மனோன்மணியம் சுந்தரனார் பல்கலைக்கழகம்

MANONMANIAM SUNDARANAR UNIVERSITY

TIRUNELVELI-627 012

தொலைநிலை தொடர் கல்வி இயக்ககம்

DIRECTORATE OF DISTANCE AND CONTINUING EDUCATION

M.Sc. Mathematics

I YEAR (Elective - I)

GRAPH THEORY AND APPLICATIONS

Sub. Code: SMAE11 **(For Private Circulation only)**

M.Sc. MATHEMATICS –I YEAR

SMAE11 – GRAPH THEORY AND APPLICATIONS

(Elective – I)

SYLLABUS

UNIT-I:

Basic Result: Subgraphs - Degrees of Vertices - Paths and Connectedness - Automorphism of a simple graph - Line graphs - Operations on graphs - Graph Products.

Chapter 1: Section 1.1 to 1.9.

UNIT-II:

Connectivity: Vertex Cuts and Edge Cuts - Connectivity and Edge Connectivity - Blocks.

Chapter 3: Section 3.1 to 3.4.

UNIT-III:

Trees: Definition, Characterization and simple properties - Centres and centroids - Counting the number of Spanning Trees - Cayley's formula.

Chapter 4: Section 4.1 to 4.5.

UNIT-IV:

Independent Sets and Matchings: Vertex Independent Sets and Vertex Coverings - Edge Independent Sets - Matchings and Factors - Matching in Bi-partite Graphs - Perfect Matching and the Tutte Matrix.

Chapter 5: Section 5.1 to 5.6.

UNIT-V:

Eulerian and Hamiltonian Graphs: Eulerian Graphs - Hamiltonian Graphs - Hamilton's "Around the World" Game.

Graph Colorings: Vertex Colorings - Applications of Graph Colorings - Critical Graphs - Brooks' Theorem.

Chapter 6: Section 6.1 to 6.3,

Chapter 7: Section 7.1 to 7.3 (up to Brooks Theorem).

Recommended Text:

R. Balakrishnan and K. Ranganathan, *Text Book of Graph Theory*, Springer Publications, 2012.

SMAE11 – GRAPH THEORY AND APPLICATIONS

(Elective – I)

CONTENTS

UNIT-1 (Basic Result)

UNIT-2 (Connectivity)

UNIT-3 (Trees)

UNIT-4 (Independent Sets and Matchings)

UNIT-5 (Eulerian and Hamiltonian Graphs)

UNIT-I:

Basic Result: Subgraphs - Degrees of Vertices - Paths and Connectedness - Automorphism of a simple graph - Line graphs - Operations on graphs - Graph Products.

Chapter 1: Section 1.1 to 1.9.

1.1 Graph Products **Introduction:**

Graphs serve as mathematical models to analyze many concrete real-world problems successfully. Certain problems in physics, chemistry, communication science, computer technology, genetics, psychology, sociology, and linguistics can be formulated as problems in graph theory. Also, many branches of mathematics, such as group theory, matrix theory, probability, and topology, have close connections with graph theory.

Some puzzles and several problems of a practical nature have been instrumental in the development of various topics in graph theory. The famous Königsberg bridge problem has been the inspiration for the development of Eulerian graph theory. The challenging Hamiltonian graph theory has been developed from the "Around the World" game of Sir William Hamilton. The theory of acyclic graphs was developed for solving problems of electrical networks, and the study of "trees" was developed for enumerating isomers of organic compounds. The well-known four-color problem formed the very basis for the development of planarity in graph theory and combinatorial topology. Problems of linear programming and operations research (such as maritime traffic problems) can be tackled by the theory of flows in networks. Kirkman's schoolgirl problem and scheduling problems are examples of problems that can be solved by graph colorings. The study of simplicial complexes can be associated with the study of graph theory. Many more such problems can be added to this list.

1.2 Basic Concepts:

Consider a road network of a town consisting of streets and street intersections. Figure (1.1 a) represents the road network of a city. Figure (1.1 b) denotes the corresponding graph of this network, where the street intersections are represented by points, and the street joining a pair of intersections is represented by an arc (not necessarily a straight line). The road network in Figure (1.1) is a typical example of a graph in which intersections and streets are, respectively, the "vertices" and "edges" of the graph. (Note that in the road network in Figure (1.1 a), there are two streets joining the intersections J_7 and J_8; and there is a loop street starting and ending at J_2.)

Figure (1.1 a) A road network and **Figure (1.1 b)** the graph corresponding to the road network in **Figure (1.1 a).**

Figure (1.1 a)

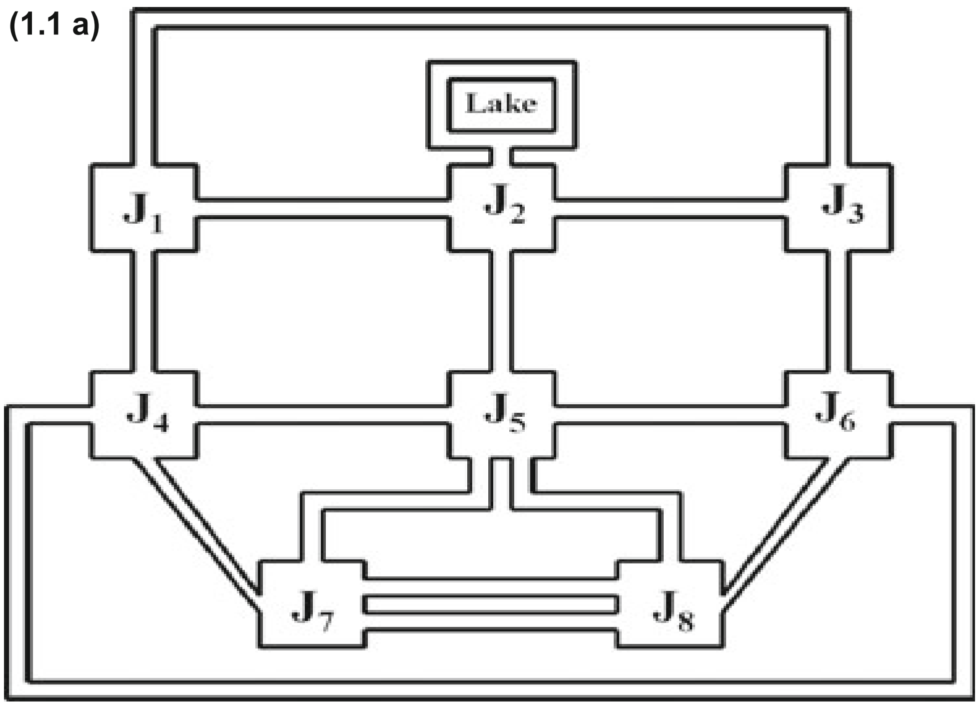
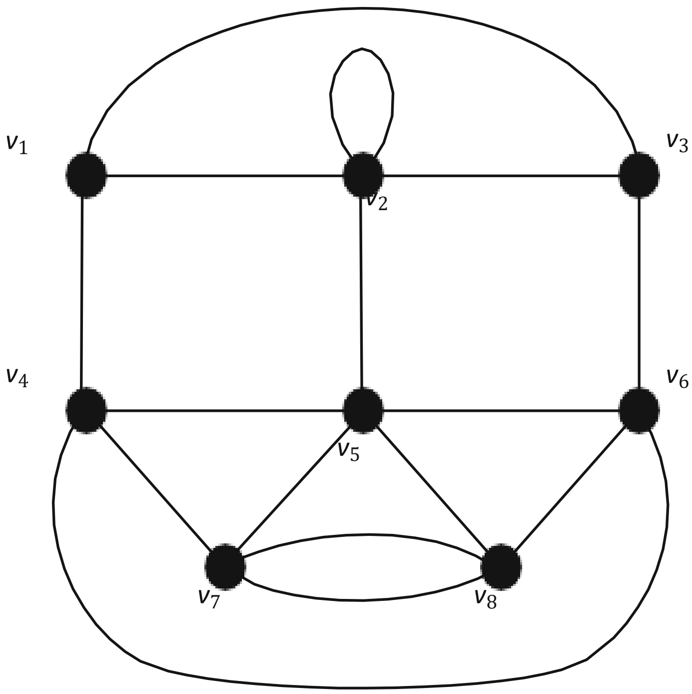


Figure (1.1 b)



We now present a formal definition of a graph.

Definition-1.2.1:

A *graph* is an ordered triple $G = (V(G), E(G), I_G)$, where $(V(G)$ is a nonempty set, $E(G)$ is a set disjoint from $V(G)$; and I_G is an "incidence" relation that associates with each element of $E(G)$, an unordered pair of elements (same or distinct) of V(G), Elements of $V(G)$ are called the *vertices* (or *nodes* or *points*) of G; and elements of $E(G)$ are called the *edges* (or *lines*) of G, $(V.G)$ and $E(G)$ are the *vertex set* and *edge set* of G, respectively. If, for the edge e of G; $I_G(e) = \{u, v\}$, we write $I_G(e) = uv$.

Example-1.2.2:

If $V(G = \{v_1,\ v_2, v_3, v_4, v_5\}$, $E(G = \{e_1,\ e_2, e_3, e_4, e_5\}$ and I_G is given by $I_G(e_1) = \{v_1, v_5\}, I_G(e_2) = \{v_2, v_3\},\ I_G(e_3) = \{v_2, v_4\}, I_G(e_4) = \{v_2, v_5\}, I_G(e_5) = \{v_2, v_5\}$ then $G = (V(G), E(G), I_G)$ is a graph.

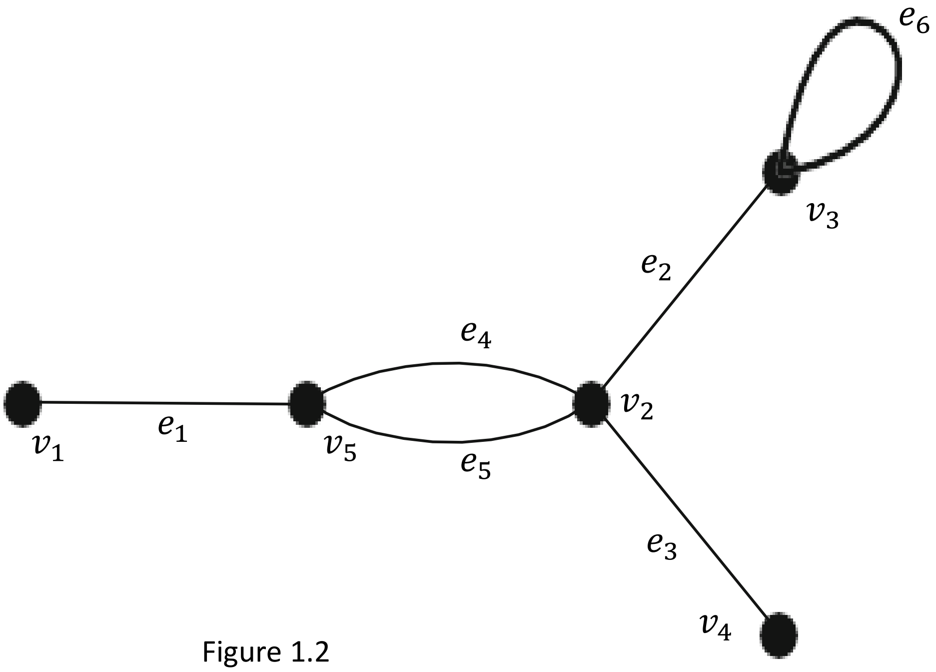


Figure 1.2

Diagrammatic Representation of a Graph-1.2.3:

Each graph can be represented by a diagram in the plane. In this diagram, each vertex of the graph is represented by a point, with distinct vertices being represented by distinct points. Each edge is represented by a simple "Jordan" arc joining two (not necessarily distinct) vertices. The diagrammatic representation of a graph aids in visualizing many concepts related to graphs and the systems of which they are models. In a diagrammatic representation of a graph, it is possible that two edges intersect at a point that is not necessarily a vertex of the graph.

Definition-1.2.4:

If $I_G(e) = \{u, v\}$, then the vertices u and v are called the *end vertices* or *ends* of the edge e. Each edge is said to join its ends; in this case, we say that e is *incident* with each one of its ends. Also, the vertices u and v are then *incident* with e. A set of two or more edges of a graph G is called a set of *multiple* or *parallel edges* if they have the same pair of distinct ends. If e is an edge with endvertices u and v, we write $e = uv$. An edge for which the two ends are the same is called a *loop* at the common vertex. A vertex u is a *neighbor* of v in G, if uv is an edge of G, and $u \neq v$. The set of all neighbors of v is the *open neighborhood* of v or the *neighbor set* of v; and is denoted by $N(v)$; the set $N(v) = N(v) \cup \{v\}$ is the closed neighborhood of v in G. When G must be explicit, these open and closed neighborhoods are denoted by $N(v)$ and $N[v]$, respectively. Vertices u and v are *adjacent* to each other in G if, and only if, there is an edge of G with u and v as its ends. Two distinct edges e and f are said to be *adjacent* if, and only if, they have a common end vertex. A graph is *simple* if it has no loops and no multiple edges. Thus, for a simple graph G, the incidence function I_G is one-to-one. Hence, an edge of a simple graph is identified with the pair of its ends. A simple graph therefore may be considered as an ordered pair $(V(G), E(G))$, where $V(G)$ is a nonempty set and $E(G)$ is a set of unordered pairs of elements of $V(G)$ (each edge of the graph being identified with the pair of its ends).

Example-1.2.5:

In the graph of Figure 1.2, edge $e_3 = v_2v_4$, edges e_4 and e_5 form multiple edges, e_6 is a loop at v_3, $N(v_2) = \{v_3, v_4, v_5\}, N(v_3) = \{v_2\}, N[v_2] = \{v_3, v_4, v_5\}$, and $N[v_2] = N(v_2) \cup \{v_2\}$. Further, v_2 and v_5 are adjacent vertices and e_3 and e_4 are adjacent edges.

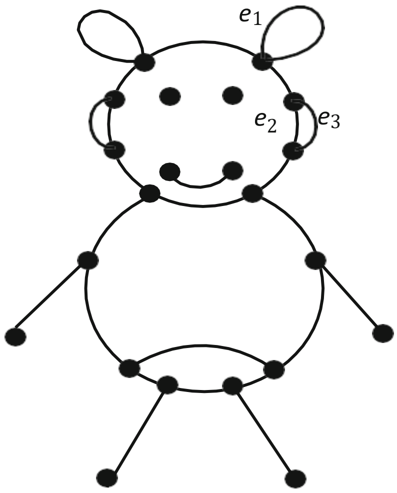


Figure 1.3 A graph diagram e_1 is a loop and $\{e_2, e_3\}$ is a set of multiple edges

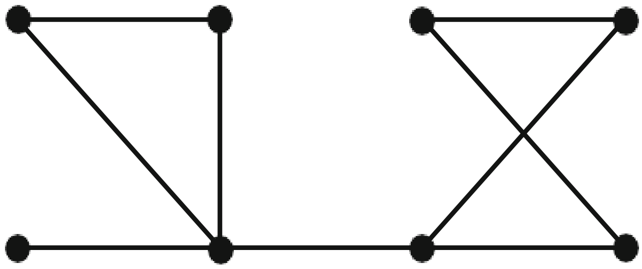


Figure 1.4 A simple graph

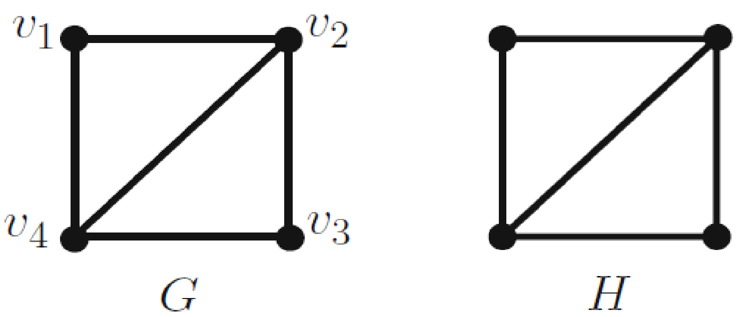


Figure 1.5 A labelled graph G and an unlabeled graph H

Definition-1.2.6:

A graph is called *finite* if both $V(G)$ and $E(G)$ are finite. A graph that is not finite is called an *infinite* graph. Unless otherwise stated, all graphs considered in this text are finite. Throughout this book, we denote by $n(G)$ and $m(G)$ the number of vertices and edges of the graph G, respectively. The number $n(G)$ is called the *order* of G and $m(G)$ is the *size* of G. When explicit reference to the graph G is not needed, $V(G), E(G), n(G)$ and $m(G)$ will be denoted simply by V, E, n and m respectively.

Figure 1.3 is a graph with loops and multiple edges, while Figure 1.4 represents a simple graph.

Remark 1.2.7:

The representation of graphs on other surfaces such as a sphere, a torus, or a Mӧbius band could also be considered. Often a diagram of a graph is identified with the graph itself.

Definition 1.2.8:

A graph is said to be *labeled* if its n vertices are distinguished from one another by labels such as $v_1, v_2, v_3, \dots, v_n$, (see Figure 1.5).

Note that there are three different labeled simple graphs on three vertices each having two edges, whereas there is only one unlabeled simple graph of the same order and size (see Figure 1.6).

Isomorphism of Graphs 1.2.9. A graph isomorphism, which we now define, is a concept similar to isomorphism in algebraic structures. Let $G = (V(G), E(G), I_G)$ and $G = (V(H), E(H), I_H)$ be two graphs. A *graph isomorphism* from G to H is a pair $(\emptyset, \theta)$, where $\emptyset: V(G) \rightarrow V(H)$ and

$\theta: E(G) \rightarrow E(H)$ are bijections with the property that $I_G(e) = \{u, v\}$ if and only if $I_H(\theta(e)) = \{\emptyset(u), \emptyset(v)\}$. If $(\emptyset, \theta)$ is a graph isomorphism, the pair of inverse mappings $(\emptyset^{-1}, \theta^{-1})$ is also a graph isomorphism. Note that the bijection $\emptyset$ satisfies the condition that u *and* v are end vertices of an edge e *of* G *if and only if* $\emptyset(u)$ and $\emptyset(v)$ are end vertices of the edge $\emptyset(e)$ *in* H. It is clear that isomorphism is an equivalence relation on the set of all graphs. Isomorphism between graphs is denoted by the symbol $\cong$ (as in algebraic structures).

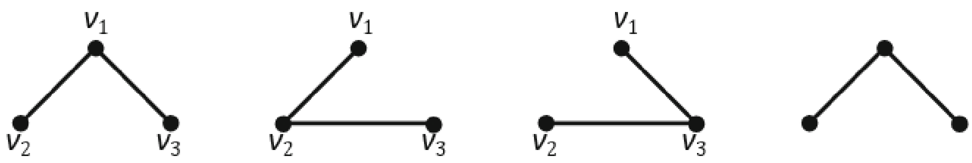


Fig. 1.6 Labeled and unlabeled simple graphs on three vertices

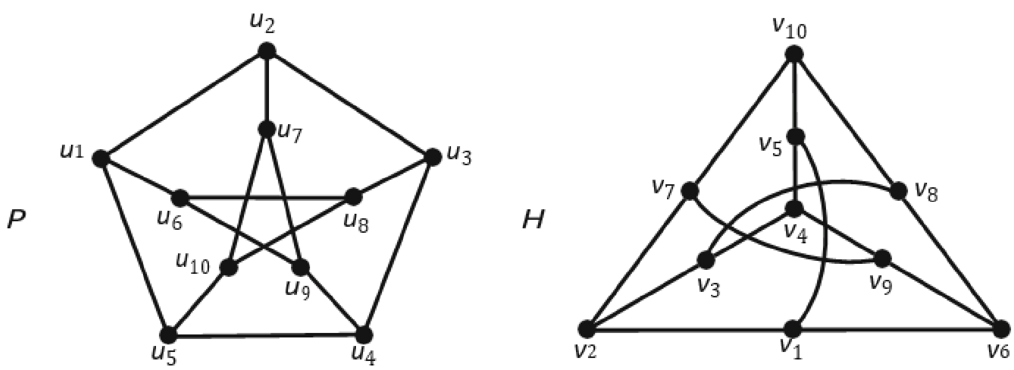


Fig. 1.7 Isomorphic graphs

Simple Graphs and Isomorphisms 1.2.10. If graphs G and H are simple, any bijection $\emptyset: V(G) \rightarrow V(H)$ such that u and v are adjacent in G if and only if $\emptyset(u)$ and $\emptyset(v)$ are adjacent in H induces a bijection $\theta: E(G) \rightarrow E(H)$ satisfying the condition that $I_G(e) = \{u, v\}$ if and only if $I_H(\theta(e)) = \{\emptyset(u), \emptyset(v)\}$ Hence, $\emptyset$ itself is referred to as an isomorphism in the case of simple graphs G and H. Thus, if G and H are simple graphs, an isomorphism from G to H is a bijection $\emptyset: V(G) \rightarrow V(H)$ such that u and

v are adjacent in G if and only if $\emptyset(u)$ and $\emptyset(v)$ are adjacent in H. Figure 1.7 exhibits two isomorphic graphs P and H, where P is the well-known Petersen graph. We observe that P is a simple graph.

Exercise 2.1:

Let G and H be simple graphs and let $\emptyset: V(G) \rightarrow V(\subset H)$ be a bijection such that $uv \in E(G)$ implies that $\emptyset(u)\emptyset(v) \in E(H)$. Show by means of an example that $\emptyset$ need not be an isomorphism from G to H.

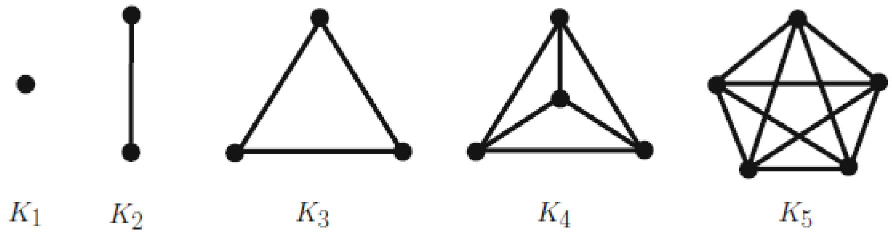
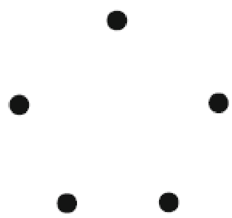


Fig. 1.8 Some complete graphs

Fig. 1.9 A totally disconnected graph on five vertices



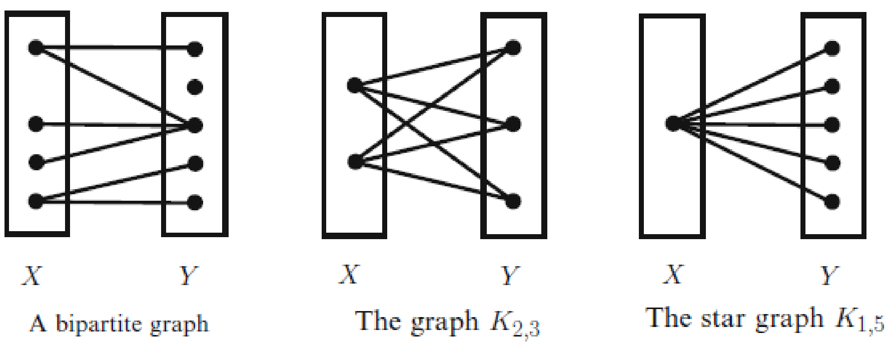


Fig. 1.10 Bipartite graphs

Definition 1.2.11.

A simple graph G is said to be complete if every pair of distinct vertices of G are adjacent in G. Any two complete graphs each on a set of n vertices are isomorphic; each such graph is denoted by K_n(Fig. 1.8).

A simple graph with n vertices can have at most $\binom{n}{2} = \frac{n(n-1)}{2}$ edges. The complete graph K_n has the maximum number of edges among all simple graphs with n vertices. At the other extreme, a graph may possess no edge at all. Such a graph is called a *totally disconnected graph* (see Fig. 1.9). Thus, for a simple graph G with n vertices, we have $0 \leq m(G) \leq \frac{n(n-1)}{2}$.

Definition 1.2.12:

A graph is *trivial* if its vertex set is a singleton and it contains no edges. A graph is *bipartite* if its vertex set can be partitioned into two nonempty subsets X and Y such that each edge of G has one end in X and the other in Y. The pair (X, Y) is called a *bipartition* of the bipartite graph. The bipartite graph G with bipartition (X, Y) is denoted by $G(X, Y)$. A simple bipartite graph $G(X, Y)$ is *complete* if each vertex of X is adjacent to all the vertices of Y. If $G(X, Y)$ is complete with $|X| = p$ and $|Y| = q$, then $G(X, Y)$ is denoted by $K_{p,q}$. A complete bipartite graph of the form $K_{1,q}$ is called a *star* (see Figure 1.10).

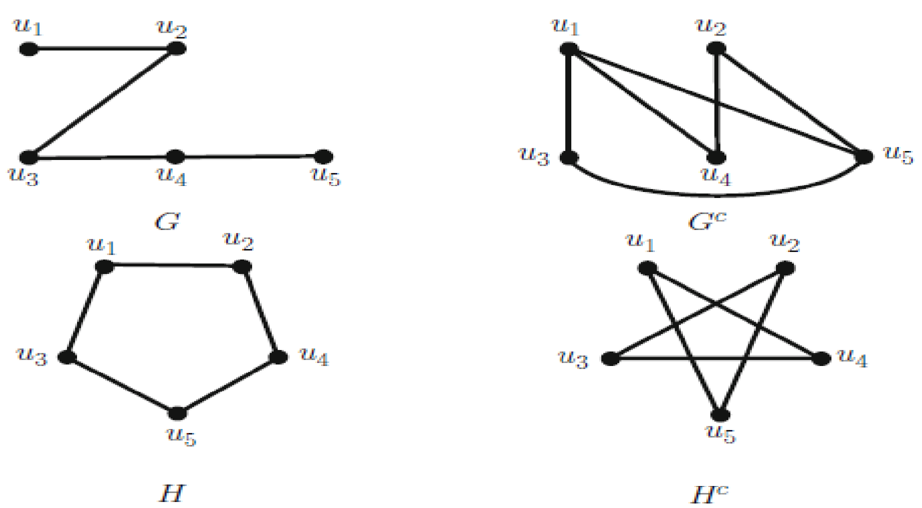


Fig. 1.11 Two simple graphs and their complements

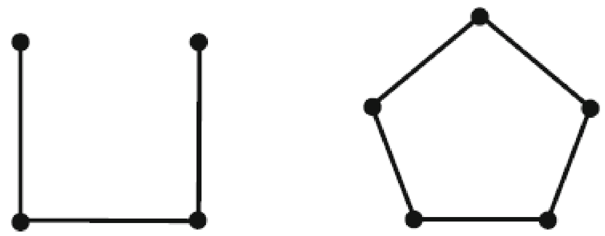


Fig. 1.12 Self-complementary graphs

Definition 1.2.13:

Let G be a simple graph. Then the *complement* G^c of G is defined by taking $V(G^c) = V(G)$ and making two vertices u and v adjacent in G^c if and only if they are non-adjacent in G (see Figure 1.11). It is clear that G^c is also a simple graph and that $(G^c)^c = G$.

If $|V(G)| = n$, then clearly, $|E(G)| + |E(G^c)| = |E(K_n)| = \frac{n(n-1)}{2}$.

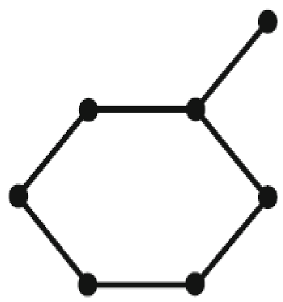
Definition 1.2.13:

A simple graph G is called *self-complementary* if $G \cong G^c$.

For example, the graphs shown in Figure 1.12 are self-complementary.

Exercise 2.2:

Find the complement of the following simple graph:



1.3 Subgraphs:

Definition 1.3.1:

A graph H is called a *subgraph* of G if $V(H) \subseteq V(G), E(H) \subseteq E(G)$, and I_H is the restriction I_G to $E(H)$. If H is a subgraph of G, then G is said to be a *super graph* of H. A subgraph H of a graph G is a proper subgraph of G if either $V(H) \neq V(G)$ or $E(H) \neq E(G)$.

A subgraph H of G is said to be *an induced subgraph* of G if each edge of G having its ends in $V(H)$ is also an edge of H.

A subgraph H of G is a *spanning subgraph* of G if $V(H) = V(G)$. The induced subgraph of G with vertex set $S \subseteq V(G)$ is called the subgraph of G induced by S and is denoted $G[S]$.

Let E' be a subset of E and let S denote the subset of V consisting of all the end vertices in G of edges in E'. Then the graph $(S, E', I_{G/E'})$ is the subgraph of G induced by the edge set E' of G. It is denoted by $G[E']$ (see Figure 1.13). Let u and v be vertices of a graph G. By $G + uv$, we mean the graph obtained by adding a new edge uv to G.

Definition 1.3.2:

A *clique* of G is a complete subgraph of G: A clique of G is a *maximal clique* of G if it is not properly contained in another clique of G (see Figure 1.13).

Definition 1.3.3:

Deletion of vertices and edges in a graph: Let G be a graph, S a proper subset of the vertex set V, and E' a subset of E. The subgraph $G[V\backslash S]$ is said to be obtained from G by the *deletion* of S. This subgraph is denoted by $G - S$. If $S = \{v\}$, $G - S$ is simply denoted by $G - v$. The spanning subgraph of G with the edge set $E\backslash E'$ is the subgraph obtained from G by deleting the edge subset E'. This subgraph is denoted by $G - E'$. Whenever $E' = \{e\}, G - E'$ is simply denoted by $G - e$. Note that when a vertex is deleted from G all the edges incident to it are also deleted from G, whereas the deletion of an edge from G does not affect the vertices of G (see Figure 1.14).

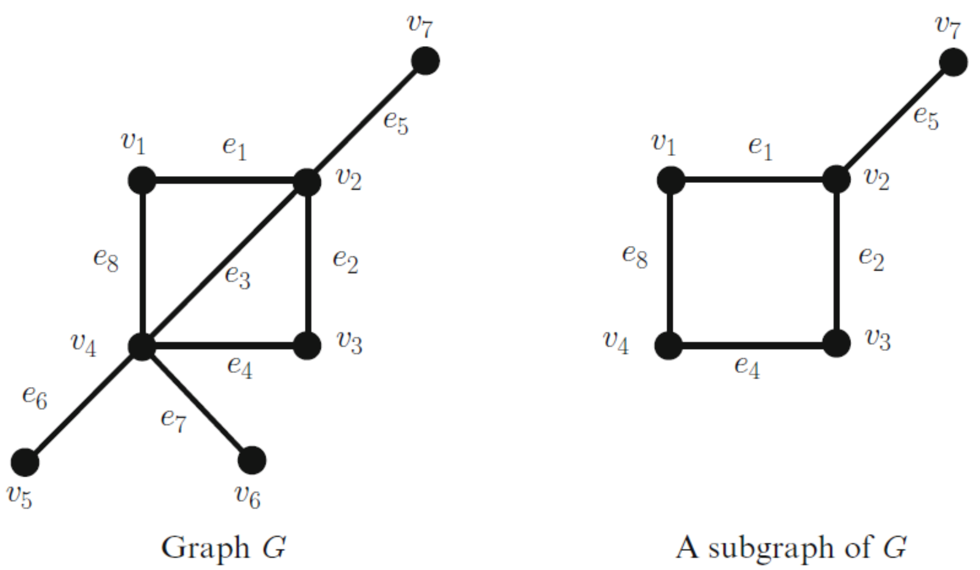


Fig. 1.13 Various subgraphs and cliques of G

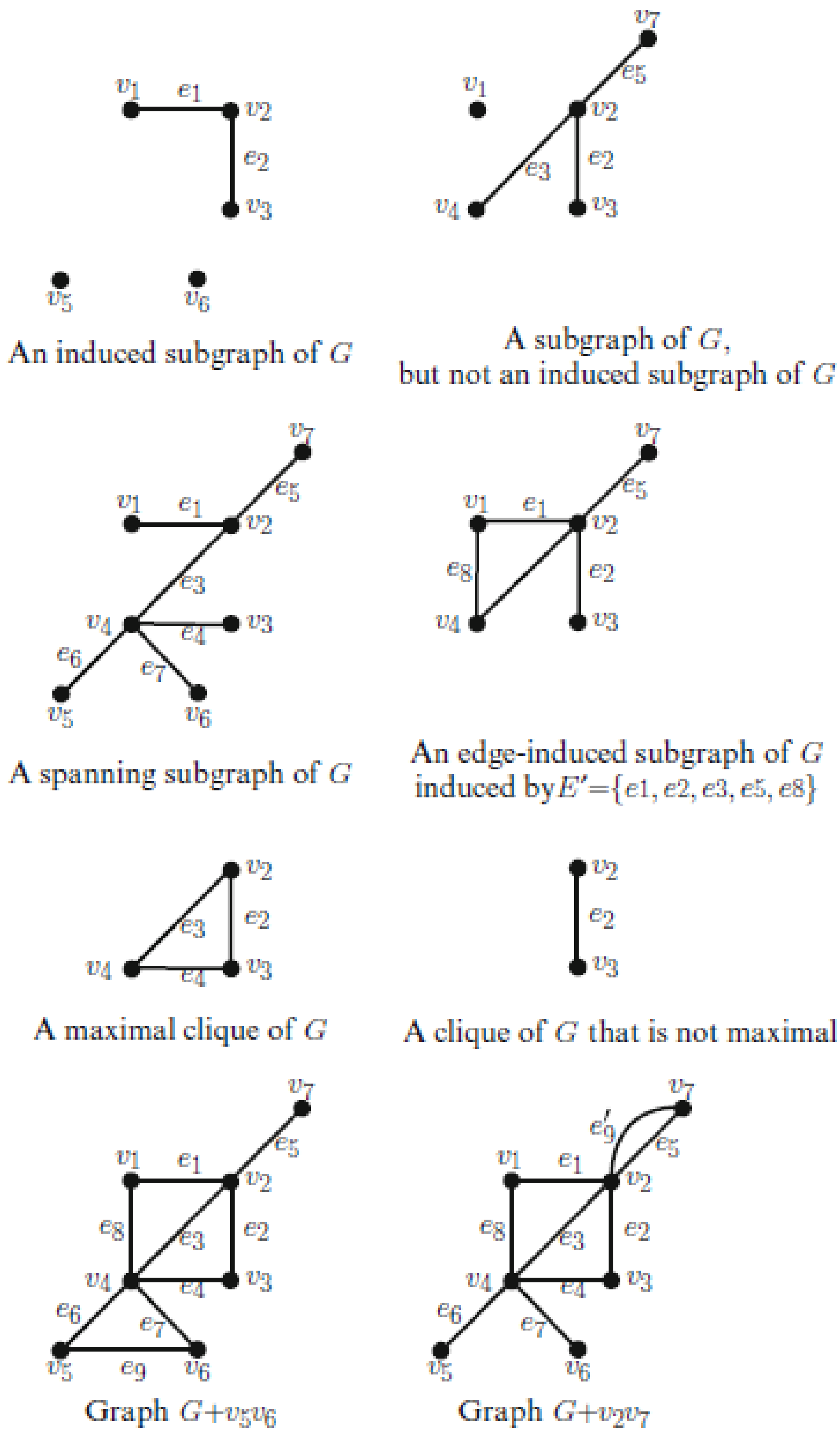


Fig. 1.13 (continued)

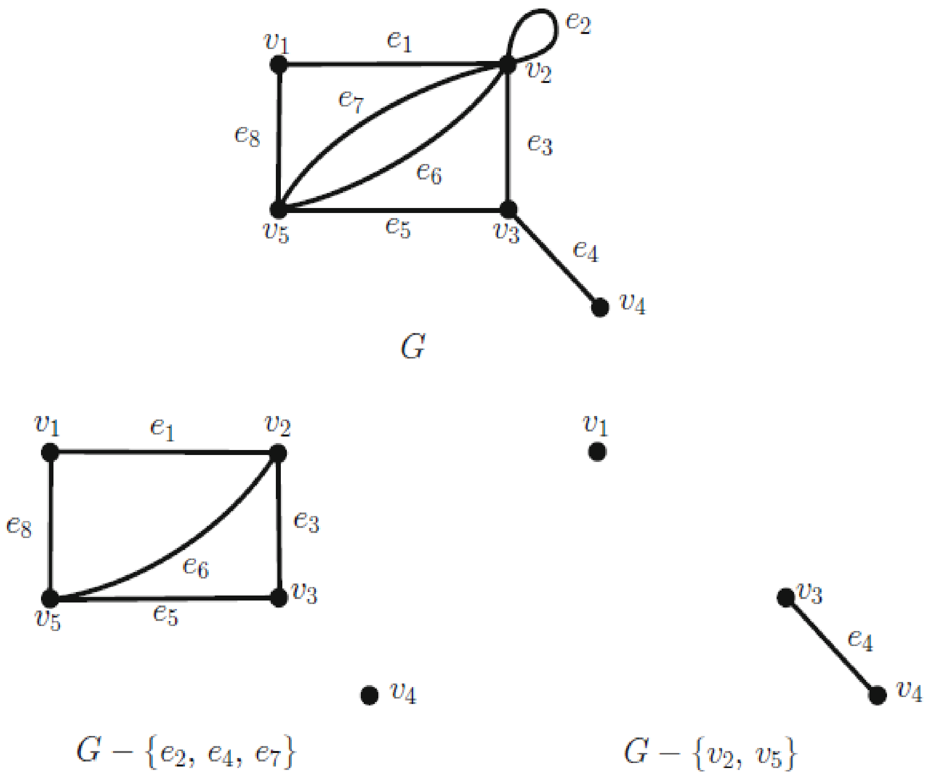


Fig. 1.14 Deletion of vertices and edges from G

1.4 Degrees of Vertices:

Definition 1.4.1:

Let G be a graph and $v \in V$. The number of edges incident at v in G is called the *degree* (or *valency*) of the vertex v in G and is denoted by $d_G(v)$, or simply $d(v)$ when G requires no explicit reference. A loop at v is to be counted twice in computing the degree of v. The minimum (respectively, maximum) of the degrees of the vertices of a graph G is denoted by $\delta(G)$ or δ (respectively, $\Delta(G)$ or Δ). A graph G is called *k-regular* if every vertex of G has degree k. A graph is said to be *regular* if it is k-regular for some non-negative integer k. In particular, a 3-regular graph is called a *cubic graph.*

Definition 1.4.2:

A spanning 1-regular subgraph of G is called a 1-*factor* or a *perfect matching* of G. For example, in the graph G of Figure 1.15, each of the pairs $\{ab, cd\}$ and $\{ad, bc\}$ is a 1-factor of G.

Definition 1.4.3:

A vertex of degree 0 is an *isolated vertex* of G. A vertex of degree 1 is called a *pendant vertex* of G, and the unique edge of G incident to such a vertex of G is a *pendant edge* of G. A sequence formed by the degrees of the vertices of G, when the vertices are taken in the same order, is called a *degree sequence* of G. It is customary to give this sequence in the nonincreasing or nondecreasing order, in which case the sequence is unique.

In the graph G of Fig. 1.16, the numbers within the parentheses indicate the degrees of the corresponding vertices. In G, v_7 is an isolated vertex, v_6 is a pendant vertex, and $v_5 v_6$ is a pendant edge. The degree sequence of G is

$(0, 1, 2, 2, 4, 4, 5)$.

The very first theorem of graph theory was due to Leonhard Euler (1707–1783). This theorem connects the degrees of the vertices and the number of edges of a graph.

Theorem 1.4.4 (Euler). *The sum of the degrees of the vertices of a graph is equal to twice the number of its edges.*

Proof:

If $e = uv$ is an edge of G, e is counted once while counting the degrees of each of u and v (even when $u = v$). Hence, each edge contributes 2 to the sum of the degrees of the vertices. Thus, the m edges of G contribute $2m$ to the degree sum.

Fig. 1.15 Graph with 1-factors

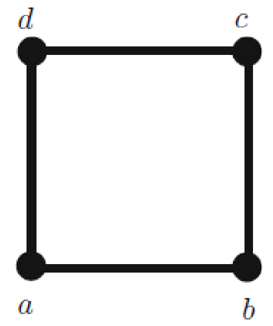
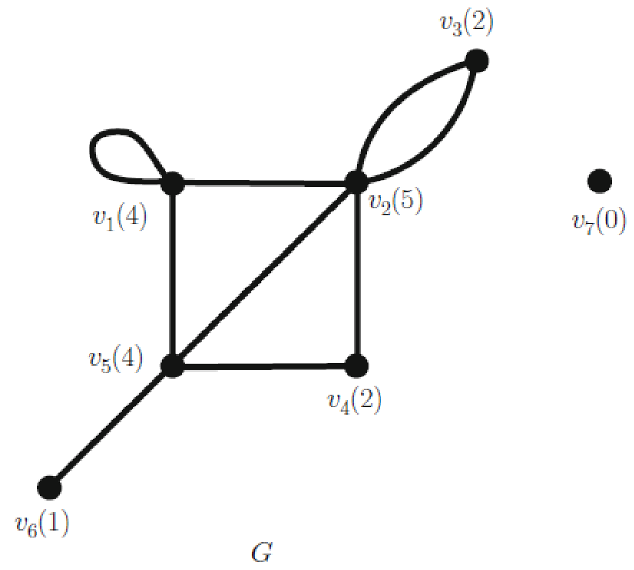


Fig. 1.16 Degrees of vertices of graph G



Remark 1.4.5:

If $d = d_1, d_2, \ldots, d_n$ is the degree sequence of G, then the above theorem gives the equation $\sum_{i=1}^{n} d_i = 2m$, where n and m are the order and the size of G, respectively.

Corollary 1.4.6:

In any graph G, the number of vertices of odd degree is even.

Proof:

Let V_1 and V_2 be the subsets of vertices of G with odd and even degrees, respectively. By Theorem 1.4.4,

$$2m(G) = \sum_{v \in V} d_G(v) = \sum_{v \in V_1} d_G(v) + \sum_{v \in V_2} d_G(v).$$

As $2m(G)$ and $\sum_{v \in V_2} d_G(v)$ are even, $\sum_{v \in V_1} d_G(v)$ is even. Since for each $v \in V_1, d_G(v)$ is odd, $|V_1|$ must be even.

Exercise 4.1. Show that if G and H are isomorphic graphs, then each pair ofcorresponding vertices of G and H has the same degree.

Exercise 4.2. Let $(d_1, d_2, \dots, d_n)$ be the degree sequence of a graph and r be positive integer. Show that $\sum_{i=1}^{n} d_i^r$ is even.

Definition 1.4.7:

Graphical sequences: A sequence of nonnegative integers $d = (d_1, d_2, \dots, d_n)$ is called *graphical* if there exists a simple graph whose degree sequence is d. Clearly, a necessary condition for $d = (d_1, d_2, \dots, d_n)$ to be graphical is that $\sum_{i=1}^{n} d_i^r$ is even and $d_1 \geq 0, 1 \leq i \leq n$. These conditions, however, are not sufficient, as Example 1.4.8 shows.

Example 1.4.8:

The sequence $d = (7,\ 6,\ 3,\ 3,\ 2,\ 1,\ 1,\ 1)$ is not graphical even though each term of d is a nonnegative integer and the sum of the terms is even. Indeed, if d were graphical, there must exist a simple graph G with eight vertices whose degree sequence is d. Let v_0 and v_1 be the vertices of G whose degrees are 7 and 6, respectively. Since G is simple, v_0 is adjacent to all the remaining vertices of G, and v_1, besides v_0, should be adjacent to another five vertices. This means that in $V - \{v_0,\ v_1\}$ there must be at least five vertices each of degree at least 2 but this is not the case.

Exercise 4.3:

If $d = (d_1, d_2, \ldots, d_n)$ is any sequence of non-negative integers with $\sum_{i=1}^{n} d_i$ even, show that there exists a graph (not necessarily simple) with d as its degree sequence.

We present a simple application whose proof just depends on the degree sequence of a graph.

Example 1.4.9:

In any group of n persons $(n \geq 2)$, there are at least two with the same number of friends.

Proof:

Denote the n persons by $v_1, v_2, \ldots, v_n$. Let G be the simple graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ in which v_i and v_j are adjacent if and only if the corresponding persons are friends. Then the number of friends of v_i is just the degree of v_i in G: Hence, to solve the problem, we must prove that there are two vertices in G with the same degree. If this were not the case, the degrees of the vertices of G must be $0, 1, 2, \ldots, (n-1)$ in some order. However, a vertex of degree $(n-1)$ must be adjacent to all the other vertices of G, and consequently there cannot be a vertex of degree 0 in G. This contradiction shows that the degrees of the vertices of G cannot all be distinct, and hence at least two of them should have the same degree.

Exercise 4.3:

Let G be a graph with n vertices and m edges. Assume that each vertex of G is of degree either k or $k+1$. Show that the number of vertices of degree k in G is $(k+1)n - 2m$.

1.5 Paths and Connectedness:

Definition 1.5.1:

A *walk* in a graph G is an alternating sequence $W: v_0 e_1 v_1 e_2 v_2 \dots e_p v_p$ of vertices and edges beginning and ending with vertices in which v_{i-1} and v_i are the ends of e_i; v_0 is the *origin* and v_p is the *terminus* of W. The walk W is said to join v_0 and v_p; it is also referred to as a $v_0 - v_p$ walk. If the graph is simple, a walk is determined by the sequence of its vertices. The walk is *closed* if $v_0 = v_6$ and is *open* otherwise. A walk is called a *trail* if all the edges appearing in the walkare distinct. It is called a *path* if all the vertices are distinct. Thus, a path in G is automatically a trail in G: When writing a path, we usually omit the edges. A *cycle* is a closed trail in which the vertices are all distinct. The *length* of a walk is the number of edges in it. A walk of length 0 consists of just a single vertex.

Example 1.5.2:

In the graph of Figure 1.17, $v_5 e_7 v_1 e_1 v_2 e_4 v_4 e_5 v_1 e_7 v_5 e_7 v_5 e_9 v_6$ is a walk but not a trail (as edge e_7 is repeated) $v_1 e_1 v_2 e_2 v_3 e_3 v_2 e_1 v_1$ is a closed walk; $v_1 e_1 v_2 e_4 v_4 e_5 v_1 e_7 v_5$ is a trail; $v_6 e_8 v_1 e_1 v_2 e_2 v_3$ is a path and $v_1 e_1 v_2 e_4 v_4 e_6 v_5 e_7 v_1$ is a cycle. Also, $v_6 v_1 v_2 v_3$ is a path, and $v_1 v_2 v_4 v_5 v_6 v_1$ is a cycle in this graph. A cycle is enclosed by ordinary parentheses.

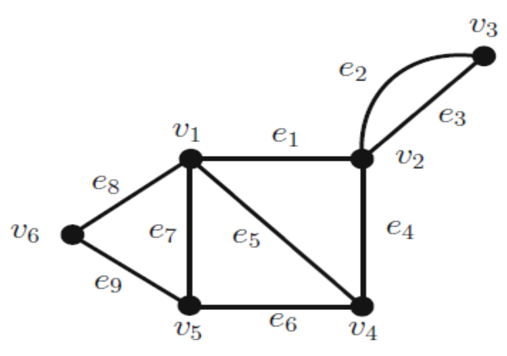


Fig. 1.17 Graph illustrating walks, trails, paths, and cycles

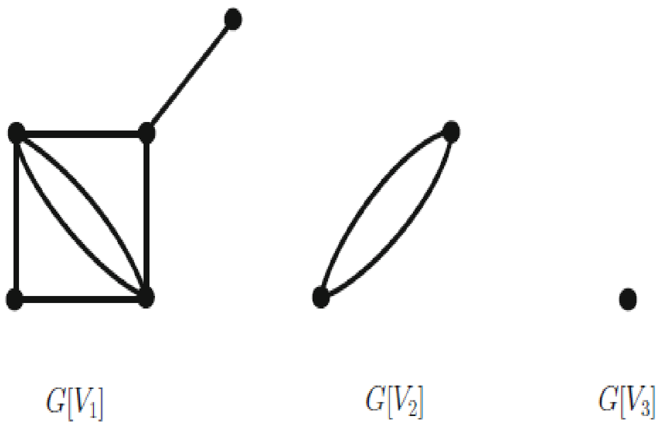
Definition 1.5.3:

A cycle of length k is denoted by C_k. Further, P_k denotes a path on k vertices. In particular, C_3 is referred to as a *triangle*, C_4 as a *square*, and C_5 as a *pentagon*. If $P = v_0e_1v_1e_2v_2 \ldots e_kv_k$ is a path, then $P^{-1} = v_ke_kv_{k-1}e_{k-1}v_{k-2} \ldots v_1e_1v_0$ is also a path and P^{-1} is called the *inverse* of the path P: The subsequence $v_ie_{i+1}v_{i+1} \ldots e_jv_j$ of P is called the $v_i - v_j$ *section* of P.

Definition 1.5.4:

Let G be a graph. Two vertices u and v of G are said to be *connected* if there is a $u - v$ path in G. The relation "connected" is an equivalence relation on $V(G)$. Let $V_1, V_2, \ldots, V_\omega$ be the equivalence classes. The subgraphs $G[V_1], G[V_2], \ldots, G[V_\omega]$ are called the *components* of G. If $\omega = 1$, the graph G is *connected*; otherwise, the graph G is *disconnected* with $\omega \geq 2$ components(see Figure 1.18).

Fig. 1.18 A graph G with three components



Definition 1.5.5:

The components of G are clearly the maximal connected sub-graphs of G. We denote the number of components of G by $\omega(G)$. Let u and v betwo vertices of G. If u and v are in the same component of G, we define $d(u,v)$ to be the length of a shortest $u-v$ path in $G;$ otherwise, we define $d(u,v)$ to be ∞. If G is a connected graph, then d is a distance function or metric on $V(G);$ that is, $d(u,v)$ satisfies the following conditions:

(i) $d(u,v) \geq 0$ and $d(u,v) = 0$ if and only if $u = v$.

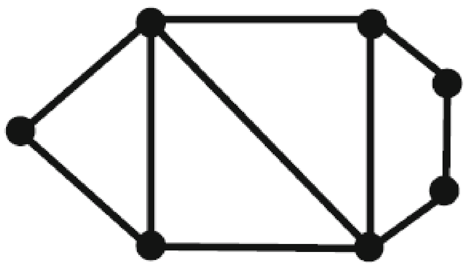
(ii) $d(u,v) = d(v,u)$

(iii) $d(u,v) \leq d(u,w) + d(w,v)$ for every *w* in *V(G)*.

Exercise 5.1:

Prove that the function d defined above is indeed a metric on *(G).*

Exercise 5.2. In the following graph, find a closed trail of length 7 that is not a cycle.



Proposition 1.5.6. *If G is simple and* $\delta \geq \frac{n-1}{2}$, *then G is connected.*

Proof: Assume the contrary.

Then G has at least two components, say G_1, G_2.

Let v be any vertex of G_1.

As $\delta \geq \frac{n-1}{2}, d(v) \geq \frac{n-1}{2}$.

All the vertices adjacent to v in G must belong to G_1.

Hence, G_1 contains at least $d(v)+1 \geq \frac{n-1}{2}+1=\frac{n+1}{2}$ vertices.

Similarly, G_2 contains at least $\frac{n+1}{2}$ vertices.

Therefore G has at least $\frac{n+1}{2}+\frac{n+1}{2}=n+1$ vertices,

which is a contradiction.

Exercise 5.3. Give an example of a nonsimple disconnected graph with $\delta \geq \frac{n-1}{2}$.

Exercise 5.4. Show by means of an example that the condition $\delta \geq \frac{n-2}{2}$ for a simple graph G need not imply that G is connected.

Exercise 5.5. In a group of six people, prove that there must be three people who are mutually acquainted or three people who are mutually nonacquainted.

Our next result shows that of the two graphs G and G^c, at least one of them must be connected.

Theorem 1.5.7. *If a simple graph G is not connected, then G^c is connected.*

Proof:

Let u and v be any two vertices of G^c (and therefore of G).

If u and v belongto different components of G, then obviously u and v are nonadjacent in G and so they are adjacent in G^c.

Thus u and v are connected in G^c.

In case u and v belong tothe same component of G, take a vertex w of G not belonging to this component of G.

Then uw and vw are not edges of G and hence they are edges of G^c.

Then uwv is a $u - v$ path in G^c.

Thus, G^c is connected.

Exercise 5.6. Show that if G is a self-complementary graph of order n then $n \equiv 0$ or $1\ (mod\ 4)$.

Exercise 5.7. Show that if a self-complementary graph contains a pendant vertex, then it must have at least another pendant vertex.

The next theorem gives an upper bound on the number of edges in a simple graph.

Theorem 1.5.8. *The number of edges of a simple graph of order n having ω components cannot exceed* $\frac{(n-\omega)(n-\omega+1)}{2}$.

Proof:

Let $G_1, G_2, \dots, G_\omega$ be the components of a simple graph G and let n_i be the number of vertices of $G_i, 1 \le i \le \omega$.

Then $m(G_i) \le \frac{n_i(n_i-1)}{2}$,

And hence $m(G_i) \le \sum_{i=1}^{\omega} \frac{n_i(n_i-1)}{2}$.

Since $n_i \ge 1$ for each $i, 1 \le i \le \omega$,

$$n_i = n - (n_1 + n_2 + \cdots + n_{i-1} + n_{i+1} + \cdots + n_\omega) \sum_{i=1}^{\omega} (n_i - 1)$$

$$= \frac{(n-\omega+1)}{2} [(\textstyle\sum_{i=1}^{\omega} n_i) - \omega]$$

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

Definition 1.5.9. A graph G is called *locally connected* if, for every vertex v of G, the subgraph $N_G(v)$ induced by the neighbor set of v in G is connected.

A cycle is *odd* or *even* depending on whether its length is odd or even. We now characterize bipartite graphs.

Theorem 1.5.10. *A graph is bipartite if and only if it contains no odd cycles.*

Proof:

Suppose that G is a bipartite graph with the bipartition (X, Y).

Let $C = v_1e_1v_2e_2v_3e_3 \dots v_ke_kv_1$ be a cycle in G.

Without loss of generality, we can suppose that $v_1 \in X$.

As v_2 is adjacent to $v_1, v_2 \in Y$.

Similarly, v_3 belongs to X, v_4 to Y, and so on.

Thus,

$v_i \in X$ or Y according as i is odd or even, $1 \le i \le k$.

Since v_kv_1 is an edge of G and $v_1 \in X, v_k \in Y$.

Accordingly, k is even and C is an even cycle.

Conversely,

Let us suppose that G contains no odd cycles.

We first assume that G is connected.

Let u be a vertex of G.

Define $X = \{v \in V |\ d(u, v)$ is even$\}$ and $Y = \{v \in V |\ d(u, v)$ is odd$\}$.

To prove: (X, Y) is a bipartition of G.

It is enough to show that no two vertices of X and no two vertices of Y are adjacent in G.

Then,

$p = d(u, v)$ and $q = d(u, w)$ are even.

Further,

As $p = d(u, u) = 0, u \in X$.

Let P be a $u - v$ shortest path of length p, and Q, a $u - w$ shortest path of length q. (See Figure 1.19.)

Let w_1 be a vertex common to P and Q such that the $w_1 - v$ section of P and the $w_1 - w$ section of Q contain no vertices common to P and Q.

Then,

$u - w_1$ section of both P and Q have the same length.

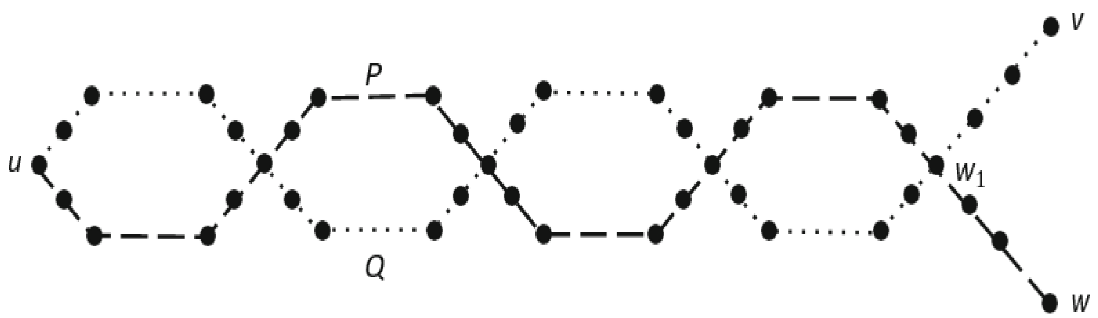


Fig. 1.19 Graph for proof of Theorem 1.5.10

Hence, the length of the $w_1 - v$ section of P and the $w_1 - w$ section of Q are both even or both odd.

Now,

If $e = uw$ is an edge of G, then the $w_1 - v$ section of P folleded by the edge vw and the $w - w_1$ section of $w - u$ path Q^{-1} is an odd cycle in G, contradicting the hypothesis.

This contradiction proves that no two vertices of X are adjacent in G.

Similarly,

no two vertices of Y are adjacent in G.

This proves the result when G is connected.

If G is not connected, let $G_1, G_2, \ldots, G_\omega$ be the components of G.

By hypothesis,

No component of G contains an odd cycle.

Hence,

By previous paragraph,

Each component of $G_i, 1 \leq i \leq \omega$, is bipartite.

Let X_i, Y_i be the bipartition of G_i.

Then,

(X, Y), where $X = \cup_{i=1}^{\omega} X_i$ and $Y = \cup_{i=1}^{\omega} Y_i$, is a bipartition of G, and G is a bipartite graph.

Exercise 5.8. Prove that a simple nontrivial graph G is connected if and only if for any partition of V into two nonempty subsets V_1 and V_2, there is an edge joining a vertex of V_1 to a vertex of V_2

Exercise 5.9. Prove that in a connected graph G with at least three vertices, any two longest paths have a vertex in common.

Exercise 5.10. Prove that in a simple graph G the union of two distinct paths joining two distinct vertices contains a cycle.

Exercise 5.11. Show by means of an example that the union of two distinct walks joining two distinct vertices of a simple graph G need not contain a cycle.

Exercise 5.12. If a simple connected graph G is not complete, prove that there exist three vertices u, v, w of G such that uv and vw are edges of G, but uw is not an edge of G.

1.6 Automorphism of a Simple Graph:

Definition 1.6.1.

An *automorphism* of a graph G is an isomorphism of G onto itself. We recall that two simple graphs G and H are isomorphic if and only if there exists a bijection $\emptyset: V(G) \to V(H)$ such that uv is an edge of G if and only if $\emptyset(u)\emptyset(v)$ is an edge of H. In this case $\emptyset$ is called an isomorphism of G onto H.

We prove in our next theorem that the set $Aut\ (G)$ of automorphisms of G is a group.

Theorem 1.6.2. *The set $Aut\ (G)$ of all automorphisms of a simple graph G is a group with respect to the composition $\circ$ of mappings as the group operation.*

Proof: We shall verify that the four axioms of a group are satisfied by the pair $(Aut\ (G),\ \circ)$.

(i) Let $\emptyset_1$ and $\emptyset_2$ be bijections on $V(G)$ preserving adjacency and nonadjacency.

Clearly, the mapping $\emptyset_1 \circ \emptyset_2$ is a bijection on $V(G)$.

If u and v are adjacent in G, then $\emptyset_2(u)$ and $\emptyset_2(v)$ are adjacent in G.

But $(\emptyset_1 \circ \emptyset_2)(u) = \emptyset_1(\emptyset_2(u))$ and $(\emptyset_1 \circ \emptyset_2)(v) = \emptyset_1(\emptyset_2(v))$.

Hence,

$(\emptyset_1 \circ \emptyset_2)(u)$ and $(\emptyset_1 \circ \emptyset_2)(v)$ are adjacent in G;

that is, $\emptyset_1 \circ \emptyset_2$ preserves adjacency.

A similar argument shows that $\emptyset_1 \circ \emptyset_2$ preserves adjacency.

Thus,

$\emptyset_1 \circ \emptyset_2$ is an automorphism of G.

(ii) It is a well-known result that the composition of mappings of a set onto itself is associative.

(iii) The identity mapping I of $V(G)$ onto itself is an automorphism of G, and it satisfies the condition

$$\emptyset \circ I = I \circ \emptyset = \emptyset \text{ for every } \emptyset \in Aut\ (G).$$

Hence,

I is the identity element of $Aut\ (G)$.

(iv) Finally, if $\emptyset$ is an automorphism of G, the inverse mapping $\emptyset^{-1}$ is also an automorphism of G.

Theorem 1.6.3. *For any simple graph* G, $Aut\ (G) = Aut\ (G^c)$.

Proof:

Since $V(G^c) = V(G)$, every bijection on $V(G)$ is also a bijection on $V(G^c)$.

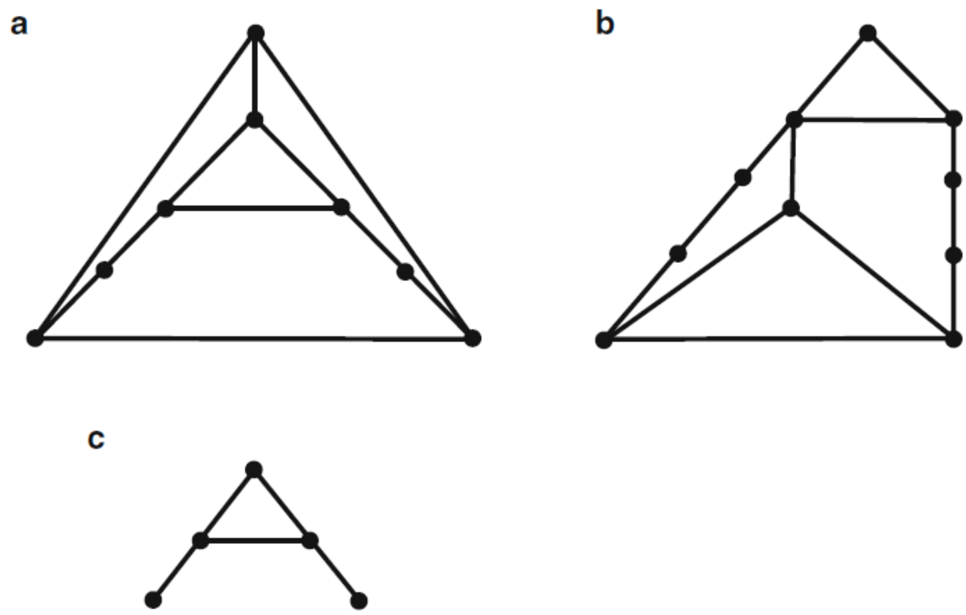
As an automorphism of G preserves the adjacency and nonadjacency of vertices of G it also preserves the adjacency and nonadjacency of vertices of G^c.

Hence,

every element of $Aut\ (G)$ is also an element of $Aut\ (G^c)$, and vice versa.

Exercise 6.1. Show that the automorphism group of K_n (or K_n^c) is isomorphic to the symmetric group S_n of degree n.

Exercise 6.2. Find the automorphism groups of the following graphs:



1.7 Line Graphs:

Let G be a loop less graph. We construct a graph $L(G)$ in the following way:

The vertex set of $L(G)$ is in *1-1* correspondence with the edge set of G and two vertices of $L(G)$ are joined by an edge if and only if the corresponding edges of G are adjacent in G. The graph $L(G)$ (which is always a simple graph) is called *the line graph* or *the edge graph* of G.

Figure 1.22 shows a graph and its line graph in which v_i of $L(G)$ corresponds to the edge e_i of G for each i. Isolated vertices of G do not have any bearing on $L(G)$, and hence we assume in this section that G has no isolated vertices. We also assume that G has no loops.

Some simple properties of the line graph $L(G)$ of a graph G follow:

1. G is connected if and only if $L(G)$ is connected.
2. If H is a subgraph of G, then $L(H)$ is a subgraph of $L(G)$.

3. The edges incident at a vertex of G give rise to a maximal complete subgraph of $L(G)$.
4. If $e = uv$ is an edge of a simple graph G, the degree of e in $L(G)$ is same as the number of edges of G adjacent to e in $L(G)$.

   This number is $d_G(u) + d_G(v) - 2$.

   Hence,

   $$d_{L(G)}(e) = d_G(u) + d_G(v) - 2.$$

5. Finally, if G is a simple graph,

$$\sum_{e \in V(L(G))} d_{L(G)}(e)$$

$$= \sum_{uv \in E(G)} (d_G(u) + d_G(v) - 2)$$

$$= \left[\sum_{u \in V(G)} d_G(u)^2\right] - 2m(G)$$

(Since uv belongs to the stars at u and v)

$$= \left[\sum_{i=1}^{n} d_i^2\right] - 2m$$

where $(d_1, d_2, \ldots, d_n)$ is the degree sequence of G, and $m = m(G)$.

By Euler's theorem,

it follows that the number of edges of $L(G)$ is given by

$$m(L(G)) = \frac{1}{2}\left[\sum_{i=1}^{n} d_i^2\right] - m$$

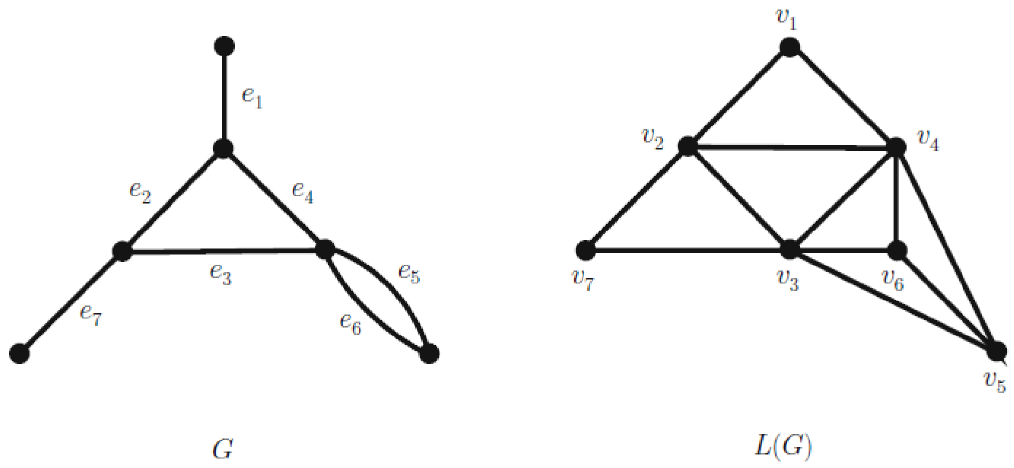


Fig. 1.22 A graph G and its line graph $L(G)$

Exercise 7.1. Show that the line graph of the star $K_{1,n}$ is the complete graph K_n.

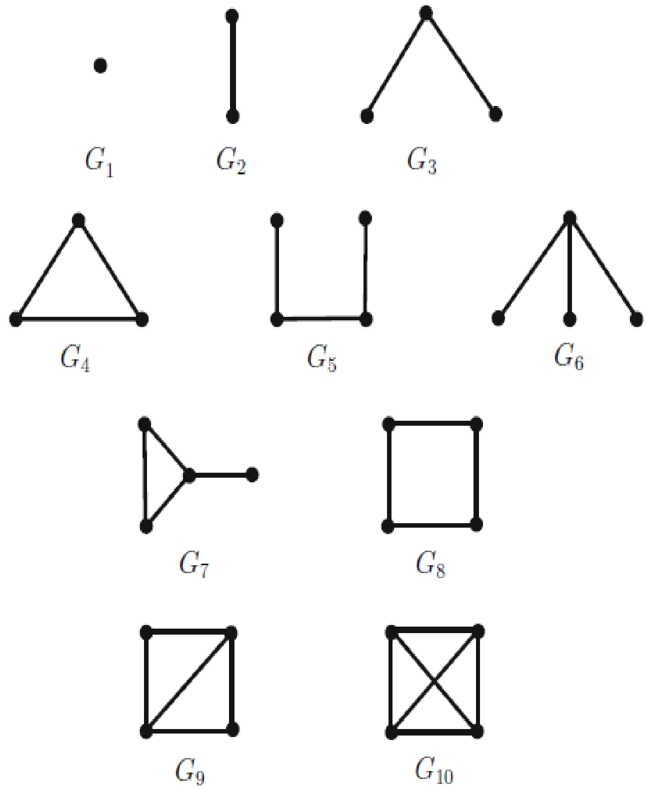
Exercise 7.2. Show that $L(C_n) \cong C_n, n \geq 3$.

Theorem 1.7.1.

The line graph of a simple graph G is a path if and only if G is a path.

Exercise 7.5. Prove that a simple connected graph G is isomorphic to its line graph if and only if it is a cycle.

Fig. 1.23 Nonisomorphic graphs on four vertices or less



Theorem 1.7.4* (H. Whitney).

Let G and G' be simple connected graphs with isomorphic line graphs. Then G and G' are isomorphic unless one of them is $K_{1,3}$ and the other is K_3

Proof:

First, suppose that $n(G)$ and $n(G')$ are less than or equal to 4. A necessary condition for $L(G)$ and $L(G')$ to be isomorphic is that $L(G)$ and $L(G')$. The only non-isomorphic connected graphs on at most four vertices are those shown in Figure 1.23.

In Figure 1.23, graphs G_4, G_5 and G_6 are the three graphs having three edges each.

We have already seen that G_4 and G_6 have isomorphic line graphs, namely, K_3.

The line graph of G_5 is a path of length 2, and hence $L(G_5)$ cannot be isomorphic to $L(G_4)$ or $L(G_6)$.

Further,

G_7 and G_8 are the only two graphs in the list having four edges each.

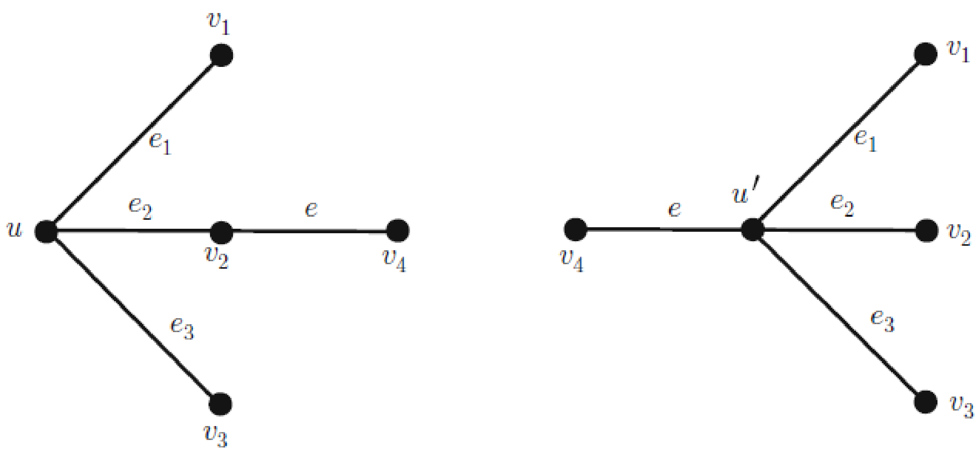


Fig. 1.24 Graphs with five vertices and edge e adjacent to one or all three other edges

Now,

$L(G_8) \cong G_8$, and $L(G_7)$ is isomorphic to G_9.

Thus,

The line graphs G_7 and G_8 are not isomorphic.

No two of the remaining graphs have the same number of edges.

Hence,

the only non-isomorphic graphs with at most four vertices having isomorphic line graphs are G_4 and G_6.

We now suppose that either G or G', say G, has at least five vertices and that $L(G)$ and $L(G')$ are isomorphic under an isomorphism $\emptyset_1$. $\emptyset_1$ is a bijection from the edge set of G onto the edge set of G'.

We now prove that $\emptyset_1$ transforms a $K_{1,3}$ subgraph of G onto a $K_{1,3}$ subgraph of G'.

Let $e_1 = uv_1, e_2 = uv_2$ and $e_3 = uv_3$ be the edges of a $K_{1,3}$ subgraph of G.

As G has at least five vertices and is connected, there exists an edge e adjacent to onlyone or all the three edges e_1, e_2 and e_3 as illustrated in Fig. 1.24.

Now,

$\emptyset_1(e_1)$, $\emptyset_1(e_2)$ and $\emptyset_1(e_3)$ from either a $K_{1,3}$ subgraph or a triangle in G'.

If $\emptyset_1(e_1)$, $\emptyset_1(e_2)$ and $\emptyset_1(e_3)$ from a triangle in G', $\emptyset_1(e)$ can be adjacent to precisely two of $\emptyset_1(e_1)$, $\emptyset_1(e_2)$ and $\emptyset_1(e_3)$ (Since $L(G')$ is simple), whereas $\emptyset_1(e)$ must be adjacent to only one or all the three.

This contradiction shows that $\{\emptyset_1(e_1), \emptyset_1(e_2), \emptyset_1(e_3)\}$ is not a triangle in G' and therefore forms a star at a vertex v' of G'.

It is clear that a similar result holds for ${\emptyset_1}^{-1}$ as well, since it is an isomorphism of $L(G')$ onto $L(G)$.

Let $S(u)$ denote the star subgraph of G formed by the edges of G incident at a vertex u of G.

We shall prove that $\emptyset_1$ maps $S(u)$ onto the star subgraph $S(u')$ of G'.

(i) First, suppose that the degree of u is at least 2.

Let f_1 and f_2 be any two edges incident at u.

The edges $\emptyset_1(f_1)$ and $\emptyset_1(f_2)$ of G' have an end vertex u' in common.

If f is any other edge of G incident with u, then $\emptyset_1(f)$ is incident with u', and conversely, for every edge f' of G' incident with u', $\emptyset_1^{-1}(f')$ is incident with u.

Thus,

$S(u)$ in G is mapped to $S(u')$ in G'.

(ii) Let the degree of u in G be 1 and $e = uv$ be the unique edge incident with u.

As G is connected and $n(G) \geq 5$, degree of v must be at least 2 in G and therefore, by (i), $S(v)$ is mapped to a star $S(v')$ in G'.

Also,

$\emptyset_1(uv) = u'v'$ for some $u' \in V(G')$.

Now,

If the degree of u' in G' is greater than 1, by paragraph (i), the star at u' in G' is transformed by ${\emptyset_1}^{-1}$ either to the star at u in G or to the star at v in G.

But as the star at v in G is mapped to the star at v' in G' by $\emptyset_1$, ${\emptyset_1}^{-1}$ should map the star at u' in G' is transformed by ${\emptyset_1}^{-1}$ either to the star at u in G or to the star at v in G.

But as star at v in G is mapped to the star at v' in G' by $\emptyset_1, {\emptyset_1}^{-1}$ should map the stars u' in G' to the star at u in G only.

As ${\emptyset_1}^{-1}$ is $1-1$, this means that $d_G(u) \geq 2$, a contradiction.

Therefore,

$d_{G'}(u') = 1$ and so $S(u)$ in G is mapped onto $S(u')$ in G'.

Thus,

G and G' are isomorphic under $\emptyset$.

Definition 1.7.5. A graph H is called a *forbidden subgraph* for a property P of graphs if it satisfies the following condition: If a graph G has property P, then G cannot contain an induced subgraph isomorphic to H.

Beineke [17] obtained a forbidden-subgraph criterion for a graph to be a line graph.

In fact, he showed that a graph G is a line graph if and only if the nine graphs of Figure 1.25 are forbidden subgraphs for G.

However, for the sake of later reference, we prove only the following result.

Theorem 1.7.6. *If* G *is a line graph, then* $K_{1,3}$ *is a forbidden subgraph of* G

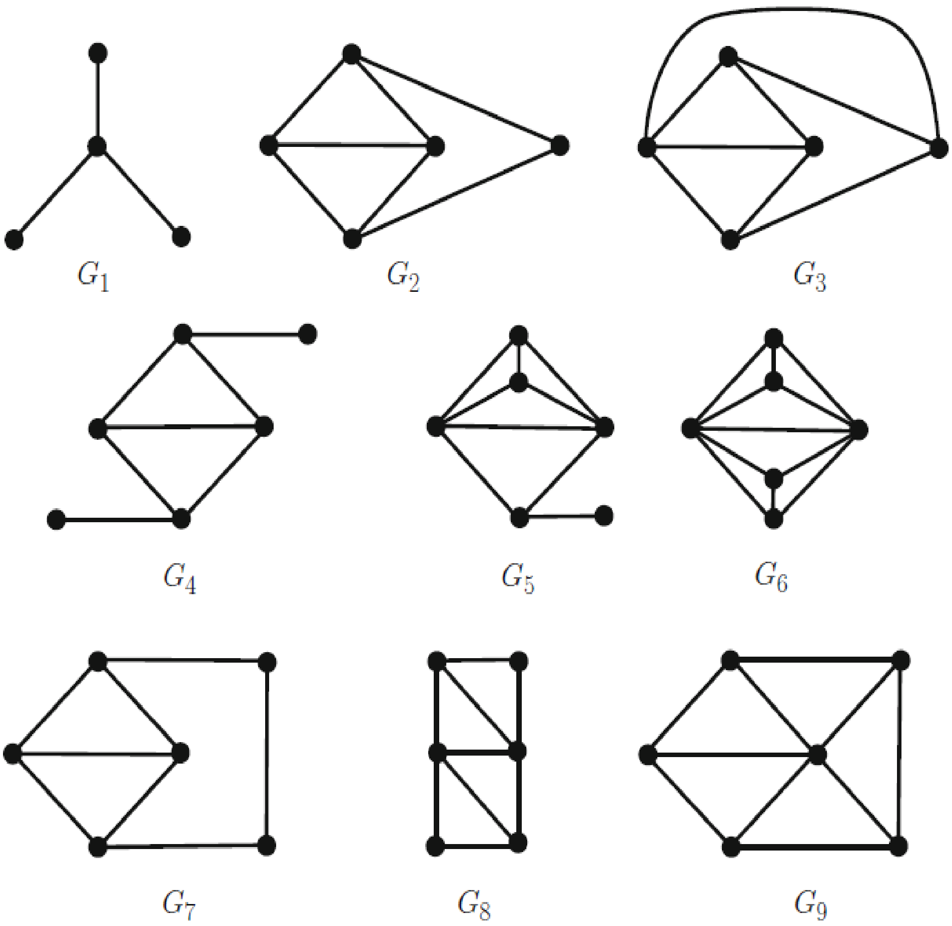


Fig. 1.25 Nine graphs of Bieneke [17]

1.8 Operations on Graphs:

In this section we consider some of the methods of generating new graphs from a given pair of graphs.

Let $G_1 = (V_1,\ E_1)$ and $G_2 = (V_2,\ E_2)$ be two simple graphs.

Definition 1.8.1.

Union of two graphs:

The graph $G = (V, E)$, where $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$ is called the union of G_1 and G_2 and is denoted by $G_1 \cup G_2$.

When G_1 and G_2 are vertex disjoint, $G_1 \cup G_2$is denoted by $G_1 + G_2$ and is called the *sum* of the graphs G_1 and G_2:

The finite union of graphs is defined by means of associativity; in particular, if $G_1, G_2, \ldots, G_r$ are pairwise vertex-disjoint graphs, each of which is isomorphic to G then $G_1 + G_2, + \ldots + G_r$ is denoted by rG.

Definition 1.8.2.

***Intersection of two graphs*:**

If $V_1 \cap V_2 \neq 0$, the graph $G = (V, E)$ where $V = V_1 \cap V_2$ and $E = E_1 \cap E_2$ is the *intersection* of G_1 and G_2 and is written as $G_1 \cap G_2$.

Definition 1.8.3.

—

***Join of two graphs*:**

Let G_1 and G_2 be two *vertex-disjoint* graphs. Then the *join* $G_1 \vee G_2$ of G_1 and G_2 is the super graph of $G_1 + G_2$ in which each vertex of G_1 is also adjacent to every vertex of G_2.

Figure 1.26 illustrates the graph $G_1 \vee G_2$. If $G_1 = K_1$ and $G_2 = C_n$, then $G_1 \vee G_2$ is called the *wheel* W_n. W_5 is shown in Fig. 1.27.

Fig. 1.26 $G_1 \vee G_2$

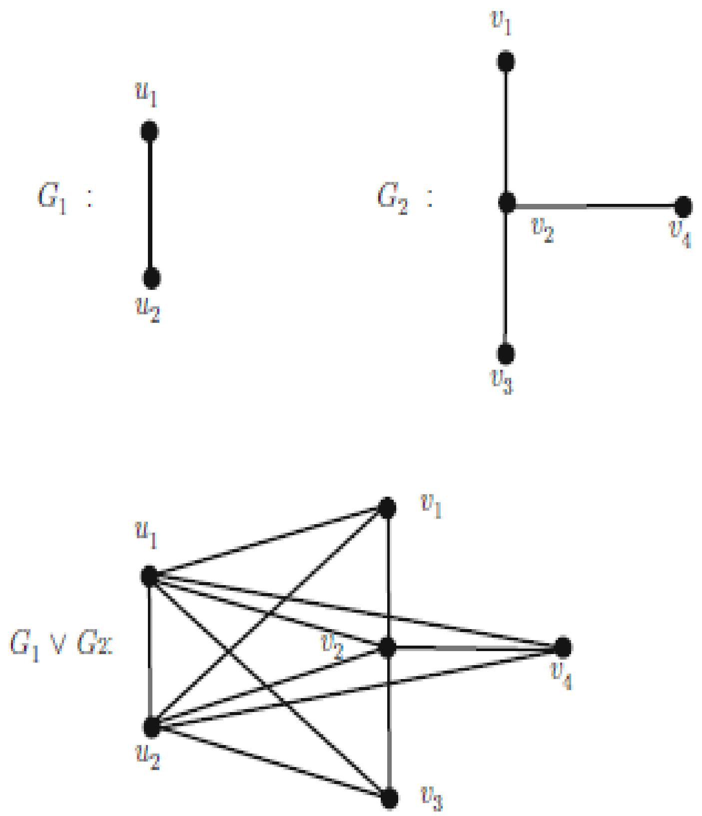
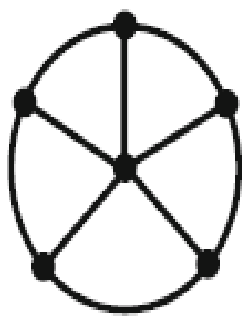


Fig. 1.27 Wheel W_5



1.9 Graph Products:

We now define graph products. Denote a general graph product of two *simple* graphs by $G * H$. We define the product in such a way that $G * H$ is also simple. Given graphs G_1 and G_2 with vertex sets V_1 and V_2 respectively, any product graph $G_1 * G_2$ has as its vertex set the Cartesian product $V(G_1) \times V(G_2)$.

For any two vertices

$(u_1, u_2), (v_1, v_2)$ *of* $G_1 * G_2$, consider the following possibilities:

(i) u_1 adjacent to v_1 in G_1 or u_1 non-adjacent to v_1 in G_1.

(ii) u_2 adjacent to v_2 in G_2 or u_2 $n0n$-adjacent to v_2 in G_2.

(iii) $u_1 = v_1$ and/or $u_2 = v_2$.

Definition 1.9.1:

Cartesian product, $G_1 \square G_2$ is defined by

(u_1, u_2) and (v_1, v_2) are adjacent in $G_1 \square G_2$ if and only if either $u_1 = v_1$ and u_2 is adjacent to v_2 in G_2, or u_1 is adjacent to v_1 in G_1 and $u_2 = v_2$.

Definition 1.9.2:

Direct (or tensor or Kronecker) product, $G_1 \times G_2$ is defined by

(u_1, u_2) is adjacent to (v_1, v_2) in $G_1 \times G_2$ if and only if either u_1 is adjacent to v_1 in G_1 and u_2 is adjacent to v_2 in G_2 and $u_2 = v_2$.

Definition 1.9.3:

Composition (or wreath or lexicographic) product, $G_1[G_2]$ is defined by

(u_1, u_2) is adjacent to (v_1, v_2) $G_1[G_2]$ if and only if u_1 is adjacent to v_1 in G_1, or $u_1 = v_1$, and u_2 is adjacent to v_2 in G_2.

Definition 1.9.4:

Strong (or normal) product, $G_1 \boxtimes G_2$.

By definition,

$G_1 \boxtimes G_2 = (G_1 \square G_2) \cup (G_1 \times G_2)$.

Example 1.9.5:

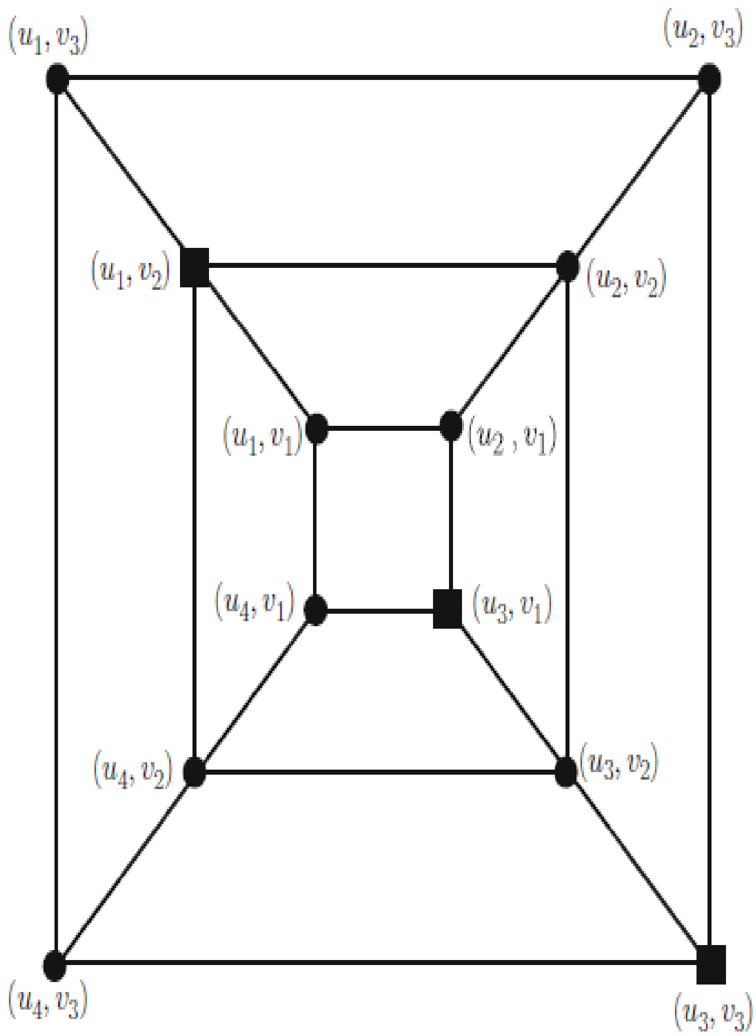
Example of Cartesian product:

Let $G_1 = C_4 = (u_1\, u_2\, u_3\, u_4)$ and

$G_2 = P_3 = (v_1\, v_2\;\; v_3)$.

Then, $G_1 \square G_2$ is the graph G_3 given in Figure 1.28.

Fig. 1.28 $G_3 = C_4 \square P_3$



Example 1.9.6:

If G_1 and G_2 are graphs in Figure 2.26, $G_1[G_2]$ and $G_2[G_1]$ are shown in Figures-1.29 and 1.30, respectively.

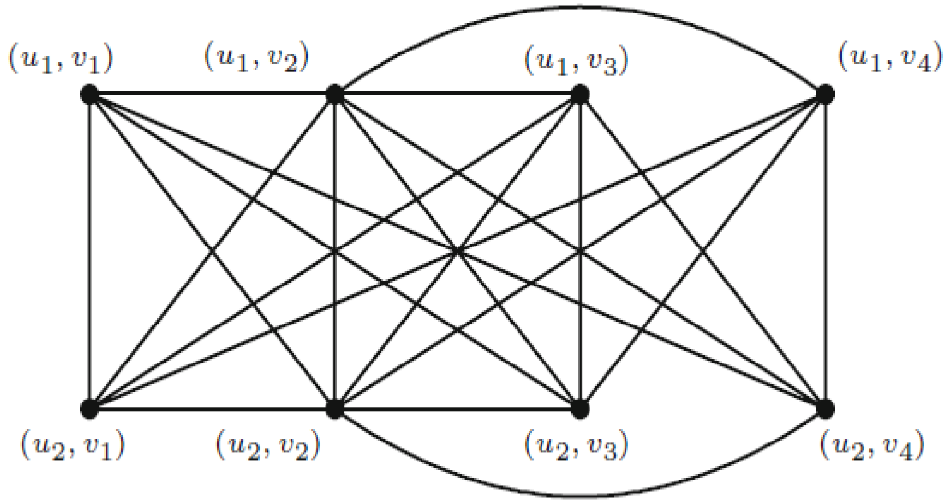
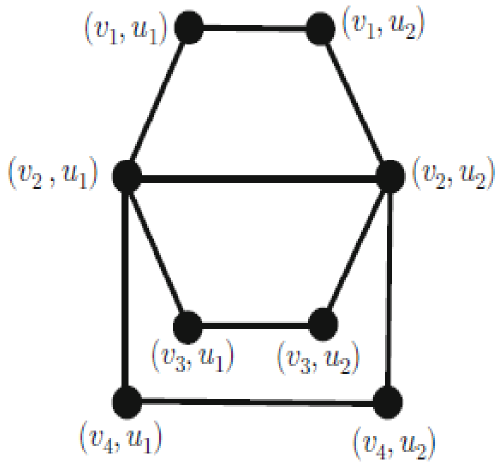


Fig. 1.29 $G_1[G_2]$

Fig. 1.30 $G_2[G_1]$



Example 1.9.7:

If G_1 and G_2 are graphs in Figure 1.26, $G_1 \boxtimes G_2$ aand $G_1 \times G_2$ are shown in Figure 1.31.

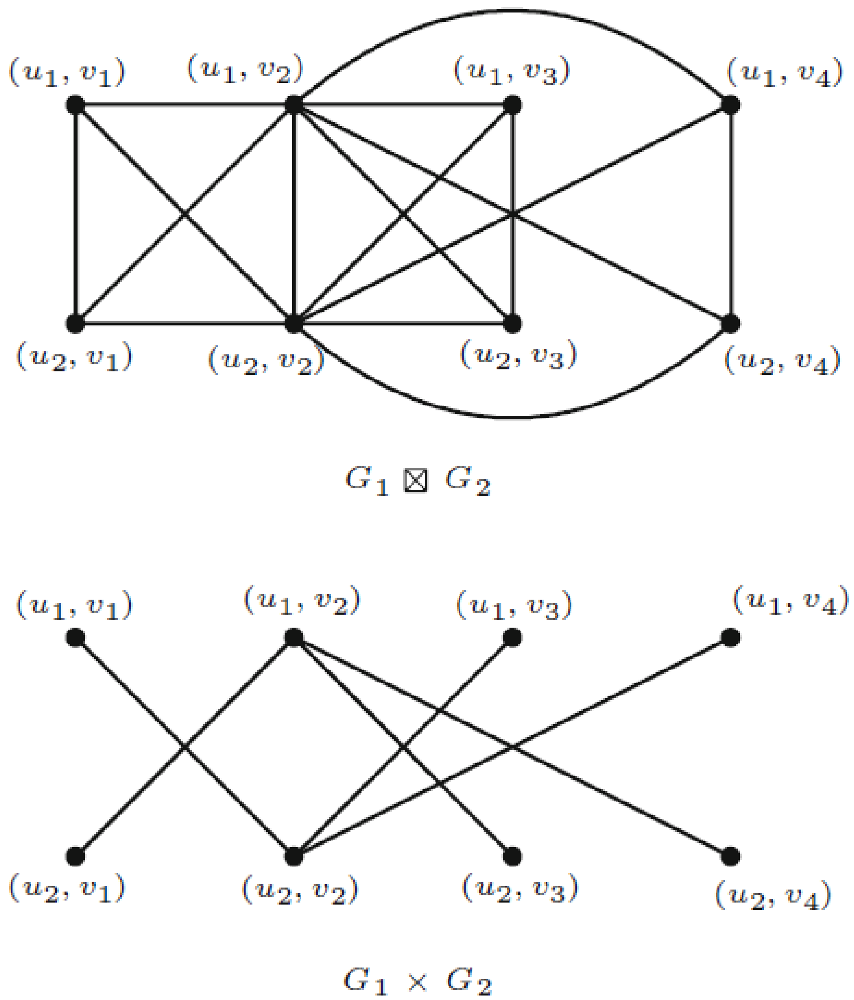


Fig. 1.31 $G_1 \boxtimes G_2$ and $G_1 \times G_2$

Exercise 9.1:

Prove that $G_1 \square G_2 \cong G_2 \square G_1$.

Unit-II:

Connectivity: Vertex Cuts and Edge Cuts - Connectivity and Edge Connectivity - Blocks.

Chapter 3: Section 3.1 to 3.4.

2. Directed Graphs:

2.1. Introduction:

Directed graphs arise in a natural way in many applications of graph theory. The street map of a city, an abstract representation of computer programs, and network flows can be represented only by directed graphs rather than by graphs. Directed graphs are also used in the study of sequential machines and system analysis in control theory.

2.2. Basic Concepts:

Definition 2.2.1.

A *directed graph* D is an ordered triple $(V(D), A(D), I_D)$ where $V(D)$ is a nonempty set called the set of *vertices* of D; $A(D)$ is a set disjoint from $V(D)$, called the set of *arcs* of D and I_D is an *incidence map* that associates with each arc of D an ordered pair of vertices of D. If a is an arc of D, and $I_D(a) = (u, v)$, u is called the *tail* of a, and v is the *head* of a. The arc a is said to join v with u. u and v are called the *ends* of a. A directed graph is also called a *digraph*.

With each digraph D, we can associate a graph G (written $G(D)$ when reference to D is needed) on the same vertex set as follows: Corresponding to each arc of D, there is an edge of G with the same ends. This graph G is called the *underlying graph* of the digraph D. Thus, every digraph D defines a unique (up to isomorphism) graph G. Conversely, given any graph G, we can obtain a digraph from G by specifying for each edge of G an order of its ends. Such a specification is called an *orientation* of G.

A digraph is represented by a diagram of its underlying graph together with arrows on its edges, the arrow pointing toward the head of the corresponding arc. A digraph and its underlying graph are shown in Figure 2.1.

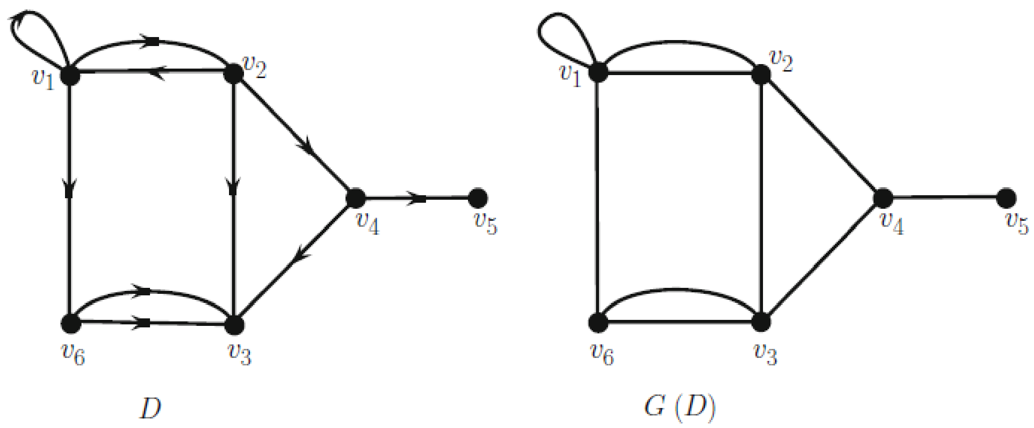


Fig. 2.1 Digraph D and its underlying graph $G(D)$

3. Connectivity:

3.1. Introduction:

The connectivity of a graph is a “measure” of its connectedness. Some connected graphs are connected rather “loosely” in the sense that the deletion of a vertex oran edge from the graph destroys the connectedness of the graph. There are graphs at the other extreme as well, such as the complete graphs $K_n, n \geq 2$, which remain connected after the removal of any k vertices, $1 \leq k \leq n-1$.

Consider a communication network. Any such network can be represented by a graph in which the vertices correspond to communication centers and the edges represent communication channels. In the communication network of Fig. 3.1a, any disruption in the communication center v will result in a communication breakdown, whereas in the network of Fig. 3.1b, at least two communication centers have to be disrupted to cause a breakdown. It is needless to stress the importance of maintaining reliable communication networks at all times, especially during times of war, and the reliability of a communication network has a direct bearing on its connectivity.

In this chapter, we study the two graph parameters, namely, vertex connectivity and edge connectivity. We also introduce the parameter cyclical edge connectivity. We prove Menger's theorem and several of its variations. In addition, the theorem of Ford and Fulkerson on flows in networks is established.

3.1 Vertex Cuts and Edges Cuts:

We now introduce the notions of vertex cuts, edge cuts, vertex connectivity, and edge connectivity.

Definitions 3.2.1.

1. A subset V' of the vertex set $V(G)$ of a connected graph G is a *vertex cut* of G if $G - V'$ is disconnected; it is a *k-vertex cut* if $|V'| = k$. V' is then called a *separating set of vertices* of G. A vertex v of G is a *cut vertex* of G if $\{v\}$ is a vertex cut of G.

2. Let G be a nontrivial connected graph with vertex set $V(G)$ and let S be a non-empty subset of $V(G)$. For $\bar{S} = V \backslash S \neq 0$, let $[S, \ \bar{S}]$ denote the set of

all edges of G that have one end vertex in S and the other in $\bar{S}$. A set of edges of G of the form $[S, \bar{S}]$ is called an *edge cut* of G. An edge e is a *cut edge* of G if $\{e\}$ is an edge cut of G. An edge cut of cardinality k is called a *k-edge cut* of G.

Fig. 3.1 Two types of communication networks

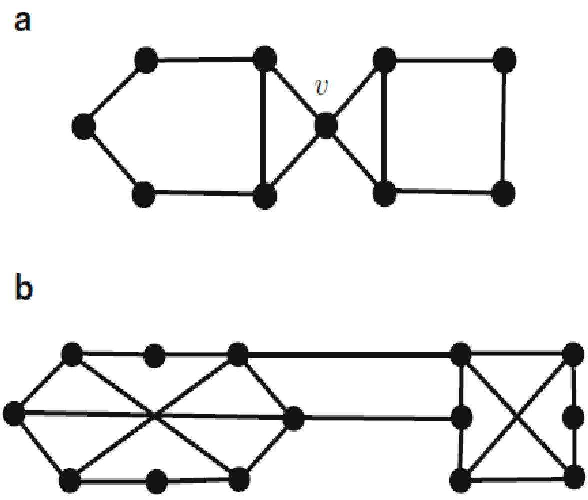
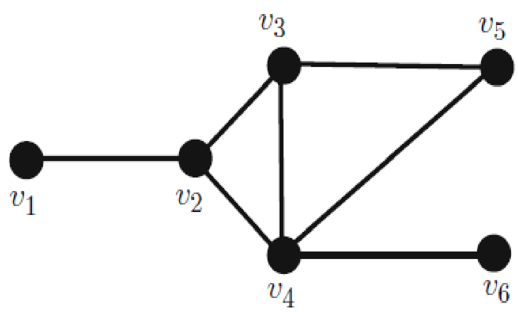


Fig. 3.2 Graph illustrating vertex cuts and edge cuts



Example 3.2.2.

For the graph of Figure 3.2, $\{v_2\}$, and $\{v_3, v_4\}$ are vertex cuts.

The edge subsets $\{v_3v_5, v_4v_5\}$, $\{v_1v_2\}$, and $\{v_4v_6\}$ are all edge cuts.

Of these, v_2 is a cut vertex, and v_1v_2 and v_4v_6 are both cut edges.

For the edge cut $\{v_3v_5, v_4v_5\}$, we may take $S = \{v_5\}$ so that

$\bar{S} = \{v_1, v_2, v_3, v_4, v_5\}$.

Remarks 3.2.3:

1. If uv is an edge of an edge cut E'; then all the edges having u and v as their ends also belong to E'.
2. No loop can belong to an edge cut.

Exercise 2.1.

If $\{x, y\}$ is a 2-edge cut of a graph G, show that every cycle of G that contains x must also contain y.

Theorem 3.2.4:

A vertex v of a connected graph G with at least three vertices is a cut vertex of G if and only if there exist vertices u and w of G distinct from v such that v is in every $u - w$ path in G.

Proof:

If v is a cut vertex of G,

then $G - v$ is disconnected and has at least two components, G_1 and G_2.

Take $u \in V(G_1)$ and $v \in V(G_2)$.

Then every $u - w$ path in G must contain v, as otherwise u and w would belong to the same component of $G - v$.

Conversely,

Suppose that the condition of the theorem holds.

Then,

the deletion of v destroys every $u - w$ path in G,

and hence u and w lie in distinct components of $G - v$.

Therefore,

$G - v$ is disconnected and v is a cut vertex of G.

Theorems 3.2.5 and 3.2.6 characterize a cut edge of a graph.

Theorem 3.2.5:

An edge $e = xy$ of a connected graph G is a cut edge of G if and only if e *belongs to no cycle of* G.

Proof:

Let e be a cut edge of G and let $[S, \bar{S}] = \{e\}$ be the partition of V defined by $G - e$ so that one of x and y belongs to S and the other to $\bar{S}$, say, $x \in S$ and $y \in \bar{S}$.

If e belongs to a cycle of G, then $[S, \bar{S}]$ must contain at least one more edge, contradicting that $\{e\} = [S, \bar{S}]$.

Hence,

e cannot belong to a cycle.

Conversely,

Assume that e is not a cut edge of G.

Then,

$G - e$ is connected,

and hence,

there exists an $x - y$ path P in $G - e$.

Then,

$P \cup \{e\}$ is a cycle in G containing e.

Theorem 3.2.6:

An edge $e = xy$ is a cut edge of a connected graph G if and only if there exist vertices u and v such that e belongs to every $u - v$ path in G.

Proof:

Let $e = xy$ be a cut edge of G

Then,

$G - e$ has two components, say, G_1 and G_2.

Let $u \in V(G_1)$ and $v \in V(G_2)$.

Then,

clearly, every $u - v$ path in G contains e.

Conversely,

Suppose that there exist vertices u and v satisfying the condition of the theorem.

Then,

there exists no $u - v$ path in $G - e$ so that $G - e$ is disconnected.

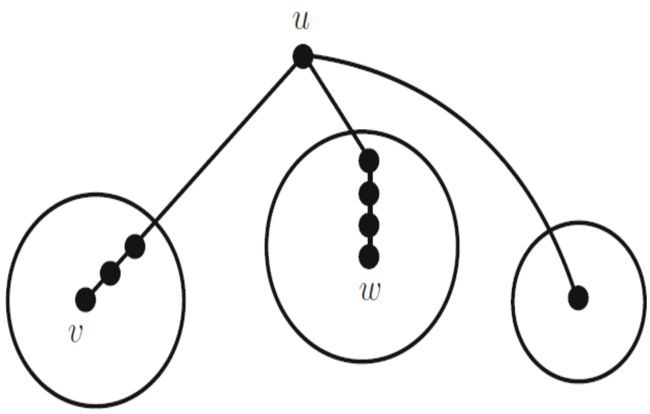
Hence,

e is a cut edge of G.

Theorem 3.2.7:

A connected graph G with at least two vertices contains at least two vertices that are not cut vertices.

Fig. 3.3 Graph for proof of Theorem 3.2.10



Proof:

First, suppose that $n(G) \geq 3$.

Let u and v be vertices of G such that $d(u, v)$ is maximum.

Then,

neither u nor v is a cut vertex of G.

For if u were a cut vertexof G, $G - u$ would be disconnected, having at least two components.

The vertex v belongs to one of these components.

Let w be any vertex belonging to a component of $G - u$ not containing v.

Then,

every $v - w$ path in G must contain u (see Figure 3.3).

Consequently,

$d(v, w) > d(v, u)$, contradicting the choice of u and v.

Hence,

u is not a cut vertex of G.

Similarly,

v is not a cut vertex of G.

If $n(G) = 2$, then K_2 is a spanning subgraph of G,

and so no vertex of G is a cut vertex of G.

This completes the proof of the theorem.

Exercise 2.1.

Find the vertex cuts and edge cuts of the graph Figure 3.2.

Exercise 2.3.

Prove or disprove: Let G be a simple connected graph with $n(G) \geq 3$. Then G has a cut edge if and only if G has a cut vertex.

Exercise 2.4. Show that in a graph, the number of edges common to a cycle and an edge cut is even.

3.3 Connectivity and Edge Connectivity:

We now introduce two parameters of a graph that in a way measure the connected-ness of the graph.

Definition 3.3.1.

For a nontrivial connected graph G having a pair of non-adjacent vertices, the minimum k for which there exists a k-vertex cut is called the *vertex connectivity* or simply the *connectivity* of G it is denoted by $\boldsymbol{k}(\mathrm{G})$ or simply $\boldsymbol{k}$ (kappa) when G is understood.

If G is trivial or disconnected, $\boldsymbol{k}(\mathrm{G})$ is taken to be zero, whereas if G contains K_n as a spanning subgraph, $\boldsymbol{k}(\mathrm{G})$ is taken to be $n-1$.

A set of vertices and/or edges of a connected graph G is said to *disconnect* G if its deletion results in a disconnected graph.

When a connected graph G (on $n \geq 3$ vertices) does not contain K_n as a spanning subgraph, $\boldsymbol{k}$ is the connectivity of G if there exists a set of $\boldsymbol{k}$ vertices of G whose deletion results in a disconnected subgraph of G while no set of $k-1$ (or fewer) vertices has this property.

Exercise 3.1:

Prove that a simple graph G with n vertices, $n \geq 2$; is complete if and only if $k(\mathrm{G}) = n-1$.

Definition 3.3.2:

The *edge connectivity* of a connected graph G is the smallest k for which there exists a k-edge cut (i.e., an edge cut having k edges). The edge connectivity of a trivial or disconnected graph is taken to be 0. The edge connectivity of G is denoted by $\lambda(G)$. If λ is the edge connectivity of a connected graph G, there exists a set of λ edges whose deletion results in a disconnected graph, and no subset of edges of G of size less than λ has this property.

Exercise 3.2. Prove that the deletion of edges of a minimum-edge cut of a connected graph G results in a disconnected graph with exactly two components. (Note that a similar result is not true for a minimum vertex cut.)

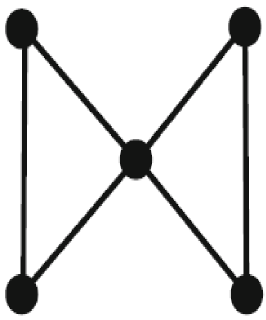
Definition 3.3.3:

A graph G is *r-connected* if $k(G) \geq r$. Also, G is r-edge connected if $\lambda(G) \geq r$.

An r-connected (respectively, r-edge-connected) graph is also ℓ-connected (respectively, ℓ-edge connected) for each ℓ, $0 \leq \ell \leq r - 1$.

For the graph G of Figure 3.5, $k(G) = 1$ and $\lambda(G) = 2$.

Fig. 3.5 A 1-connected graph



G

We now derive inequalities connecting $k(G), \lambda(G)$ and $\delta(G)$.

Theorem 3.3.4.

For any loop-less connected graph G; $k(G) \leq \lambda(G) \leq \delta(G)$.

Proof:

We observe that $k = 0$ if and only if $\lambda = 0$.

Also,

$\delta = 0$ implies that $k = 0$ and $\lambda = 0$.

Hence,

we may assume that k, λ and δ are all at least 1.

Let $\mathcal{E}$ be an edge cut of G with λ edges.

Let u and v be the end vertices of an edge of $\mathcal{E}$.

For each edge of $\mathcal{E}$ that does not have both u and v as end vertices, remove an end vertex that is different from u and v.

If there are t such edges, at most t vertices have been removed.

If the resulting graph, say H, is disconnected, then $k \leq t < \lambda$.

Otherwise,

there will remain a subset of edges of E having u and v as end vertices, the removal of which from H would disconnect G.

Hence,

in addition to the already removed vertices, the removal of one of u and v will result in either a disconnected graph or a trivial graph.

In the process,

a set of at most $t+1$ vertices has been removed and $k \leq t+1 \leq \lambda$.

Finally,

it is clear that $\lambda \leq \delta$.

In fact,

if v is a vertex of G with $d_G(v) = \delta$, thenthe set$[\{v\}, V \setminus \{v\}]$ of δ edges of G incident at v forms an edge cut of G.

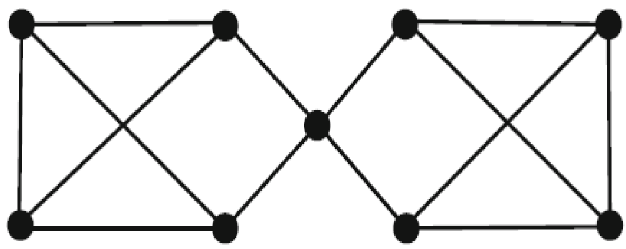
Thus,

$$\lambda \leq \delta.$$

Note:

It is possible that the inequalities in Theorem 3.3.4 can be strict. See the graph G of Figure 3.6, for which $k = 1,\ \lambda = 2$, and $\delta = 3$.

Fig. 3.6 Graph G with $\kappa = 1$, $\lambda = 2$ and $\delta = 3$



G

Exercise 3.3:

Prove or disprove: If H is a subgraph of G, then

(i) $k(H) \leq k(G)$ and

(ii) $\lambda(H) \leq \lambda(G)$.

Exercise 3.4:

Determine $\lambda(K_n)$.

Theorem 3.3.7 (Whitney [193]). *A graph G with at least three vertices is* 2-*connected if and only if any two vertices of G are connected by at least two internally disjoint paths.*

Proof:

Let G be 2-connected.

Then,

G contains no cut vertex.

Let u and v be two distinct vertices of G.

We now use induction on $d(u, v)$ to prove that u and v are joined by two internally disjoint paths.

If $d(u, v) = 1$,

let $e = uv$.

As G is 2-connected and $n(G) \geq 3$, e cannot be a cut edge of G, since if e were a cut edge, at least one of u and v must be a cut vertex.

By Theorem 3.2.5,

e belongs to a cycle C in G.

Then,

$C - e$ is a u-v path in G, internally disjoint from the path uv.

Now,

assume that any two vertices x and y of G with $d(x, y) = k - 1, k \geq 2$, are joined by two internally disjoint x-y paths in G.

Let $d(u, v) = k$.

Let P be a u-v path of length k and w be the vertex of G just preceding v on P. Then,

$$d(u, w) = k - 1.$$

By an induction hypothesis,

there are two internally disjoint u-w paths, say P_1 and P_2, in G.

As G has no cut vertex, $G - w$ is connected and hence there exists a u-v path Q in $G - w$.

Q is clearly a u-v path in G not containing w.

Let x be the vertex of Q such that the x-v section of Q contains only the vertex x in common with $P_1 \cup P_2$ (see Figure 3.8).

We may suppose,

without loss of generality, that x belongs to P_1.

Then,

the union of the u-x section of P_1 and x-v section of Q and $P_2 \cup (wv)$ are two internally disjoint u-v paths in G.

This gives the proof in one direction.

In the other direction,

assume that any two distinct vertices of G are connected by at least two internally disjoint paths.

Then,

G is connected.

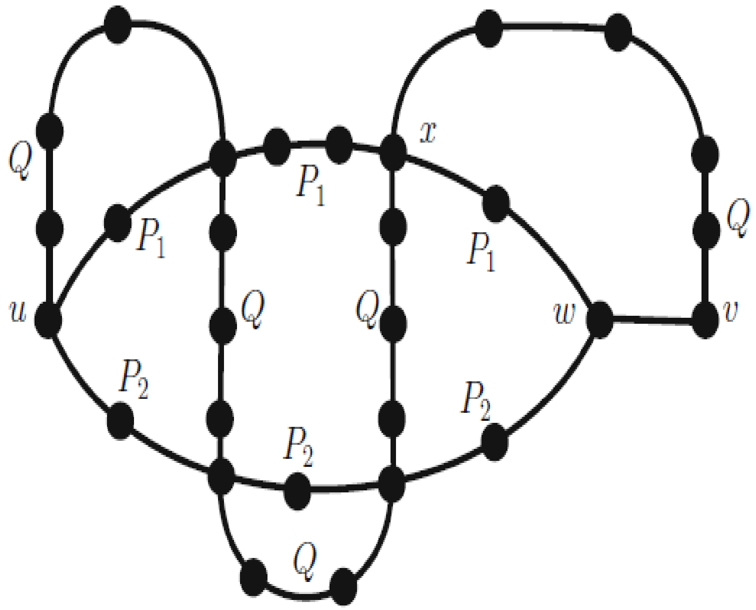
Further,

G cannot contain a cut vertex, since if v were a cut vertex of G, there must exist vertices u and w such that every u-w path contains v (compare with Theorem 3.2.4), contradicting the hypothesis.

Hence,

G is 2-connected.

Fig. 3.8 Graph for proof of Theorem 3.3.7



3.4 Blocks:

In this section, we focus on connected graphs without cut vertices.

Definition 3.4.1:

A graph G is *non-separable* if it is nontrivial and connected and has no cut vertices. A *block of a graph* is a maximal non-separable subgraph of G. If G has no cut vertex, G itself is a block.

In Figure 3.12,

a graph G and its blocks B_1, B_2, B_3, and B_4 are indicated.

B_1, B_3, and B_4 are the *end blocks* of G

(i.e., blocks having exactly one cut vertex of G).

The following facts are worthy of observation.

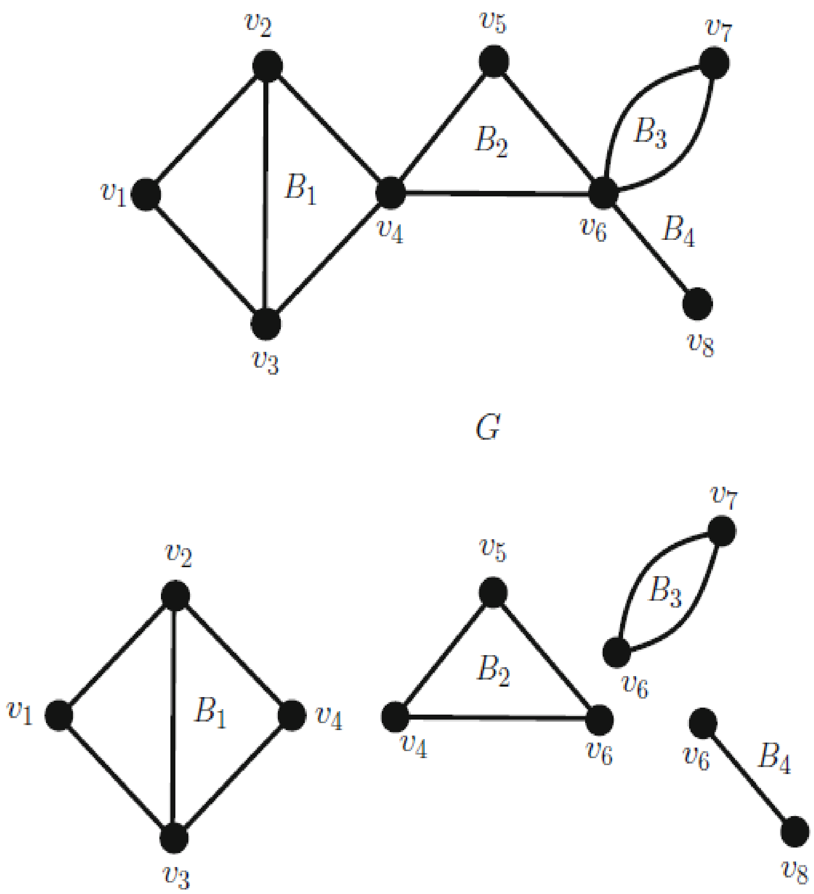


Fig. 3.12 A graph G and its blocks

Remarks 3.4.2:

Let G be a connected graph with $n \geq 3$.

1. Each block of G with at least three vertices is a 2-connected subgraph of G:
2. Each edge of G belongs to one and only one of its blocks.

   Hence G is an edge-disjoint union of its blocks.
3. Any two blocks of G have at most one vertex in common.

   (Such a common vertex is a cut vertex of G.)
4. A vertex of G that is not a cut vertex belongs to exactly one of its blocks.
5. A vertex of G is a cut vertex of G if and only if it belongs to at least two blocks of G.

Whitney's theorem (Theorem 3.3.7) implies that a graph with at least three vertices is a block if and only if any two vertices of the graph are connected byat least two internally disjoint paths.

Also, any two vertices of a block with at least three vertices belong to a common cycle.

Thus, a block with at least three vertices contains a cycle.

UNIT-III:

Trees: Definition, Characterization and simple properties - Centres and centroids - Counting the number of Spanning Trees - Cayley's formula.

Chapter 4: Section 4.1 to 4.5.

4.1 Introduction:

"Trees" form an important class of graphs. Of late, their importance has grown considerably in view of their wide applicability in theoretical computer science.

In this chapter, we present the basic structural properties of trees, their centers and centroids. In addition, we present two interesting consequences of the Tutte– Nash–Williams theorem on the existence of k pairwise edge-disjoint spanning trees in a simple connected graph. We also present Cayley's formula for the number of spanning trees in the labeled complete graph K_n.

4.2 Definition, Characterization, and Simple Properties:

Certain graphs derive their names from their diagrams. A "tree" is one such graph. Formally, a connected graph without cycles is defined as a *tree.* A graph without cycles is called an *acyclic graph* or a *forest.* So, each component of a forest is a tree. A forest may consist of just a single tree! Figure 4.1 displays two pairs of isomorphic trees.

Remarks 4.2.1:

1. It follows from the definition that a forest (and hence a tree) is a simple graph.
2. A subgraph of a tree is a forest and a connected subgraph of a tree T is a *subtree* of T.

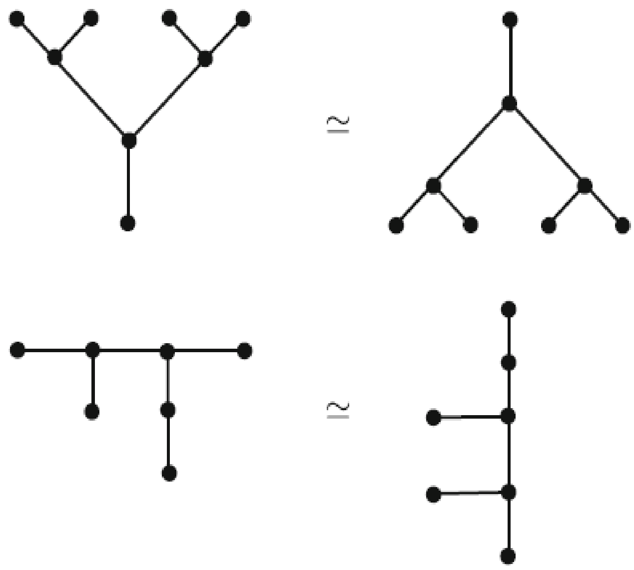


Fig. 4.1 Examples of isomorphic trees

In a connected graph,

any two distinct vertices are connected by at least one path.

Trees are precisely those simple connected graphs in which every pair of distinct vertices is joined by a unique path.

Theorem 4.2.2. *A simple graph is a tree if and only if any two distinct vertices are connected by a unique path.*

Proof:

Let T be a tree.

Suppose that two distinct vertices u and v are connected by two distinct u-v paths.

Then,

their union contains a cycle in T,

contradicting that T is a tree.

Conversely,

suppose that any two vertices of a graph G are connected by a unique path.

Then,

G is obviously connected.

Also,

G cannot contain a cycle, since any two distinct vertices of a cycle are connected by two distinct paths.

Hence,

G is a tree.

Note:

A spanning subgraph of a graph G, which is also a tree, is called a *spanning tree* of G. A connected graph G and two of its spanning trees T_1 and T_2 are shown in Figure 4.2.

The graph G of Figure 4.2 shows that a graph may contain more than one spanning tree; each of the trees T_1 and T_2 is a spanning tree of G.

A loop cannot be an edge of any spanning tree, since such a loop constitutes a cycle (of length 1). On the other hand, a cut edge of G must be an edge of every spanning tree of G. Theorem 4.2.3 shows that every connected graph contains a spanning tree.

Theorem 4.2.3. *Every connected graph contains a spanning tree.*

Proof:

Let G be a connected graph.

Let $\mathscr{J}$ be the collection of all connected spanning subgraphs of G.

$\mathscr{J}$ is nonempty as $G \in \mathscr{J}$.

Let $T \in \mathscr{J}$ have the fewest number of edges.

Then,

T must be a spanning tree of G.

If not,

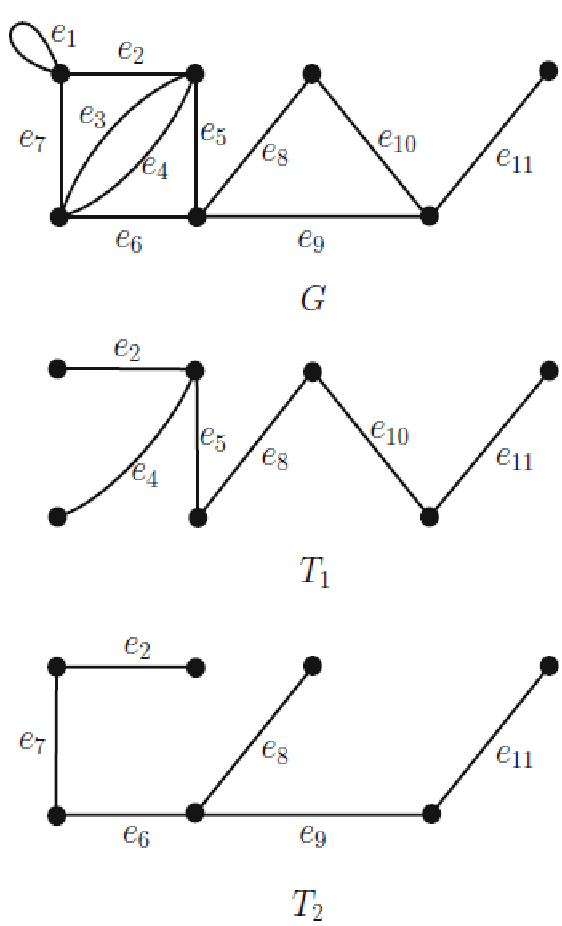
T would contain a cycle of G, and the deletion of any edge of this cycle would give a (spanning) subgraph in $\mathcal{J}$ having one edge less than that of T.

This contradicts the choice of T.

Hence,

T has no cycles and is therefore a spanning tree of G.

Fig. 4.2 Graph G and two of its spanning trees T_1 and T_2



Now, we will see that, there is a relation between the number of vertices and the number of edges of any tree.

Theorem 4.2.7. *A connected graph G is a tree if and only if every edge of G is a cut edge of G.*

Proof:

If G is a tree, there are no cycles in G.

Hence,

no edge of G can belong to a cycle.

By Theorem 3.2.7,

each edge of G is a cut edge of G.

Conversely,

if every edge of a connected graph G is a cut edge of G, then G cannot contain a cycle, since no edge of a cycle is a cut edge of G.

Hence,

G is a tree.

Exercise 2.7. Prove that the following statements are equivalent:

1. G is connected and unicyclic (i.e., G has exactly one cycle).
2. G is connected and $n = m$.
3. For some edge e of G, $G - e$ is a tree.
4. G is connected and the set of edges of G that are not cut edges forms a cycle.

4.3 Centers and Centroids:

Definitions 4.3.1. Let G be a connected graph.

1. The *diameter* of G is defined as max $\{d(u,v): u,v \in V(G)\}$ and is denoted by $diam\ (G)$.
2. If v is a vertex of G, its *eccentricity* $e(v)$ is defined by $e(v) = \max\{d(v,u)\}: u \in V(G)\}$.

3. The *radius* $r(G)$ of G is the minimum eccentricity of G;

   that is, $r(G) = \min\{e(v): v \in V(G)\}$.

   Note that $diam\ (G) = \max\{e(v): v \in V(G)\}$.

4. A vertex v of G is called a *central vertex* if $e(v) = r(G)$.

   The set of central vertices of G is called the *center* of G.

Example 4.3.2:

Figure 4.3 displays two graphs T and G with the eccentricities of their vertices.

We find that $r(T) = 4$ and $diam(T) = 7$. Each of u and v is a central vertex of T.

Also,

$r(G) = 3$ and $diam(G) = 4$. Further, G has five central vertices.

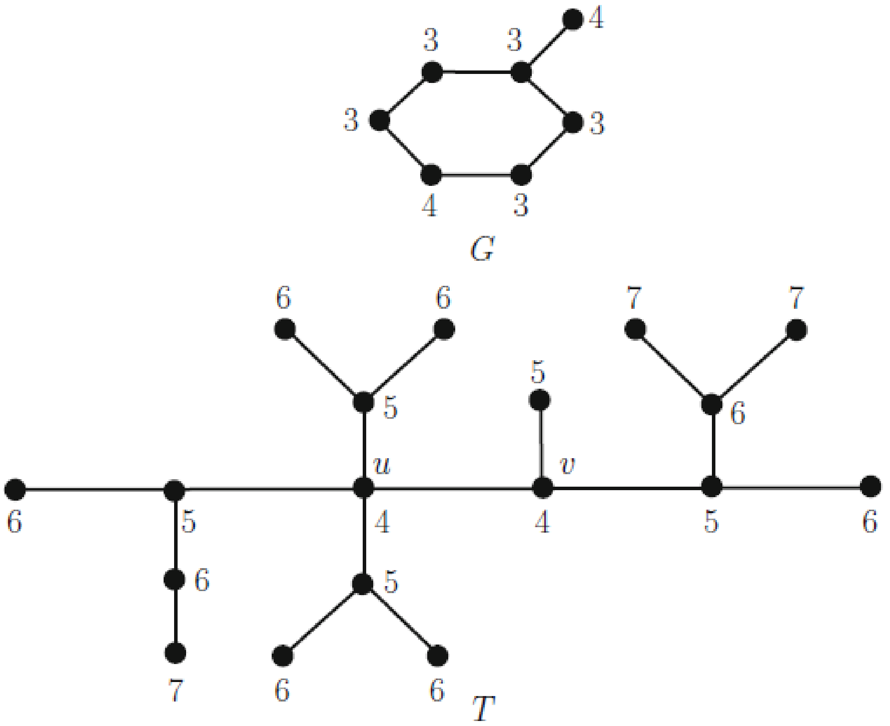


Fig. 4.3 Eccentricities of vertices for graphs G and T

Theorem 4.3.4 (Jordan [117]):

Every tree has a center consisting of either a single vertex or two adjacent vertices.

Proof:

The result is obvious for the trees K_1 and K_2.

The vertices of K_1 and K_2 are central vertices.

Now,

let T be a tree with $n(T) \geq 3$.

Then,

T has at least two pendant vertices.

Clearly,

the pendant vertices of T cannot be central vertices.

Delete all the pendant vertices from T.

This results in a subtree T' of T.

As any maximum-distance path in T from any vertex of T' ends at a pendant vertex of T, the eccentricity of each vertex of T' is one less than the eccentricity of the same vertex in T.

Hence,

the vertices of minimum eccentricity of T' are the same as those of T.

In other words,

T and T' have the same center.

Now,

If T'' is the tree obtained from T' by deleting all the pendant vertices of T', then T'' and T' have the same center.

Hence the centers of T'' and T are the same.

Repeat the process of deleting the pendant vertices in the successive subtrees of T until there results a K_1 or K_2.

This will always be the case as T is finite.

Hence,

the center of T is either a single vertex or a pair of adjacent vertices.

Note:

The process of determining the center described above is illustrated in Figure 4.4 for the tree T of Figure 4.3. We observe that the center of T consists of the pair of adjacent vertices v_2 and v_3.

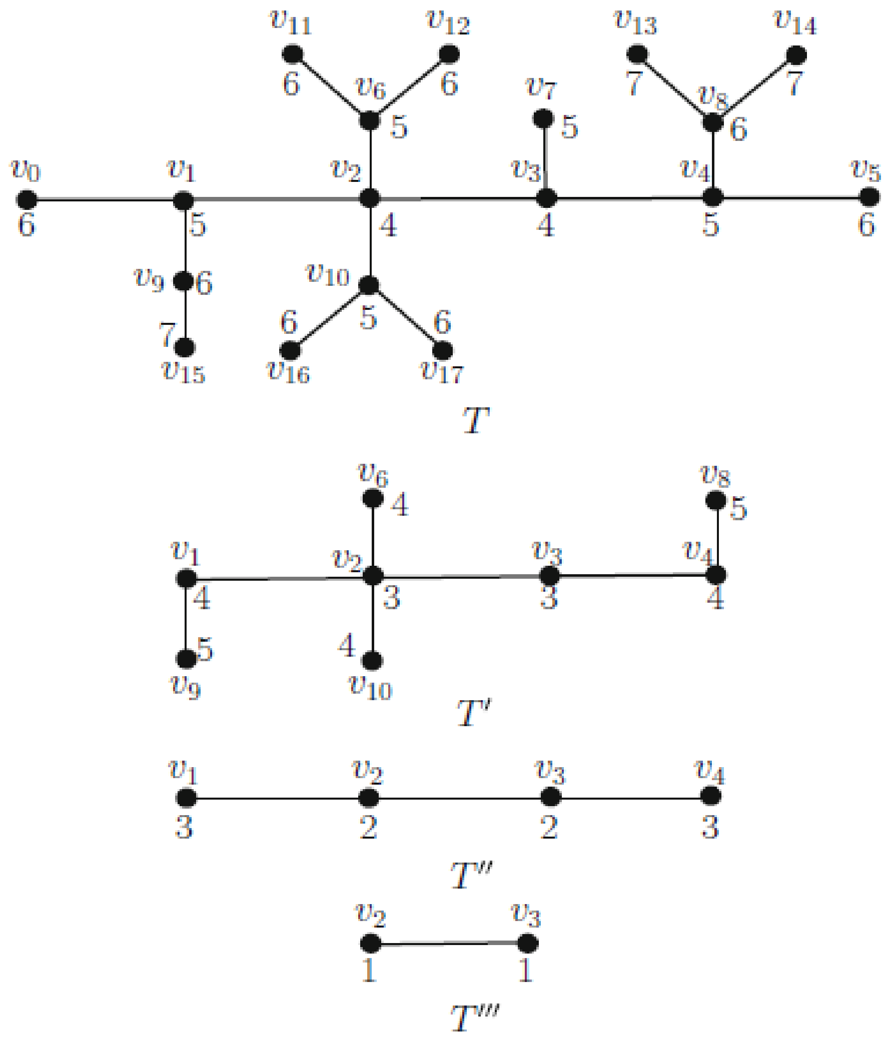


Fig. 4.4 Determining the center of tree T

Exercise 3.1. Construct a tree with *85* vertices that has $\Delta = 5$ and the center consisting of a single vertex.

4.4 Counting the Number of Spanning Trees:

Counting the number of spanning trees in a graph occurs as a natural problem in many branches of science. Spanning trees were used by Kirchoff to generate a "cycle basis" for the cycles in the graphs of electrical networks. In this section, we consider the enumeration of spanning trees in graphs.

The number of spanning trees of a connected labeled graph G will be denoted by $\tau(G)$. If G is disconnected, we take $\tau(G) = 0$. There is a recursive formula for $\tau(G)$. Before we establish this formula, we shall define the concept of *edge contraction* in graphs.

Definition 4.4.1:

An edge e of a graph G is said to be *contracted* if it is deleted from G and its ends are identified. The resulting graph is denoted by $G \circ e$.

Edge contraction is illustrated in Figure 4.7.

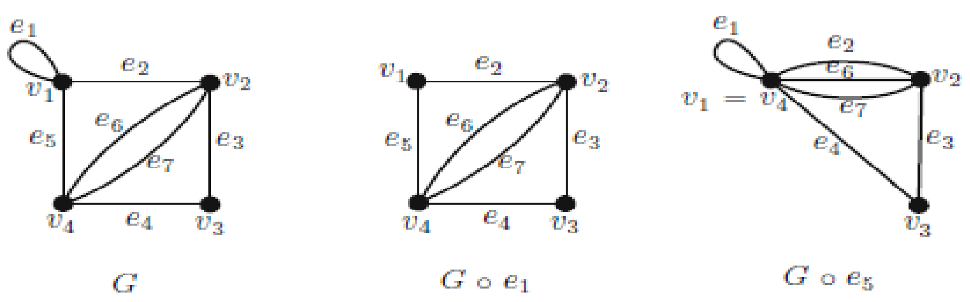


Fig. 4.7 Edge contraction

If e is not a loop of G,

Then,

$n(G \cdot e) = n(G) - 1,\ m(G \cdot e) = m(G) - 1$ and $\omega(G \cdot e) = \omega(G)$.

For a loop e,

$n(G \cdot e) = n(G),\ m(G \cdot e) = m(G) - 1$ and $\omega(G \cdot e) = \omega(G)$.

Theorem 4.4.2 gives a recursive formula for $\tau(G)$.

Theorem 4.4.2.

If e is not a loop of a connected graph $G, \tau(G) = \tau(G - e) + \tau(G \cdot e)$.

Proof:

$\tau(G)$ is the sum of the number of spanning trees of G containing e and the number of spanning trees of G not containing e.

Since $V(G - e) = V(G)$,

every spanning tree of $G - e$ is a spanning tree of G not containing e, and conversely,

any spanning tree of G for which e is not anedge is also a spanning tree of $G - e$.

Hence,

the number of spanning trees of G not containing e is precisely the number of spanning trees of $G - e$, that is, $\tau(G - e)$.

If T is a spanning tree of G containing e,

the contraction of e in both T and G results in a spanning tree $T \cdot e$ of $G \cdot e$.

Conversely,

if T_0 is a spanning tree of $G \cdot e$,

there exists a unique spanning tree T of G containing e such that $T \cdot e = T_0$.

Thus,

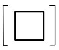
the number of spanning trees of G containing e is $\tau(G \cdot e)$.

Hence,

$$\tau(G) = \tau(G - e) + \tau(G \cdot e).$$

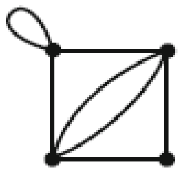
We illustrate below the use of Theorem 4.4.2 in calculating the number of spanning trees. In this illustration, each graph within parentheses stands for the number of its spanning trees.

For example,

 stands for the number of spanning trees of C_4.

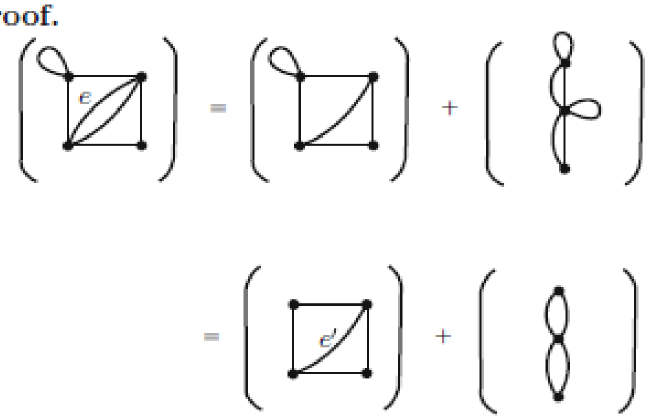
Example 4.4.3:

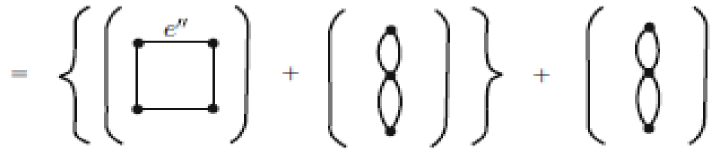
Find $\tau(G)$ for the following graph G:

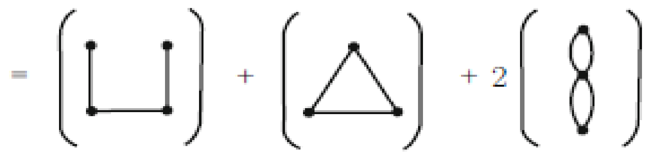


G

Proof.



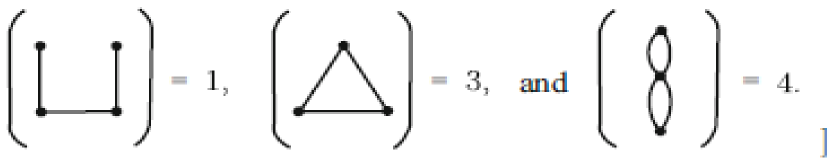




$= 1 + 3 + 2(4)$

$= 12.$

[By enumeration,



Hence,

$$\tau(G) = 12.$$

We have seen that every connected graph has a spanning tree. When will it have k edge-disjoint spanning trees? An answer to this interesting question was given by both Tutte [181] and Nash-Williams [145] at just about the same time.

Theorem 4.4.4 (Tutte [181]; Nash-Williams [145]):

A simple connected graph G contains k pairwise edge-disjoint spanning trees if and only if for each partition P of $V(G)$ into p parts, the number m(P) of edges of G joining distinct parts is at least $k(p-1), 2 \leq p \leq |V(G)|$.

Proof:

We prove only the easier part of the theorem (necessity of the condition). Suppose G has k pairwise edge-disjoint spanning trees.

If T is one of them and if $P = \{V_1, V_2, \dots, V_p\}$ is a partition of $V(G)$ into p

parts, then G must have at least $|P|-1$ edges of T .

As this is true for each of the k pairwise edge-disjoint trees of G, the number of edges joining distinct parts of P is at least $k(p-1)$.

4.5 Cayley's Formula:

Cayley was the first mathematician to obtain a formula for the number of spanning trees of a labeled complete graph.

Theorem 4.5.1 (Cayley [33]):

$\tau(K_n) = n^{n-2}$, where K_n is a labeled complete graph on n vertices, $n \geq 2$.

Proof:

First to prove the following two lemmas.

1. Let $\{d_1, d_2, \dots, d_n\}$ be the sequence of positive integers such that $\sum_{i=1}^{n} d_i = 2(n-1)$. Then there exist a tree T with vertex set $\{v_1, v_2, \dots, v_n\}$ and $d(v_i) = d_i, 1 \leq i \leq n$.

2. Let $\{v_1, v_2, \dots, v_n\}, n \geq 2$ be given and let $\{d_1, d_2, \dots, d_n\}$ be the sequence of positive integers such that $\sum_{i=1}^{n} d_i = 2(n-1)$. Then the number of trees with $\{v_1, v_2, \dots, v_n\}$ as the vertex set in which v_i has degree $d_i, 1 \leq i \leq n$ is $\frac{(n-2)!}{(d_1-1)!(d_2-1)!\dots(d_n-1)!}$.

Proof of the theorem:

The total number of trees T_n with vertex set $\{v_1, v_2, \dots, v_n\}$ is obtained by summing over all possible sequences $\{d_1, d_2, \dots, d_n\}$ with $\sum_{i=1}^{n} d_i = 2n-2$

Hence,

$$\tau(K_n) = \sum_{d_i \geq 1} \frac{(n-2)!}{(d_1-1)!(d_2-1)!\ldots(d_n-1)!} \text{ with } \sum_{i=1}^{n} d_i = 2n-2$$

$$= \sum_{d_i \geq 1} \frac{(n-2)!}{k_1!\,k_2!\ldots.k_n!} \sum_{i=1}^{n} k_i = n-2 \text{ where } k_i = d_i - 1, 1 \leq i \leq n.$$

Putting $x_1 = x_2 = \cdots = x_n = 1$ and $m = n-2$ in the multinomial expansion

$$(x_1 + x_2 + \cdots + x_n)^m = \sum_{d_i \geq 1} \frac{{x_1}^{k_1}{x_2}^{k_2}\ldots{x_n}^{k_n}}{k_1!\,k_2!\ldots.k_n!} m! \text{ with } (k_1 + k_2 + \cdots + k_n) = m$$

we get,

$$n^{n-2} = \sum_{k_i \geq 1} \frac{(n-2!)}{k_1!\,k_2!\ldots.k_n!} m! \text{ with } (k_1 + k_2 + \cdots + k_n) = n-2.$$

Thus,

$$\tau(K_n) = n^{n-2}.$$

UNIT-IV:

Independent Sets and Matchings: Vertex Independent Sets and Vertex Coverings - Edge Independent Sets - Matchings and Factors - Matching in Bi-partite Graphs - Perfect Matching and the Tutte Matrix.

Chapter 5: Section 5.1 to 5.6.

5.Independent Sets and Matchings:

5.1 Introduction:

Vertex-independent sets and vertex coverings as also edge-independent sets and edge coverings of graphs occur very naturally in many practical situations and hence have several potential applications. In this chapter, we study the properties of these sets. In addition, we discuss matchings in graphs and, in particular, in bipartite graphs. Matchings in bipartite graphs have varied applications in operations research. We also present two celebrated theorems of graph theory, namely, Tutte's *1*-factor theorem and Hall's matching theorem. All graphs considered in this chapter are loopless.

5.2 Vertex-Independent Sets and Vertex Coverings:

Definition 5.2.1.

A subset S of the vertex set V of a graph G is called *independent* if no two vertices of S are adjacent in $G. S \subseteq V$ is a *maximum independent set* of G if G has no independent set S' with $|S'| > |S|$. A *maximal independent set* of G is an independent set that is not a proper subset of another independent set of G.

For example,

in the graph of Figure 5.1, $\{u, v, w\}$ is a maximum independent set and

$\{x, y\}$ is a maximal independent set that is not maximum.

Definition 5.2.2. A subset K of V is called a *covering* of G if every edge of G is incident with at least one vertex of K. A covering K is *minimum* if there is no covering K' of G such that $|K'| < |K|$ it is *minimal* if there is no covering K_1 of G such that K_1 is a proper subset of K.

In the graph W_5 of Figure 5.2, $\{v_1, v_2, v_3, v_4, v_5\}$ is a covering of W_5 and $\{v_1, v_3, v_4, v_6\}$ is a minimal covering. Also, the set $\{x, y\}$ is a minimum covering of Figure 5.1.

Fig. 5.1 Graph with maximum independent set $\{u, v, w\}$ and maximal independent set $\{x, y\}$

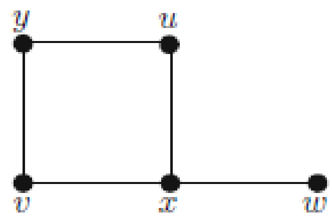
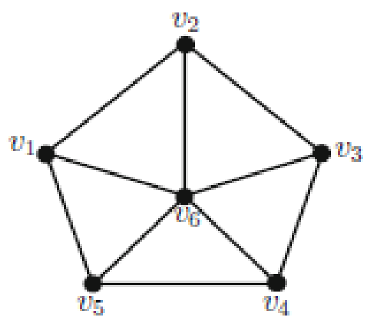


Fig. 5.2 Wheel W_5



The concepts of covering and independent sets of a graph arise very naturally in practical problems. Suppose we want to store a set of chemicals in different rooms. Naturally, we would like to store incompatible chemicals, that is, chemicals that are likely to react violently when brought together, in distinct rooms. Let G be a graph whose vertex set represents the set of chemicals and let two vertices be made adjacent in G if and only if the corresponding chemicals are incompatible. Then any set of vertices representing compatible chemicals forms an independent set of G.

Now consider the graph G whose vertices represent the various locations in a factory and whose edges represent the pathways between pairs of such locations. A light source placed at a location supplies light to all the pathways incident to that location. A set of light sources that supplies light to all the pathways in the factory forms a covering of G.

Theorem 5.2.3:

A subset S of V is independent if and only if $V \backslash S$ is a covering of G.

Proof:

S is independent if and only if no two vertices in S are adjacent in G.

Hence,

every edge of G must be incident to a vertex of $V \backslash S$.

This is the case if and only if $V \backslash S$ is a covering of G.

5.3 Edge-Independent Sets:

Definitions 5.3.1.

1. A subset M of the edge set E of a loop-less graph G is called *independent* if no two edges of M are adjacent in G. A *matching* in G is a set of independent edges.
2. An *edge covering* of G is a subset L of E such that every vertex of G is incident to some edge of L. Hence, an edge covering of G exists if and only if $\delta > 0$.
3. A matching M of G is *maximum* if G has no matching M' with $|M'| > |M|$. M is *maximal* if G has no matching M' strictly containing M. $\alpha'(G)$ is the cardinality of a maximum matching and $\beta'(G)$ is the size of a minimum edge covering of G.
4. A set S of vertices of G is said to be *saturated* by a matching M of G or *M- saturated* if every vertex of S is incident to some edge of M. A vertex

v of G is *M-saturated* if $\{v\}$ is M-saturated. v is *M-unsaturated* if it is not M-saturated.

For example, in the wheel W_5 (Figure 5.2), $M = \{v_1v_2, v_4v_6\}$ is a maximal matching; $\{v_1v_5, v_2v_3, v_4v_6\}$ is a maximum matching and a minimum edge covering; the vertices v_1, v_2, v_4, and v_6 are M-saturated, whereas v_3 and v_5 are M-unsaturated

5.4 Matchings and Factors:

Definition 5.4.1.

A *matching* of a graph G is a set of independent edges of G. If $e = uv$ is an edge of a matching M of G, the end vertices u and v of e are said to be *matched* by M.

If M_1 and M_2 are matchings of G, the edge subgraph defined by $M_1 \Delta M_2$, the symmetric difference of M_1 and M_2, is a subgraph H of G whose components are paths or even cycles of G in which the edges alternate between M_1 and M_2.

Definition 5.4.2. An *M-augmenting path* in G is a path in which the edges alternate between $E\backslash M$ and M and its end vertices are M-unsaturated. An *M-alternating path* in G is a path whose edges alternate between $E\backslash M$ and M.

Example 5.4.3:

In the graph G of Figure 5.2, $M_1 = \{v_1v_2, v_3v_4, v_5v_6\}$,

$M_2 = \{v_1v_2, v_3v_6, v_4v_5\}$ and $M_1 = \{v_3v_4, v_5v_6\}$ are matching G.

Moreover,

G[$M_1 \Delta M_2$]is an even cycle $(v_3 v_4 v_5 v_6 v_3)$.

The path $v_2 v_3 v_4 v_6 v_5$ is an M_5-augmenting path in G.

Maximum matchings have been characterized by Berge [19].

Theorem 5.4.4. *A matching M of a graph G is maximum if and only if G has no M-augmenting path.*

Proof:

Assume first that M is maximum.

If G has an M-augmenting path *P:* $v_0 v_1 v_2 \dots v_{2t+1}$ in which the edges alternate between $E \backslash M$ and M, then P has one edge of $E \backslash M$ more than that of M.

Define $M' = M \cup \{ v_0 v_1, v_2 v_3, \dots, v_{2t} v_{2t+1} \} \backslash v_1 v_2, v_3 v_4, \dots, v_{2t-1} v_{2t}$

Clearly, M' is a maximum matching of G.

Conversely,

assume that G has no M-augmenting path.

Then M must be maximum.

If not, there exists a matching M' of G with $|M'| > |M|$.

Let H be the edge subgraph $G[M \Delta M']$ defined by the symmetric difference of M and M'.

Then,

the components of H are paths or even cycles in which the edges alternate between M and M'.

Since $|M'| > |M|$, at least one of the components of H must be a path starting and ending with edges of M'.

But then such a path is an M -augmenting path of G, contradicting the assumption (see Figure 5.5).

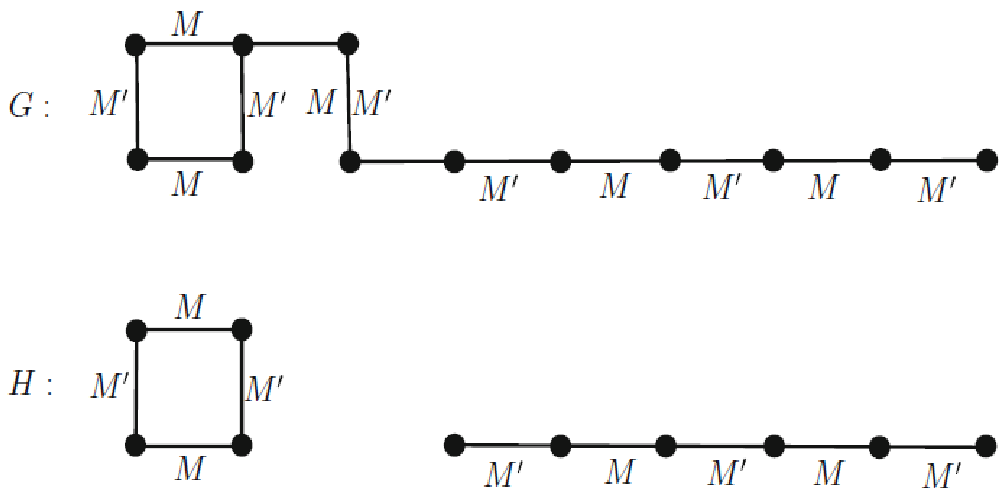


Fig. 5.5 Graphs for proof of Theorem 5.4.4

Definition 5.4.5:

A *factor* of a graph G is a spanning subgraph of G. A *k-factor* of G is a factor of G that is k-regular. Thus, a *1-factor* of G is a matching that saturates all the vertices of G. For this reason, a 1-factor of G is called a *perfect matching* of G. A 2-factor of G is a factor of G that is a disjoint union of cycles of G. A graph G is *k-factorable* if G is an edge-disjoint union of k-factors of G.

Example 5.4.6:

In Figure 5.6,

G_1 is 1-factorable and G_2 is 2-factorable, whereas G_3 has neither a 1-factor nor a 2-factor.

The dotted, solid, and ordinary lines of G_1 give the three distinct 1-factors, and the dotted and ordinary lines of G_2 give its two distinct 2-factors.

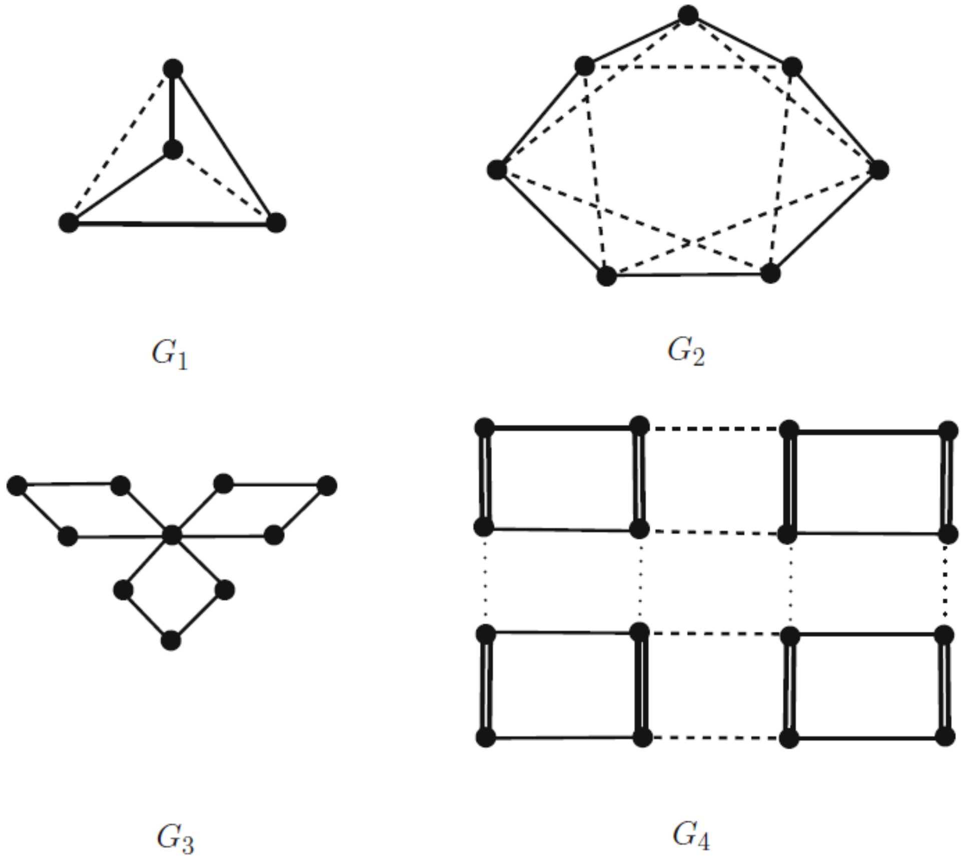


Fig. 5.6 Graphs illustrating factorability

Exercise 4.1. Give an example of a cubic graph having no *1*-factor.

Exercise 4.2. Show that $K_{n,n}$ and K_{2n} are *1*-factorable.

Exercise 4.3. Show that the number of *1*-factors of

(i) $K_{n,n}$ is n!

(ii) K_{2n} is $\frac{(2n)!}{2^n n!}$.

5.5 Matchings in Bipartite Graphs:

Assignment Problem 5.5.1.

Suppose in a factory there are n jobs $j_1, j_2, \dots, j_n$ and s workers $w_1, w_2, \dots, w_n$. Also suppose that each job j_i can be performed by a certain number of workers and that each worker w_j has been trained to do a certain number of jobs. Is it possible to assign each of the n jobs to a worker who can do that job so that no two jobs are assigned to the same worker?

We convert this job assignment problem into a problem in graphs as follows:

Form a bipartite graph G with bipartition (J, W), where $J = \{j_1, j_2, \dots, j_n\}$ and $W = \{w_1, w_2, \dots, w_n\}$ and make j_i adjacent to w_j if and only if worker w_j can do the job j_i. Then our assignment problem translates into the following graph problem: Is it possible to find a matching in G that saturates all the vertices of J ?

A solution to the above matching problem in bipartite graphs has been given by Hall [90] (see also Hall, Jr. [91]).

For a subset $S \subseteq V$ in a graph G, $N(S)$ denotes the neighbor set of S, that is, the set of all vertices each of which is adjacent to at least one vertex in S.

Theorem 5.5.2 (Hall). *Let G be a bipartite graph with bipartition (X, Y). Then G has a matching that saturates all the vertices of X if and only if*

$$|N(S)| \geq |S| \qquad \dots\dots\dots\dots\dots(5.3)$$

for every subset S of X.

Proof:

If G has a matching that saturates all the vertices of X, then distinct vertices of X are matched to distinct vertices of Y.

Hence,

trivially,

$|N(S)| \geq |S|$ for every subset S X.

Conversely,

assume that the condition (5.3) above holds but that G has no matching that saturates all the vertices of X.

Let M be a maximum matching of G.

As M does not saturate all the vertices of X, there exists a vertex $x_0 \in X$ that is M -unsaturated.

Let Z denote the set of all vertices of G connected to x_0 by M - alternating paths.

Since M is a maximum matching,

by Theorem,

G has no M -augmenting path.

As x_0 is M -unsaturated,

x_0 is the only vertex of Z that is M - unsaturated.

Let $A = Z \cap X$ and $B = Z \cap Y$.

Then the vertices of $A \backslash \{x_0\}$ get matched under M to the vertices of B, and $N(A) = B$.

Thus,

since $|B| = |A| - 1$, $|N(A)| = |B| = |A| - 1 < |A|$

and this contradicts the assumption (5.3).

5.6 Perfect Matchings and the Tutte Matrix:

It has been established by Tutte that the existence of a perfect matching in a simple graph is related to the non-singularity of a certain square matrix. This matrix is called the "Tutte matrix" of the graph. We now define the Tutte matrix

Definition 5.6.1:

Let $G = (V, E)$ be a simple graph of order n and let $V = \{v_1, v_2, \ldots, v_n\}$.

Let $\{x_{ij}: 1 \leq i \leq j \leq n\}$ be a set of indeterminates. Then, the Tutte matrix of G is defined to be the n by n matrix $T = (t_{ij})$, where,

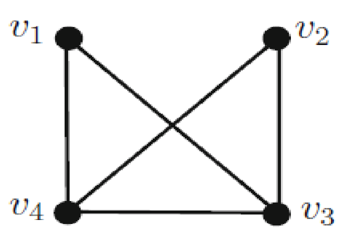
$$t_{ij} = \begin{cases} x_{ij} & if \quad v_i v_j \in E(G) \; and \; i < j \\ -x_{ij} & if \quad v_i v_j \in E(G) \; and \; i > j \\ 0 & \quad otherwise \end{cases}$$

Thus,

T is a skew-symmetric matrix of order n.

Example 5.6.2:

For example, if G is the graph



G

then

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} = \begin{bmatrix} 0 & 0 & x_{13} & x_{14} \\ 0 & 0 & x_{23} & x_{24} \\ -x_{13} & -x_{23} & 0 & x_{34} \\ -x_{14} & -x_{24} & -x_{34} & 0 \end{bmatrix}.$$

UNIT-V:

Eulerian and Hamiltonian Graphs: Eulerian Graphs - Hamiltonian Graphs - Hamilton's "Around the World" Game.

Graph Colorings: Vertex Colorings - Applications of Graph Colorings - Critical Graphs - Brooks' Theorem.

Chapter 6: Section 6.1 to 6.3,

Chapter 7: Section 7.1 to 7.3 (up to Brooks Theorem).

6. Eulerian and Hamiltonian Graphs:

6.1Introduction:

The study of Eulerian graphs was initiated in the 18th century and that of Hamiltonian graphs in the 19th century. These graphs possess rich structures; hence, their study is a very fertile field of research for graph theorists. In this chapter, we present several structure theorems for these graphs.

6.2 Eulerian Graphs:

Definition 6.2.1:

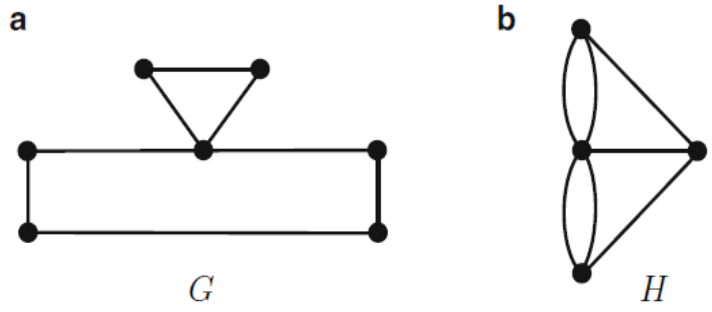
An *Euler trail* in a graph G is a spanning trail in G that contains all the edges of G. An *Euler tour* of G is a closed Euler trail of G. G is called *Eulerian* (Figure 6.1a) if G has an *Euler tour*. It was Euler who first considered these graphs, and hence their name.

It is clear that an Euler tour of G, if it exists, can be described from any vertexof G. Clearly, every Eulerian graph is connected.

Euler showed in 1736 that the celebrated *Königsberg bridge problem* has no solution. The city of Königsberg (now called Kaliningrad) has seven bridges linking two islands A and B and the banks C and D of the Pregel (now called Pregalya) River, as shown in Figure 6.2.

The problem was to start from any one of the four land areas, take a stroll across the seven bridges, and get back to the starting point without crossing any bridge a second time. This problem can be converted into one concerning the graph obtained by representing each land area by a vertex and each bridge by an edge. The resulting graph H is the graph of Figure 6.1b. The Königsberg bridge problem will have a solution provided that this graph H is Eulerian. But this is not the case since it has vertices of odd degrees (see Theorem 6.2.2).
Eulerian graphs admit, among others, the following two elegant characterizations.

Fig. 6.1 (**a**) Eulerian graph G; (**b**) non-Eulerian graph H



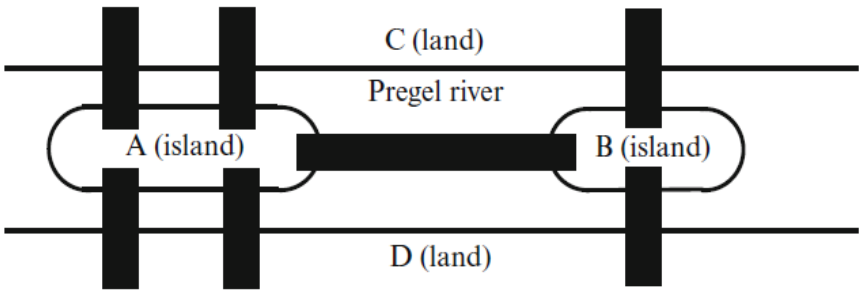


Fig. 6.2 Königsberg bridge problem

Theorem 6.2.2:

For a nontrivial connected graph G, the following statements are equivalent:

(i) G is Eulerian.

(ii) The degree of each vertex of G is an even positive integer.

(iii) G is an edge-disjoint union of cycles.

Proof:

To Prove: $(i) \Rightarrow (ii)$

Let T be an Euler tour of G described from some vertex $v_0 \in V(G)$.

If $v \in V(G)$, and $v \neq v_0$,

then every time T enters v, it must move out of v to get back to v_0.

Hence,

two edges incident with v are used during a visit to v,

and therefore, $d(v)$ is even.

At v_0, every time T moves out of v_0, it must get back to v_0.

Consequently, $d(v_0)$ is also even.

Thus,

the degree of each vertex of G is even.

To Prove: $(ii) \Rightarrow (iii)$

As $\delta(G) \geq 2$,

G contains a cycle C_1.

In $G \setminus E(C_1)$,

remove the isolated vertices if there are any.

Let the resulting subgraph of G be G_1.

If G_1 is nonempty, each vertex of G_1 is again of even positive degree. Hence $\delta(G_1) \geq 2$,

and so G_1 contains a cycle C_2.

It follows that after a finite number, say r, of steps, $G \backslash E(C_1 \cup C_2 \cup \ldots \cup C_r)$ is totally disconnected.

Then,

G is the edge- disjoint union of the cycles $C_1, C_2, \ldots, C_r$.

To Prove: $(iii) \Rightarrow (i)$

Assume that G is an edge-disjoint union of cycles.

Since any cycle is Eulerian, G certainly contains an Eulerian subgraph.

Let G_1 be a longest closed trail in G.

Then,

G_1 must be G.

If not,

let $G_2 = G \backslash E(G_1)$.

Since G is an edge- disjoint union of cycles,

every vertex of G is of even degree ≥ 2.

Further,

since G_1 is Eulerian, each vertex of G_1 is of even degree ≥ 2.

Hence,

each vertex of G_2 is of even degree.

Since G_2 is not totally disconnected and G is connected,

G_2 contains a cycle C having a vertex v in common with G_1.

Describe the Euler tour of G_1 starting and ending at v and follow it by C.

Then,

$G_1 \cup C$ is a closed trail in G longer than G_1.

This contradicts the choice of G_1,

and so G_1 must be G.

Hence,

G is Eulerian.

Remark:

If $G_1, G_2, \dots, G_r$ are subgraphs of a graph G that are pairwise edge-disjoint and their union is G, then this fact is denoted by writing

$$G = G_1 \oplus G_2 \oplus \dots \oplus G_r.$$

In the above equation, if $G_i = C_i$, a cycle of G for each i,

then

$$G = C_1 \oplus C_2 \oplus \dots \oplus C_r.$$

The set of cycles $S = \{C_1, C_2, \dots, C_r\}$ is then called a *cycle decomposition* of G.

Thus,

Theorem 6.2.2 implies that *a connected graph is Eulerian if and only if it admits a cycle decomposition.*

Theorem 6.2.3*:

A graph G is Eulerian if and only if each edge e of G belongs to an odd number of cycles of G.

For instance, in Fig. 6.3, e belongs to the three cycles $P_1 \cup e, P_2 \cup e$, and $P_3 \cup e$.

Proof:

Denote by γ_e the number of cycles of *G* containing e.

Assume that γ_e is odd for each edge *e* of *G*.

Since a loop at any vertex *v* of *G* is in exactly one cycle of *G* and contributes 2 to the degree of *v* in *G*,

we may suppose that *G* is loop-less.

Let $S = \{C_1, C_2, \ldots, C_p\}$ be the set of cycles of *G*.

Replace each edge e of *G* by γ_e parallel edges and replace *e* in each of the γ_e cycles containing *e* by one of these parallel edges, making sure that none of the parallel edges is repeated.

Let the resulting graph be G_0 and let the new set of cycles be

$S_0 = \{{C_1}^0, {C_2}^0, \ldots, {C_p}^0\}$.

Clearly,

S_0 is a cycle decomposition of G_0.

Hence, by Theorem 6.2.2, G_0 is Eulerian.

But,

then $d_{G_0}(v) \equiv 0 \ (mod \ \ 2)$ for each $v \in V(G_0) = V(G)$.

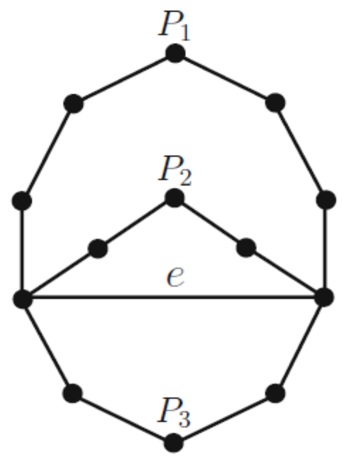
Moreover,

$d_G(v) \equiv d_{G_0}(v) - \sum_e(\gamma_e - 1)$, where *e* is incident at *v* in *G* and hence $d_G(v) \equiv 0 \ (mod \ 2)$, γ_e being odd for each $e \in E(G)$.

Thus,

G is Eulerian.

Fig. 6.3 Eulerian graph with edge e belonging to three cycles



Conversely,

assume that G is Eulerian.

We proceed by induction on $n = |V(G)|$.

If $n = 1$, each edge is a loop and hence belongs to exactly one cycle of G.

Assume the result for graphs with fewer than $n(\geq 2)$ vertices.

Let G be a graph with n vertices.

Let $e = xy$ be an edge of G and let $\lambda(e)$ be the multiplicity of e in G.

The graph G ⍰ e obtained from G by contracting the edge e is also Eulerian.

Denote by z the new vertex of G ⍰ e obtained by identifying the vertices x and y of G.

The set of edges incident with z in G ⍰ e is partitioned into three subsets (see Figure 6.4):

1. $E_z(x)$ = set of edges arising out of edges of G incident with x but not with y
2. $E_z(y)$ = set of edges arising out of edges of G incident with y but not with x
3. $E_z(xy)$ =set of $\lambda(e) - 1$ loops of G ⍰ e corresponding to the edges parallel to e in G

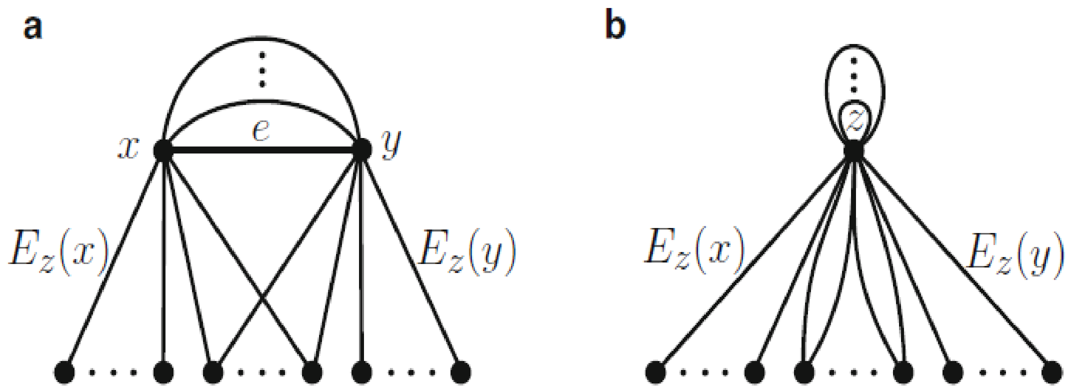


Fig. 6.4 Graph for proof of Theorem 6.2.3

Let $k = |E_z(x)|$.

Since G is Eulerian,

$$k + \lambda(e) = d_G(x) \equiv 0 \pmod 2.$$

Let Γ_f and $\Gamma(e_i, e_j)$ denote, respectively, the number of cycles in G ⍰ e containing the edge f and the pair (e_i, e_j) of edges.

Since $|V(G$ ⍰ $e)| = n - 1$,

and since G ⍰ e is Eulerian by the induction assumption, Γ_f is odd for each edge f of G ⍰ e.

Now,

any cycle of G containing e either consists of e and an edge parallel to e in G (and there are $\lambda(e) - 1$ of them) or contains e, an edge e_i of $E_z(x)$, and an edge $e_j{}'$ of $E_z(y)$.

These correspond in G ⍰ e, respectively, to a loop at z and to a cycle containing the edges of G ⍰ e that correspond to the edges e_i and $e_j{}'$ of G.

By abuse of notation, we denote these corresponding edges of G ⍰ e also by e_i and $e_j{}'$, respectively.

Moreover,

any cycle of G □ e containing an edge e_i of $E_z(x)$ will also contain either an edge e_j of $E_z(x)$ or an edge $e_j{}'$ of $E_z(y)$ but not both.

A cycle of the former type is counted once in Γ'_{e_i} and once in Γ'_{e_j}, and these will not give rise to cycles in G containing e.

Thus,

$$\gamma_e = (\lambda(e) - 1) + \sum_{e_i \in E_z(x)} \Gamma'_{e_i} - \sum_{\substack{\{i,j\} \\ i \neq j \\ e_i, e_j \in E_z(x)}} \Gamma'(e_i, e_j)$$

Now,

by the induction hypothesis,

$\Gamma'_{e_i} \equiv 1 \ (mod\ \ 2)$ for each e_i, and $\Gamma'\left(e_i, e_j\right) = \Gamma'(e_j, e_i)$ in the last sum on the right,

and hence this latter sum is even.

Thus,

$$(\lambda(e) - 1) + k \ (mod\ \ 2) \equiv 1 \ (mod\ \ 2).$$

Corollary 6.2.4*. *A graph is Eulerian if and only if it has an odd number of* cycle decompositions.

Proof:

In one direction, the proof is trivial.

If G has an odd number of cycle decompositions,

then it has at least one,

and hence G is Eulerian.

Conversely,

assume that G is Eulerian.

Let $e \in E(G)$ and let $C_1, C_2, \dots, C_r$ be the cycles containing e.

By Theorem 6.2.3,

r is odd. We proceed by induction on $m = |E(G)|$ with G Eulerian.

If G is just a cycle,

then the result is true.

Assume then that G is not a cycle.

This means that for each $i, 1 \leq i \leq r$,

by the induction assumption, $G_i = G - E(C_i)$ has an odd number, say s_i, of cycle decompositions.

(If G_i is disconnected, apply the induction assumption to each of the nontrivial components of G_i.)

The union of each of these cycle decompositions of G_i and C_i yields a cycle decomposition of G. Hence the number of cycle decompositions of G containing C_i is s_i $1 \leq i \leq r$.

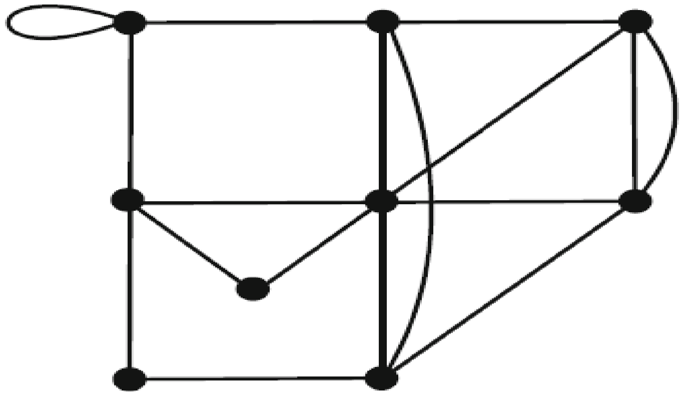
Let $s(G)$ denote the number of cycle decompositions of G.

Then,

$$s(G) = \sum_{i=1}^{r} s_i \equiv r \ (mod \ \ 2) \text{ (Since } s_i \equiv 1 \ (mod \ \ 2))$$

$$\equiv 1 \ (mod \ \ 2)$$

Exercise 2.1. Find an Euler tour in the graph G below.



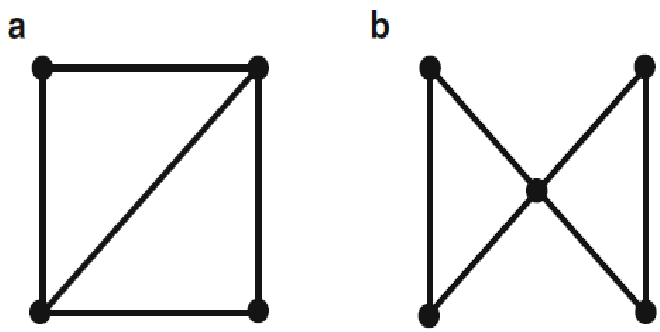
G

6.3 Hamiltonian Graphs:

Definition 6.3.1. A graph is called *Hamiltonian* if it has a spanning cycle (see Figure 6.5a). These graphs were first studied by Sir William Hamilton, a mathematician. A spanning cycle of a graph G, when it exists, is often called a *Hamilton cycle* (or *Hamiltonian cycle*) of G.

Definition 6.3.2. A graph G is called *traceable* if it has a spanning path of G (see Figure 6.5b). A spanning path of G is also called a *Hamilton path* (or *Hamiltonian path*) of G.

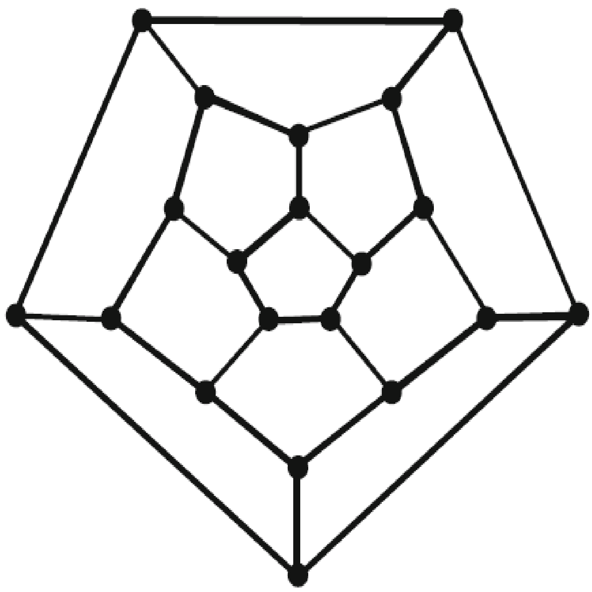
Fig. 6.5 (**a**) Hamiltonian graph; (**b**) non-Hamiltonian but traceable graph



6.3.1 Hamilton's "Around the World" Game:

Hamilton introduced these graphs in 1859 through a game that used a solid dodecahedron (Figure 6.6). A dodecahedron has 20 vertices and 12 pentagonal faces. At each vertex of the solid, a peg was attached. The vertices were marked Amsterdam, Ann Arbor, Berlin, Budapest, Dublin, Edinburgh, Jerusalem, London, Melbourne, Moscow, Novosibirsk, New York, Paris, Peking, Prague, Rio di Janeiro, Rome, San Francisco, Tokyo, and Warsaw. Further, a string was also provided. The object of the game was to start from any one of the vertices and keep on attaching the string to the pegs as we move from one vertex to another along a particular edge with the condition that we have to get back to the starting city without visiting any intermediate city more than once.

Fig. 6.6 Solid dodecahedron for Hamilton's "Around the World" problem



In other words, the problem asks one to find a Hamilton cycle in the graph of the dodecahedron (see Figure 6.6). Hamilton solved this problem as follows: When a traveler arrives at a city, he has the choice of taking the edge to his right or left. Denote the choice of taking the edge to the right by R and that of taking the edge to the left by L. Let 1 denote the operation of staying where he is.

Define the product O_1O_2 of two operations O_1 and O_2 as O_1 followed by O_2. For example, LR denotes going left first and then going right. Two sequences of operations are *equal* if, after starting at a vertex, the two sequences lead to the same vertex. The product defined above is associative but not commutative. Further, it is clear (see Figure 6.6) that

$$R^5 = L^5 = 1$$

$$RL^2R = LRL,$$

$$LR^2L = RLR,$$

$$RL^3R = L^2, \text{ and}$$

$$LR^3L = R^2.$$

These relations give

$$1 = R^5 = R^2R^3 = (LR^3L)R^3 — (LR^3)(LR^3) = (LR^3)^2 = (LR^2R)^2$$

$$= (L(LR^3L)R)^2 = (L^2R^3LR)^2 = (L^2((LR^3L)R)LR)^2$$

$$= (L^3R^3LRLR)^2$$

$$= LLLRRRLRLRLLLRRRLRLR$$

The last sequence of operations contains 20 operations and contains no partial sequence equal to 1. Hence, this sequence must represent a Hamilton cycle. Thus, starting from any vertex and following the sequence of operations (6.2), we do indeed get a Hamilton cycle of the graph of Fig. 6.6.

Knight's Tour in a Chessboard 6.3.3:

The knight's tour problem is the problem of determining a closed tour through all *64* squares of an *8 8* chessboard by a knight with the condition that the knight does not visit any intermediate square more than once. This is equivalent to finding a Hamilton cycle in the corresponding graph of $64(= 8 \times 8)$ vertices in which two vertices are adjacent if and only if the knight can move from one vertex to the other following the rules of the chess game. Figure 6.7 displays a knight's tour.

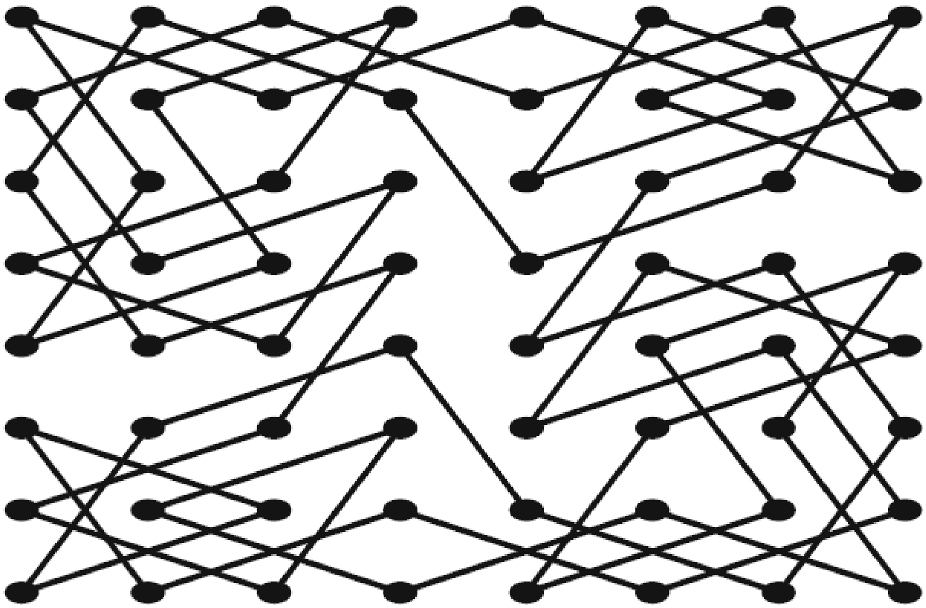


Fig. 6.7 A knight's tour in a chessboard

Even though Eulerian graphs admit an elegant characterization, no decent characterization of Hamiltonian graphs is known as yet. In fact, it is one of the most difficult unsolved problems in graph theory. (Actually, it is an NP-complete problem; see reference [71].) Many sufficient conditions for a graph to be Hamiltonian are known; however, none of them happens to be an elegant necessary condition.

We begin with a necessary condition. Recall that $\omega(H)$ stands for the number of components of the graph H.

Theorem 6.3.4:

If G is Hamiltonian, then for every nonempty proper subset S of V,
$\omega(G-S) \leq |S|$.

Proof:

Let C be a Hamilton cycle in G.

Then,

since C is a spanning subgraph of G, $\omega(G-S) \leq \omega(C-S)$.

If $|S| = 1, C - S$ is a path, and therefore $\omega(C-S) = 1 = |S|$

The removal of a vertex from a path P results in one or two components, according to whether the removed vertex is an end vertex or an internal vertex of P.

Hence,

by induction, the number of components in $C - S$ cannot exceed $|S|$.

This proves that $\omega(G-S) \leq \omega(C-S) \leq |S|$.

It follows directly from the definition of a Hamiltonian graph or from Theorem 6.3.4 that any Hamiltonian graph must be 2-connected. [If G has a cut vertex v, then taking $S = \{v\}$, we see that $\omega(G-S) > |S|$.]

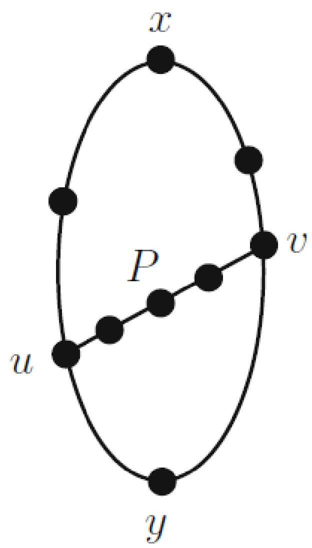
The converse,

however, is not true.

For example, the theta graph of Figure 6.8 is *2*-connected but not Hamiltonian. Here,

P stands for a *u*-*v* path of any length ≥ *2* containing neither *x* nor *y*.

Fig. 6.8 Theta graph



Exercise 3.1. Show by means of an example that the condition in Theorem 6.3.4 is not sufficient for *G* to be Hamiltonian.

7.1. Graph Colorings:

Graph theory would not be what it is today if there had been no coloring problems. In fact, a major portion of the 20th-century research in graph theory has its origin in the four-color problem.

In this chapter, we present the basic results concerning vertex colorings and edge colorings of graphs. We present one important theorem on graph colorings, namely, Brooks' theorem.

Applications of Graph Coloring:

7.2 Vertex Colorings:

Definition 7.2.1:

The *chromatic number* $\chi(G)$ of a graph G is the minimum number of independent subsets that partition the vertex set of *G*. Any such minimum partition is called a *chromatic partition* of *V(G).*

Definition 7.2.2:

The *chromatic number* of a graph *G* is the minimum number of colors needed for a proper vertex coloring of *G*. *G* is *k-chromatic* if $\chi(G) = k$.

Definition 7.2.3: A *k-coloring* of a graph *G* is a vertex coloring of *G* that uses at most *k* colors.

Definition 7.2.4:

A graph *G* is said to be *k-colorable* if *G* admits a *proper* vertex coloring using at most *k* colors.

7.3Critical Graphs:

Definition 7.3.1:

A graph *G* is called *critical* if for every proper subgraph *H* of *G*,

$\chi(H) < \chi(G)$.

Equivalently,

$\chi(G-e) < \chi(G)$ for each edge e of G.

Also,

G is *k-critical* if it is k-chromatic and critical.

Exercise 3.1. Prove that any critical graph is connected.

Theorem 7.3.3:

If G is k-critical, then $\delta(G) \geq k-1$.

Proof:

Suppose $\delta(G) \leq k-2$.

Let v be a vertex of minimum degree in G.

Since G is k-critical,

$$\chi(G-v) = \chi(G) - 1 = k-1.$$

Hence,

in any proper $(k-1)$- coloring of $G-v$, at most $(k-2)$ colors would have been used to color the neighbors of v in G.

Thus,

there is at least one color, say c, that is left out of these k— 1 colors.

If v is given the color c,

a proper $(k-1)$-coloring of G is obtained.

This is impossible since G is k-chromatic.

Hence,

$\delta(G) \geq k-1$.

Theorem 7.3.5:

In a critical graph G, no vertex cut is a clique.

Proof:

Suppose G is a k-critical graph and S is a vertex cut of G that is a clique of G

(i.e., a complete subgraph of G).

Let $H_i, 1 \leq i \leq r,$ be the components of $G \setminus S,$ and let $G_i = G[V(H_i) \cup S]$. Then,

each G_i is a proper subgraph of G and hence admits a proper $(k - 1)$-coloring.

Since S is a clique,

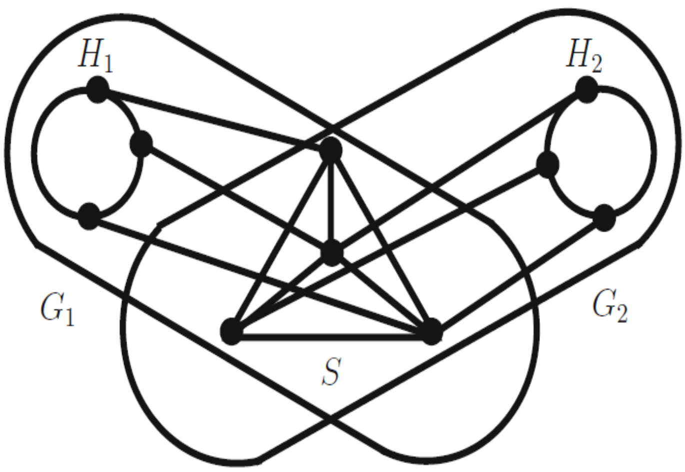
its vertices must receive distinct colors in any proper $(k - 1)$-coloring of G_i.

Hence,

by fixing the colors for the vertices of S, and coloring for each i the remaining vertices of G_i so as to give a proper $(k - 1)$-coloring of G_i, we obtain a proper $(k - 1)$-coloring of G.

This contradicts the fact that G is k-chromatic. (See Figure 7.2).

Fig. 7.2 $G[S] \simeq K_4$ $(r = 2)$



7.3.1 Brooks' Theorem:

Theorem 7.3.7 (Brooks' theorem):

If a connected graph G is neither an odd cycle nor a complete graph, then $\chi(G) \leq \Delta(G)$.

Proof:

If $\Delta(G) \leq 2$, then G is either a path or a cycle.

For a path G (other than K_1 and K_2), and for an even cycle G, $\chi(G) = 2 = \Delta(G)$.

According to our assumption,

G is not an odd cycle.

So let $\Delta(G) \geq 3$.

The proof is by contradiction.

Suppose the result is not true.

Then,

there exists a minimal graph G of maximum degree $\Delta(G) = \Delta \geq 3$ such that G is not Δ- colorable, but for any vertex v of G, $G - v$ is Δ-colorable.

Claim 1:

Let v be any vertex of G. Then in any proper Δ*-coloring of* $G - v$*, all the* Δ *colors must be used for coloring the neighbors v in G.*

Otherwise,

if some color i is not represented in $N_G(v)$, then v could be colored using i, and this would give a Δ-coloring of G,

a contradiction to the choice of G.

Thus,

G is a Δ-regular graph satisfying Claim 1.

For $v \in V(G)$, let $N(v) = \{v_1, v_2, \ldots, v_\Delta\}$.

In a proper Δ-coloring of $G - v = H$, let v_i receive color $i, 1 \leq i \leq \Delta$.

For $i \neq j$, let H_{ij} be the subgraph of H induced by the vertices receiving the i^{th} and j^{th} colors.

Claim 2:

v_i and v_j belong to the same component of H_{ij}. Otherwise, the colors i and j can be interchanged in the component of H_{ij} that contains the vertex v_j.

Such an interchange of colors once again yields a proper Δ-coloring of H.

In this new coloring, both v_i and v_j receive the same color, namely, i, a contradiction to Claim 1.

This proves Claim 2.

Claim 3:

If C_{ij} is the component of H_{ij} containing v_i and *v_j, then C_{ij} is a path in H_{ij}.*

As before,

$N_H(v_i)$ contains exactly one vertex of color *j*.

Further,

C_{ij} cannot contain a vertex, say y, of degree at least *3;* for, if y is the first such vertex on a $v_i - v_j$ path in C_{ij} that has been colored, say, with i, then at least three neighbors of y in C_{ij} have the color *j*.

Hence,

we can recolor y in H with a color different from both i and j; and in this new coloring of H; $v_i - v_j$ would belong to distinct components of H_{ij} (see Figure 7.3a).

(Note that by our choice of y, any $v_i - v_j$ path in H_{ij} must contain y.)

But this contradicts Claim 3.

Claim 4:

$C_{ij} \cap C_{ik} = \{v_i\}$ for $j \neq k$.

Indeed, if $w \in C_{ij} \cap C_{ik}, w \neq v_i$, then w is adjacent to two vertices of color j on C_{ij} and two vertices of color k on C_{ik}. (see Figure 7.3b)

Again,

we can recolor w in H by giving a color different from the colors of the neighbors of w in H.

In this new coloring of H, v_i *and* v_j belong to distinct components of H_{ij}, a contradiction to Claim 2.

This completes the proof ofClaim 4.

We are now in a position to complete the proof of the theorem.

By hypothesis,

G isnot complete.

Hence,

G has a vertex v; and a pair of nonadjacent vertices v_1 *and* v_2 in $N_G(v)$.

Then,

the $v_1 - v_2$ path C_{12} in H_{12} of $H = G - v$ contains a vertex y ($\neq v_2$) adjacent to v_1.

Naturally,

y would receive color *2*.

Since $\Delta \geq 3$,

by Claim 1, there exists a vertex $v_3 \in N_G(v)$.

Now,

interchange colors *1* and *3* in the path C_{13} of H_{13}.

This would result in a new coloring of $H = G - v$.

Denote the $v_i - v_j$ path in $\boldsymbol{H}$ under this new coloring by C_{ij}' (see Figure 7.3c).

Then,

$y \in C_{23}'$

since v_1 receives color *3* in the new coloring (whereas y retains color *2*).

Also,

$y \in C_{12} - v_1 \subset C_{12}'$.

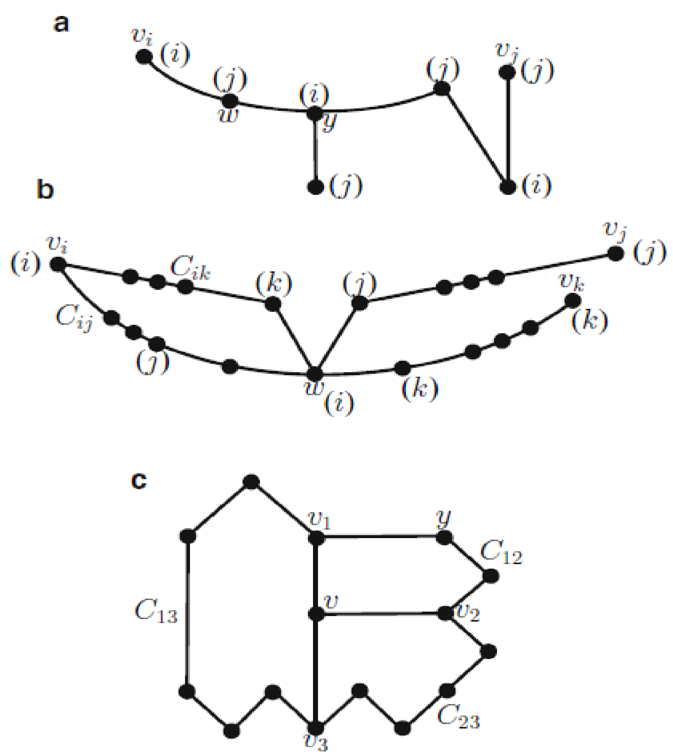
Thus,

$y \in C_{23}' \cap C_{12}'$.

This contradicts Claim 4 (since $y \neq v_2$),

and the proof is complete.

Fig. 7.3 Graphs for proof of Theorem 7.3.7 (The numbers inside the parentheses denote the vertex colors)



Exercise 3.1. Prove that any critical graph is connected.

Exercise 3.2. Show that a graph is 3-critical if and only if it is an odd cycle. It is clear that any k-chromatic graph contains a k-critical subgraph.

Exercise 3.3. If $\chi(G) = k$, show that G contains at least k vertices each of degree at least $k - 1$.

Exercise 3.4. Prove or disprove: If G is k-chromatic, then G contains a K_4.

Study Learning Material Prepared by

Dr. I. VALLIAMMAL, M.Sc., M.Phil., Ph.D.,

Assistant Professor, Department of Mathematics,

Manonmaniam Sundaranar University,Tirunelveli-12,

Tamil Nadu, India.