NEP NEW SYLLABUS

GRAPH THEORY

for

First Year B.Sc. (Computer Science) : Semester - II

[Course Code : MTS-151-T - Credit : 02]

2024 Patter as per NEP Syllabus

w.e.f. Academic Year 2024-25

M. D. Bhagat
Ex-Head Dept. of Mathematics
Tuljaram Chaturchand College
Baramati (Dist. Pune)

R. S. Bhamare
Ex-Head of P.G. Dept. of Mathematics
New Arts, Science & Commerce College
Ahmednagar.

N. M. Phatangare
Assistant Prof. Dept. of Mathematics
Fergusson College,
Pune 411 004.

Dr. A. S. Khairnar
Assistant Professor & HOD of Mathematics,
MES's Abasaheb Garware College,
Pune 411 004.

Dr. Anjali K. Shinde
M.Sc., Ph.D., SET, M.Ed.
Prof. & Head
Dept. of Mathematics,
K.R.T. Arts, B.H. Commerce and A.M. Science College,
Nashik.

Price ₹ 200.00

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N8317

GRAPH THEORY **ISBN 978-93-6109-915-1**

First Edition : January 2025

© : Authors

The text of this publication, or any part thereof, should not be reproduced or transmitted in any for or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom. The reader must cross check all the facts and contents with original Government notification or publications.

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312, Shivaji Nagar,
Off J.M. Road, Pune – 411005
Tel - (020) 25512336/37/39
Email : niralipune@pragationline.com

Polyplate

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Survey No. 10/1A, Ghule Industrial Estate
Nanded Gaon Road
Nanded, Pune - 411041

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan
(For orders outside Pune)
S. No. 28/27, Dhayari Narhe Road, Near Asian College
Pune 411041, Maharashtra
Tel : (020) 24690204; Mobile : 9049672225
Email : bookorder@pragationline.com

Nirali Prakashan
(For orders within Pune)
119, Budhwar Peth, Jogeshwari Mandir Lane
Pune 411002, Maharashtra
Mobile : 9657703145, 9890997937
Email : niralilocal@pragationline.com

MUMBAI

Nirali Prakashan
Rasdhara Co-op. Hsg. Society Ltd., 'D' Wing Ground Floor, 385 S.V.P. Road
Girgaum, Mumbai 400004, Maharashtra
Mobile : 7045821020, Tel : (022) 2385 6339 / 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

DELHI

Nirali Prakashan
Room No. 2 Ground Floor
4575/15 Omkar Tower, Agarwal Road
Darya Ganj, New Delhi 110002
Mobile : 9555778814/9818561840
Email : delhi@niralibooks.com

BENGALURU

Nirali Prakashan
Maitri Ground Floor, Jaya Apartments,
No. 99, 6th Cross, 6th Main,
Malleswaram, Bengaluru 560003
Karnataka; Mob : 9686821074
Email : bengaluru@niralibooks.com

NAGPUR

Nirali Prakashan
Above Maratha Mandir, Shop No. 3,
First Floor, Rani Jhanshi Square,
Sitabuldi Nagpur 440012 (MAH)
Tel : (0712) 254 7129
Email : nagpur@niralibooks.com

KOLHAPUR

Nirali Prakashan
438/2, Bhosale Plaza, Ground Floor
Khasbag, Opp. Balgopal Talim
Kolhapur 416 012, Maharashtra
Mob : 9850046155
Email : kolhapur@niralibooks.com

JALGAON

Nirali Prakashan
34, V. V. Golani Market, Navi Peth,
Jalgaon 425001, Maharashtra
Tel : (0257) 222 0395
Mob : 94234 91860
Email : jalgaon@niralibooks.com

SOLAPUR

Nirali Prakashan
R-158/2, Avanti Nagar, Near Golden
Gate, Pune Naka Chowk
Solapur 413001, Maharashtra
Mobile 9890918687
Email : solapur@niralibooks.com

marketing@pragationline.com | www.niralibooks.com

Also find us on www.facebook.com/niralibooks

Preface ...

We have great pleasure in presenting this text book on **Graph Theory** to the students of F.Y.B.Sc. (Computer Science) (Semester - II). This book is written according to the new revised syllabus of University under New National Education Policy (NEP) 2020 to be implemented from June 2024.

We have taken utmost care to present the matter systematically and with proper flow of mathematical concepts. We begin the Chapter by Introduction and at the end the Summary of the Chapter is provided. We have added one significant feature: **"Think Over It"** in this new **edition**. Here, we have posed questions of simple, difficult and intuitive type in nature. It is expected that the students should think over it and try to find the answers. This will assess the understanding of the knowledge of the Chapter.

The book contains several selected solved examples and an ample number of graded problems in the exercises.

We are thankful to **Shri Dineshbhai Furia, Shri Jignesh Furia**, Mrs. Anagha (Co-ordinator), Mr. Ilyas Shaikh, (D.T.P.), Mrs. Anjali Muley (Fig. Drawing) and the staff of Nirali Prakashan for the efforts they have taken to publish the book in time.

We welcome the valuable suggestions from our colleagues' and readers for the improvement of the book.

Authors

✍✍✍

Syllabus ...

1. An Introduction to Graph **(08 Hrs.)**

1.1 Definitions, Basic terminologies and properties of graphs.

1.2 Special types of graphs, Some applications of special types of graph.

1.3 Matrix representation and elementary results, Isomorphism of graphs.

2. Connected Graph **(07 Hrs.)**

2.1 Walk, trail, path, cycle, more definitions and elementary properties of connectedness.

2.2 Cut edge (Bridge), Cut vertex, Cut set, Vertex connectivity, Edge connectivity, Definitions and Properties.

2.3 Shortest path problem, Dijkstra's algorithm.

3. Euler and Hamilton Path **(07 Hrs.)**

3.1 The Konigsberg bridge problem, Euler trail, path, circuit and tour, elementary properties and examples, Fleury's algorithm.

3.2 Hamilton path, circuit, definitions, elementary properties and examples.

4. Trees **(08 Hrs.)**

4.1 Definitions, basic terminologies, properties and applications of trees.

4.2 Weighted graph, definition and properties of spanning tree, shortest spanning tree, Kruskal's algorithm, Prim's algorithm.

4.3 Binary tree, definitions and properties, tree traversal: Preoder, inorder, postorder, infix, prefix, postfix notations and examples.



Contents . . .

1.	**An Introduction to Graph**	**1.1 – 1.42**
2.	**Connected Graph**	**2.1 – 2.44**
3.	**Eulerian and Hamiltonian Graphs**	**3.1 – 3.17**
4.	**Trees**	**4.1 – 4.51**



CHAPTER 1

An Introduction to Graph

Objectives...

At the end of this chapter, students will be able to understand:

- The definitions and basic teminologies of graph theory.
- Identify and classify special types of graphs and explore their applications.
- Matrix representations of graphs.
- How to apply the concept of isomorphism to identify equivalent graphs?



Frank Harary

Frank Harary (March 11, 1921 - January 4, 2005) was a prolific American mathematician, who specialized in graph theory. He was widely recognized as one of the "fathers" of modern graph theory. Harary was a master of clear exposition and together with his many doctoral students, he standardized the terminology of graphs.

He broadened the reach of this field to include physics, psychology, sociology, and even anthropology. Gifted with a keen sense of humor. Harary challenged and entertained audiences at levels of mathematical sophistication. A particular trick he employed was to turn theorems into games - for instance, students would try to add red edges to a graph on six vertices in order to create a red triangle, while another group of students tried to add edges to create a blue triangle (and each edge of the graph had to be either blue or red). Because of the simplest case of Ramsey theory, one team or the other would have to win.

1.1 INTRODUCTION

A graph consists of two types of objects viz. vertices and edges such that each edge is associated with unordered pair of vertices. Consider the situation which describes the road map of a city joining some important places like garden (g), hospital (h), bus station (b), railway station (r), swimming tank (s), temple (t) and telephone booth (p). By taking into consideration their locations, we can represent this situation diagrammatically as shown below:

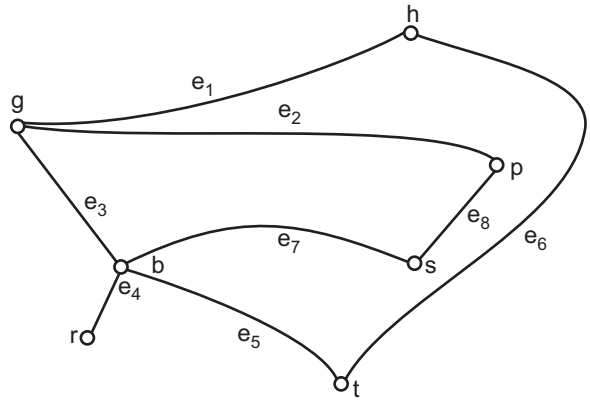


Fig. 1.1

This diagram represents a graph with 7 vertices g, b, r, t, s, p, h and 8 edges e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8.

Some other examples of graphs are :

(i) Communication lines joining military stations of a country.

(ii) A river together with its tributories and subtributories.

(iii) Ranking of workers in an office according to their ranks.

(iv) A plot of land with the walls to prevent the water from flowing.

(v) Railway route map of the country.

(vi) Nerves in the human body.

Definition: A graph G = G (V, E) consists of two sets : A non-empty set V called vertex set whose elements are called vertices (or nodes) and a set E called edge set whose elements are called edges, such that each edge $e \in E$ is associated with an unordered pair (v_1, v_2) of vertices $v_1, v_2 \in V$.

If both the sets V and E are finite then the graph is called a **finite graph**. If both the sets V and E or either of them is infinite set then the graph is called an **infinite graph**. For our purpose, the word graph would mean a finite graph.

In the above definition whether the edge e_i is associated with vertex pair (v_j, v_k) or (v_k, v_j) is immaterial. This means we are not yet assigning the directions to the edges. Such graphs in which the edges are not assigned the directions are called undirected graphs.

Fig. 1.1 shows the diagrammatic representation of a graph. In the diagrammatic representation of a graph, the vertices are denoted by small circles and the edges are denoted either by straight lines or curved lines. Another way of representing a graph is to list the edges as vertex pairs. Thus, a graph shown in Fig. 1.1 can be represented as {(g, h), (g, p), (g, b), (h, t), (p, s), (s, b), (b, t), (b, r)}.

If the edge set E of a graph G is empty then G is called a null graph. The null graphs N_1, N_2, N_3, N_4 of 1, 2, 3, 4 vertices respectively are shown below.

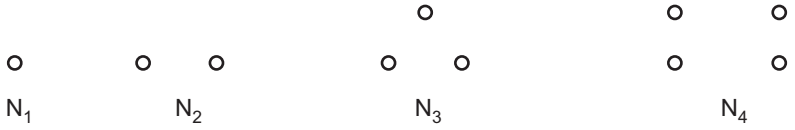
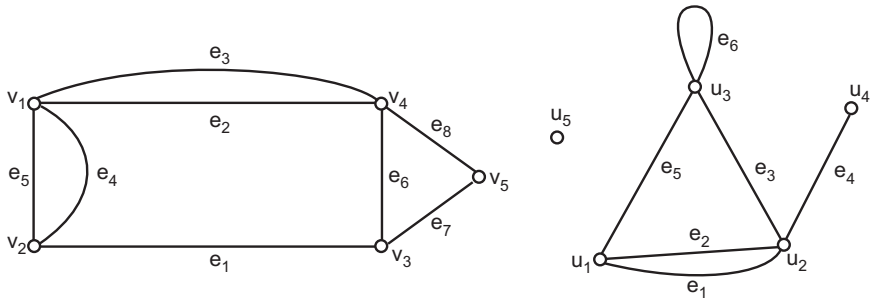


Fig. 1.2

In the graph G (V, E), if the edge e_i is associated with the vertex pair (v_j, v_k) then v_j, v_k are end vertices of e_i and equivalently we say that edge e_i is incident on v_j and v_k. Also, in this case, the vertices v_j and v_k are said to be adjacent vertices. The two edges e_i and e_j are called adjacent if they have a vertex in common. In the graph of Fig. 1.1, the vertices h and t are adjacent vertices but h and p are not adjacent because there is no edge joining h and p. Also the edges e_2 and e_8 are adjacent edges as they have a vertex p in common.

In the graph G (V, E) if two edges have the same end vertices then they are called parallel edges. If an edge e_i of a graph G has its two end vertices coincident then e_i is said to form a loop at that vertex. A graph without loops and without parallel edges is called a simple graph (or linear graph). A graph with parallel edges but having no loop is called a multigraph. A graph having parallel edges and loops is called a pseudograph.



(a) **(b)**

Fig. 1.3

Fig. 1.3 (a) shows a multigraph having e_2 and e_3 as parallel edges and also e_4 and e_5 as parallel edges.

Fig. 1.3 (b) shows a pseudograph. Here e_1 and e_2 are parallel edges and the edge e_6 forms a loop at the vertex u_3.

Degree of a vertex : Let v be a vertex in a graph G(V, E). The number of edges incident on v is called the degree of v and is denoted by d (v).

We note that every edge of the graph G contributes two degrees, one at each end vertex. Also the loop at the vertex v contributes two degrees at v. The total degree of the graph G is the sum of the degrees of all the vertices. Hence, we have the following result.

Theorem 1 The total degree of a graph G is equal to twice the number of edges in G.

This result is known as Hand Shaking Lemma; the name follows from the fact that at a party if n hand shakes occur then because each hand shake involves two hands, the total number of hands involved in hand shakes is 2n.

A vertex of degree 1 in a graph is called a pendant vertex and the corresponding edge is called a pendant edge. A vertex of degree 0 is called isolated vertex.

In Fig. 1.3 (a) we have d (v_1) = 4, d (v_2) = 3, d (v_3) = 3, d (v_4) = 4, d (v_5) = 2. The total degree is 4 + 3 + 3 + 4 + 2 = 16. Also there are 8 edges and 2 × 8 = 16. This verifies Hand Shaking Lemma.

In Fig. 1.3 (b), d (u_1) = 3, d (u_2) = 4, d (u_3) = 4, d (u_4) = 1, d (u_5) = 0. Here u_4 is a pendant vertex and e_4 is a pendant edge. The vertex u_5 is isolated vertex. We note that the edge e_6 which forms a loop at u_3 contributes 2 degrees at u_3.

As a corollary to Hand Shaking Lemma i.e. theorem 1, we have the following result. ■

Theorem 2 In a graph G (V, E) the number of vertices of odd degree is even.

Proof : Let G (V, E) be a graph with p vertices and q edges. Then by Hand Shaking Lemma, the total degree of G is 2q.

i.e. $$\sum_{i=1}^{p} d(v_i) = 2q$$

We separate the sum on left hand side into two parts viz. in the first part we consider vertices of even degree and in the second part the vertices of odd degree.

$$\therefore \quad \sum_{\substack{\text{even degree} \\ \text{vertices}}} d(v_i) + \sum_{\substack{\text{odd degree} \\ \text{vertices}}} d(v_j) = 2q$$

The first sum on left side is even. Also the right side 2q is even. Therefore, the second sum on left side must be even.

i.e. $$\sum_{\substack{\text{odd degree} \\ \text{vertices}}} d(v_j)$$ is even number.

Hence, the number of odd degree vertices must be even. ■

Example 1 : The Holywood graph

The Hollywood graph represents actors by vertices and connects two vertices when the actors represented by these vertices have worked together on a movie. This graph is a simple graph because its edges are undirected, it contains no multiple edges, and it contains no loops.

Example 2 : Round-Robin Tournaments

A tournament where each team plays with each other team exactly once is called a round-robin tournament. Such tournaments can be modeled using Directed graphs where each team is represented by a vertex. Note that (a, b) is an edge if team a beats team b. This graph is a simple directed graph, containing no loops or multiple directed edges. In Fig. 1.4, we see that Team 1 is undefeated in this tournament, and Team 3 is winless.

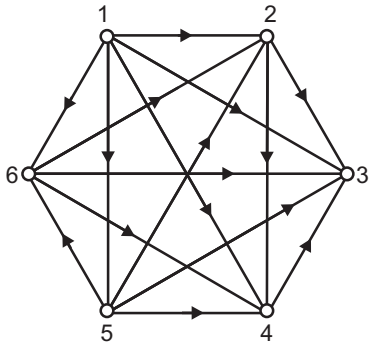


Fig. 1.4

Example 3 : The Web Graph

The world wide web can be modeled as a directed graph where each web page is represented by a vertex and where an edge starts at the web page a and ends at other removed somewhere on the web almost every second, the web graph changes on an almost continual basis. Many people are studying the properties of the web graph to better understand the nature of the web.

1.2 TYPES OF GRAPHS

(i) Weighted graph : A weighted graph is a graph in which every edge is assigned to it some positive number called *weight of that edge*. If we interpret the vertices of the graph as cities and the edges as the roads joining the cities, then the weight w (e_i) of edge e_i joining the cities v_j and v_k may be actual length of the road or it may be the cost of constructing that road or it may be the bus fare for travelling from v_j to v_k.

The total weight of the graph G is the sum of the weights of all edges of G.

(ii) Complete graph : A simple graph with n vertices such that there exists an edge between every pair of vertices is called a *complete graph* and it is denoted by K_n.

The graphs K_1, K_2, K_3, K_4, K_5 are shown below.

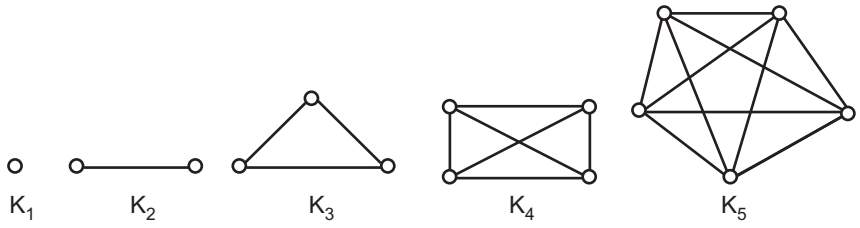


Fig. 1.5

We note that in a complete graph K_n, each vertex is joined to remaining n – 1 vertices by an edge. Hence, the degree of each vertex is n – 1. The total degree is n(n – 1). Therefore, the number of edges in K_n is $\frac{n(n-1)}{2}$.

(iii) Regular graph : Let K be a positive integer. A simple graph G in which every vertex is of degree K is called k-regular graph. A graph G is called regular if it is K-regular for some positive integer K. A graph G shown below is 2-regular.

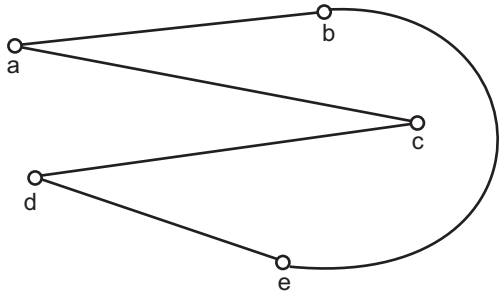


Fig. 1.6

A complete graph K_n is $n-1$ regular. We see that a complete graph is regular but regular graph need not be complete.

(iv) Bipartite graph : Let G(V, E) be a simple graph. If the vertex set V of G can be partitioned into two non-empty disjoint subsets V_1 and V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \phi$) in such a way that every edge of G has its one end vertex in V_1 and the other end vertex in V_2; then G is called a bipartite graph and the collection $\{V_1, V_2\}$ is called a bipartition.

A bipartite graph in which there exists an edge between every vertex in V_1 and every vertex in V_2 is called a complete bipartite graph. A complete bipartite graph with bipartition V_1, V_2 having m vertices in V_1 and n vertices in V_2 is denoted by $K_{m, n}$. Following figures illustrate this.

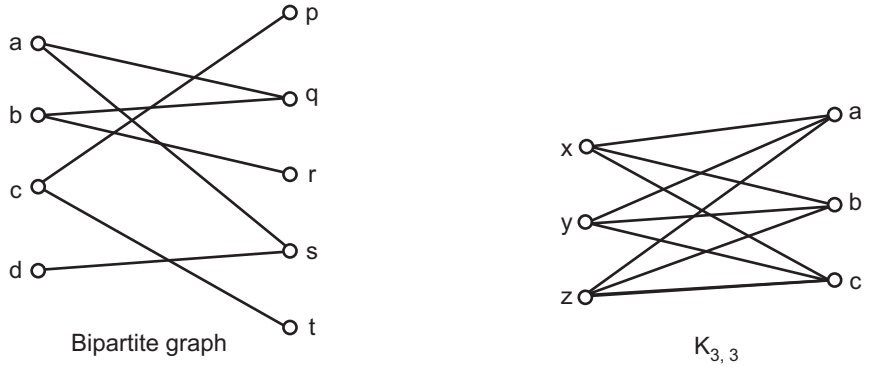


Fig. 1.7

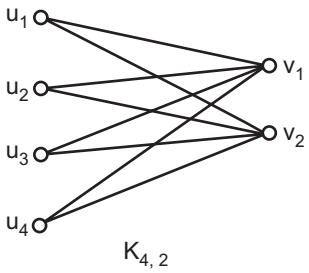
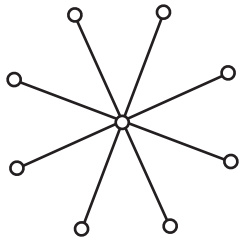


Fig. 1.8

In $K_{m, n}$ every vertex in

$V_1 = \{u_1, u_2, ..., u_m\}$ is joined to every vertex in $V_2 = \{v_1, v_2, ... v_n\}$. Therefore $K_{m, n}$ has mn edges.

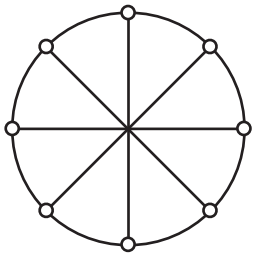
The graph $K_{1, n}$ is called a star.



$K_{1, 8}$ (8 star)

Fig. 1.9

From the star graph, we get a wheel graph.



w_8 (8 wheel)

Fig. 1.10

Consider the graph shown below.

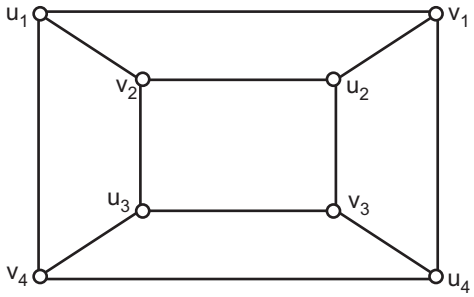


Fig. 1.11

In this graph, we observe that every edge is such that it joins u type vertex to v type vertex.

Therefore, $V_1 = \{u_1, u_2, u_3, u_4\}$ and $V_2 = \{v_1, v_2, v_3, v_4\}$ is a bipartition of the vertex set. We can redraw the above graph as below.

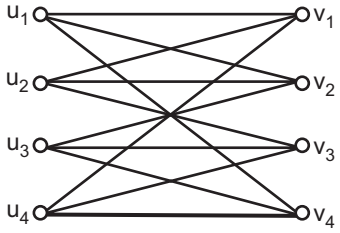


Fig. 1.12

It is a bipartite graph.

Cycles : The cycle C_n, $n \geq 3$, consists of n vertices 1, 2, …, n and edges {1, 2}, {2, 3}, …, {n – 1, n}, and {n, 1}. The cycles C_3, C_4, C_5 and C_6 are shown in the following figure.

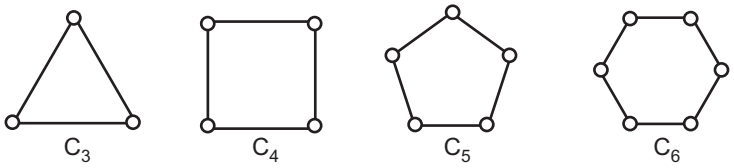


Fig. 1.13

Wheels : We obtain the wheel W_n when we add an additional vertex to the cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges. The wheels W_3, W_4, W_5 and W_6 are shown in the following figure.

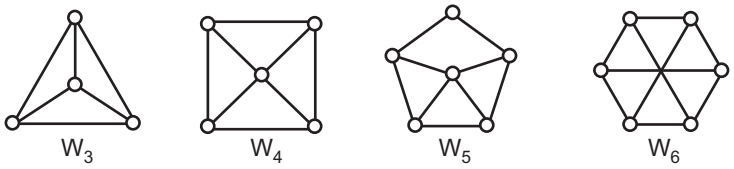


Fig. 1.14

n-Cubes : The n-dimensional hypercube, or n-cube, denoted by Q_n, is the graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position. The graphs Q_1, Q_2 and Q_3 are shown in the following figure.

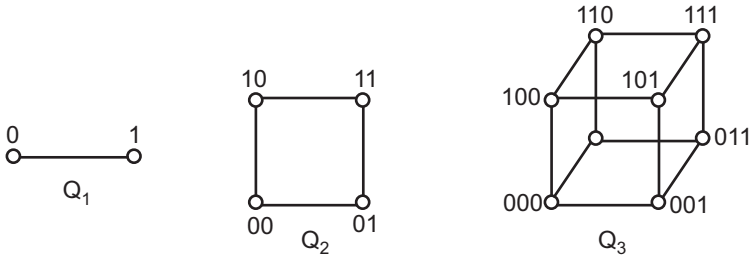


Fig. 1.15

Complete Bipartite Graphs : The complete bipartite graph $K_{m;\,n}$ is the graph that has its vertex set partitioned into two subsets of m and n vertices, respectively. There is an edge between two vertices if and only if one vertex is in the first subset and the other

vertex is in the second subset. The complete bipartite graphs $K_{2,3}$, $K_{3,3}$, $K_{3,5}$ and $K_{2,6}$ are shown in the following figure.

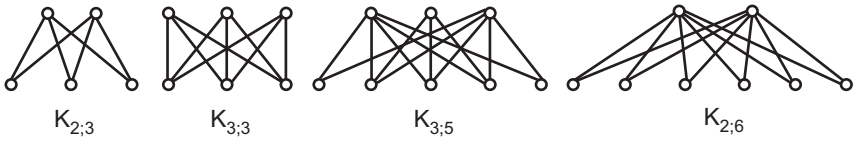


Fig. 1.16

Subgraph : A subgraph of a graph G = (V, E) is a graph H = (W, F), where $W \subseteq V$ and $F \subseteq E$. A subgraph H of G is a proper subgraph of G if $H \neq G$.

Example : Few subgraphs of K_4 are shown in the following figure.

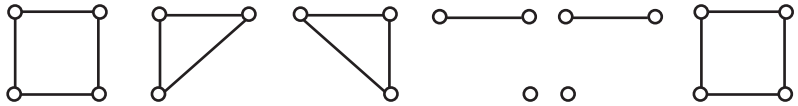


Fig. 1.17

Union of Two Graphs : The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.

Example : The two graphs G_1 and G_2 and their union is shown below.

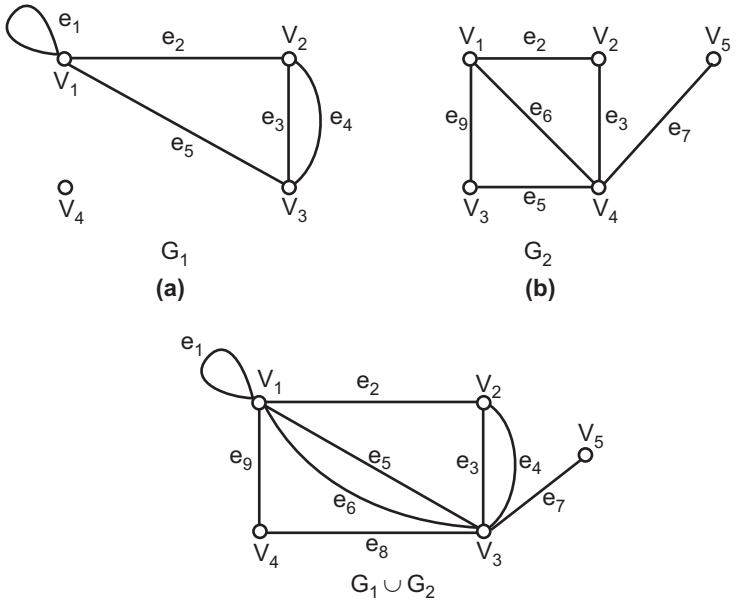


Fig. 1.18

Applications of Special Types of Graphs:

Graphs play an essential role in modeling real-world problems. Here are some simple applications of special types of graphs.

1. Job Assignment using Bipartite Graphs:

A bipartite graph is a graph in which the set of vertices is divided into two disjoint sets, and every edge connects a vertex from one set to a vertex from the other set. This type of graph is used to model problems where we need to assign jobs to workers.

For example, consider a scenario where we have a set of workers and a set of jobs. Each worker can do certain jobs. The bipartite graph can represent this situation, where one set of vertices represents workers, and the other set represents jobs. An age between a worker represents workers, and the other set represents jobs. An edge between a worker and a job indicates that the worker can do that job. We can use the bipartite graph to find a way to assign workers to jobs efficiently, ensuring that each job is assigned to one worker, and every worker is assigned a job.

2. Local Area Networks (LANs) using Complete Bipartite Graphs:

In computer networks, a complete bipartite graph $K_{1, n}$, also known as a star graph, is often used to model a simple Local Area Network (LAN). In this model, one central node (such as a router or server) is connected to several peripheral nodes (such as computers or devices), but the peripheral nodes are not connected to each other.

A compete bipartite graph $K_{1, n}$ consists of two sets of vertices: one set has one vertex (representing the central node), and the other set has n vertices (representing the peripheral devices). This kind of structure ensures that all devices are connected to the central node, but not directly to each other, which is typical for a star network.

3. Local Area Networks Using Cycles C_n:

Another common graph used to model LANs is the cycle graph C_n. A cycle graph consists of a set of n vertices, where each vertex is connected to two other vertices, forming a closed loop. This type of network can represent situations where devices are connected in a ring, such as in certain types of networking protocols.

In a cycle graph C_n, each device (node) is connected to two other devices, forming a circular structure. This is useful for modeling certain types of communication networks where messages are passed from one device to another in a circular manner until they reach their destination.

4. Interconnection Networks for Parallel Computation:

In parallel computing efficient communication between different processors is crucial. Special types of graphs, such as hypercube graphs and complete graphs, are used to model interconnection networks that allow processors to communicate efficiently.

For example, the hypercube graph is used to represent a high-dimensional network where each vertex represents a processor and edges represent direct communication between processors. These types of graphs are designed to ensure that each processor can quickly communicate with others, making them ideal for parallel computing applications.