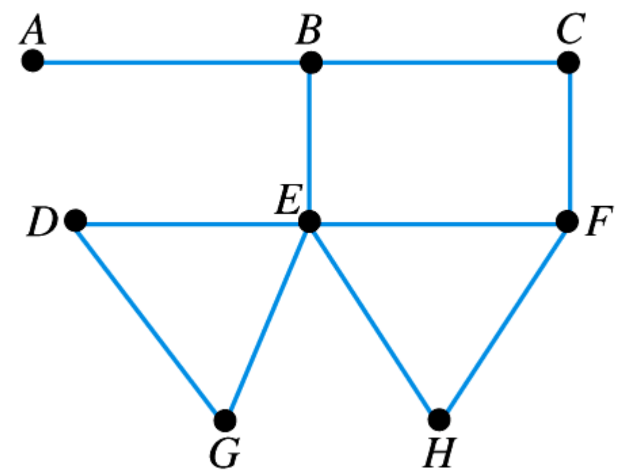
- A **Hamilton path** is a path that contains each *vertex* of a graph exactly once.
- A **Hamilton circuit** is a path that begins and ends at the same vertex and passes through all other vertices of the graph exactly one time.

- Graph (a) shown has Hamilton path *A, B, C, E, D.* The graph also has Hamilton path *C, B, A, D, E*. Can you find some others?

- Graph (b) shown has Hamilton path *A, B, C, F, H, E, G, D.* The graph also has Hamilton path *G, D, E, H, F, C, B, A*. Can you find some others?



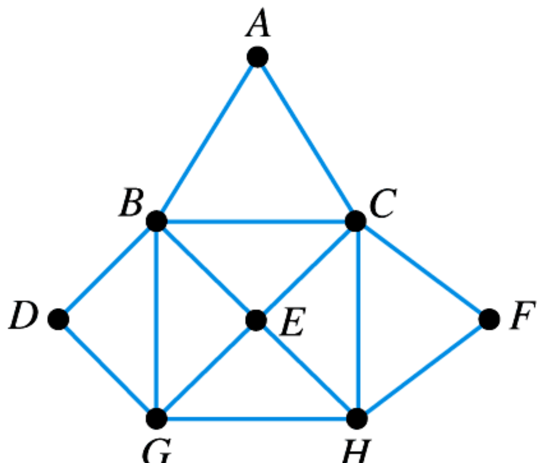
(a)



(b)

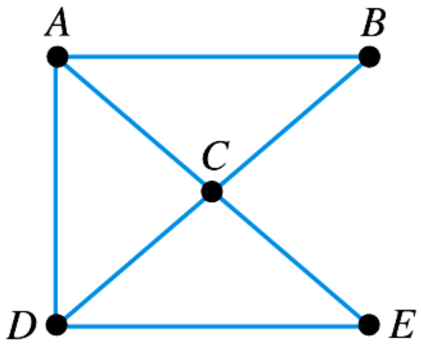
Example: Hamilton Circuit

- Graph (a) shown has Hamilton circuit *A, B, D, G, E, H, F, C, A.*
- A Hamilton circuit starts and ends at the same vertex.



(a)

- Graph (b) shown has Hamilton circuit *A, B, C, E, D, A.*
- Can you find another



(b)

Number of Unique Hamilton Circuits in a Complete Graph

- The number of unique Hamilton circuits in a complete graph with *n* vertices is $(n - 1)!$ where

$$(n-1)! = (n-1)(n-2)(n-3)\ldots(3)(2)(1)$$

Example: Number of Hamilton Circuits

- How many unique Hamilton circuits are there in a complete graph with the following number of vertices?
- a) 4 b) 9

- a) 4 = (4 – 1)! = •3•2•1 = 6
- b) 9 = (9 – 1)! = 8•7•6•5•4•3•2•1 = 40,320

Example 3. *Draw a graph with six vertices containing a Hamiltonian circuit but not an Eulerian circuit. Justify your answer.*

[Indore (Vth Sem.), 2012; Indore (VIth Sem.), 2018]

Solution. The required graph with six vertices may be drawn as in Fig. 19.

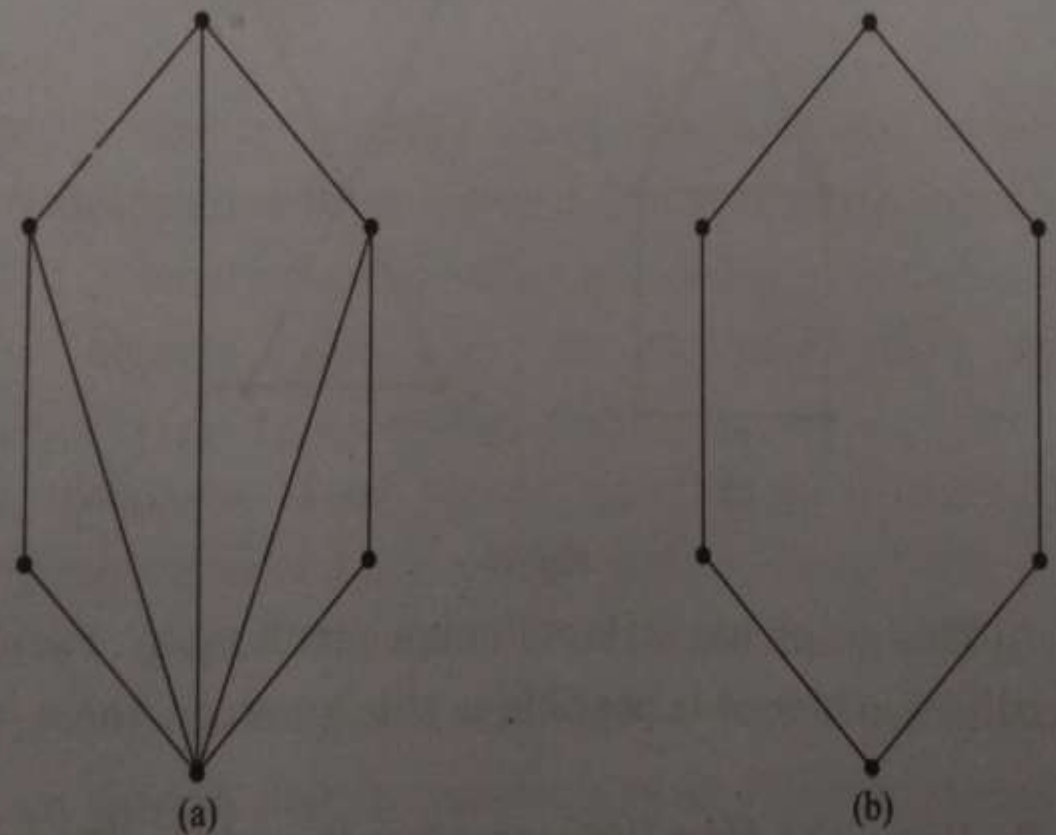


Fig. 19

The graph in the fig has a Hamiltonian circuit, but it does not have an Eulerian circuit since each vertex in the graph is not of even degree.

Example 4. *Draw a graph with six vertices containing an Eulerian circuit but not Hamiltonian circuit.*

Solution. The required graph is as shown in Fig. 20.

Fig. 20

Example 5. *Show that the following graphs have Hamiltonian circuits*

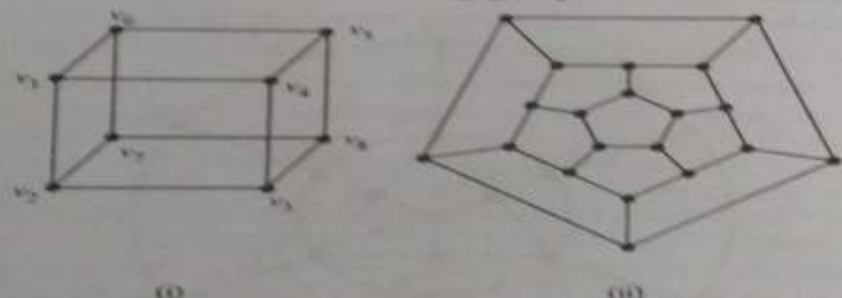


Fig. 21

Solution. (i) This graph has a Hamiltonian circuit $v_1, v_2, v_3, v_8, v_7, v_6, v_5, v_4, v_1$ as shown by dark line in Fig. below :

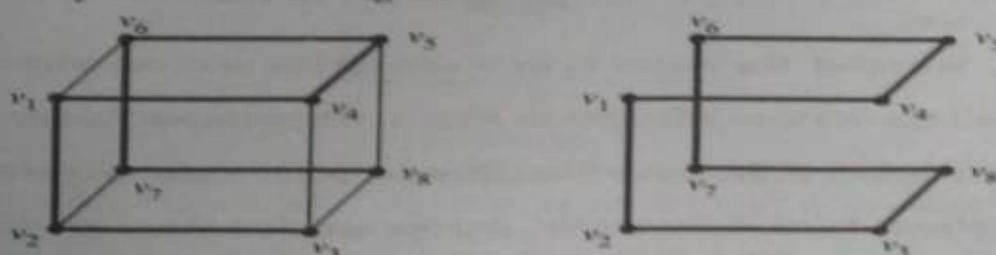


Fig. 22. Graph and its Hamiltonian Circuit

Dijkstra Prcedure for Shortest path

- Dijkstra’s algorithm is unique for many reasons, which we’ll soon see as we start to understand how it works. But the one that has always come as a slight surprise is the fact that this algorithm isn’t just used to find the shortest path between two specific nodes in a graph data structure. ***Dijkstra’s algorithm*** can be used to determine the shortest path from one node in a graph to *every other node* within the same graph data structure, provided that the nodes are reachable from the starting node.

Rules for running Dijkstra's algorithm:

1/ From the starting node, visit the vertex with the smallest known distance/cost.

2/ Once we've moved to the smallest-cost vertex, check each of its neighboring nodes.

3/ Calculate the distance/cost for the neighboring nodes by summing the cost of the edges leading from the start vertex.

4/ If the distance/cost to a vertex we are checking is less than a known distance, update the shortest distance for that vertex.

Example 2. ***Find the shortest path from a to z in the graph shown in Fig. 4 where numbers associated with the edges are the weights.***

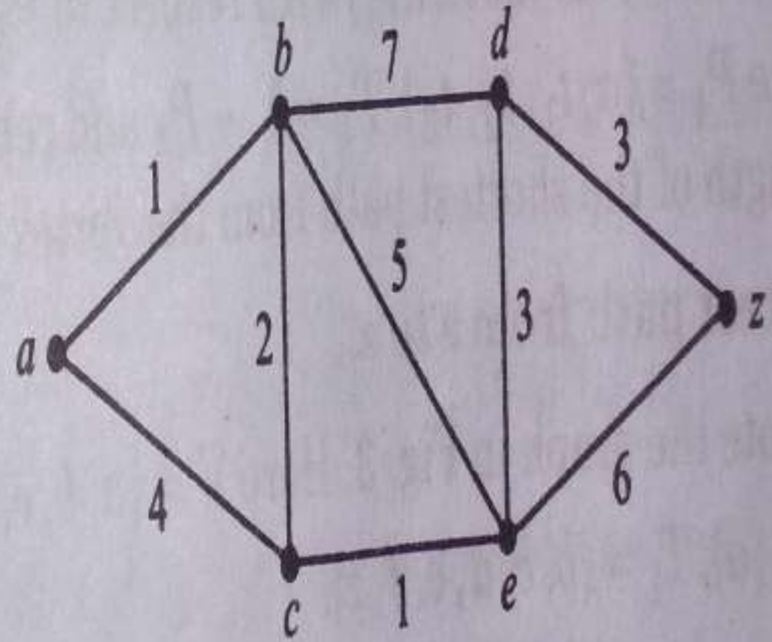


Fig. 4

Solution. Let $G = (V, E)$ denote the graph in Fig. 4.

Here $$V = \{a, b, c, d, e, z\}.$$

Step I. Taking $P_1 = \{a\}, T_1 = \{b, c, d, e, z\}$. Now $l(b) = 1$, $l(c) = 4$, $l(d) = \infty$, $l(e) = \infty$, $l(z) = \infty$.

Thus $b \in T_1$ has the minimum index 1.

Step II. Taking $P_2 = \{a, b\}, T_2 = \{c, d, e, z\}$

$$l(c) = \min. (4, 1 + 2) = 3$$
$$l(d) = \min. (\infty, 1 + 7) = 8$$
$$l(e) = \min. (\infty, 1 + 5) = 6$$
$$l(z) = \min. (\infty, 1 + \infty) = \infty.$$

Thus $c \in T_2$ has the minimum index 3.

Step III. Taking $P_3 = \{a, b, c\}, T_3 = \{d, e, z\}$

$$l(d) = \min. (8, 3 + \infty) = 8$$
$$l(e) = \min. (6, 3 + 1) = 4$$
$$l(z) = \min. (\infty, 3 + \infty) = \infty.$$

Thus $e \in T_3$ has the minimum index 4.

Step IV. Taking $P_4 = \{a, b, c, e\}, T_4 = \{d, z\}$

$$l(d) = \min. (8, 4 + 3) = 7$$
$$l(z) = \min. (\infty, 4 + 6) = 10.$$

Thus $d \in T_4$ has the minimum index 7.

Step V. Taking $P_5 = \{a, b, c, e, d\}, T_5 = \{z\}$

$$l(z) = \min. (10, 7 + 3) = 10.$$

Hence the length of the shortest path from a to z is 10. The shortest path in this example is $a \to b \to c \to e \to d \to z$.

Example 3. ***Find the shortest path between a and z for the graph in Fig. 5, where the numbers associated with the edge are the distances between vertices :***

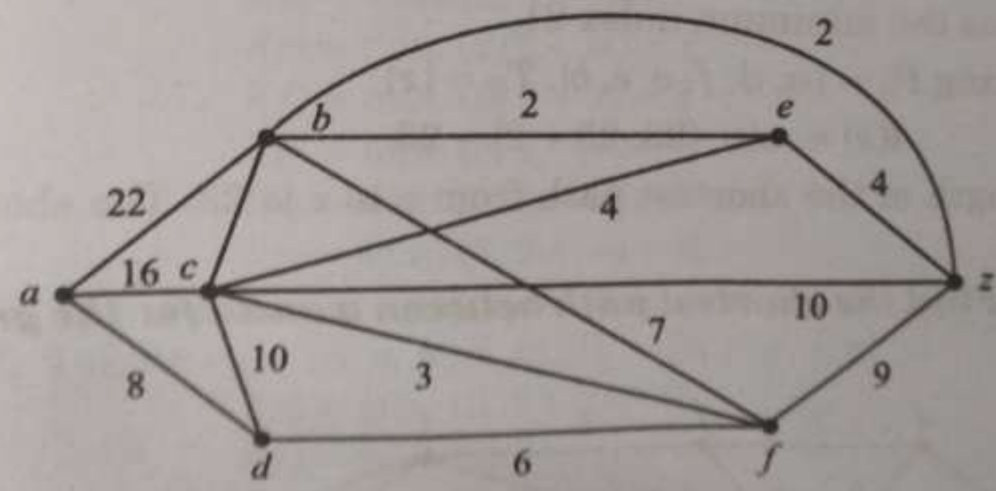


Fig. 5

[Sagar (Vth Sem.), 2013]

[Sagar (Vth Sem.), 2

(If some data is missing you may assume that)

Solution. Let $G = (V, E)$ denote the graph in Fig. 5.

Here $$V = \{a, b, c, d, e, f, z\}.$$

Step I. Taking $P_1 = \{a\}, T_1 = \{b, c, d, e, f, z\}$

$$l(b) = 22, l(c) = 16, l(d) = 8$$
$$l(e) = \infty, l(f) = \infty, l(z) = \infty.$$

Thus $d \in T_1$ has the minimum index 8.

Step II. Taking $P_2 = \{a, d\}, T_2 = \{b, c, e, f, z\}$

$$l(b) = \min.(22, 8 + \infty) = 22$$
$$l(c) = \min.(16, 8 + 10) = 16$$
$$l(e) = \min.(\infty, 8 + \infty) = \infty$$
$$l(f) = \min.(\infty, 8 + 6) = 14$$
$$l(z) = \min.(\infty, 8 + \infty) = \infty.$$

Thus $f \in T_2$ has the minimum index 14.

Step III. Taking $P_3 = \{a, d, f\}, T_3 = \{b, c, e, z\}$

$$l(b) = \min.(22, 14 + 7) = 21$$
$$l(c) = \min.(16, 14 + 3) = 16$$
$$l(e) = \min.(\infty, 14 + \infty) = \infty$$
$$l(z) = \min.(\infty, 14 + 9) = 23.$$

Thus $c \in T_3$ has the minimum index 16.

Step IV. Taking $P_4 = \{a, d, f, c\}, T_4 = \{b, e, z\}$

$$l(b) = \min.(21, 17 + 20) = 21$$

[Taking edge $cb = 20$, note that some other number may also be assumed as length of edge cb]

$$l(e) = \min.(\infty, 17 + 4) = 21$$
$$l(z) = \min.(23, 17 + 10) = 23.$$

Thus $e \in T_4$ has the minimum index 21.

Step V. Taking $P_5 = \{a, d, f, c, e\}, T_5 = \{b, z\}$

$$l(b) = \min.(21, 21 + 2) = 21$$
$$l(z) = \min.(23, 21 + 4) = 23.$$

Thus $b \in T_5$ has the minimum index 21.

Step VI. Taking $P_6 = \{a, d, f, c, e, b\}, T_6 = \{z\}$

$$l(z) = \min.(23, 23 + 2) = 23.$$

Hence the length of the shortest path from a to z is 23. The shortest path is $a \to d \to f \to z$.

Example 6. ***Find the shortest path between a to z of the following graph :***

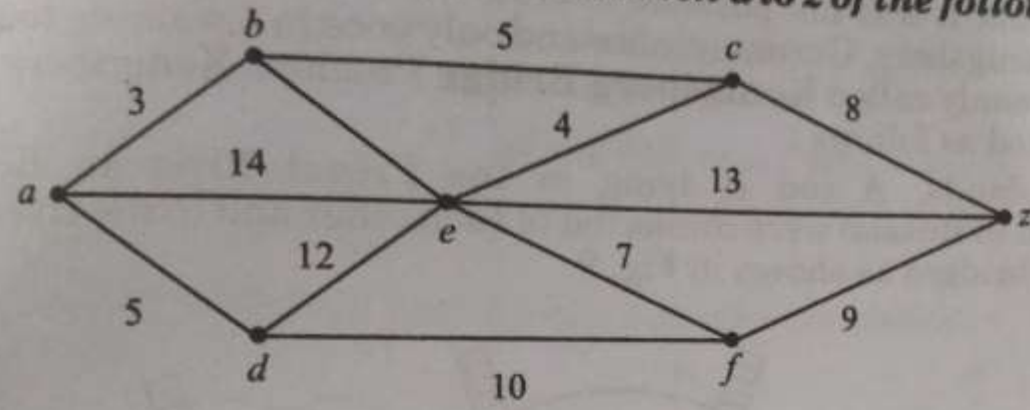


Fig. 8

[Indore (Vth Sem.), 2012]

Solution. Let $G = (V, E)$ denote the graph in Fig. 8.

Here $V = \{a, b, c, d, e, f, z\}$

Step I. Taking $P_1 = \{a\}$, $T_1 = \{b, c, d, e, f, z\}$.

$l(b) = 3$, $l(c) = \infty$, $l(d) = 5$, $l(e) = 14$, $l(f) = \infty$, $l(z) = \infty$

Thus $b \in T_1$ has the minimum index 3.

Step II. Taking $P_2 = \{a, b\}$, $T_2 = \{c, d, e, f, z\}$

$$l(c) = \min(\infty, 3 + 5) = 8$$
$$l(d) = \min(5, 3 + \infty) = 5$$
$$l(e) = \min(14, 3 + 8) = 11$$
$$l(f) = \min(\infty, 3 + \infty) = \infty$$
$$l(z) = \min(\infty, 3 + \infty) = \infty.$$

Thus $d \in T_2$ has the minimum index 5.

Step III. Taking $P_3 = \{a, b, d\}$, $T_3 = \{c, e, f, z\}$

$$l(c) = \min(8, 5 + \infty) = 8$$
$$l(e) = \min(11, 5 + 12) = 11$$
$$l(f) = \min(\infty, 5 + 10) = 15$$
$$l(z) = \min(\infty, 5 + \infty) = \infty.$$

Thus $c \in T_3$ has the minimum index 8.

Step IV. Taking $P_4 = \{a, b, d, c\}$, $T_4 = \{e, f, z\}$

$$l(e) = \min(11, 8 + 4) = 11$$
$$l(f) = \min(15, 8 + \infty) = 15$$
$$l(z) = \min(\infty, 8 + 8) = 16.$$

Thus $e \in T_4$ has the minimum index 11.

Step V. Taking $P_5 = \{a, b, d, c, e\}$, $T_5 = \{f, z\}$

$$l(f) = \min(15, 11 + 7) = 15$$
$$l(z) = \min(16, 11 + 13) = 16.$$

Thus $f \in T_5$ has the minimum index 15.

Step VI. Taking $P_6 = \{a, b, d, c, e, f\}$, $T_6 = \{z\}$

$$l(z) = \min(16, 15 + 9) = 16.$$

Hence the length of the shortest path from a to z is 16. The shortest path is

$$a \to b \to c \to z.$$

Chinese Postman Problem -

A postman starts from his post-office in a town to deliver his letters and returns to his starting point every day. He wants to follow the route so that he visits each road at least once and total distance covered is least.

This problem can be reformulated in terms of a weighted graph, where the graph represents the network of roads and the weight of each edge is the length of the corresponding road.

We wish to find a closed walk that includes each edge at least once and whose total weight is minimum. If the graph is Eulerian then any Euler circuit is a closed walk of least total weight since each road is visited exactly once.

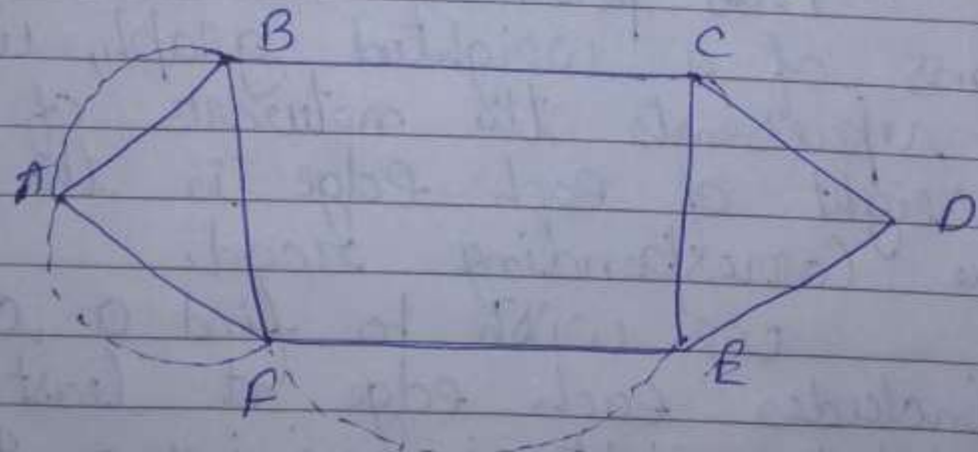
If the graph is not Eulerian then we take a particular graph in which exactly two vertices have odd degree.

among the several paths from E to B, the path

$$E \to F \to A \to B$$

is shortest.

The solution of the chinese postman problem is obtained by taking this shortest path from E to B together with the Euler Circuit from B to E. It gives Euler graph.



Theorem - An undirected graph possesses an Eulerian path if and only if it is connected and has two vertices of odd degree.

proof -

Only if condition -

Suppose the graph possesses an Eulerian path. Then the graph is connected. Since in a Euler path every edge is traveled only once. It follows that the degree of any vertex in the graph must be even with only exception for the two vertices at the two ends of the path. ie. if the two vertices at the two ends of the Euler path are distinct, they are

the only two vertices with odd degrees. If they coincide, all vertices have even degree. And, the Euler path becomes an Euler circuit.
— This proves the necessary condition.

If Condition –
Suppose the graph is connected and has two vertices of odd degree. We construct an Eulerian path by starting at one of the two vertices that are of odd degree and going through the edges of the graph in such a way that no edge will be traced more than once. So it is clear that when we passes through the vertex of even degree during the construction of Euler path, one edge remains untraced. As a result when the construction comes to an end, we must have reached the other vertex of odd degree.

of each path, one edge remains untraced. As a result when the construction comes to an end, we must have reached the other vertex of odd degree.

Thus, it is clear that if all the edges in the graph were traced, we would have an Eulerian path. If all the edges were not traced, we shall remove those edges that have been traced and get a subgraph formed by the remaining edges. In this subgraph the degree of each vertex is even.

Since the original graph is connected, it follows that this subgraph must touch the path that we have tracked at one or more vertices. Now we construct a path from one of those vertices. Since the degrees of vertices are even, so this path will return to the vertex at which it starts.

Now, we Combine these two paths to get a new path that starts and ends at the two vertices of odd degree. This process may be repeated, untill we obtain a path that traverses all the edges in the graph.

Theorem –
An Undirected graph possesses an Eulerian circuit if and only if it is Connected and its vertices are all of even degree.

Example 2. ***Solve the travelling salesman problem for the following graph :***

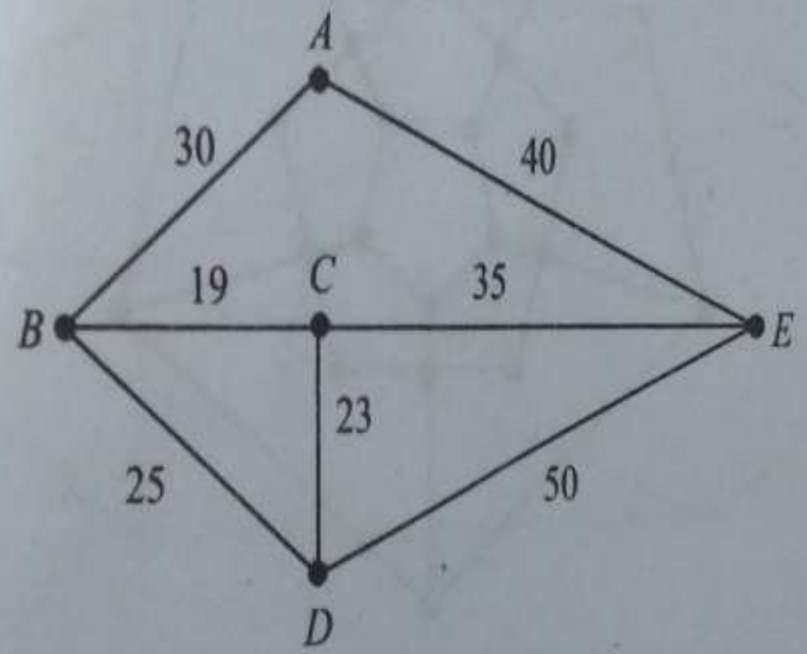


Fig. 31

[Indore 2008]

Solution. The given graph has the following two Hamiltonian circuits :

(a) $A \to B \to C \to D \to E \to A$

(b) $A \to B \to D \to C \to E \to A$.

Let $w(v_i, v_j)$ denote the length of the edge $\{v_i, v_j\}$; *i.e.*, distance between two cities v_i and v_j. Then

the total-distance of the Hamilonian circuit described in (a)

$$= w(A, B) + w(B, C) + w(C, D) + w(D, E) + w(E, A)$$

$$= 30 + 19 + 23 + 50 + 40 = 162$$

the total distance of the Hamiltonian circuit described in (b)

$$= w(A, B) + w(B, C) + w(D, C) + w(C, E) + w(E, A)$$

$$= 30 + 25 + 23 + 35 + 40 = 153.$$

Since the total distance of Hamiltonian circuit in (b) is minimum, salesman should travel according to circuit in (b).

Example 1. ***Solve the travelling salesman problem for the following weighted graph :***

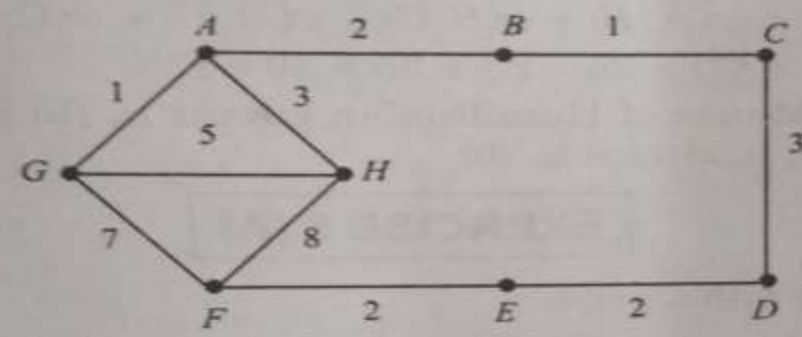


Fig. 30

Solution. The given graph has the following two Hamiltonian circuits :

(a) $A \to B \to C \to D \to E \to F \to H \to G \to A$

(b) $A \to B \to C \to D \to E \to F \to G \to H \to A.$

[Indore (Vth Sem.), 2012; Jabalpur (Vth Sem.), 2012-13]

Let $w(v_i, v_j)$ denote the weight of the edge $\{v_i, v_j\}$. Then the total weights of Hamiltonian circuit described in (a)

$$= w(A, B) + w(B, C) + w(C, D) + w(D, E) + w(E, F) + w(F, H) + w(H, G) + w(G, A)$$

$$= 2 + 1 + 3 + 2 + 2 + 6 + 5 + 1 = 22$$

the total weights of Hamiltonian circuit described in (b)

$$= w(A, B) + w(B, C) + w(C, D) + w(D, E) + w(E, F) + w(F, G) + w(G, H) + w(H, A)$$

$$= 2 + 1 + 3 + 2 + 2 + 7 + 5 + 3 = 25.$$

Since the total weight of Hamiltonian circuit in (a) is minimum, salesman should travel accoring to circuit in (a).

problem for the following

TREES AND ITS PROPERTIES

§ 6.1. INTRODUCTION

The concept of a tree is the most important tool for the study of graph theory, especially for those interested in application of graph.

In this chapter, first of all we shall define a *tree* and study its properties. We shall also introduce and discuss other graph theoretic terms related to trees. Thereafter, we shall introduce the concept of spanning tree-another important notion in the theory of graphs. The relationships among circuits, trees, and so on, in a graph are explored.

§ 6.2. TREE

[Jiwaji (Vth Sem.), 2012]

In the study of graph of a network we obtain several circuit (closed paths) through which current can flow. Such circuits, are necessary for the flow of current.

However, by deleting some branches from the graph we can check the current to flow. By deleting branches and rejecting the closed path, the residual graph is known as a "tree".

Definition. A **tree** is a connected graph without any circuits. **[Indore 2001; Rewa 2001; Jiwaji 2001; Sagar 2001]**

Example 1. All the trees with at most five vertices are shown in Fig. 1.

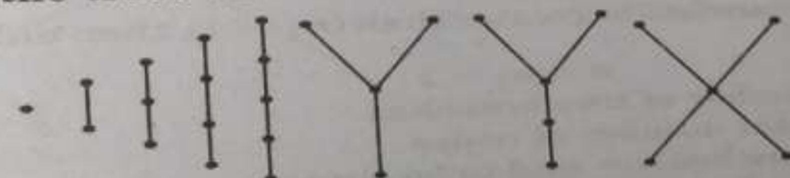


Fig. 1

Example 2. The adjoining Fig. 2 shows the six trees with six vertices :

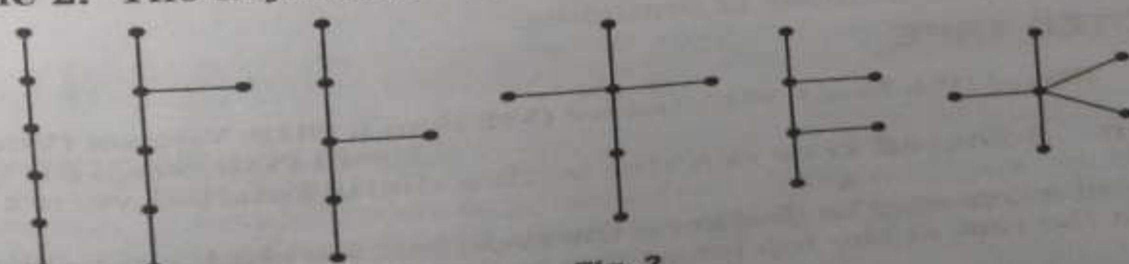


Fig. 2

Example 3. The adjoining Fig. 3 shows a river with its tributaries and subtributaries :

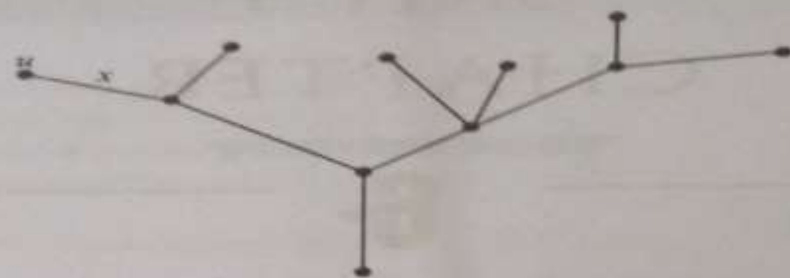


Fig. 3. River tree

The process of deleting the branches from a graph can be done by several methods. In the adjoining Fig. 4 (a), a graph is shown and in Fig. 4 (b) and (c) two trees are shown amongst several possibilities.

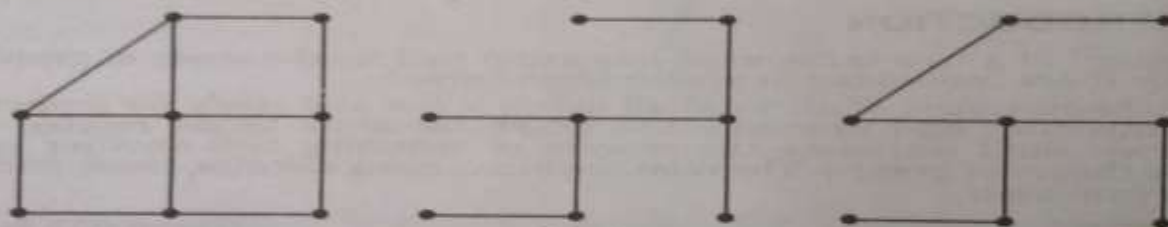


Fig. 4

In every tree, the remaining branches of the tree are just sufficient to join all nodes. Hence we can define precisely a tree as follows :

Definition. Any set of branches in the original graph, just sufficient in number to connect all the nodes, is called a **tree.**

If n_i denotes the total number of nodes, then it can easily be seen that the minimum number of branches to join all nodes is $(n_i - 1)$ and we shall not get any closed path (circuit).

If the number of branches is greater than $(n_i - 1)$, then only we shall get closed path.

Hence $$n = n_i - 1$$

where n = number of tree branches

n_i = total number of nodes.

The remaining branches are said to be **lines.**

Thus, if l is the number of lines, then

$$b = l + n$$

where b denotes the total number of branches.

Example 1. In the adjoining Fig. 5 rooted trees with 4-vertices are shown :

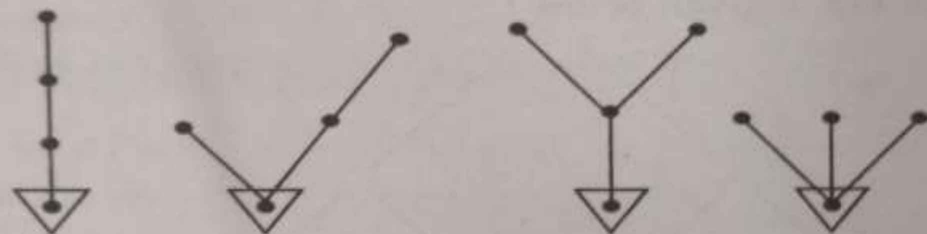


Fig. 5. Rooted Trees with 4 Vertices

Example 2. In the adjoining Fig. 6 all rooted trees with 5 vertices are shown :

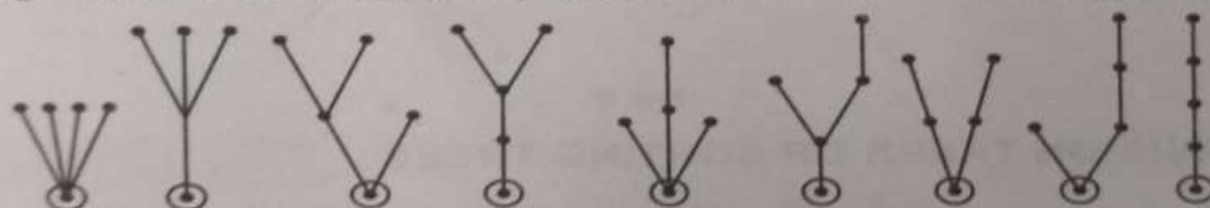


Fig. 6. Rooted Trees with Five Vertices

ORDERED ROOTED TREE

[Sagar (Vth Sem.), 2013]

Definition. If edges leaving each vertex of a rooted tree T are labelled, then T is called an **ordered rooted tree**. The vertices of an ordered rooted tree can be labelled as follows : we assign 0 to the root of the tree. We next assign 1, 2, 3, 4, ... to the vertices immediately following the root of T according as the edges were ordered. The remaining vertices can be ordered as follows : If p is the label of a vertex v of T then $p.0$, $p.1$, $p.2$, ... are assigned to the vertices immediately following v according as the edges were ordered. An ordered rooted tree is shown in the Fig. 7 given below :

THANK YOU