BLOCK II:
GRAPH THEORY

Unit 1 : Introduction to Graph
Unit 2 : Paths and Circuits-I
Unit 3 : Paths and Circuits-II
Unit 4 : Trees
Unit 5 : Graph Representation

UNIT 1: INTRODUCTION TO GRAPH

Space for learners:

Unit Structure:

1.1 Introduction
1.2 Unit Objectives
1.3 Brief history on development of graph theory
1.4 Basic Concepts
1.4.1 Definition of a graph
1.4.2 Basic terminologies
1.4.3 Finite and infinite graphs
1.4.4 Directed and undirected graphs
1.4.5 Different types of Digraphs
1.4.6 Incidence and Degree
1.4.7 Out-Degree and in-Degree in Directed Graph
1.4.8 Isolated Vertex, Pendant Vertex and Null Graph
1.4.9 Some Results
1.5 Summing Up
1.6 Answers to Check Your Progress
1.7 Possible questions
1.8 References and Suggested Readings

1.1 INTRODUCTION

Space for learners:

In this unit, you will learn the fundamental aspects of graph theory. You will also learn about finite and infinite graph, directed and undirected graphs, incidence and degree, isolated and pendant vertices, null graph. You will also learn the history of graph theory in this unit. Graph theory is an area of mathematics which is realistic in its nature. The purpose of graph theory is to solve day to day problems of human beings. In this unit, you will also learn several properties related to finite and infinite graphs. This unit tries to simplify the ideas related to directed and undirected graphs. Incidence and degree of a vertex are two of the main building blocks of graph theory. You will learn some fundamental properties related to degree. Moreover, various examples will be discussed in this unit. These examples will help your knowledge to grow. Applications of graph theory can be found in various areas of mathematics, computer science, biology, theoretical chemistry, social networks, etc. since the scopes of applications are limited in this unit, thus we will skip applications and we will mainly focus on theoretical foundations only.

1.2 UNIT OBJECTIVES

After going through this unit, you will be able to:

- know the history of development of graph theory in concise manner
- understand the fundamental concepts and notions of graph theory
- define graph and its different types viz. finite and infinite graphs, directed and undirected graphs. Incidence and degree, isolated and pendant vertices, null graphs etc. will be discussed.
- solve problems related to above graphs.

1.3 BRIEF HISTORY ON DEVELOPMENT OF GRAPH THEORY

Space for learners:

The subject “graph theory” was initiated from a real problem. The problem is known as “Konigsberg bridge problem”. It was one of the unsolved problems of 18^{th} century. But, mathematician Leonhard Euler (1707-1782) solved this famous problem in 1736. The problem is discussed below.

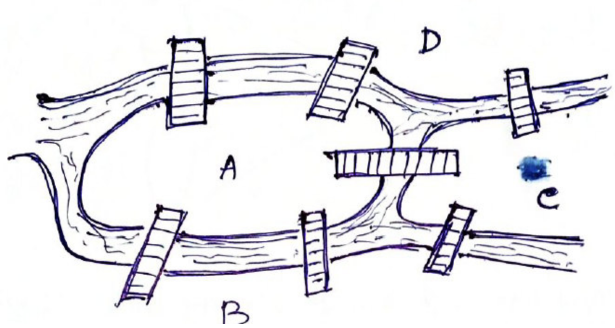


Fig. 1: Konigsberg bridge problem

Two islands, A and C, of the Pregel River in Konigsberg were linked to each other and to the banks, B and D, of the Pregel River by seven bridges as shown in Fig 1. The problem was to start at any of the four land areas A, B, C or D, walk across each of the seven bridges exactly once, and return to the starting point. Euler proved that there is no solution of the Konigsberg bridge problem. To give proof, Euler simplified the problem. He represented each land area by a point and each bridge by a line joining the corresponding points. Thus, this simplified representation produces a graph. Euler’s representation of the Konigsberg bridge problem is shown in Fig2.

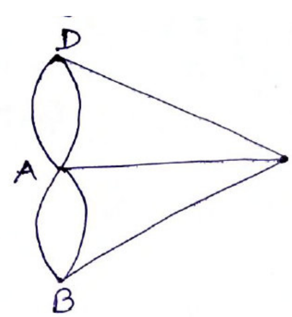


Fig. 2: The graph of the Konigsberg bridge problem

Since the development of graph theory by Euler, it has been using in many areas. Physicist Gustav Kirchhoff used theory of tree in 1845 to solve the system of simultaneous linear equations representing the current in each branch and around each circuit of an electric network. Kirchhoff used simple representations of the electric networks of the circuits using only points and lines without indicating electrical elements of the circuits. Similarly, Caylay discovered trees and his ideas of trees were applied to enumerate the isomers of the saturated hydrocarbons C_nH_{2n+2}, where n represents the number of carbons atoms. Some of the saturated hydrocarbons are given below.

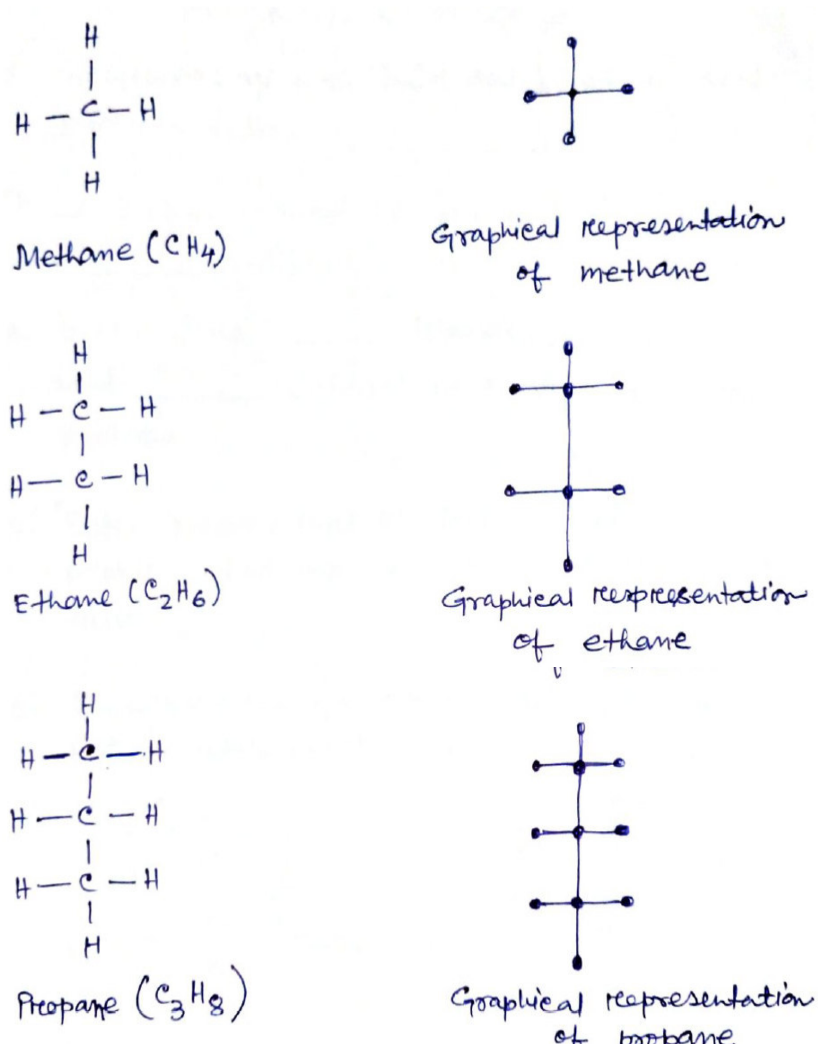


Fig. 3: Some saturated hydrocarbons and their graphical representations.

Space for learners:

These are some of the fields where graph theory has been applied. But, the reach of graph theory to any area of science and social science can be found easily. Computer networking, an area of computer science, has its foundation based on graph theory.

Space for learners:

CHECK YOUR PROGRESS-I

1. Graph theory was initiated from a real problem entitled …………………
2. ………………. Solved Konigsberg bridge problem in ……………….
3. There were ……………. Islands, ……………... banks and ……………. Bridges in Konigsberg bridge problem.
4. "Euler proved that Konigsberg bridge problem has no solution" ……. is the statement true?

1.4 BASIC CONCEPTS

In this section, we study the basic concepts of graph theory. It is important to note that graph theory does not require any other sophisticated area of mathematics other than basic set theory to introduce fundamental notions and definitions of graph theory. Thus, it is expected that our readers are familiar to basic set theory.

1.4.1 Definition of A Graph

A graph (or simply a linear graph) G = (V, E) consists of a set of points V= $\{v_1, v_2, v_3, \dots \dots\}$ and a set E= $\{e_1, e_2, e_3, \dots \dots\}$ of unordered pairs of points of V.

The points $v_1, v_2, v_3, \dots \dots$ are called vertices and e_1, e_2, e_3,…. are called edges. Several books use the words 'nodes' in lieu of vertices and 'lines' in lieu of edges.

Each edges e_k is represented by an unordered pair (v_i, v_j). It is also denoted as $\{v_i, v_j\}$. Thus, (v_i, v_j) and $\{v_i, v_j\}$ are not different in graph theory. $e_k = (v_i, v_j)$ is said to join vertex v_i and vertex v_j. We write $e_k = v_iv_j$ and say that e_i and e_j are adjacent points. It is sometimes denoted as $e_i adj e_j$. The points v_i and line e_k are incident with each other. Similarly, v_j and the line e_k are incident with each other. Two lines e_k and e_m are adjacent lines if they are incident with a common point.

Example 1: If G = (V, E) be a graph, where V = {x, y, z, p, q} and E={(x, y), (y, z), (x, q), (x, p), (y, p)}, then the graph G = (V,E) can be represented as given below.

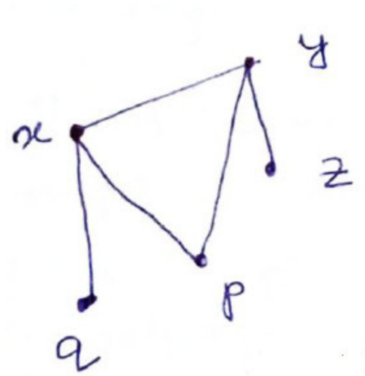


Fig. 4: Graph of G=(V,E)

If we represent e_1=(x,y), e_2=(y,z), e_3=(x,q), e_4=(x,p) and e_5=(y,p), then the graph can be represented as given below

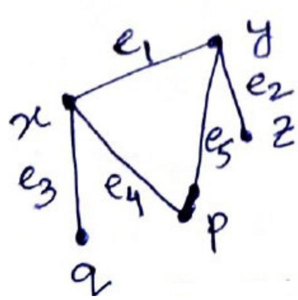


Fig. 5: Graph of G = (V, E)

Space for learners:

CHECK YOUR PROGRESS-II

5. A graph G = (V, E) consist of ……… and ……………

6. The elements of the set V of a graph G = (V, E) are called………. .

7. The elements of the set E of a graph G = (V, E) are called ……….. .

__Space for learners:__

1.4.2 Basic Terminologies

In this section, we study basic terminologies of graph theory.

Definition1: A graph with p-vertices and q- edges is called a (p,q) graph.

Example 2: A (3,3) graph is represented as given below

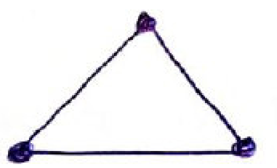


Fig. 6: (3,3)-graph

Definition 2: The (1,0)-graph is called trivial graph

Example 3: The trivial graph is shown below

Definition 3: In graph G=(V, E); an edge to be associated with a vertex pair (vivi) is permissible. Such an edge is called a loop (or self-loop).

Example 4: Let G=(V, E) be a graph, where V={x,y,z} and E= {(x, x), (x, y), (x, z), (y, z), (z, z)}.

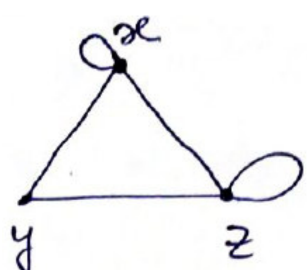


Fig. 7: Graph of G=(V, E).

If we denote e1=(x1, x), e2=(x, y), e3=(x, z), e4=(y, z) and e5=(z, z), then figure 7 can be re-drawn as below.

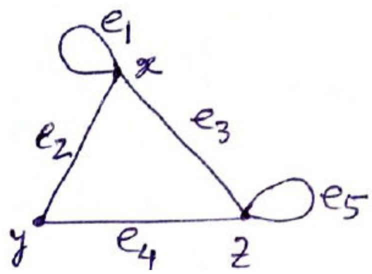


Fig. 8: Graph of G=(V, E) with distinct edges.

Definition 4: In a multigraph, no loops are allowed but more than one edge can join two vertices. These edges are called multiple edges (or multiple lines).

Example 5: If G = (V, E) be a graph, where V= {x, y, z, p} and E={(x, y), (x, z), (x, p), (y, z), (y, z)} then G=(V,E) is a multigraph. This graph is shown below.

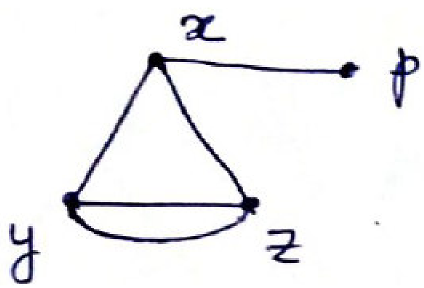


Fig. 9: Multigraph G=(V, E)

Here, (y, z) is considered twice in the set e. If we represent G = (x, y), e_2=(x, z), e_3=(x, p), e_4=(y, z) and e_5=(y, z), then fig 9 can be represented as given below.

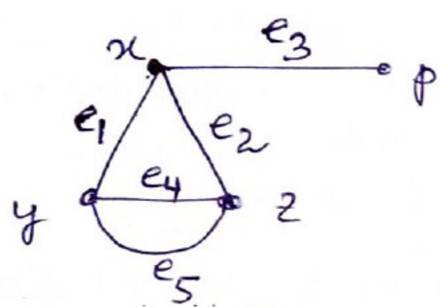


Fig. 10: Multigraph G=(V, E)

Space for learners:

Since e_4 and e_5 are distinct edges, thus it is not problematic for our readers to write (y, z) twice in the set E. Such edges are known as parallel edges.

Definition 5: A multigraph having loop(s) is called a pseudo graph.

Example 6: If G = (V, E) be a graph, where V = {x, y, z, p} and E = {(x, x), (x, y), (x, z), (x, p), (y, z), (y, z)}, then G = (V, E) is a pseudo graph.

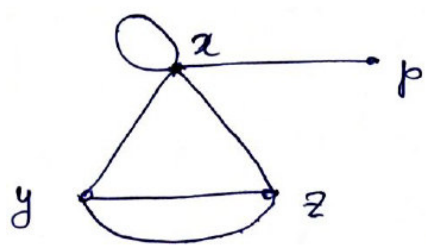


Fig. 11: Pseudo graph G=(V, E)

Definition 6: A graph which has neither loops nor multiple edges is called a simple graph.

Example7: Fig5, fig6 are example of simple graph.

CHECK YOUR PROGRESS-III

8. Euler's graph representing Konigsberg bridge problem is a ……… graph.
9. In a multigraph, ……….. are allowed.
10. Is loop allowed in a multigraph?
11. Is loop allowed in a simple graph?
12. Are multi edges allowed in a simple graph?
13. "Every pseudo graph is a simple graph"- true or false?

Space for learners:

1.4.3 Finite and Infinite Graphs

In this section, we study finite and infinite graph.

Definition7: A graph with a finite number of vertices as well as a finite number of edges is called a finite graph; otherwise, it is an infinite graph.

The graphs, which we considered earlier, are all finite graphs.

Example 8:

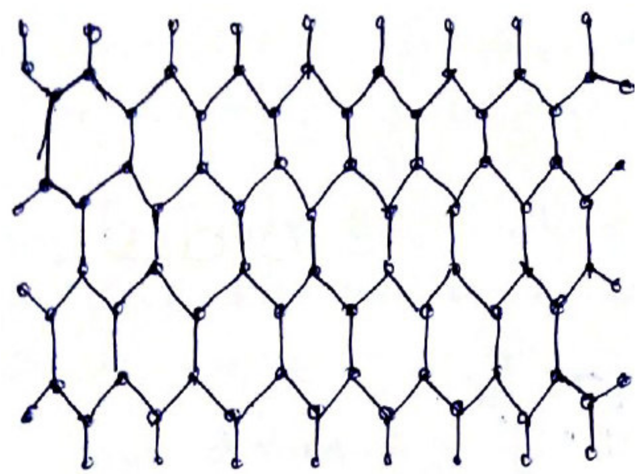


Fig. 12: Portion of an infinite graph

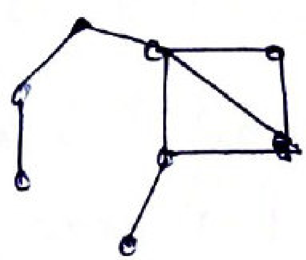


Fig. 13: a finite graph

In reality, Fig 12 is the graphical representation of Graphene. Graphene is one of the very important atomic-scale hexagonal lattices made of carbon-atoms. The discoverers of Graphene were awarded Nobel prize in physics in 2010.

Space for learners:

1.4.4 Directed and Undirected Graphs

Space for learners:

In this section, we discuss directed and undirected graphs. These graphs are used in several real-life problems.

Definition 8: A directed graph (or digraph) G=(V,E) consists of a set of vertices $V=\{v_1, v_2, v_3, ...\}$, a set of edges $E=\{e_1,e_2,e_3,....\}$ and a mapping of that maps every edges onto some ordered pair of vertices $(v_{i,}, v_j)$. A diagraph is also referred to as oriented graph.

We often make a distinction between the terms “oriented graph” and “directed graph” by considering only these digraphs which have at most one directed edge between a pair of vertices (for digraphs).

The elements of E are called directed edges (or directed lines or areas).

In digraphs, a vertex is represented by a point and an edge by a line segment between v_i and v_j with an arrow directed from vertex v_i to vertex v_j.

Example 9:

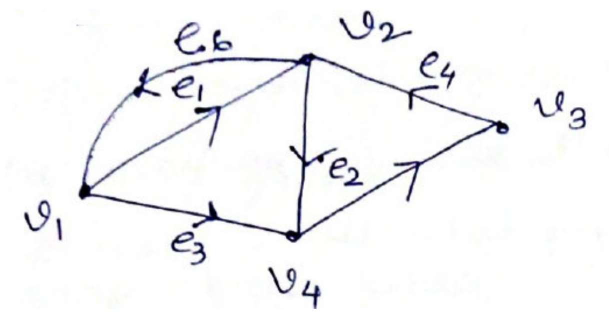


Fig. 14: A digraph with four vertices and six edges.

Suppose $e_k=(v_i, v_j)$ is a directed edge in a digraph G = (V, E); then v_i is called the initial vertex of e_k and v_j is called the terminal vertex of e_k. In this case, e_k is said to be incident from v_i and to be incident to v_j. Also, v_i is adjacent to v_j; and v_j is adjacent from v_i.

Definition 9: An undirected graph G=(V, E) consist of a set $V=\{v_1, v_2, v_3, ...\}$ of vertices and a set $E= \{e_1,e_2, e_3,......\}$ of edges such that each edges $e_k \in E$ is associated with an unordered pair of vertices.

Example 10:

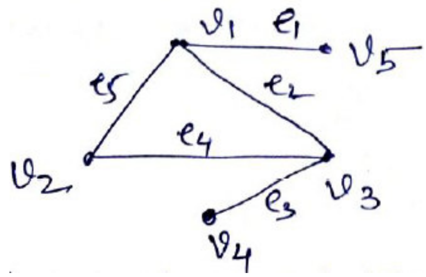


Fig. 15: An undirected graph with five vertices and five edges.

1.4.5 Different Types of Digraphs

There are various types of digraphs available in literature. Here we discuss some of them.

Definition 10: A digraph that has no parallel edges or self-loops is called a simple digraph.

Example 11:

Fig. 16: A simple digraph

Definition 11: A digraph that has at most one directed edge between a pair of vertices, but is allowed to have loops, is called an asymmetric graph or anti symmetric digraph.

Example 12:

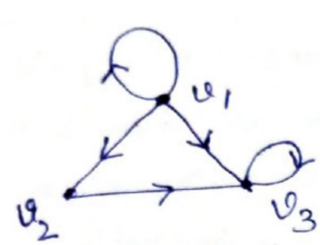


Fig. 17. An asymmetric digraph

Space for learners:

Definition 12: A digraph that is both simple and symmetric is called a simple symmetric digraph.

Definition 13: A digraph that is both simple and asymmetric is called simple asymmetric is called simple asymmetric digraph.

Definition 14: A simple digraph in which there is exactly one edge directed from every vertex to other vertex is said to be a complete symmetric digraph.

Space for learners:

1.4.6 Incidence and Degree

In this section, we discuss incidence and degree.

Definition 15: A vertex v_i and an edge e_k are said to be incident with (on or to) each other, if v_i is an end vertex of the edge e_k.

Example 13:

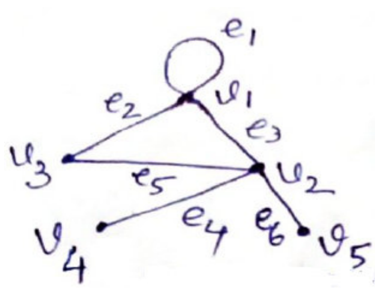


Fig. 18: Vertex incident

Here, Vertex v_4 is incident with edge e_4. The edges e_5, e_3, e_4 and e_6 are incident with vertex v_2.

Definition 16: Two non-parallel edges are said to be adjacent if they are incident on a common vertex.

Example 14: In example 13, the edges e_3, e_4, e_5 and e_6 are adjacent.

Definition 17: Two vertices are said to be adjacent if there is an edge between them.

Example 15:

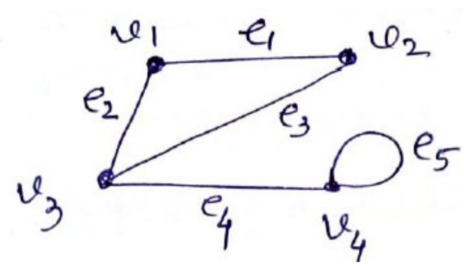


Fig. 19: v_1 and v_2 are adjacent

Here, vertices v_1 and v_2 are adjacent because there is an edge e_1 between them.

Definition 18: Degree of a vertex v_i is the number of edges incident on a vertex v_i, with loops counted twice. It is denoted as $d(v_i)$

Example 16: In example 15, $d(v_1) =2$, $d(v_2) =2$, $d(v_3) =3$ and $d(v_4) =3$

Definition 19: A regular graph is a graph in which all vertices are of equal degree.

Example 17:

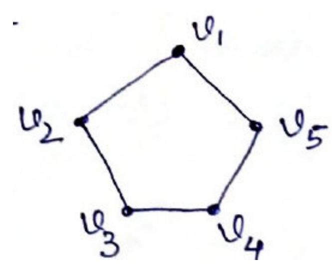


Fig. 20: A regular graph G = (V, E) of degree two.

Here, $d(v_i) = 2 \ \forall \ i = 1,2,3,4,5$. Thus, G=(V, E) is a regular graph of degree two.

1.4.7 Out-Degree and in-Degree in Directed Graph

Definition 20: In a digraph, out-degree of a vertex v_i is the number of edges incident out of a vertex v_i. It is denoted by $d^+(v_i)$.

Space for learners:

Example 18:

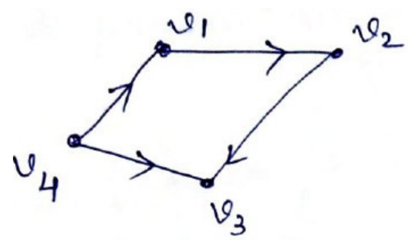


Fig. 21: Out-Degree of G = (V, E)

Here, in the graph G = (V, E); $d^+(v_1) = 1$, $d^+(v_2) = 1$, $d^+(v_3) = 0$ and $d^+(v_4) = 2$.

Definition 21: In a digraph, indegree of a vertex v_i is the number of edges incident into v_i. It is denoted by $d^-(v_i)$.

Example 19: In example 18, $d^+(v_1) = 1$, $d^+(v_2) = 1$, $d^+(v_3) = 2$ and $d^+(v_4) = 0$.

1.4.8 Isolated Vertex, Pendant Vertex and Null Graph

Definition 22: A vertex is said to be an isolated vertex if it has degree zero.

Definition 23: A vertex having degree one is called a pendant vertex (or an end vertex)

Example 20:

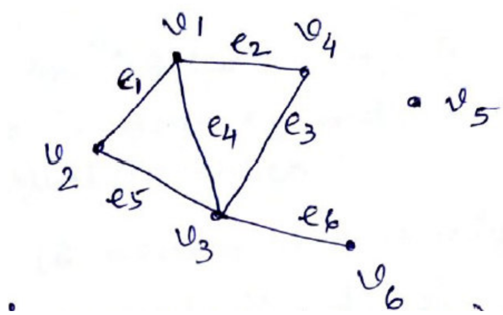


Fig. 22: A graph G = (V, E) with six vertices and six edges.

In the above graph, v_5 is isolated vertex and v_6 is the pendant vertex.

Space for learners:

Definition 24: If two adjacent edges have their common vertex of degree two then the two edges are said to be in series.

Example 21: In example 20, the edges e_1 and e_5 are in series.

Definition 25: A graph is said to be a null graph if every vertex of it has degree zero.

Example 22:

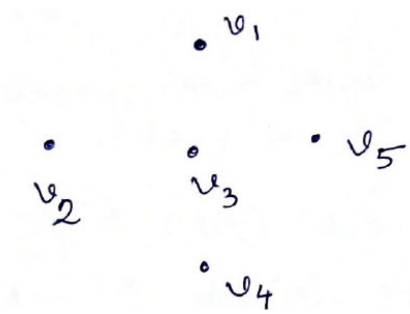


Fig. 23: Null graph of five vertices

Definition 26: An isolated vertex is a vertex in which the in-degree and the out-degree are both equal to zero.

Definition 27: A vertex v_i in a digraph is said to be pendant if $d^+(v_i) + d^-(v_i) = 1$.

1.4.9 Some Results

In this section, we discuss some theorems, problems, etc. related to previous sections.

Theorem1: If G = (V, E) be an undirected graph, then

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

Or,

The sum of the degrees of all vertices in an undirected graph G = (V, E) is twice the number of edges in G.

Proof: Let, G = (V, E) be an undirected graph. In G, every edge is incident with exactly two vertices. Thus, each edge gets counted twice, once at each end. Moreover, degree of a vertex is the number of edges incident with that vertex. Thus, sum of the degrees of all vertices counts

Space for learners:

the total number of times an edge is incident with a vertex. Thus,

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

.

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

Proof: Let G = (V, E) be a graph. We write $V = V_1 \cup V_2$, where V_1 and V_2 are the sets of vertices with odd and even degrees respectively.

Then,

$$\sum_{v \in V} d(v) = \sum_{v_i \in V_1} d(v_i) + \sum_{v_j \in V_2} d(v_j)$$

$$\Rightarrow 2|E| = \sum_{v_i \in V_1} d(v_i) + \text{an even number}$$

$$\Rightarrow \sum_{v_i \in V_1} d(v_i) = 2|E| - \text{an even number}$$

$$\Rightarrow \sum_{v_i \in V_1}' d(v_i) = \text{an even number.}$$

Thus, the number of vertices of odd degree in G = (V, E) is even.

Theorem 3: In a directed graph G = (V, E);

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

Theorem 4: If G is a directed graph, then

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

Proof: Let, G = (V, E) be a directed graph

Space for learners:

then $$\sum_{v_i \in V} d(v_i) = \sum_{v_i \in V}' d^+(v_i) + \sum_{v_i \in V}' d^-(v_i)$$

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

$$\Rightarrow \sum_{v_i \in V}' d^+(v_i) = \sum_{v_i \in V}' d^-(v_i) = |E|. \quad //$$

Problem 1: Determine the number of edges in a graph with 5 vertices, 2 vertices of degree 4, 2 vertices of degree 3 and 1 vertex of degree2.

Solution: Let, G = (V, E) be a graph where |v| = 5. Let v_1, v_2, v_3, v_5 and v_5 are five vertices of G = (V, E).

Now, $$\sum_{v_i \in V}' d(v_i) = 2|E|$$

$$\Rightarrow (4+4)+(3+3)+2 = 2|E|$$

$$\Rightarrow |E| = \frac{16}{2} = 8$$

Thus, the number of edges of G = (V, E) is 8.

Problem 2: How many vertices are required to draw a graph with 7 edges in which each vertex is of degree2.

Solution: Let there are 'x' number of vertices in the graph.

then $$\sum_{v_i \in V}' d(v_i) = 2|E|$$

$$\Rightarrow 2x = 2 \times 7$$

$$\Rightarrow x = 7$$

So, 7 vertices are required.

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

Solution: let G = (V, E) be a simple graph then $\sum_{v_i \in V} d(v_i) = 2|E|$.

Given, |v| = n also, the maximum degree of each vertex in a simple graph can be (n-1)

Space for learners:

Therefore; $\sum_{i=1}^{n} d(v_i) = 2|E|$

$$\Rightarrow \quad n(n-1) = 2|E|$$

$$\Rightarrow \quad |E| = \frac{n(n-1)}{2}.$$

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

1.5 SUMMING UP

- "Konigsberg bridge problem" has two islands, two banks and seven bridges.
- Leonhard Euler initially represented "Konigsberg bridge problem" using a graph.
- A linear graph G = (V, E) consists of a set of vertices and edges. A graph with p-vertices and q- edges is called (p, q) graph. The (1,0)-graph is called trivial graph.
- In a multigraph, no loops are allowed but more than one edge can join two vertices. These edges are called multiple edges (or multiple lines). A multigraph having loop(s) is called a pseudo graph. A simple graph which has neither loops nor multiple edges.
- A graph with a finite number of vertices as well as a finite number of edges is called a finite graph; otherwise, it is an infinite graph.
- A simple digraph has no parallel edges or self-loops.
- A simple digraph in which there is exactly one edge directed from every vertex to other vertex is said to be a complete symmetric digraph. A digraph that has at most one directed edge between a pair of vertices, but is allowed to have loops, is called an asymmetric graph.
- In a digraph, indegree of a vertex v_i is the number of edges incident into v_i. In a digraph, out-degree of a vertex v_i is the number of edges incident out of a vertex v_i.

Space for learners:

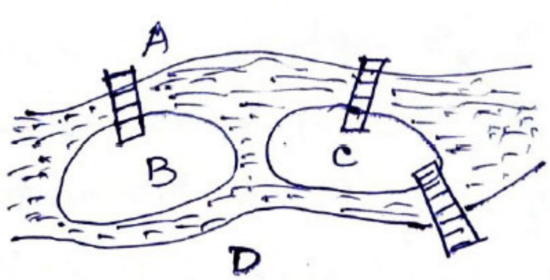
- An isolated vertex if it has degree zero. A pendant vertex has degree one. A graph is said to be a null graph if every vertex of it has degree zero.

1.6 ANSWERS TO CHECK YOUR PROGRESS

1. Konigsberg bridge problem
2. Leonhard Euler, 1736
3. Two, two, seven
4. True
5. Vertices, edges
6. Vertices
7. Edges
8. Eulerian
9. multiple edges
10. No
11. No
12. No
13. False

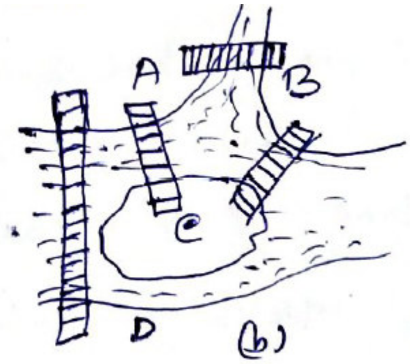
1.7 POSSIBLE QUESTIONS

1. Represent the following figures as of Euler's representation process.

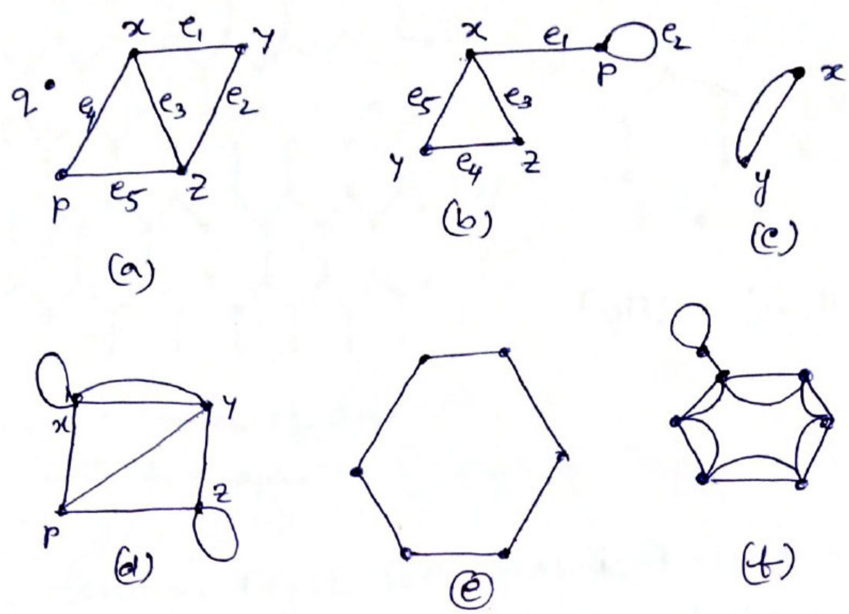


(a)

Space for learners:



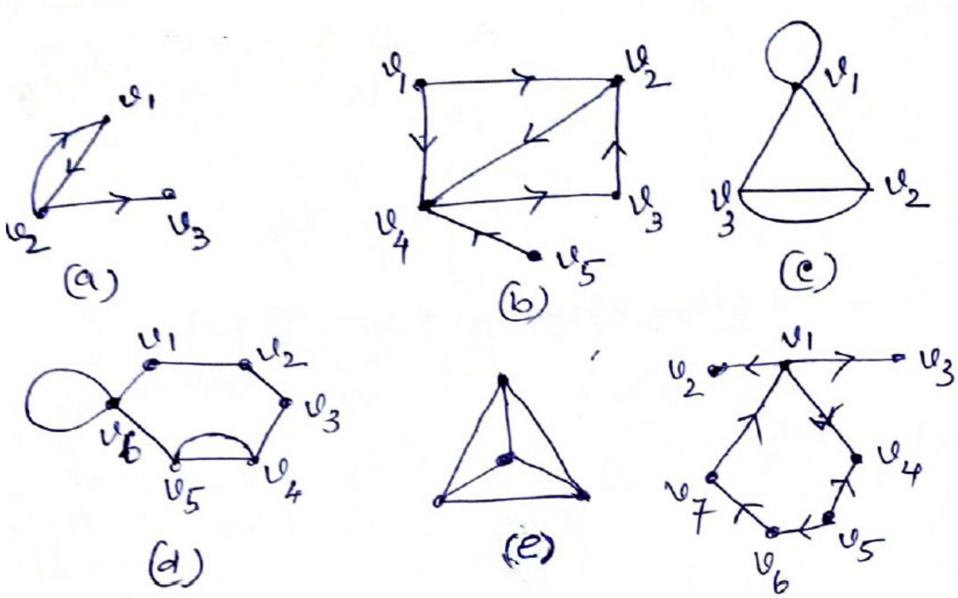
2. Represent graphically C_4H_{10}.
3. Draw the graph G = (V, E), where V = {x, y, z, p} and E= {(x, y), (x, p), (x, z), (y, z)}.
4. What is the size of an r-regular (m, n) graph?
5. Prove that the degree of a vertex of a simple graph G on n vertices cannot exceed (n-1).
6. Is it possible to draw a simple graph with 5vertices and 13 edges? Justify your answer.
7. Identify simple graphs, multigraph, pseudo graphs from the figures given below.



8. The graphical representation of C_2H_6 is a ………… graph.
9. Draw a portion of an infinite graph.

Space for learners:

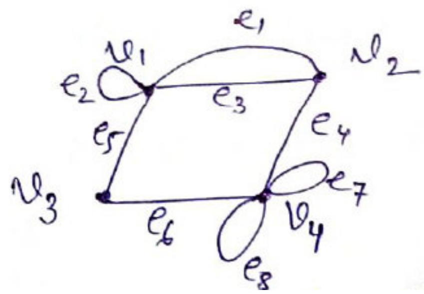
10. Draw a finite graph.
11. In a finite graph, number of vertices and number of edges are both ………. .
12. In an infinite graph, number of vertices and number of edges are both…….. .
13. Define digraph.
14. Define undirected graph.
15. Is every graph is a digraph?
16. Choose directed graphs and undirected graphs from below.



17. Define a simple digraph.
18. Define an asymmetric digraph.
19. Define a simple symmetric digraph.
20. Define a simple asymmetric digraph.
21. Define a complete symmetric digraph.
22. Draw a simple digraph, asymmetric digraph, a simple symmetric digraph, a simple asymmetric digraph, a complete symmetric digraph.
23. Define degree of a vertex.

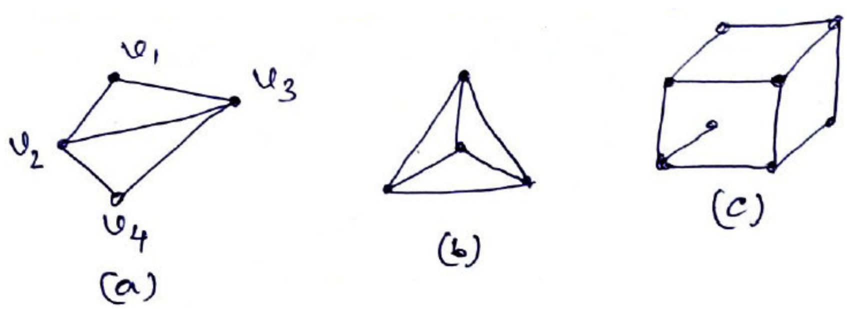
Space for learners:

24. Find degree of vertex ((()))) of the following graph.

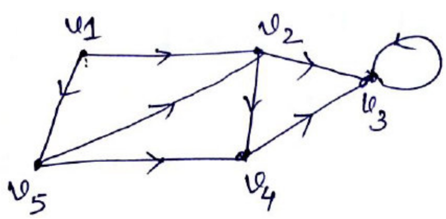


25. Degree of the vertex in (1,0) graph is …………. .

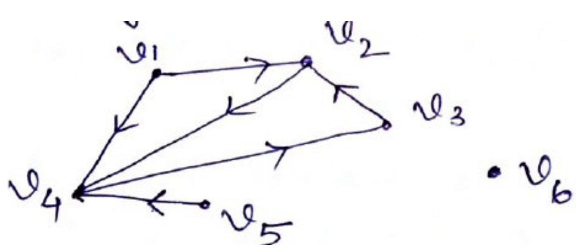
26. Identify the regular graph.



27. Find out-degree and in-degree of each vertex of the following graph.



28. Define isolated vertex.
29. Define pendant vertex
30. Find $d^+(v_i)$ and $d^-(v_i)$ $\forall i=1,2,3$ of the following graph.



Space for learners:

31. Prove that in a diagraph,

i) If v_i is an isolated vertex, then

$$d^+(v_i) = 0 \text{ and } d^-(v_i) = 0.$$

ii) If v_i is a pendant vertex, then

$$d^+(v_i) + d^-(v_i) = 1$$

Space for learners:

1.8 REFERENCES AND SUGGESTED READINGS

- Harary, F. Graph theory. Narosa Publishing House, 2001
- Deo, Narsingh, Graph theory with applications to engineering and computer science, PHI Learning Pvt. Ltd., 2013
- Diestel, Reinhard, Graph theory, Springer, 2006
- Vasudev, C. Graph theory with applications, New age international Publishers, 2018.

UNIT 2: PATHS AND CIRCUITS-I

Space for learners:

Unit Structure:

2.1. Learning Objectives

2.2. Introduction

2.3. Isomorphism in Graphs

2.4. Subgraphs

2.5. Walks, Trails, Paths and Circuit

2.6. Connected and Disconnected Graphs

2.7. Summing Up

2.8. Answers to Check Your Progress

2.9. Possible Questions

2.10. References and Suggested Readings

2.1. INTRODUCTION

Space for learners:

Many tangible real-world issues may be successfully analyzed using graphs as mathematical models. Graph theory may be used to formulate issues in physics, chemistry, communication science, computer technology, genetics, psychology, sociology, and linguistics. Graph theory also has strong ties to several disciplines of mathematics, including group theory, matrix theory, probability, and topology. The development of different subjects in graph theory has been aided by several puzzles and issues of a practical character. The classic Konigsberg bridge issue served as a model for the creation of Eulerian graph theory. The Hamiltonian graph theory was derived from Sir William Hamilton's "Around the World" game. The study of "trees" was created to enumerate isomers of chemical compounds, and the idea of acyclic graphs was developed to solve difficulties with electrical networks. In this unit, we present some fundamental concepts of graph theory which include graph isomorphism, various types of subgraphs, walks, trails, paths and circuit.

2.2. UNIT OBJECTIVES

After going through this unit, you will be able to:

- Explain the definition, concept, and properties of graph isomorphism.
- Explain and differentiate various types of subgraphs.
- Define walks, trails, paths and circuits.
- Differentiate between connected and disconnected graph.

2.3. ISOMORPHISM IN GRAPHS

In graph theory, a graph G can be called as equivalent to another graph G' if both the graphs are identical in terms of their vertices and edges. This concept is called as graph ***isomorphism***. Two ***isomorphic*** graphs may use different labels for the vertices and may have drawn differently, but they have exactly the same number of vertices and same sets of edges. The formal definition on graph isomorphism is presented below.

Definition 2.3.1: Graph ***Isomorphism*** is a concept in graph theory which states that any two graphs, G and G' are called as ***Isomorphic*** if there is a bijection between the vertex sets of G and G'. Formally, a graph $G(V, E)$ is isomorphic to another graph $G'(V', E')$ if there exists a bijective function $\Phi: V \rightarrow V'$ such that if any vertices, $u, v \in V$ are there is an edge from u to v in G then there must be an edge from $\Phi(u)$ to $\Phi(v)$ in G'. Mathematically two isomorphic graphs G and G' are denoted as $G \simeq G'$. The map Φ is termed as an isomorphism from G to G'.

Example 2.3.1: The graphs shown in figure 2.1 are examples of isomorphic graphs. Both the graphs have equal number of vertices and edges. The graph on left has the vertices, $< u_1, u_2, \ldots \ldots \ldots, u_n >$ and the graph on right has the vertices $<v_1, v_2, \ldots \ldots \ldots, v_n >$. In this example, $\Phi(u_i) = v_i$. The adjacency matrices of both the graphs are presented in Table 2.1. The adjacency matrices of both the graphs are identical as they are isomorphic.

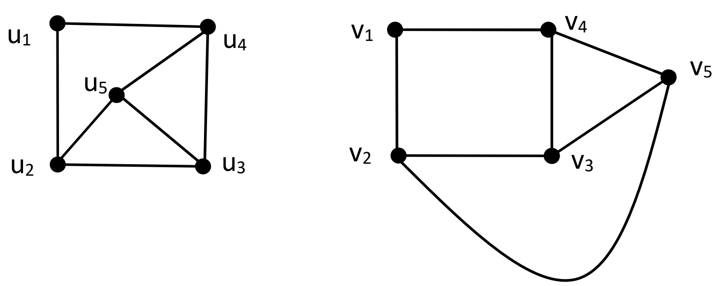


Fig. 2.1: Isomorphic Graphs

Table 2.1: Adjacency matrices of the graphs shown in Fig. 2.1

	u1	***u2***	***u3***	***u4***	***u5***
u1	0	1	0	1	0
u2	1	0	1	0	1
u3	0	1	0	1	1
u4	0	1	1	0	1
u5	0	1	1	1	0

	v1	***v2***	***v3***	***v4***	***v5***
v1	0	1	0	1	0
v2	1	0	1	0	1
v3	0	1	0	1	1
v4	0	1	1	0	1
v5	0	1	1	1	0

Space for learners:

Theorem 2.3.1: Consider ϕ be an isomorphism of the graph $G = (V, E)$ to the graph $G' = (V', E')$. Consider a vertex $v \in V$. Then $degree(v) = degree(\phi(v))$. i.e., the degree of vertices is preserved by isomorphism.

Proof: Consider two vertices $u, v \in V$. If u is adjacent to $v$in graph G, then $\phi(u)$ must be adjacent to $\phi(v)$ in graph G'. So, the number adjacent vertices of $v$in G is equal to the number of adjacent vertices of $\phi(v)$ in G'. Hence, $degree(v) = degree(\phi(v))$.

Properties: If a graph G is isomorphic to another graph G' by a bijection ϕ, then the following properties hold true.

- Number of vertices in G is same as the number of vertices in G'.
- Number of edges in G is same as the number of edges in G'.
- Both in-degree and out-degree of a vertex $v$is same as the in-degree and out-degree of $\phi(v)$.

Definition 2.3.2: An ***automorphism*** of a graph is a type of symmetry in graph theory in which the graph is mapped onto itself while retaining the edge–vertex connection. In other words, a graph G is isomorphic onto itself.

Space for learners:

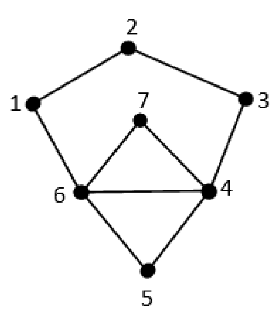
2.4. SUBGRAPHS

A graph $G'$is a subgraph of another graph G if all the vertices and edges of G' belong to the G and each edge in G' has the same source and destination in G as G'. A subgraph can be called as a subpart of another graph. The formal definition of subgraph is presented in Definition 2.4.1.

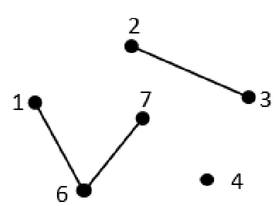
Definition 2.4.1: A graph $G'(V', E')$ is called as a subgraph of another graph $G\ (V, E)$if $V' \subseteq V$,$E' \subseteq E$ and $R_{G'}$ is the restriction ofR_G toE'. The graph $G'$is a ***proper subgraph*** of G if $V' \subset V$ or $E' \subset E$. The graph G can be called as ***supergraph*** of G' if $G'$is a subgraph of G. A graph G' is called as an ***induced subgraph*** of G if a vertex $v \in V$ is adjacent to another vertex u in G and $u, v \in V'$ then v must be adjacent to v in G'as well. If $G'$is an induced subgraph of G and vertex set of G', $V' \subseteq V$ then G' is called as the subgraph of G induced by V' and is denoted by $G[V']$. If $G'(V', E')$ is an induced subgraph of $G\ (V, E)$ and $E' \subseteq E$ then G' is called as the subgraph of G induced by E' and is denoted by $G[E']$. A subgraph $G'(V', E')$of $G\ (V, E)$ is called as a ***spanning subgraph*** of G if $V' = V$.

Example 2.4.1: Figure 2.2 (b)-(d) shows various types of subgraphs of the graph G. The graph G has the vertex set, $V = \{1,2,3,4,5,6,7\}$. The graph in (b) is a subgraph of G but not an induced subgraph because in G the vertex 1 is adjacent to 2 and vertex 4 is adjacent to 3 and 7; but the same is not true in this subgraph. Also, this is not a spanning subgraph as it does not include all the vertices of G. The graph shown in (c) is an induced subgraph of G as this subgraph doesn't include any pair of vertices which are adjacent in G but not in this subgraph. The subgraph in (d) includes all the vertices of G but not all the edges. So, this is a spanning subgraph of G.

Definition 2.4.2: A ***clique*** is a subgraph of a graph G whose vertex set is a subset of the vertex set of G and any two vertices of the clique are adjacent. Informally, a clique is a complete subgraph of another graph. That means, all the vertices of the clique are adjacent to each other. A clique is called as ***maximal clique*** if no adjacent vertex can be added to expand the clique. A ***maximum clique*** is a clique which contains maximum possible vertex.

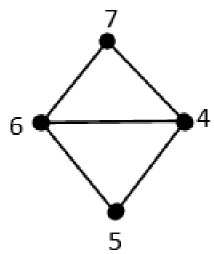


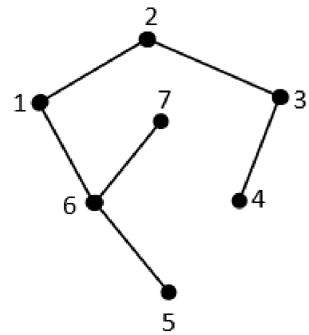
(a) Graph G



(b) Subgraph of G

Space for learners:





(c) Induced Subgraph of G (d) Spanning subgraph of G

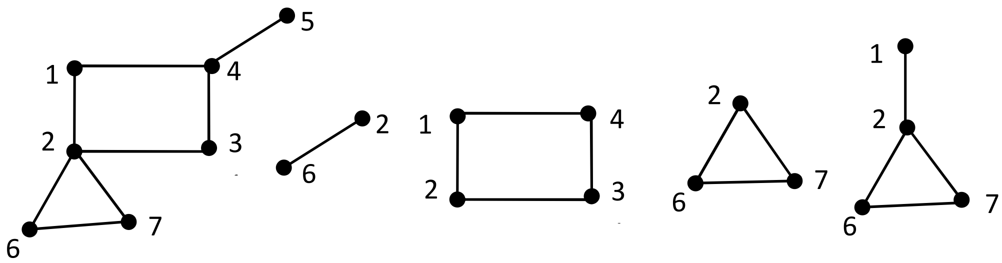
Fig. 2.2: Various types of subgraphs

Example 2.4.2: Figure 2.3 presents some examples of cliques of the graph shown in the subfigure (a). The subgraph shown in subfigure (b) is a clique as there is an edge between any two vertices in the subgraph. Similarly, the subgraphs shown in subfigures (c) and (d) are also cliques. While the subgraph in subfigure (e) is not a clique as the vertex 1 is adjacent to vertex 2 only. The cliques in (c) and (d) are maximal cliques as if we add any other vertex to them, then they will no longer be cliques. The clique in (b) is not maximal as there is a possibility of adding another vertex (vertex 7) to expand the clique.

2.5. WALKS, TRAILS, PATHS AND CIRCUIT

A ***walk*** is a finite alternating series of vertices and edges that starts and ends with vertices, with each edge connecting the vertices before and after it. A walk may have repeated vertices but not edges. A walk is called as a ***closed*** walk if the starting vertex and the ending vertex is the same, otherwise, it is called as ***open***. An open walk with no repeated edges is called as a ***trail***. The vertices may repeat in a trail. A trail with non-repeated vertices is called as a ***path***. A non-empty trail in which the starting and ending vertices are the only vertices that are repeated is called as a ***circuit*** or a ***cycle***. Definition 2.5.1 gives the formal definition of walk, trail, path and circuit.

Space for learners:



(a) Main Graph, G (b) clique of G (c) clique of G (d) clique of G (e) subgraph of

Fig. 2.3: Various subgraphs of the graph shown in (a). The subgraphs (b)-(d) are cliques but the one in (e) is not a clique. The cliques (c) and (d) are maximal cliques but not the one in (b).

Definition 2.5.1: Consider a graph $G = (V, E)$ with the vertex set, $V = \{v_0, v_1, v_2, \dots \dots \dots, v_n\}$ and the set of edges, $E = \{e_0, e_1, e_2, \dots \dots \dots, e_m\}$. A walk, W of the graph G can be defined as an alternating sequence of vertices and edges,$W =< v_0, e_1, v_1, e_2 v_2, \dots \dots \dots, e_p, v_p >$such that e_i is the edge from the vertex v_{i-1} to v_i. The vertex v_0 is called as the ***origin*** and the v_p is called as the ***terminal*** of W. The walk, W joins the vertex v_0 to v_p and the walk is called as $v_0 - v_p$ walk. The walk which terminates at origin, i.e., $v_0 = v_p$ is termed as a ***closed*** walk otherwise termed as ***open***. When the edges of a walk are distinct, the walk is called as a ***trail*** and when the vertices are distinct then it is called a ***path***. A closed trail with distinct vertices is called as a ***circuit***. The number of edges present in a walk can be referred as the ***length*** of the walk.

Definition 2.5.2: A cycle which has n vertices and $n \geq 3$has a length of n. A graph having a cycle of length n is denoted as C_n. A cycle with length 3 i.e., C_3 is called as a ***triangle***, C_4 is termed as a ***square*** and C_5 as ***pentagon***.

Lemma 2.5.1: Every $u - v$ walk of a graph G contains a $u - v$ path.

Proof: We prove this lemma by the method of induction on length of the walk $u - v$. Let l be the length of the walk $u - v, W$.

Base case: $l = 0$.The walk contains a single vertex $u = v$ with no edge. Then there is obviously a $u - v$ path of length 0.

Space for learners:

Induction step: $l \geq 1$. Let us try to prove the lemma for walks of length less than l. If W is a $u - v$ walk with length l contains no repeated vertex then the walk is already a path. If not, then suppose there exist some vertex i in Wwhich occur more than once in the walk then removing the all the occurrences of i(and the corresponding edges) leaving one then we will get a walk W' of length less than l. By induction hypothesis there exists a $u - v$ path in W' and as W' is contained in W hence there is a $u - v$path in W.

Lemma 2.5.2: A closed walk of odd length contains a cycle.

Proof: Let W be a closed walk with odd length l. Using the method of induction, we can prove that W contains a cycle.

Base case: $l = 1$. If the length of the walk is 1 then there is a self- loop and W contains only a single vertex, hence there is a cycle.

Induction step: $l \geq 3$. Consider the walk, W consists of a vertex set $V = < v_0, v_1, v_2, \ldots \ldots, v_n >$ and $v_0 = v_n$. If each vertex$v_i (0 \leq i \leq n)$ is distinct, then the walk itself is a cycle. If not, then there exist two positive terms i, j such that $i < j$, $v_i, v_j \in V - \{v_0, v_n\}$and $v_i = v_j$. Now we can split this walk W into two closed walks W_1 and W_2 at v_i such that W_1 includes the vertices $v_i, v_{i+1}, v_{i+2}, \ldots \ldots, v_j$ $(v_i = v_j)$ and $v_0, v_1, .. v_i, v_{j+1} \ldots \ldots, v_n$. So, sum of the lengths of W_1 and W_2 will be equal to l. Since the length l is odd, one of these closed walks will be odd and by induction hypothesis, it has a cycle.

Example 2.5.1: The graphs in figure 2.4 illustrates various subtypes of walk. The subfigure (b) presents an open walk (3→2→1→3→4→1) of the graph shown in subfigure (a). In this walk, 3 is the origin and 2 is the terminal. This is called as an open walk as the origin and the terminal is not the same.

The walk shown in (c) (1→2→3→4→5→3→1) is a closed walk as the origin and the terminal vertices are the same which is 1. This cannot be called as a circuit or cycle as the vertex 3is repeated twice in the walk.

The walk shown in (d) is (3→2→1→3→4→5→6) represents a trail of the graph in (a). Here, the vertex 3 repeated twice but all the edges are distinct which satisfies the properties of a trail.

Space for learners:

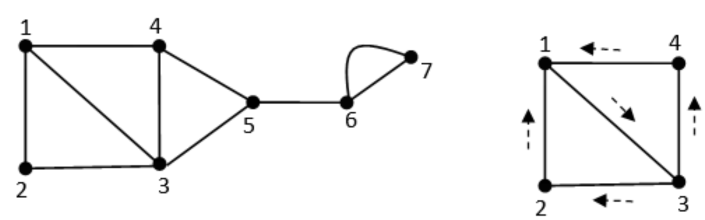
The subfigure (e) presents a path of graph in (a). All the vertices and edges along the path 3→2→1→4→5→6→7 are distinct which satisfies the properties of a path.

The example shown in (f) represent a circuit of the graph in (a). A circuit must have distinct edges and distinct vertices except for the starting and ending vertices. The walk 3→2→1→4→5→3 satisfies the properties of a circuit, so we can term this walk as a circuit.

2.6. CONNECTED AND DISCONNECTED GRAPHS

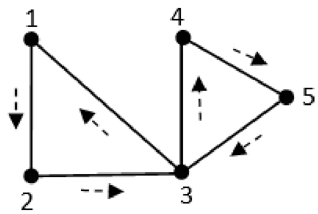
A graph is called as connected if each vertex of the graph is reachable from all other vertices. Otherwise, the graph is called as a disconnected graph. A disconnected graph contains more than one connected subgraph. Such subgraphs are called as components of a graph. Formal definition of connected graphs, disconnected graphs and components are given by Definition 2.6.1.

Definition 2.6.1: Consider a graph, $G(V, E)$. If there exist a $u - v$ path in G such that $u, v \in V$, then u is said to ***connected*** to v. The relation ***connected*** is an equivalence relation on the vertex set V of graph G. Suppose $V_1, V_2, \dots, V_k$ are equivalence classes of V. Then a subgraph with vertex set V_i, $1 \leq i \leq k$ is a ***component*** of G. If $k = 1$, then the graph G is ***connected graph*** and the graph G will be called as ***disconnected graph*** if $k \geq 2$. In simple words, a ***connected graph*** can have at most one component. In case of a ***connected graph***,Gthere will be a path $u - v$ for any pair of vertices $u, v \in V$.

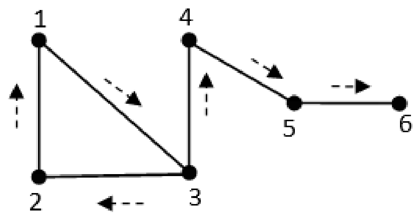


(a) A sample graph G (b) An open walk of G: 3→2→1→3→4→1

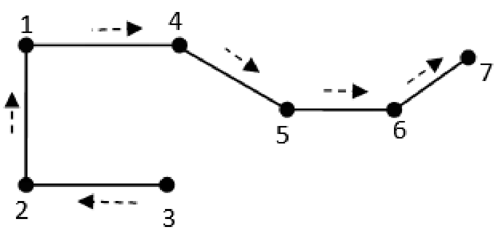
Space for learners:



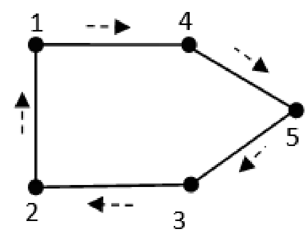
(c) A close walk of G:
1→2→3→4→5→3→1



(d) A trail of G: 3→2→1→3→4→5→6



(e) A path of G: 3→2→1→4→5→6→7



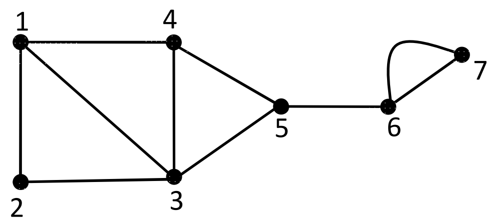
(f) A circuit of G:
3→2→1→4→5→3

Fig. 2.4: Example to illustrate walk, trail, path and circuit

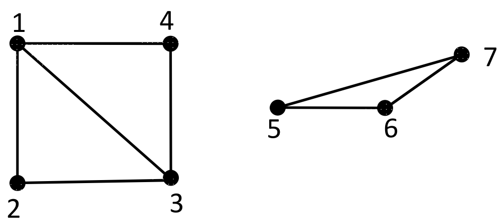
Definition 2.6.2: Let $G'(V', E')$ be a subgraph of a graph $G(V, E)$. The subgraph G' will be a ***maximally connected component*** of G if G' is connected and for any vertex v such that $v \in V$ and $v \notin V'$ there is no vertex $u \in V$ which is adjacent to v. Informally, if there exist no vertex in G which can be added to G' and G' still be connected.

Example 2.6.1: The figures shown in figure 2.4 illustrates the graph connectedness. The graph shown in (a) represents a connected a graph. This graph has 7 vertices and each vertex is reachable from all remaining 6 vertices. The graph in (b) also has 7 vertices, but the vertices 1-4 are not reachable from the vertices 5-7. Thus, this graph is disconnected. The subfigure (c) shows the components of the graph in (b). The components are enclosed within the rectangular boxes. One component has the vertex set {1,2,3,4} and the other has the vertex set {5,6,7}. Both the components are connected graphs individually.

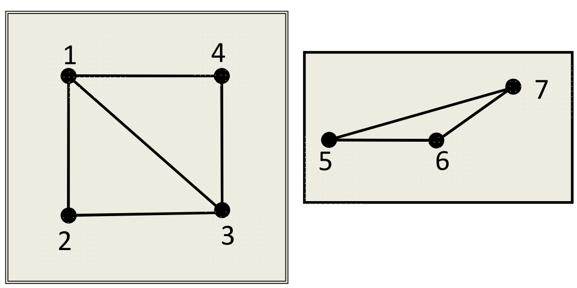
Space for learners:



(a) A connected graph



(b) A disconnected graph



(c) Components of the graph shown in (b)

Fig. 2.5: Sample graphs illustrating connected graph, disconnected graph and components of a graph

Theorem 2.6.1: A simple graph G with n vertices and minimum degree $\delta \geq \frac{n-1}{2}$ is connected.

Proof: We shall prove this theorem by contradiction. Suppose G is not connected and has at least two components, say G_1 and G_2. Let us consider v be any vertex of G_1. The degree of v, $d(v) \geq \frac{n-1}{2}$ as $\delta \geq \frac{n-1}{2}$. Hence, v has at least $\frac{n-1}{2}$ adjacent vertices in $G_1$and so, G_1contains at least $\frac{n-1}{2} + 1 = \frac{n+1}{2}$vetrices. Similarly, G_2 also contains minimum $\frac{n+1}{2}$ vertices. Hence, the graph, G has a minimum of $\frac{n+1}{2} + \frac{n+1}{2} = n + 1$ vertices, which is a contradiction.

Space for learners:

Theorem 2.6.2: If a simple graph G is not connected then $\overline{G}$ is connected.

Proof: Let $G(V,E)$ has more than one component. Let u,v be any two vertices of G (and of $\overline{G}$). If u,v belongs to two different components of G (u is not adjacent to v in G), then they are adjacent in $\overline{G}$. So, u and v are connected in $\overline{G}$. If u,v belongs to the same component of G then let us select a vertex w from a different component. The edges uw and vw do not belong to G but they belong to $\overline{G}$. Then there exists a path uwv in $\overline{G}$, which is nothing but a $u-v$ path. Hence $\overline{G}$ is connected.

Theorem 2.6.3: A graph with n vertices and k components can have at most $\frac{(n-k)(n-k+1)}{2}$ edges.

Proof: Let $G_1, G_2, \dots G_k$ be the components of a graph G and let n_i be the number of vertices of the i^{th} component of G such that $1 \le i \le k$ and $e(G_i)$ represents the number of edges present in G_i.

Any graph of n vertices can have at most $\frac{n(n-1)}{2}$ vertices (this happens when the graph is a complete graph which mean each vertex is connected with each other).

Thus, for any $G_i, 1 \le i \le k$, $e(G_i) \le \frac{n_i(n_i-1)}{2}$, and hence $e(G) \le \sum_{i=1}^{k} \frac{n_i(n_i-1)}{2}$.

Since each component has at most one vertex, for any G_i, $n_i > 1$ and

$n_i = n - (sum\ of\ the\ vertices\ in\ all\ the\ components\ of\ G\ except\ G_i)$

Hence, $n_i \le n-k+1$, so $\sum_{i=1}^{k} \frac{n_i(n_i-1)}{2} \le \sum_{i=1}^{k} \frac{(n-k+1)(n-k)}{2} = \frac{(n-k+1)(n-k)}{2} = \frac{(n-k)(n-k+1)}{2}$,

Hence proved, $e(G) \le \frac{(n-k)(n-k+1)}{2}$.

Theorem 2.6.3: A graph $G(V,E)$ is connected if and only if for any partition of vertex set V into subsets V_1 and V_2, there is an edge from any vertex of V_1 to any vertex of V_2.

Proof: Let a graph $G(V,E)$ is connected and let $V = V_1 \cup V_2$. Let us consider two vertices u,v such that $u \in V_1$ and $v \in V_2$. There exists a

Space for learners:

$u - v$ path in G, say$< u, w_0, w_2, \dots, w_k >$($w_k = v$) as G is connected. Let i be the smallest positive integer such that $w_i \in V_2$, then $w_{i-1} \in V_1$. Since w_{i-1} and w_i are adjacent, thus there is an edge from $w_{i-1} \in V_1$to $w_i \in V_2$. Conversely, let G is not connected. Thus, G has at least two components. Let V_1 represents the set of one component and V_2 represents the other component. It is obvious that there is no edge from any vertex of V_1 to any vertex of V_2. Hence it proves the theorem.

Space for learners:

CHECK YOUR PROGRESS

1. Let two groups G and H are two isomorphic graphs. Which of the following is true in terms of G and H?
   a. Number of vertices in G is same as the number of vertices in H
   b. Number of edges in G is same as the number of edges in H.
   c. Both in-degree and out-degree of a vertex $v$is same as the in-degree and out-degree of $\phi(v)$.
   d. All of the above

2. A subgraph containing all the vertices is called as
   a. Induced subgraph
   b. Spanning graph
   c. Clique
   d. None of the above

3. The subgraph in which all the vertices are adjacent to each other is called as
   a. Induced subgraph
   b. Spanning graph
   c. Clique
   d. None of the above

4. What will be the number of edges in a walk with n vertices?
   a. n-1
   b. n
   c. $n + 1$
   d. $2n$

5. Which of the following is true in terms of a walk?
   a. All the vertices must be distinct.

b. All the edges must be distinct.
c. Both (a) and (b)
d. None of the above.

6. A walk with same starting and ending vertex is called as a
   a. Open walk
   b. Closed walk
   c. Cycle
   d. Trail
7. A walk with no repeated vertex is called as a
   a. Open walk
   b. Closed walk
   c. Path
   d. Trail
8. A closed walk with distinct vertices is called as a
   a. Cycle
   b. Path
   c. Trail
   d. Clique
9. The maximum number of edges a graph with n vertices and k edges is
   a. $\frac{(n-k)(n-k+1)}{2}$
   b. $\frac{(n-k)(n-k-1)}{2}$
   c. $\frac{(n-k)n}{2}$
   d. $\frac{(n-k-1)(n-k+1)}{2}$
10. The maximum number of components that a connected graph with n vertice scan have is
   a. 0
   b. $n/2$
   c. n-1
   d. n

Space for learners:

2.7. SUMMING UP

Space for learners:

- ***Graph Isomorphism*** is a concept in graph theory which states that any two graphs, G and G' are called as *Isomorphic* if there is a bijection between the vertex sets of G and G'. Two isomorphic graphs have equal number of vertices and edges. The degree of a vertex v in G is same as the degree of its corresponding vertex in G'.
- A ***subgraph*** can be called as a subpart of another graph. An ***induced subgraph*** of a graph is another graph generated from a subset of the graph's vertices and all of the edges joining pairs of vertices in that subset. A ***spanning subgraph*** is a subgraph of another graph if the vertex set remains the same in both the subgraph and the original graph. A ***clique*** is a complete subgraph. of another graph.
- A ***walk*** is a finite alternating series of vertices and edges that starts and ends with vertices, with each edge connecting the vertices before and after it. A walk may have repeated vertices but not the edges of another graph. An open walk with no repeated edges is called as a ***trail***. The vertices may repeat in a trail. A trail with non-repeated vertices is called as a ***path***. A non-empty trail in which the starting and ending vertices are the only vertices that are repeated is called as a ***circuit*** or a ***cycle***.
- A graph is called as connected if each vertex of the graph is reachable from all other vertices. Otherwise, the graph is called as a disconnected graph. A disconnected graph contains more than one connected subgraph. Such subgraphs are called as components of a graph. A trail is a walk in which the starting and ending vertices are the only vertices that are repeated is called as a ***circuit*** or a ***cycle***.

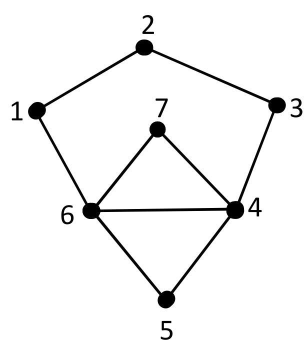
2.8. ANSWERS TO CHECK YOUR PROGRESS

1. d	2. b	3. c	4. a	5. b
6. b	7. c	8. a	9. a	10. a

2.9. POSSIBLE QUESTIONS

Space for learners:

1. What are the properties of graph isomorphism?
2. What is meant by subgraph? What are the different types of subgraphs available?
3. What is meant by an induced subgraph? Explain with an example.
4. What do mean by a walk of a graph? What is the difference between a trail and a path?
5. What is the difference between a closed walk and a cycle?
6. What is meant by components of a graph? How is it related to graph connectedness?
7. Find a trail, a cycle and a path in the graph given below.



8. Verify if the sequence given below can be considered as a trail. Justify your answer.
   $2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 2 \rightarrow 1$
9. Verify if the sequence given below can be considered as a path. Justify your answer.
   $2 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 1$
10. Verify if the sequence given below can be considered as a cycle. Justify your answer.
   $2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 2$

2.10. REFERENCES AND SUGGESTED READINGS

Space for learners:

- Graph Theory with Applications to Engineering and Computer Science by Narsingh Deo, published by Prentice Hall India Learning Private Limited.
- Introduction to Graph Theory by Richard J Trudeau, Published by Courier Corporation.
- A First Course in Graph Theory by Gary Chartrand and Ping Zhang, Published by Courier Corporation.
- Graph theory with applications by John Adrian Bondy, Published by Elsevier Publishing Company

UNIT 3: PATHS AND CIRCUITS-II

Space for learners:

Unit Structure:

3.1 Introduction

3.2 Unit Objectives

3.3 Euler Graphs

3.3.1 Definitions

3.3.2 Theorems on Euler Graph

3.3.3 Arbitrarily Traceable Graphs

3.4 Hamiltonian Graphs

3.4.1 Definitions

3.4.2 Theorems on Hamiltonian Circuits

3.5 Bipartite Graphs

3.5.1 Properties of Bipartite Graph

3.5.2 Matching in Bipartite Graph

3.6 Summing Up

3.7 Answers to Check Your Progress

3.8 Possible Questions

3.9 References and Suggested Readings

3.1 INTRODUCTION

This unit focuses on three very important concepts of graph theory- Euler graph, Hamiltonian graph, and Bipartite graph. Almost every real-world problem involving discrete groupings of items, where the focus is on the relationship between them rather than the intrinsic features of the items, may be translated into one of these graphs. Thus, these graphs find a wide range of applications such as in computer science, communication science, economics, computer graphics, electronic circuit design, mapping genomes, operation research, error correction code etc. The goal of this unit is to familiarize the students with the various terms and definitions related to these graphs and to introduce some important theorems.

3.2 UNIT OBJECTIVES

After going through the unit, you will be able to-

- define the terms Euler graph, Hamiltonian graph, and Bipartite graph.
- determine whether a graph is an Euler graph or not.
- determine if a graph is a Hamiltonian graph.
- list the properties of the Bipartite graph.
- Determine whether there exists a perfect match in a graph or not.

3.3 EULER GRAPHS

In 1736, Swiss mathematician Leonhard Euler in his famous paper, where he solved the Königsberg bridge problem, raised an interesting problem. The problem was, given a graph G, is it possible to find a walk, with the same staring and the end vertices and includes each edge of G exactly once. In the same paper, he also presented the solution of the problem and introduced the concept of the Euler graph which now is extensively used in many fields- ranging from DNA sequence reconstruction to circuit designs.

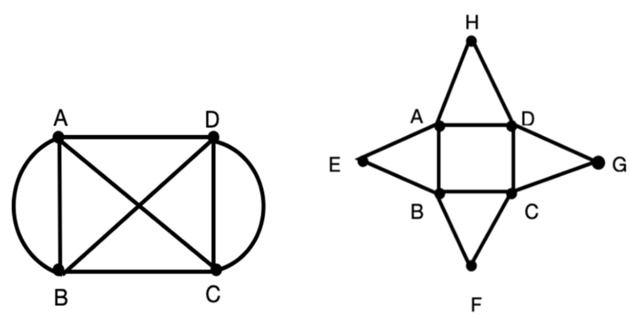
Space for learners:

3.3.1 Definitions

Space for learners:

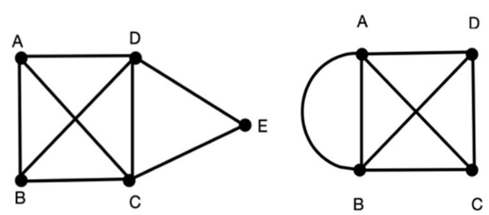
- **Euler line:** in a given graph *G*, if there exists a closed walk (having the same starting and end vertex) such that it contains each edge of *G* exactly once, then the walk is called an Euler line.
- **Euler Graph:** A graph that contains an Euler line is called an Euler Graph.
- **Unicursal Line:** A walk that contains all the edges of the graph exactly once, with different starting and end vertices, is called a unicursal line. A unicursal line is an Euler line with the dropped constraint of the walk being closed. Hence, it is also referred to as the open Euler line.
- **Unicursal Graph:** A graph that contains a unicursal line is called the Unicursal Graph.

Example: The graphs in figure 3.1 are examples of Euler graphs. The graph in figure 3.1(a) consists of four vertices and eight edges. If we start from vertex A, the walk (A, B, C, D, B, A, C, D, A) contains all the edges of the graph. Similarly, the graph in figure 3.1 (b) is also an Euler graph and one possible Euler line is-(A, B, C, D, A, H, D, G, C, F, B, E, A) starting and ending with vertex A.



(a) (b)

Fig. 3.1: Examples of Euler Graph

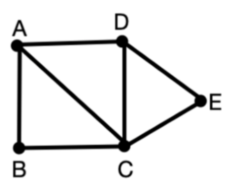


(a) (b)

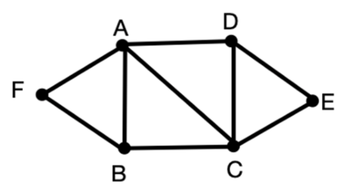
Fig. 3.2: Examples of non-Euler Graph

However, the graphs in figure 3.2 are not Euler graphs. It is not possible to get any Euler line starting from any vertex of these graphs.

The graphs in figure 3.3 are also not Euler graphs. But they are unicursal graphs. For the graph in figure 3.3 (a), one possible unicursal line is (A, B, C, A, D, C, E, D, C). Similarly, (B, F, A, B, C, A, D, C, E, D) is a possible unicursal line for the graph in figure 3.3 (b).



(a)



(b)

Fig 3.3: Example of Unicursal Graph and Unicursal Line

3.3.2 Theorems on Euler Graph

Theorem 3.1: A connected graph G is an Euler graph if and only if each vertex in G is of even degree.

Proof:

Part A- Necessary:

Assume that G is a Euler graph. By the definition of the Euler graph, there must exist a Euler line in G, which is a closed walk containing each edge of G exactly once. This implies that- every time a vertex v is encountered while tracing the Euler line, there must be

1. A new edge incident on v which serves as the entry edge and
2. Another new edge incident out of v which serves as the exit edge.

This holds true for each intermediate vertex v indicating that the intermediate vertices must be of even degree. As an Euler line is a closed walk, there exists only one terminal vertex which is both the starting and the end vertex. Thus, the walk starts from the terminal vertex and came back to the same vertex at the end. This indicates that the terminal vertex must also be of even degree.

Space for learners:

Thus, we may conclude that if a graph G is an Euler graph then each vertex must be of even degree.

Part B- Sufficiency:

Assume that each vertex of G is of even degree. Let's start with any vertex v of $G$and then arbitrarily trace a walk in such a way that an edge is traversed only once. As each vertex of G is of even degree, we can exit from any vertex u that we enter in this walk. The walk may come to end only when we eventually reach v. Let this closed walk be termed as h. If h contains all the edges of G then it's an Euler line, otherwise, we will remove h from G, which results in a subgraph G'. As the vertices in both h and G are of even degree, the vertices in G' must also be of even degree. As the graph G was connected, G'must have at least one common vertex with G. Let this common vertex be w. Now, starting from w, again we trace another arbitrary walk h'containing an edge of G' only once. As h' also has vertices only of even degree, this walk may also come to an end only on encountering w. Now we remove h' from G' and join with h, which results in a new walk that starts and ends with v but with a greater number of edges. We can apply this process recursively until we obtain a closed walk that contains all the edges of G. Thus, G is a Euler graph.

Example: If we consider the graphs in figure 3.1(a), the graph has four vertices (A, B, C, D) and eight edges. All the vertices of the graph are of degree 4, which is even. Thus, the graph is an Euler graph.

Similarly, the graph in figure 3.1(b) has eight vertices (A, B, C, D, E, F, G, H). The vertices A, B, C, and D are of degree 4; and the vertices E, E, G, and H are of degree 2. Thus, all the vertices in this graph are also of even degree. Thus, the graph is a Euler graph.

Now, if we consider the graphs in figure 3.2, there exists at least one vertex in each graph which is of odd degree. For example, in the graph of figure 3.2 (a), the degree of vertices A and B is 3. Similarly, in graph 3.2 (b), the vertices C and D are of degree 3. Thus, the graphs are not Euler graphs.

Königsberg bridge problem: The famous Königsberg bridge problem stated that whether it is possible to cross the seven bridges, connecting the two islands of the city Königsberg, exactly once in a single traversal. The additional requirement of the problem was that the traversal must end at the same point from where it started. The

Space for learners:

problem may be represented graphically as in figure 3.4(a). An equivalent representation of the same problem in terms of a graph is given in figure 3.4(b).

As it can be seen from figure 3.4 (b) that the vertices of the graph are not of even degree. Hence, it is not an Euler graph and a walk that starts and ends at the same point, by crossing each edge of the graph exactly once, is not possible.

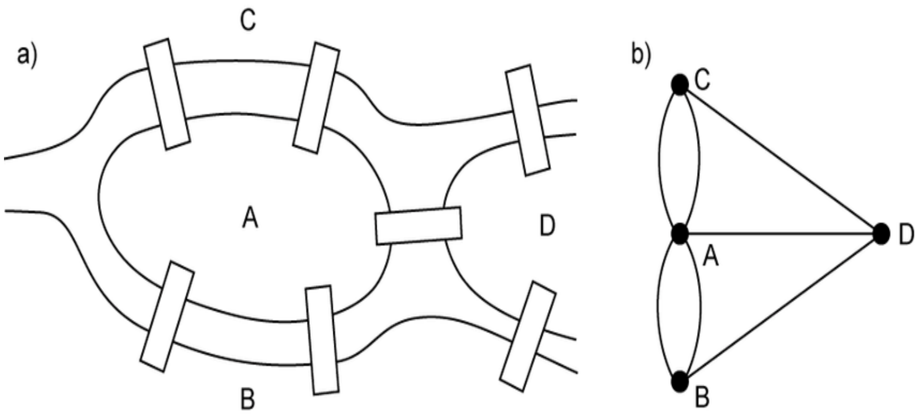


Fig. 3.4: Graphical representation of Königsberg bridge problem

Theorem 3.2: There exist exactly k edge-disjoint subgraphs, in a graph G with exactly $2k$ odd degree vertices, such that each subgraph is a unicursal graph and all the subgraphs together include all the edges of G.

Proof: Let the odd degree vertices in G be ($v_1, v_2, v_3, \ldots\ldots, v_k; u_1, u_2, u_3, \ldots\ldots, u_k$). Now, let's add k edges ($e_1, e_2, e_3, \ldots\ldots e_k$) in between a pair of vertices (v_1, u_1), (v_2, u_2), (v_3, u_3)….. and (v_k, u_k). This results in a new graph G' where each vertex is of even degree. Thus, G' is a Euler graph.

Let, $\boldsymbol{\rho}$ be a Euler line in G'. If we now deleting e_1 from $\boldsymbol{\rho}$ will result in a unicursal line $\boldsymbol{\gamma}$. Deleting e_2 from that from $\boldsymbol{\gamma}$will split it into two unicursal lines $\boldsymbol{\gamma}1$ and $\boldsymbol{\gamma}$'; removal of e_3 from whichever unicursal line that it belongs to, will split that line again into two more unicursal lines resulting in a total 3 unicursal lines. Continuing this process, until we delete all other remaining extra k-3 edges, i.e., $e_4, e_5, \ldots\ldots e_k$ will finally result in k unicursal lines. As we removed only the extra edges that we had added to G, all the unicursal lines together will still contain all the original edges of G.

Space for learners:

Space for learners:

Theorem 3.3: For a connected graph to be an Euler Graph, if and only if it contains edge-disjoint circuits.

Proof:

If Part:

Let's G is a connected graph containing circuits. All the circuits in G are edge-disjoint and thus G can be decomposed into circuits. As in a circuit, all the vertices are of degree 2, it can be concluded that all the vertices in G have even degree. Hence, G is a Euler graph.

Only if path:

Let G is a Euler graph. Now let's randomly take any vertex v_1 from G, it must be involved with at least two edges as it is of even degree. Let one of these edges be (v_1, v_2) incident on the vertex v_2. Due to the same reason, v_2 must also be part of at least two edges. Let (v_2, v_3) be an edge connecting the vertices v_2 and v_3. If we continue the process, it will end only when we reach the starting vertex v_1 resulting in a circuit C. Now, removing C from G will result in a subgraph G' where all the vertices are of even degree. Thus, we can repeat the same process in G' and remove another circuit from it. This we can continue until we get a Null graph.

3.3.3 Arbitrarily Traceable Graphs

In a Euler graph, starting from any vertex v, if we start tracing the edges in such a way that no edge is repeated, it may not always result in a Euler line. For example, consider the graph in figure 3.5. If we now start from vertex A and start tracing the edges in the sequence (A, C, B, A); we will get back to the vertex A, after which we don't have any option to visit a new edge. However, the sequence (A, B, C, A) is a circuit, not a Euler line as it does not cover all the edges of the graph. On the contrary, if we choose the starting vertex as C, and take a walk by visiting a new edge every time, we are guaranteed to get a Euler line, does not matter in what sequence we visit the edges. One such sequence is (C, E, D, C, B, A, C). From this example, it is clear that in a Euler graph, starting from any random vertex $v,$ if we take an arbitrary walk by simply visiting a new edge every time, we may not get a Euler line.

For any vertex *u* in an Euler Graph *G*, if it is always possible to start from that vertex and then take an arbitrary walk by randomly selecting a new edge every time, and get back to *u* by traversing all the edges in *G*, then *G* is said to be ***arbitrarily traceable*** with respect to *u*.

Theorem 3.4: A Euler graph *G* is arbitrarily traceable with respect to a vertex *v*, if and only if *v* is a part of every circuit in *G*.

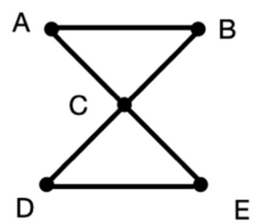


Fig. 3.5: Arbitrarily Traceable Graph with respect to Vertex 'C'

Proof: Let the Eulerian graph *G* can be traced arbitrarily from a vertex *v*. Assume that circuit *C* does not pass through *v*. Let *H* be a subgraph of *G* that does not contain the edges of *C*. As *G* is Euler graph all its vertices are of even degree and *C* being a circuit all its vertices are also of degree 2. Therefore, all the vertices in *H* also have an even degree meaning that it's an Euler graph. So, in *H* if we start from *v*, then it is possible to traverse all the edges of *H* exactly once and then come back to *v*. Now, according to our initial assumption, as *C* does not contain *v*, this walk cannot be extended to contain the edges of *C*.

Example: The graph in figure 3.5 contains two circuits C1 (A, B, C, A) and C2 (C, D, E, C). As we can see that C is the only vertex that is common in both the circuits, the graph is arbitrarily traceable with respect to C only. For the other vertices namely- A, B, D, and E we may not always get a Euler line by randomly walking through a new edge every time.

Space for learners:

CHECK YOUR PROGRESS-I

1. **Fill in the blanks:**
   a. A connected graph is a Euler graph if and only if all its vertices are of ______ degree.
   b. In a graph G, with 6 odd degree vertices, there exist at least _____ subgraphs such that each subgraph is a unicursal graph and all the subgraphs together include all the edges of G.
2. **State true or false:**
   a. A Euler graph is arbitrarily traceable with respect to any vertex in the graph.
   b. In a graph G, the sum of the degrees of vertices is18. G is an Euler graph.
   c. A graph where G all the vertices are of degree 6. The graph is a Euler graph.

Space for learners:

3.4 HAMILTONIAN GRAPHS

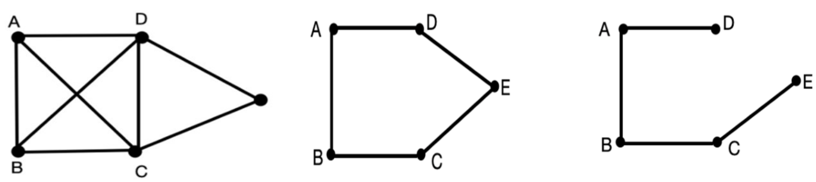
Sir William Hamilton, an Irish mathematician (1805-1865), created the Icosian game, where a dodecahedron was used with each of the 20 vertices labelled with the name of a different capital city across the world. The objective of the game was to create a closed walk across all the cities using the edges of the dodecahedron that visited each city precisely once, beginning and finishing in the same city. The term- "Hamiltonian Graph" originated from this problem and became one of the most important and interesting concepts in Graph Theory.

3.4.1 Definitions

- **Hamiltonian Circuits**: In a graph G, if there exists a circuit that passes through all the vertices of G exactly once, then the circuit is called a Hamiltonian Circuit. If G contains n vertices, then a Hamiltonian circuit of G will always contain exactly n edges.
- **Hamiltonian Graph:** A graph that possesses a Hamiltonian circuit is called a Hamiltonian graph.

- **Hamiltonian Paths:** If there exists a path in a graph *G*, such that it starts with a vertex *v* and ends with vertex *u*; containing all the vertices of *G* exactly once, then that path is called Hamiltonian path. Dropping an edge from a Hamiltonian circuit result in a Hamiltonian path. In a Hamiltonian graph *G*, each Hamiltonian path contains exactly *n-1* edges.

Example: Consider the graph in figure 3.6(a). It contains five vertices and 8 edges. The graph is Hamiltonian and a possible Hamiltonian circuit is shown in figure 3.6 (b). The figure in 3.6(c) presents a possible Hamiltonian path for the graph.



(a) (b) (c)

Fig. 3.6: Examples of Hamiltonian graph, Hamiltonian circuit and Hamiltonian path

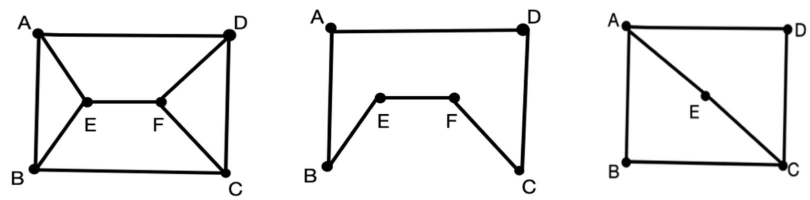
We may observe that the Hamiltonian Circuit contains all the 5 vertices of the original graph and has exactly 5 edges. The Hamiltonian path on the other hand contains exactly 4 edges and 5 vertices.

3.4.2 Theorems on Hamiltonian Circuits

Theorem 3.5 (Dirac's Theorem): In a simple graph *G*, with *n* vertices ($n \geq 3$), if the degree of each vertex is greater than or equal to $n/2$, then *G* is a Hamiltonian graph.

Example: The graph in figure 3.7(a) has 6 vertices. Each vertex in the graph has degree $3 \geq (6/2)$. Thus, the graph is Hamiltonian. Figure 3.7(b) presents a Hamiltonian circuit for the same. Now, if we consider the graph in figure 3.7 (c) it has 5 vertices. The degree of the vertex E is 2 which is less than 5/2. Thus, this graph is not Hamiltonian.

Space for learners:



(a) (b) (c)

Fig. 3.7: Example of Hamiltonian graph and non-Hamiltonian Graph

Theorem 3.6 (Oreo's theorem): If in a simple Graph G with n vertices, where $n \geq 2$, for each pair of non-adjacent vertices u and v, degree(u)+degree(v)$\geq n$, then the graph G is a Hamiltonian graph.

Example: Consider the graph in figure 3.6 (a). The pairs of non-adjacent vertices in this graph are- (A, C), (A, F), (B, F), (B, D), (C, E), and (D, E). All the pair of vertices has a sum of degrees equal to 6≥6. Thus, the graph is Hamiltonian.

For the graph in figure 3.7(c), (E, D) is a non-adjacent pair of vertices. The sump of degrees of the vertices E and D is 4 which is less than 5. Thus, we can claim that the graph is not a Hamiltonian graph.

CHECK YOUR PROGRESS-II

3. **Fill in the blanks:**
   a. A Hamiltonian path traverse each vertex of the graph exactly______.
   b. A Hamiltonian path for a Hamiltonian graph with 6 vertices has exactly _____ edges.
4. **State true or false:**
   a. A graph G has 6 vertices with degrees 2, 2, 4, 1, 3, 3 and 3. The graph is a Hamiltonian graph.
   b. A Hamiltonian circuit contains all the edges of the graph.

3.5 BIPARTITE GRAPHS

In graph theory, a graph G = (V, E) is said to be a bipartite graph if, the set of vertices V can be divided into two disjoint sets V1 and V2 such that each edge e belonging to E, connects a pair of vertices (u,

Space for learners:

v) such that $u \in V1$ and $v \in V2$. In other words, there does not exist any edge in *G* that connects vertices of the same set.

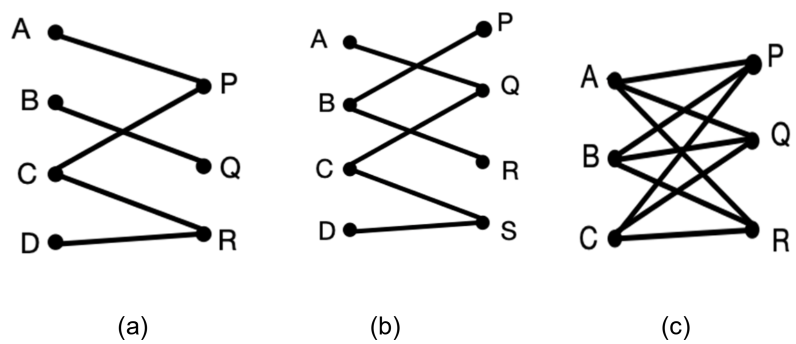


Fig 3.8: Examples of (a) Bipartite graph, (b) Balanced Bipartite graph and (c) Complete Bipartite Graph

Example: The graphs in figure 3.8 are bipartite. In all the graphs the set of vertices can be divided into two disjoint sets and none of the edges connects vertices from the same set. In graph 3.8(a), the bipartition of the vertex set is V1 = {A, B, C, D} and V2 = {P, Q, R}. As we may see that there no edge connecting two vertices from V1 or V2. For the graph in figure 3.8(b), the two disjoint sets of vertices are V1 = {A, B, C, D} and V2 = {P, Q, R, S}. In the last graph in figure 3.8(c) the bipartition of the vertices is V1 = {A, B, C} and V2 = {P, Q, R}.

Following are some terms related to bipartite graph-

- **Balanced Bipartite Graph:** If the two sets V1 and V2 have the same number of vertices then, the graph *G* is called a balanced bipartite graph.
- **Complete Bipartite Graph:** A bipartite graph, *G* is referred to as a complete bipartite graph if each vertex in one set is connected to every vertex in the other set. In other words, for each vertex belong to V1, there exists an edge to each *v* belonging to V2 and vice versa. A complete bipartite graph is denoted by $K_{m,\, n}$ where m and n are the cardinalities (number of vertices) of set V1 and V2 respectively.

Example: The graph in figure 3.8(b) is a balanced bipartite graph as the bipartition of the graph V1 and V2 contains an equal number of vertices. The graph in figure 3.8(c) is an example of a complete

Space for learners:

bipartite graph as each vertex in V1 is connected to all the vertices in V2.

Space for learners:

3.5.1 Properties of Bipartite Graph

Lemma 3.1: In a bipartite graph *G* with the vertex petitions sets V1 and V2, $\sum_{u \in V1} deg(u) = \sum_{v \in V2} deg(v)$

Proof: Let's *G'* be a subgraph of *G* that contains only the vertices of *G* and doesn't contain contains any edge. Hence, initially for this subgraph, the degree of each vertex is zero. As *G* is a bipartite graph, each edge connects a vertex in V1 and to a vertex in V2. Thus, if we now add an edge of *G* to *G'*, say between the vertex $u \in$ V1 and $v \in$ V2, this will increase the sums of degrees of the vertices in both sets to 1. Thus, $\sum_{u \in V1} deg(u) = \sum_{v \in V2} deg(v) = 1$ after adding a single edge. Adding the second edge will connect another vertex from V1 to a vertex in V2. This will further increase the sums of the degrees of the vertices by 1. Thus, after adding the second edge, $\sum_{u \in V1} deg(u) = \sum_{v \in V2} deg(v) = 2$. Since, every edge contributes exactly one to the sum of the degrees of vertices in each side, continuing the process until we add all the edges of *G* will still maintain the equality.

Example: For the graph in figure 3.8(a), the bipartition of the vertices are, V1 = {A, B, C, D} and V2 = {P, Q, R}. The sum of degrees of vertices in

V1=deg(A)+deg(B)+deg(C)+deg(D)=1+1+2+1=5

The sum of degrees of vertices in V2 = deg(P)+deg(Q)+deg(R) =2+1+2 =5. Thus, we may see that the sum of degrees of vertices in both the sets is the same. We can establish the same for the other two graphs in figure 3.8.

Theorem 3.7: If *G* is a *k*-regular (*k*>0) bipartite graph with bipartition V1 and V2, then |V1|=|V2|, i.e., number of vertices in V1 must be equal to number of vertices in V2.

Proof: A graph is *k*-regular if all the vertices in the graph are of degree *k*. As, *G* is a *k*-regular bipartite graph, all the vertices in *G* are of equal degree, i.e., *k*. Thus,

$$\sum_{u \in V1} deg(u) = k|V1| \text{ and } \sum_{v \in V2} deg(v) = k|V2|.$$

Form the Lemma 3.1-

$$\sum_{u \in V1} deg(u) = \sum_{v \in V2} deg(v).$$

This implies that,

$$k|V1| = k|V2| \Rightarrow |V1|=|V2|$$

Example: The bipartite graph in figure 3.8(c) is a 3-regular graph, as all the vertices in that graph are of degree 3. The bipartition of the graph is V1 = {A, B, C} and V2 = {P, Q, R}. As we may see that both the partitions have an equal number of vertices, i.e., 3.

Theorem 3.8: All the circuits in a bipartite graph are of even length.

Proof: Let *G* be a bipartitely graph with set of vertices partitioned into V1 and V2. Let, $C = (u, v_1, v_2 \ldots . v_{2k}, u)$ be a circuit in *G* with an odd length $2k+1$. Let the vertex $u \in$ V1. As *G* is bipartite, *u* must be connected to a vertex in V2.

Now, starting from *u*, following the sequence in *C*, the first edge connects *u* to v_1. Thusv_1 must belong to V2. The second edge connects vertex v_1 with v_2. Using the same argument, we can say that $v_2 \in$ V1. If we continue following the sequence, the $2k^{th}$ edge in *C* connects the vertex $v_{2k-1} \in$ V2 and $v_{2k} \in$ V1. The final edge $2k+1$ connects $v_{2k} \in$ V1 to $u \in$ V2, which contradicts our initial assumption that $u \in$ V1. Thus, we can conclude that all the circuits in a bipartite graph must be of even length.

Theorem 3.9: Every subgraph of a bipartite graph is a bipartite graph,

Proof: Let *G* be a bipartite graph with the set of vertices partitioned into V1 and V2. Let *G'* be a valid subgraph of *G*. Then let V1'=V1 ∩ *G'* and V2'=V2 ∩ *G'*. If (V1', V2') is an invalid bipartition of *G'* then there must exist an edge that connects the vertices *u* and *v* such that $u, v \in$ V1' or $u,v \in$ V2'. However, as *G'* is a valid subgraph of *G* it must not contain any edge that is not in *G*. Thus, the edge (*u*, *v*) is not a valid edge which implies that V1' and V2' is a valid bipartition. So, G' is also a bipartite graph.

Space for learners:

Theorem 3.10: A bipartite graph with at least one edge is *2*-colourable.

Proof: Let *G* be a bipartite graph with a set of vertices partitioned into V1 and V2. None of the vertices in V1 are adjacent to each other. Thus, they can be coloured with 1-colour, say colour-1. The same is true for the vertices in set V2 and all the vertices in set V2 are coloured with the same colour, colour-2.

Now, we will have to prove that colour-1 and colour -2 cannot be the same. As there exist a positive number of edges in the graph, there is at least one edge in the bipartite graph that connects a vertex *u* in set V1 to a vertex *v* in V2. This implies that *u* and *v* are adjacent to each other and thus they cannot be coloured with the same colour. Thus colour -1 and colour-2 must be different.

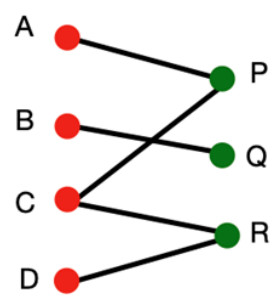


Fig. 3.9: Example of Colouring a bipartite graph

Example: Consider the graph in figure 3.8(a). It can be coloured using *2*-colours as shown in figure 3.9. In this case, we have used red and green colours. As we may observe that none of the adjacent vertices are coloured with the same colour. The red colour has been used for the vertices A, B, C, and D, which are not adjacent to each other. The green colour has been used for the vertices P, Q, and R; none of which are adjacent to each other. Thus, with two colours, we can properly colour the graph. The same can be shown for the other two graphs in figure 3.8.

Space for learners:

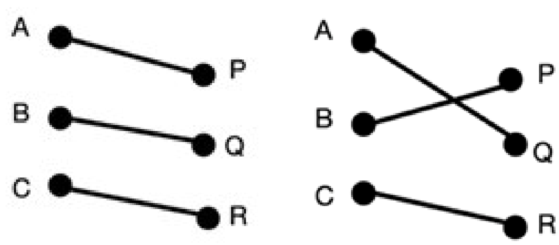
CHECK YOUR PROGRESS-III

5. **Fill in the blanks:**
   a. A bipartite graph can be coloured using ___ colours.
   b. In a bipartite graph, the sum of degrees of vertices in one set is 8. The same of the other set is____.
6. **State true or false:**
   a. A 6- regular bipartite graph contains equal number of vertices in both the set of bipartitions.
   b. The circuits of a bipartite graph can be of odd as well as of even length.

Space for learners:

3.5.2 Matching in Bipartite Graph

In a graph *G*, a matching *M* is a subgraph, with a set of edges such that no two edges have a common vertex. Thus, in a matching each vertex has degree exactly 1.



(a) **(b)**

Fig. 3.10: Example of Matching

A matching *M* is said to be maximal if it contains the largest possible number of edges from *G*. A perfect matching is a maximal matching that contains all the vertices of *G*.

The graph in figure 3.10(a) is a matching for graph 3.8(a). It is also the maximal matching for the graph. However, it is not a perfect matching as it does not include the vertex D. On the other hand, the graph in figure 3.10(b) is an example of matching for the graph in figure 3.8(c). It is a perfect matching as it includes all the vertices of the graph.

Lemma 3.2: In bipartite graph *G*, with bipartition V1 and V2, there does not exist a perfect match if $|V1| \neq |V2|$

Proof: Suppose, there exists a perfect matching M for G. Now, let's construct a subgraph G' which contains all the vertices of G and the edges of M. According to theorem 3.9, G' is also a bipartite graph. Since G' contains the edges of M, all the vertices in G' are of degree 1. Thus, G' is a 1-regular bipartite graph, and applying the theorem 3.7, we can conclude that $|V1|=|V2|$.

Example: It is not possible to have a perfect match for the graph in figure 3.8(a), as the bipartition does not contain an equal number of vertices. On the contrary, if we examine the graph in figure 3.8(c), it contains an equal number of vertices in both sets. Thus, a perfect match is possible in this graph. The graph in figure 3.10(b) is an example of a perfect match for this graph.

3.6 SUMMING UP

- In this module we have discussed - Euler graph, Hamiltonian graph, and Bipartite graph.
- A Euler line is a closed walk that contains all the edges of the graph exactly once. A graph containing a Euler line is an Euler graph.
- A graph is a Euler graph if and only if all the vertices are of even degree.
- A unicursal line is an open Euler line.
- A connected graph is a Euler graph if and only f it contains edge-disjoint circuits.
- A Euler graph is arbitrarily traceable with respect to a vertex v, if v is a part of every circuit in the graph.
- A Hamiltonian circuit contains all the vertices of a graph exactly once. A graph containing a Hamiltonian circuit is called a Hamiltonian graph.
- Using Dirac's theorem and Oreo's theorem we can check if a graph is Hamiltonian or not.
- A bipartite graph, G, is a graph, where the vertices can be partitioned into two disjoint sets such that no edge of G connects two vertices from the same set.
- The subgraph of a bipartite graph is also a bipartite graph.

Space for learners:

- A bipartite graph is *2*-colourable.
- In a bipartite graph, a perfect match exists if both set of vertices have equal cardinality.

Space for learners:

3.7 ANSWERS TO CHECK YOUR PROGRESS

1.

a. Even

b. 3

2.

a. False

b. False.

c. True.

3.

a. once

b. 5

4.

a. False.

b. False.

5.

a. 2

b. 8

6.

a. True

b. False

3.8 POSSIBLE QUESTIONS

1. Short Answer Type Questions:

a. Define Hamiltonian graph. List some of its applications.

b. Define Euler graph and list some its applications.

c. For a graph to be arbitrarily traceable with respect to a vertex v, what constraint v must satisfy?

d. Define unicursal line. Why is it also called an open Euler line?

e. What is a complete bipartite graph. Give an example.

2. **Long Answer Type Questions:**

    a. State the Königsberg bridge problem and illustrate Euler's solution to this problem.

    b. What is a matching? Explain with an example the concept of perfect matching. Prove that perfect matching is not possible in a bipartite graph having different number of vertices in the bipartition.

    c. Discuss the Dirac's theorem and Oreo's theorem for Hamiltonian gram with the help of examples.

    d. State some applications of Bipartite graph. Prove that a bipartite graph is 2-colourable.

    e. Prove that a subgraph of a bipartite graph is also a bipartite graph.

Space for learners:

3.9 REFERENCES AND SUGGESTED READINGS

- M.E Van Valkenburg, "Network Analysis", Prentice Hall, 2006.
- Abhijit Chakrabarti, "Circuit Theory (Analysis and Synthesis)", Dhanpat Rai & Co, 7th Edition, 2018.
- C. K Alexander and M.N.O Sadiku, "Electric Circuits", McGraw Hill Education, 2004

UNIT 4: TREES

Space for learners:

Unit Structure:

4.1 Introduction

4.2 Unit Objectives

4.3 Properties of Trees

4.4 Distance and Center of Trees

4.5 Rooted and Binary Trees

4.6 Counting Binary Trees

4.7 Fundamental Circuit

4.8 Spanning Trees in Weighted Graphs

4.9 Cut Sets

4.10 Summing Up

4.11 Answers to Check Your Progress

4.12 Possible Questions

4.13 References and Suggested Readings

4.1 INTRODUCTION

A tree is a nonlinear discrete data structure. This unit gives an overview of the tree and its properties. The types of trees such as rooted and binary trees are also discussed in this unit. A binary tree has a maximum of two leaf nodes. The counting tree with its properties is also reported in the unit. The concept of a circuit along with the minimum spanning tree is also discussed in the unit. A minimum spanning tree contains the minimum weight of the graph. The graph cut set and the weighted graph are also discussed in this unit.

4.2 UNIT OBJECTIVES

After going through this unit, you will be able to know

- About trees and their properties.
- About the rooted tree, counting tree, and binary trees.
- About the circuit and weighted graph.
- About the spanning tree.

4.3 TREES AND THEIR PROPERTIES

The tree is a discrete nonlinear structure that represents hierarchical relationships between nodes. It is a connected and acyclic undirected graph. There is a path between the nodes of a tree. A tree with n nodes contains (n-1) numbers of edges. Every node has a degree. The node which has o degree, known as the root of the tree. The node with degrees 1 and 2 is known as the leaf and internal node of the tree.

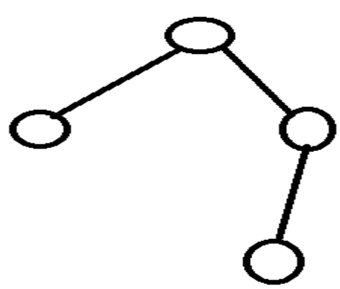


Fig. 4.1: Example of a tree

Space for learners:

The properties of a tree are explained below:

i) A tree is a nonlinear data structure.

ii) Every tree has a root node with degree 2.

iii) The degree of a leaf node is 1.

iv) The degree of an internal node is at most 2.

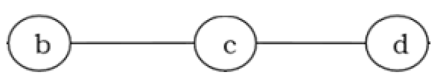
v) Every tree has n-1 numbers of edges.

4.4 DISTANCE AND CENTER OF TREES

The vertex with the minimal eccentricity of a tree is known as the center of the tree. The eccentricity of a vertex is the maximum distance from that respective vertex to other vertexes in the tree and it is the diameter of the tree. Some trees may contain only one center and this type of tree is known as a central tree. Some trees may contain more than one tree and this type of tree is known as a bi-central tree.

To understand the Center of a tree, let's consider the following tree.

In the above tree, five nodes are present. Initially, a node with degree 1 and its adjacent edges should remove from the tree to find the center of the tree. So, you remove node a and e as both nodes have degree 0. After removing a and e along with its adjacent edges, the resultant graph will be as follows.

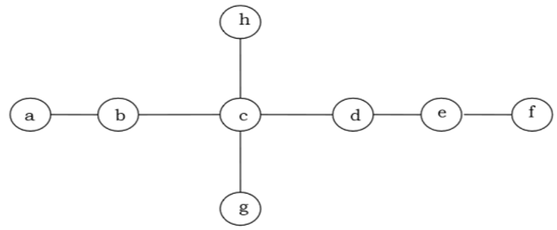


Then apply the same procedure on the graph and remove b and d from the graph, Final the graph contains only one vertex and that is C. So, it is a central tree.

Space for learners:

CHECK YOUR PROGRESS - I

1. What is degree of a tree?
2. What is the degree of the root node?
3. How many edges are there in a tree of n nodes?
4. The following tree is central (True or False).



Space for learners:

4.5 ROOTED AND BINARY TREE

A rooted tree is a connected and acyclic graph. The tree has a special node known as the root of the tree and each node is directly or indirectly connected to the root of the tree where children of the internal nodes are ordered. The internal node of the rooted tree may have fewer or exactly m children. The rooted tree in which m =2, is known as the binary tree.

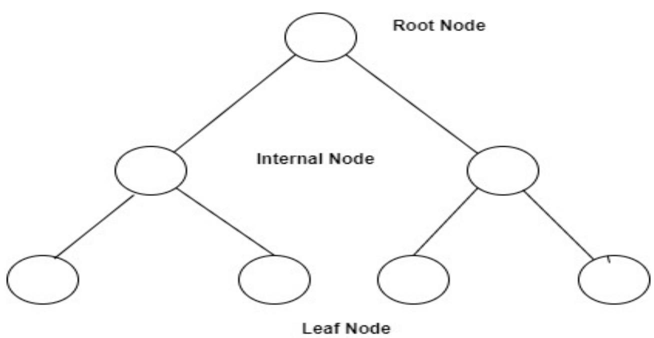


Fig. 4.2: Rooted tree

The Binary tree is that rooted that has a maximum of two children. It means that each node can have either 0, 1, or 2 children.

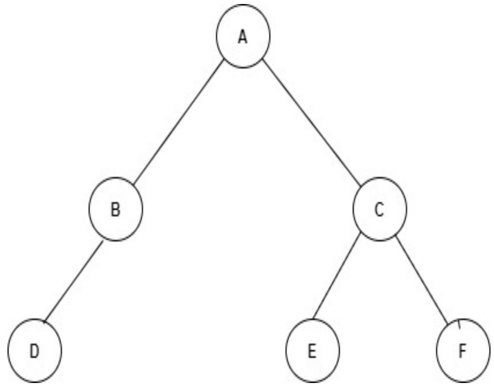


Fig. 4.3: Rooted tree

In Fig. 4.3, the root node is A and it has a maximum of 2 children, i.e., B and C. Node B has only one child and node C have two children. The leaf nodes D, E, and F have no children. So, it is a binary tree.

The properties of the binary tree are presented below.

i) The maximum number of nodes in a level "i" is $2^{i.}$

ii) The height of the tree is the longest path from the root node to the leaf node.

iii) The minimum number of nodes at height h is h+1.

iv) The maximum of nodes in a binary tree of height h is $2^{(h+1)-}1$

v) The height and number of nodes of the binary tree are inversely related.

The height of a binary tree with n nodes can be calculated as follows.

As you know that $n = 2^{h+1} - 1$

- ⇨ $n+1 = 2^{h+1}$
- ⇨ Now Taking log on both the sides
- ⇨ $\log_2(n+1) = \log2(2^{h+1})$
- ⇨ $\log_2(n+1) = h+1$
- ⇨ $h = \log_2(n+1) - 1$

Depending on the number of children of a node, the binary tree is further divided into the following categories.

Space for learners:

Space for learners:

i) **Full or Strict Binary Tree:**

The full or strict binary tree is one that each node must have either 0 or 2 children. The full binary tree can also be defined as the tree in which each node must contain 2 children except the leaf nodes.

ii) **Complete Binary Tree:**

The complete binary tree is one where all the nodes In a complete binary tree, the nodes should be added from the left.

iii) **Perfect Binary Tree:**

A perfect binary tree is one where all the internal nodes have 2 children, and all the leaf nodes are at the same level.

iv) **Degenerate Binary Tree:**

In this binary tree, the internal nodes have only one child.

v) **Balanced Binary Tree:**

The balanced binary tree is one where the left and right subtree differ by at most 1. For example, AVL and Red-Black trees.

4.6 COUNTING BINARY TREE

Let's have a binary tree. How do you count the number of nodes and the nodes have two children (two children) without or without using recursion.

i) Let's you have a binary tree, and you can count all the nodes in the binary tree using the following approach

   a. Do post-order traversal of the tree.

   b. If the root is null, then perform return 0.

   c. If the root is not null then you can make a recursive call to the left child and right child. The result of these with 1 will be return

Space for learners:

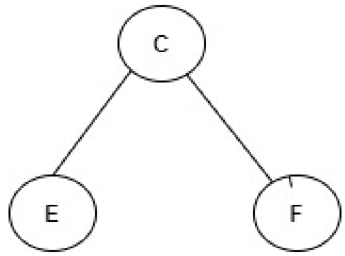


Fig. 4.4: Binary tree

In Fig. 4.4, the number of nodes is 3.

The number of nodes that have both children or null can be count using the following approach.

1) Create an empty Queue and push the root node to Queue.
2) Do following while Queue is not empty.
   a. Pop an item from Queue and process it.
      i) If it is a full node then increment the counter.
   b. Push left child of the popped item to Queue, if available.
   c. Push the right child of the popped item to Queue, if available.

4.7 FUNDAMENTAL CIRCUITS

The fundamental circuit is related to the spanning tree. Let you have a connected graph G and T be a spanning tree. Then a circuit formed by adding a chord T in the spanning tree T is known as a fundamental circuit.

To understand it, lets you have a graph G. Now, form a spanning tree T from the graph. A spanning tree is a that tree which contains all vertices of the graph without any cycle.

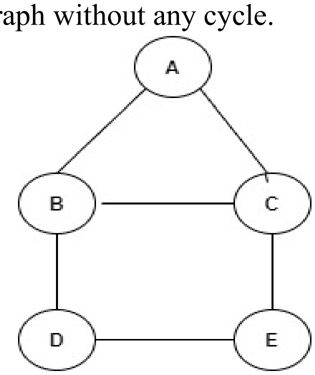


Fig. 4.5: Example of Graph

__Space for learners:__

Now, the spanning of the graph is

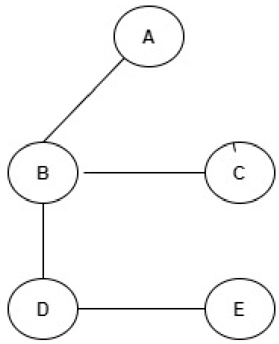


Fig. 4.6: Example of Spanning Tree

Now, you have to find the branch and chord set from the spanning tree. The branch set is that set that contains the edges of the spanning tree. The chord set is that one which does not present in the spanning tree.

So, the branch set = {AB, BC, BD, DE}

The chord set = {AC, CE}

Now, if you add AC in the spanning tree, then it will form a circuit (AB, BC, AC). So, it is known as a fundamental circuit. Again, if we add CE, then it will create another fundamental circuit (BC, CE, ED, DB).

CHECK YOUR PROGRESS- II

5. What is rooted tree?
6. What is the maximum number of nodes associated with a height h?
7. What is the height of a binary tree with n number of nodes?
8. Is perfect and full binary being the same?
9. Can you form a fundamental circuit from a spanning tree?
10. What is chord and branch set?

4.8 SPANNING-TREE IN WEIGHTED GRAPHS

A spanning tree contains all vertices of a graph without having any cycles. A spanning tree cannot be disconnected.

So, you can say that every connected and undirected graph has at least one spanning tree. Let's you have the following graph.

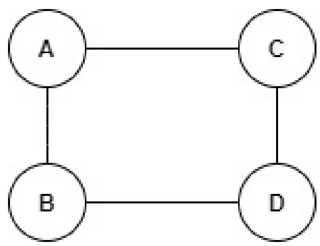


Fig. 4.7: Example of Graph

In the graph, four vertices (A, B, C, and D) are present. From the graph, you can draw the following four spanning trees.

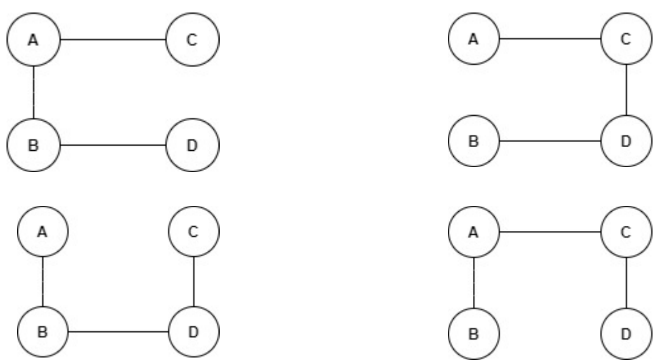


Fig. 4.8: Different Spanning Tree of Fig. 4.7

The above is graph is not complete. So in this graph, you can apply the Kirchhoff theorem to count the number of spanning trees. But if you have a complete undirected graph, you can count the number of spanning-tree using the formula n^{n-2}. Let's consider the following graph.

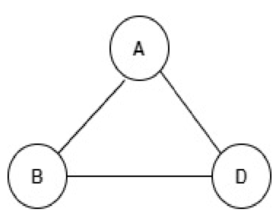


Fig. 4.9: Example of Graph

Space for learners:

The above graph is completely undirected. So, you can draw n^{n-2} = 3^{3-2} = 3. The spanning trees are as follows.

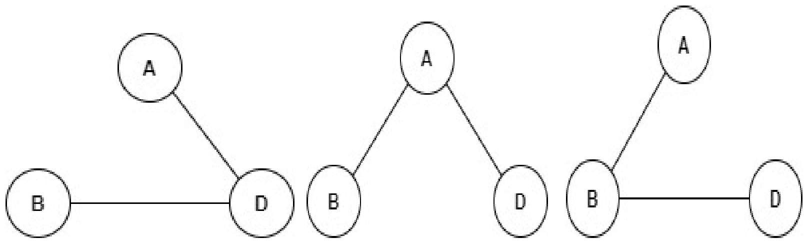


Fig. 4.10 Different Spanning Tree of Fig. 4.9

The properties of a Spanning tree are presented below.

i) A connected graph can have more than one spanning tree.

ii) All possible spanning trees must have many edges and vertices.

iii) A spanning tree does not have a closed circuit.

iv) A spanning will be disconnected after removing one edge.

v) The addition of an extra edge in the spanning tree creates the fundamental circuit.

A minimum spanning tree contains the minimum cost of the graph. A minimum spanning tree can be found in a weighted graph. A weighted graph must have one weight associated with each edge. When you create spanning trees from this type of graph, one spanning must have minimum weight, which is known as a minimum spanning tree (MST).

You can find the minimum spanning tree from a graph using the Kruskal Algorithm.

Let's you have the following graph.

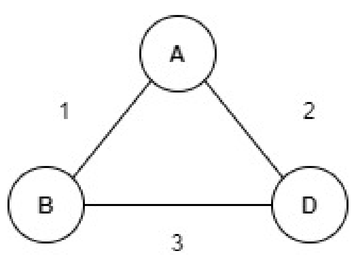


Fig. 4.11: Example of Graph

Space for learners:

In the above graph, three edges are there which have edge weights 1, 2, and 3, accordingly. So, this is a weighted graph. From the above-weighted graph, you can draw the following spanning trees.

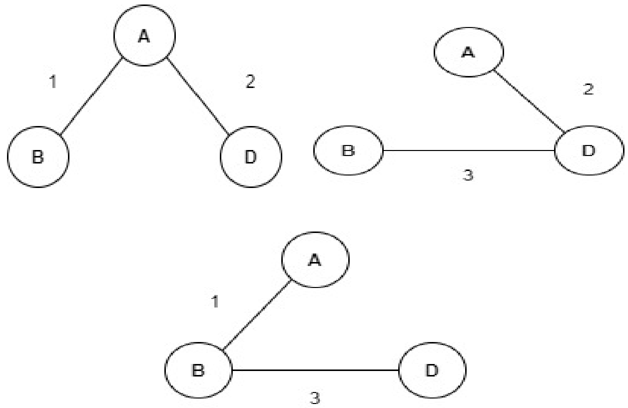


Fig. 4.12: Different Spanning Tree with the weight of Fig. 4.11

In the above three spanning, the first spanning tree has the total weight = 1 + 2 = 3. The second spanning tree has the total weight = 2 + 3 = 5. The last spanning tree has the total weight = 1 + 3 =4.

So, now, you have three spanning three with three different weights. Among all, the first spanning tree has the minimum weight. So it is known as MST.

As mentioned above, you can find the MST using Kruskal's algorithm as follows.

i) Sort all the edge weight in ascending order.

ii) Consider and add one by one edge from the sorted list.

iii) Do not add an edge it creates a cycle.

If you apply the above steps in the above graph, then the execution will be as follows.

i) After sorting the edge, you will get 1, 2, 3.

ii) Now consider the first edge weight 1 and add it to the tree, as it will not create any cycle.

iii) Then you can add edge 2, as it will also not create any cycle.

iv) Finally, consider edge weight 3. But you can not add it as it will create a cycle.

Space for learners:

So, your spanning tree will contain only the edge weight 1 and 2. So, the MST is 3.

Space for learners:

4.9 CUT SETS

Before discussing the cutsets, you first know about the cut edge and cut vertice. A cut vertice is that upon removing of which the graph will be a disconnect. Like vertice, upon removing of which edge, two or more graphs will be a disconnect, is known as the cut edge.

Let's have a graph G (V, E). A subset EE of E is called a cut set of G, if deletion of all the edges of EE from G, the G will disconnect. If deleting edges from a graph makes it disconnected, is known as cut sets.

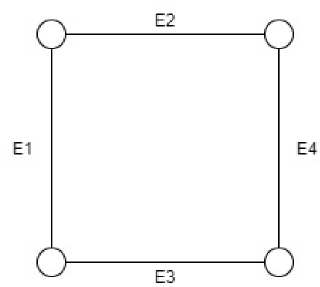


Fig. 4.13: Example of Graph

In the above graph, the graph contains 4 edges i.e., {E1, E2, E3, E4}. Let the cut set = { E1, E4}. Upon removing E1, and E4 from the graph will look like two graphs (below). So, it is the cut set.

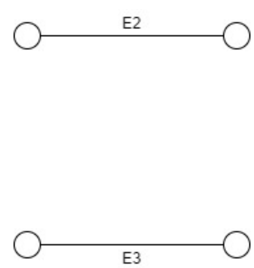


Fig. 4.14: Example of Cut set

Depending on the size of the cut, a cut set may be minimum, and maximum. If the size of the cut set is minimum as compared to the other cut set, then it is minimum otherwise the cut set may equal or maximum.

CHECK YOUR PROGRESS-III

11. What is MST?
12. Which algorithm is used to count the number of spanning tree of a graph?
13. A Spanning has a cycle (True or False).
14. What is cut vertex and cut edge?

Space for learners:

4.10 SUMMING UP

- The tree is a discrete nonlinear structure that represents hierarchical relationships between nodes.
- A tree with n nodes contains (n-1) numbers of edges. Every node has a degree.
- Every tree has a root node with degree 2.
- The degree of an internal node is at most 2.
- The vertex with the minimal eccentricity of a tree is known as the center of the tree.
- Some trees may contain only one center and this type of tree is known as a central tree. Some trees may contain more than one tree and this type of tree is known as a bi-central tree.
- A rooted tree is a connected and acyclic graph. The tree has a special node known as the root of the tree and each node is directly or indirectly connected to the root of the tree.
- The Binary tree is that rooted that has a maximum of two children. It means that each node can have either 0, 1, or 2 children.
- The minimum number of nodes at height h is h+1.
- The maximum of nodes in a binary tree of height h is 2^(h+1)-1
- The full or strict binary tree is one that each node must have either 0 or 2 children.
- A perfect binary tree is one where all the internal nodes have 2 children, and all the leaf nodes are at the same level.

- The fundamental circuit is related to the spanning tree. Let you have a connected graph G and T be a spanning tree. Then a circuit formed by adding a chord T in the spanning tree T is known as a fundamental circuit.
- A spanning tree contains all vertices of a graph without having any cycles.
- A minimum spanning tree contains the minimum cost of the graph. A minimum spanning tree can be found in a weighted graph.
- A cut vertice is that upon removing of which the graph will be a disconnect. Like vertice, upon removing of which edge, two or more graphs will be a disconnect, is known as the cut edge.

4.11 ANSWERS TO CHECK YOUR PROGRESS

1) The number of edges associated with a vertex is known as the degree of a vertex.
2) 2
3) n-1
4) False
5) A rooted tree is a connected and acyclic graph. The tree has a special node known as the root of the tree and each node is directly or indirectly connected to the root of the treewhere children of the internal nodes are ordered.
6) $2^{(h+1)-}1$
7) h = log2(n+1) – 1
8) No
9) Yes
10) The branch set is that set that contains the edges of the spanning tree. The chord set is that one which does not present in the spanning tree.
11) A minimum spanning tree contains the minimum cost of the graph. A minimum spanning tree can be found in a weighted graph.
12) Kirchhoff theorem.

Space for learners:

13) False

14) A cut vertice is that upon removing of which the graph will be a disconnect. Like vertice, upon removing of which edge, two or more graphs will be a disconnect, is known as the cut edge.

Space for learners:

4.12 POSSIBLE QUESTIONS

Short answer type questions:

i) What is a tree? What are the properties of a tree?

ii) What is the center of the tree?

iii) What is the difference between centric and bicentric trees?

iv) What is a binary tree?

v) What are the properties of a binary tree?

vi) Show that the height of a binary tree of node n is h = log2(n+1) – 1

vii) What is the difference between a full and perfect binary tree?

viii) How do you count the number of nodes in a binary tree?

ix) What isa fundamental circuit? How do you form a fundamental circuit from a spanning tree?

x) What is MST?

xi) How many spanning trees will be formed from a connected graph with n vertices?

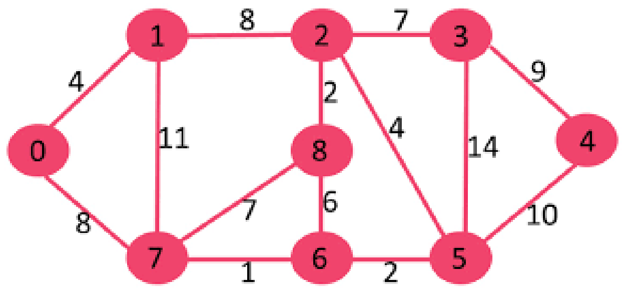
xii) What are the properties of a spanning tree?

xiii) What is cur set?

Long answer type questions:

i) Explain the MST with an example.

ii) Find the MST for the following tree



iii) Explain binary trees with their types with examples.

4.13 REFERENCES AND SUGGESTED READINGS

- Data Structures Using C by Reema Theraja Publisher: Oxford Publication

Space for learners:

UNIT 5: GRAPH REPRESENTATION

Space for learners:

Unit Structure:

5.1 Introduction
5.2 Unit Objectives
5.3 Matrix Representation of Graphs
5.4 Adjacency Matrix
5.5 Adjacency List
5.6 Incidence Matrix
5.7 Basic Concept of Graph Coloring, Covering and Partitioning
5.8 Summing Up
5.9 Answers to Check Your Progress
5.10 Possible Questions
5.11 References and Suggested Readings

5.1 INTRODUCTION

Space for learners:

Graph theory has evolved into a powerful tool that can be used to a wide range of areas. Engineering mathematics, computer programming, networking and marketing are only a few of them. Paths generated by travelling along the edges of a graph can be used to simulate a variety of issues. Models that incorporate pathways in graphs can be used to address problems such as efficiently designing routes for parcel delivery, waste collection, and finding shortest path. Graphs may grow exceedingly complicated when faced with these types of problems, necessitating a more efficient means of expressing them in practice. The adjacency matrix and adjacency list are used to solve this problem.

5.2 UNIT OBJECTIVES

After going through this unit, you will be able to:

- To understand and apply the fundamental concepts in graph representations
- To explain the basic concepts on graph colouring and solving practical problems.
- To explain the basic concepts on graph covering and solving problems.

5.3 MATRIX REPRESENTATION OF GRAPHS

In a computer, there are several methods to represent a graph. Graphs are typically depicted diagrammatically, although this is only viable when the number of vertices and edges is minimal. As a result, the notion of graph matrix representation is established. The computation of paths and cycles in graphs problems such as communication networks, power distribution, and transportation, among others, is one of the primary advantages of this representation. However, this format has the drawback of reducing the visual appeal of graphs.

5.4 ADJACENCY MATRIX

Space for learners:

The most convenient way of representing any graph is the matrix representation. It is a square matrix of order (n xn) where n is the number of vertices in the graph. Generally represented by $M[a_{ij}]$ where a_{ij} is the i^{th} row, j^{th} column element. The general form of adjacency matrix is given below.

$$\text{Where, } a_{ij} = \begin{cases} 1; \text{ if an edge in the graph between the vertex } v_i \text{ and } v_j \\ 0; \text{ otherwise} \end{cases}$$

$$\mathbf{M} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots\cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{22} & \cdots\cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \cdots\cdots & a_{3n} \\ . & . & . & . & \cdots\cdots & . \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \cdots\cdots & a_{nn} \end{pmatrix}$$

The matrix is termed as adjacency matrix, because each entry in the matrix stores the information between the vertices as adjacent or not. The entry can be either 0 or 1.

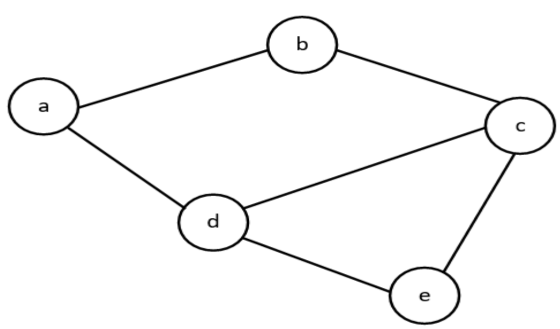


Fig. 5.1: A Simple Graph

Consider the graph, G given in Figure 5.1, the adjacency matrix with respect to the vertices a, b, c, d, e is shown below. As there is an edge between vertex ‘a’ and vertex ‘b’, the corresponding position in the adjacency matrix is having the entry 1. As there is no edge between vertex ‘a’ and vertex ‘c’, the corresponding position in the adjacency matrix is having the entry 0.

$$
\mathbf{I} = \begin{array}{c} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \end{array}
\begin{array}{c} \begin{array}{ccccc} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} \end{array} \\
\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}
$$

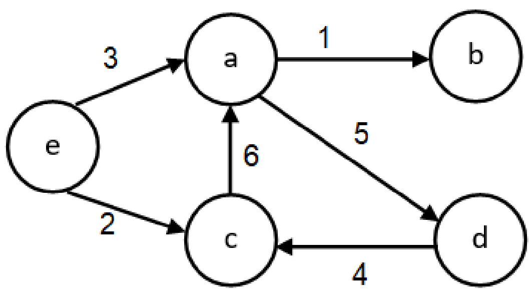


Fig. 5.2: A Weighted Graph

Consider the graph, G given in Figure 5.2, the adjacency matrix with respect to the vertices a, b, c, d, e is shown below. As there is a directed edge from vertex ‘a’ to vertex ‘d’ having weight ‘5’ the corresponding position in the adjacency matrix is having the entry 5. As there is no directed edge from vertex ‘a’ to vertex ‘c’, the corresponding position in the adjacency matrix is having the entry 0.

Space for learners:

$$M = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 3 & 0 & 2 & 0 & 0 \end{bmatrix} \end{array}$$

Space for learners:

STOP TO CONSIDER

In an adjacency matrix, if the diagonal elements are zero, then the graph is called simple graph.

In a multi graph i.e. a graph having parallel edges, adjacency matrix can be found using

$$a_{ij} = \begin{cases} n; \text{ number of edges between a pair of vertex } v_i \text{ and } v_j \\ 0; \text{ otherwise} \end{cases}$$

In a weighted graph, adjacency matrix can be found using

$$a_{ij} = \begin{cases} w; \ w \text{ is the weight of edge between the vertex between } v_i \text{ and } v_j \\ 0; \text{ otherwise} \end{cases}$$

5.5 ADJACENCY LIST

An adjacency list is a group of unordered lists that is used to describe a finite graph. Each unordered list in an adjacency list describes a vertex's collection of neighbors in the graph. This is one of the graph representations frequently used in computer systems.

In Adjacency List, an array of a list "Adjlist[i]"is used to represent the graph. The list size is equal to the number of vertex(n).

If we assume that graph has *n* vertex, then

Adjlist[0] will have all the vertices that are connected to vertex 0.

Adjlist[1] will have all the vertices that are connected to vertex 1 and so on.

Consider the undirected graph G in Figure 5.3.

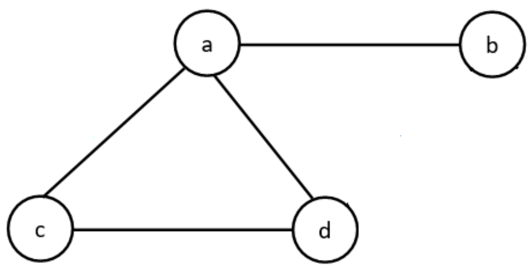
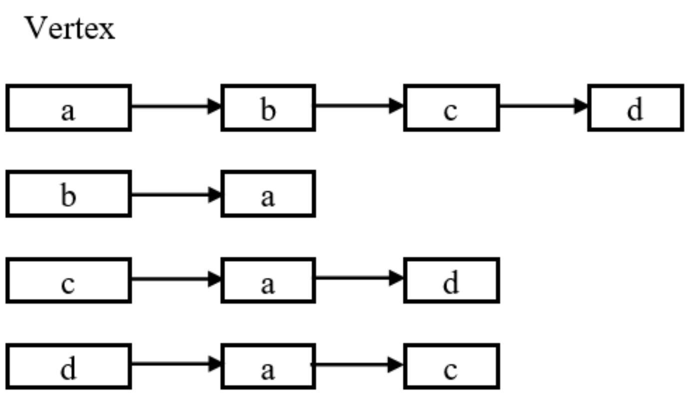


Fig. 5.3: An undirected Graph

The adjacency list for the undirected graph is shown below, here the adjacency list for vertex 'a' are the vertices adjacent to 'a' that is there is an edge connecting 'a' with vertices 'b', 'c' and 'd'. Similarly, adjacency list for vertex 'c' is 'a' and 'd', adjacency list for vertex 'd' is 'a' and 'c'. Finally, adjacency list for vertex 'b' is 'a'.



Consider the directed graph G in **Fig.5.4**

Space for learners:

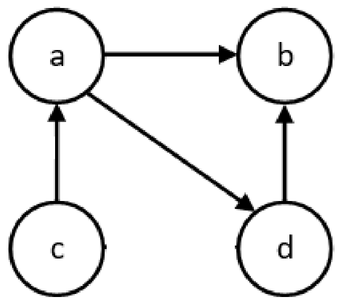
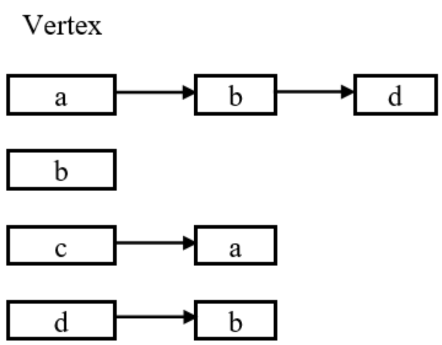


Fig. 5.4: A directed Graph

The adjacency list for directed graph is shown below, here the adjacency list for vertex ‘a’ are the vertices adjacent to ‘a’ that is there is an outgoing edge from ‘a’ to vertex ‘b’ and ‘d’. Similarly, adjacency list for vertex ‘c’ is the outgoing edge from ‘c’ to ‘a’ and adjacency list for vertex ‘d’ is the outgoing edge from ‘d’ to ‘b’. Finally, adjacency list for vertex ‘b’ is nil, as there is no outgoing edge from vertex ‘b’.



5.6 INCIDENCE MATRIX

Consider a graph G with n vertices and e edges, then the incidence matrix I[a_{ij}] is a matrix of order (n x e) where the element a_{ij}, where rows correspond to its vertices and columns correspond to its edges is defined as

$$a_{ij} = \begin{cases} 1; \text{ if vertex } i \text{ belongs to edge } j \\ 0; \text{ otherwise} \end{cases}$$

Space for learners:

Consider the graph G in Figure 5.5

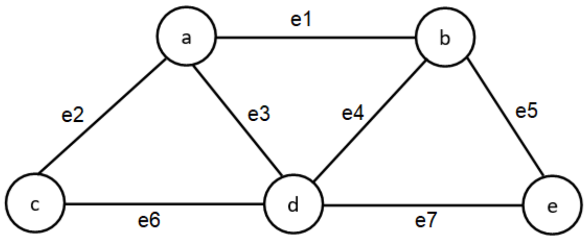


Fig. 5.5: A Graph

The incidence matrix with respect to the vertices a, b, c, d, e and edges e1, e2, e3, e4, e5, e6, e7 is shown below. As there is an edge incident on vertex 'a' and vertex 'b' the corresponding position in the incidence matrix is having the entry 1. As there is no edge e4, e5, e6 and e7 incident on vertex 'a', the corresponding position in the incidence matrix is having the entry 0.

$$\mathbf{I} = \begin{array}{c} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{e1} & \mathbf{e2} & \mathbf{e3} & \mathbf{e4} & \mathbf{e5} & \mathbf{e6} & \mathbf{e7} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Similarly, the incidence matrix $I[a_{ij}]$ of a digraph G is defined as

$$a_{ij} = \begin{cases} 1; \text{ if edge } j \text{ is incident out of vertex } i \\ -1; \text{ if edge } j \text{ is incident into vertex } i \\ 0; \text{ otherwise} \end{cases}$$

Consider the graph G in Figure 5.6.

Space for learners:

Space for learners:

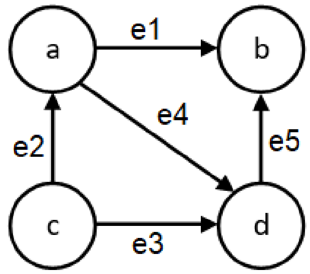


Fig. 5.6: A Graph

The incidence matrix with respect to the vertices a, b, c, d and edges e1, e2, e3, e4, e5 is shown below. As there is an edge 'e1' and 'e4' incident out of vertex 'a' the corresponding position in the incidence matrix is having the entry 1. Whereas there is an edge 'e2' incident into vertex 'a', the corresponding position in the incidence matrix is having the entry -1. Finally, as neither the edge 'e3' and 'e4' is incident into or out of vertex 'a', the corresponding entry in the incidence matrix is marked as 0.

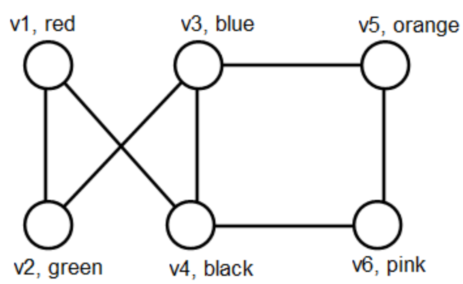
$$
\begin{array}{cc}
 & \begin{array}{ccccc} \mathbf{e1} & \mathbf{e2} & \mathbf{e3} & \mathbf{e4} & \mathbf{e5} \end{array} \\
\mathbf{I} = \begin{array}{c} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{array} & \begin{bmatrix} 1 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 \end{bmatrix}
\end{array}
$$

5.7 BASIC CONCEPT OF GRAPH COLORING, COVERING AND PARTITIONING

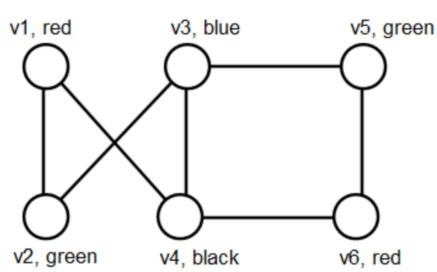
Graph Coloring: Consider a graph G having n vertices. If we want to paint all the vertices such that no two adjacent vertices are of same colour then a question can be asked as to what should be the minimum number of colours required in such case? This type of problem constitutes graph colouring problem. Similarly, colouring problem in a graph can be applied also to the edges. One application of graph colouring is Map Coloring where geographical map of states where no two adjacent states can be assigned same color.

As an example, in Figure 5.7 (a), assigning all the vertices with colours such that no two adjacent vertices are assigned same colour is called proper colouring. In some cases, proper colouring with minimum number of colours may be required. In Figure 5.7 (b), 4 different colours are used compared to six in Figure 5.7 (a).

The minimum number of colours required to colour a graph G is called its chromatic number. If a graph requires *k* different colours for its proper colouring then it is known as *k*-chromatic or *k*-colourable.



(a)



(b)

Fig. 5.7: Proper Colouring of a Graph

Space for learners:

STOP TO CONSIDER

A complete graph where each vertex is connected to every other vertex having n vertices, has chromatic number $k = n$.

A cycle graph having n vertices, has chromatic number $k = 3$ if n is odd or $k = 2$ if n is even.

Space for learners:

Graph Covering: A covering graph C is a subgraph which contains either all the vertices or all the edges corresponding to some other graph G.

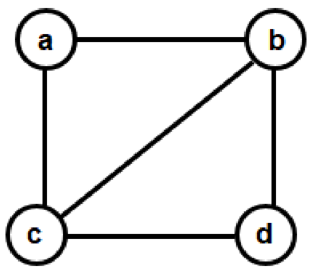


Fig. 5.8: A simple Graph

A subset is called a line covering of a graph G if every vertex of G is incident with at least one edge. For example, in the graph given in Figure 5.8, subset S_1, S_2, S_3, S_4 are line covering as all the vertices are covered using the edges in each of the subset. However, subset S_5 is not line covering due to the fact that vertex 'c' is not covered.

$S_1 = \{(a, b), (c, d)\}$

$S_2 = \{(a, c), (b, d)\}$

$S_3 = \{(a, b), (b, d), (c, d)\}$

$S_4 = \{(a, b), (b, c), (b, d)\}$

$S_5 = \{(a, b), (b, d)\}$

A subset K of V is called a vertex covering of a graph G (V, E), if every edge of 'G' is incident with or covered by a vertex in 'K'. For example, in the graph given in Figure 5.8, subset K_1 contains vertex 'b' and 'c' which covers the edges that is 'ba', 'bc', 'bd' and 'ca', 'cb', 'cd' respectively. Thus, all the edges are covered by vertex {b, c} and so is K_1 vertex covering. Similarly, subset K_2 contains vertex 'a' 'b' and 'c' which covers all the edges in the graph G. Also, subset K_3

contains vertex 'a' 'd' and 'c' which covers all the edges in the graph G. So, both K_2 and K_3 are vertex covering. However, subset K_4contains vertex 'a' and 'c' which do not cover the edge 'bd', therefore K_4 is not vertex covering.

K1 = {b,c}

K2 = {a, b,c}

K3 = {a,d, c}

K4= {a, c}

Graph Partitioning: A graph partition is the process of reducing a large graph to a smaller one by grouping its nodes into mutually exclusive groups.

Graph chromatic partitioning: A proper colouring of a graph induces partitioning of vertices into different subsets such as the graph shown in Figure 5.7 (b) can be portioned into {v1, v6}, {v2, v5}, {v3} and {v4}. As it can be observed that no two vertices in the four subsets are adjacent. Such a subset of vertices is called an independent set.

A maximal independent set is an independent set to which no other vertex can be added without compromising its independence property. There can be many maximal independent sets of different sizes, however the one with largest number of vertices is of particular importance.

CHECK YOUR PROGRESS

1 In a graph G, number of vertices is 5. What is the total number of elements in the adjacency matrix?
   a) 5
   b) 25
   c) 10
   d) 125

2 Which of these adjacency matrices represents a simple graph?
   a) [[0, 0, 1], [1, 0, 1], [1, 0, 0]]
   b) [[1, 0, 0], [0, 1, 0], [0, 1, 1]]
   c) [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
   d) [[0, 0, 1], [0, 0, 0], [0, 0, 1]]

Space for learners:

3 In a simple graph, sum of the column in an incidence matrix is __________

a) number of edges
b) greater than 2
c) number of edges + 1
d) equal to 2

4 The dimensions of an incidence matrix for graph having v as the number of vertices and e as the number of edges is given by ___________.

a) e x e
b) v x e
c) v x v
d) ex (v + e)

5 Incidence matrix and Adjacency matrix of a graph G will always have ________?

a) Same dimension
b) Different dimension
c) Some cases may have different dimension
d) None of the above

6 Vertex coloring of a graph is_______________.

a) Adjacent vertices do not have same color
b) Adjacent vertices always have same color
c) All vertices should have a different color
d) All vertices should have same color

7 Minimum number of unique colors required so that adjacent vertices do not have the same colour is given by_____________.

a) chromatic key
b) chromatic index
c) chromatic number
d) color number

8 In an empty graph having n vertices_________ number of unique colours will be needed for vertex colouring.

a) n + 1
b) 1
c) 2
d) n

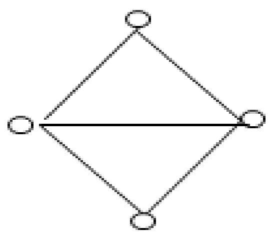
Space for learners:

Space for learners:

9 In an empty graph having n vertices_________ number of unique colours will be needed for vertex colouring.

a) n-1
b) 1
c) n+1
d) n

10 How many unique colors will be required for vertex coloring of the following graph?



a) 2
b) 3
c) 4
d) 5

5.8 SUMMING UP

- Adjacency matrix is represented by $M[a_{ij}]$ where a_{ij} is the i^{th} row, j^{th} column element. The general form is given by:

$$a_{ij} = \begin{cases} 1; \text{ if an edge in the graph between the vertex } v_i \text{ and } v_j \\ 0; \text{ otherwise} \end{cases}$$

- Incidence matrix $I[a_{ij}]$ is a matrix of order (n x e) where the element a_{ij}, where rows correspond to its vertices and columns correspond to its edges is defined as

$$a_{ij} = \begin{cases} 1; \text{ if vertex } i \text{ belongs to edge } j \\ 0; \text{ otherwise} \end{cases}$$

- An adjacency list is a group of unordered lists that is used to describe a finite graph. Each unordered list in an adjacency list describes a vertex's collection of neighbours in the graph.
- Graph Coloring problem is to paint all the vertices such that no two adjacent vertices are of same colour.
- Graph Covering: A covering graph C is a subgraph which contains either all the vertices or all the edges corresponding to some other graph G.
- A graph partition is the process of reducing a large graph to a smaller one by grouping its nodes into mutually exclusive groups.

5.9 ANSWERS TO CHECK YOUR PROGRESS

1. b	2.a	3.d	4.b	5.b
6.a	7. c	8. b	9. d	10. b

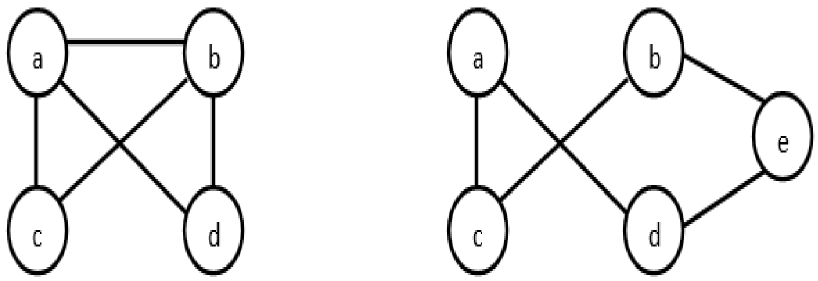
5.10 POSSIBLE QUESTIONS

1. Draw the graph having the following matrix as its adjacency matrix.

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 2 & 3 \\ 4 & 1 & 3 & 1 & 2 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

2. Draw the adjacency matrix and adjacency list of the following graphs

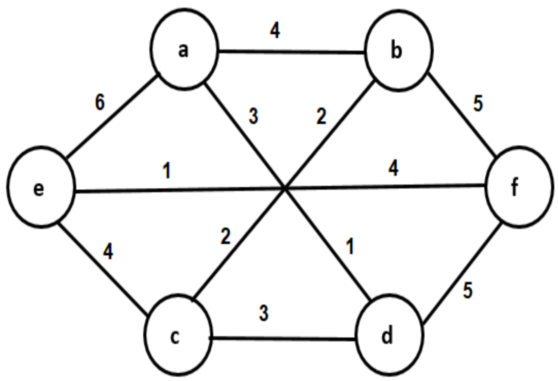
Space for learners:



(a) (b)

Space for learners:

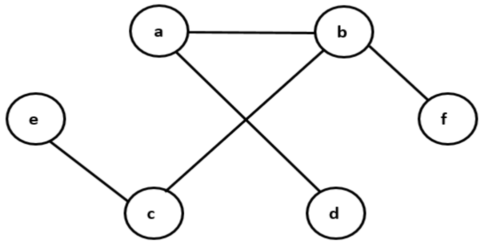
3. Write the adjacency matrix of the graph given below.



4. Draw the graph for the incidence matrix given below:

$$I = \begin{array}{c} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{e1} & \mathbf{e2} & \mathbf{e3} & \mathbf{e4} & \mathbf{e5} & \mathbf{e6} & \mathbf{e7} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

5. Draw the incidence matrix for the graph given below.

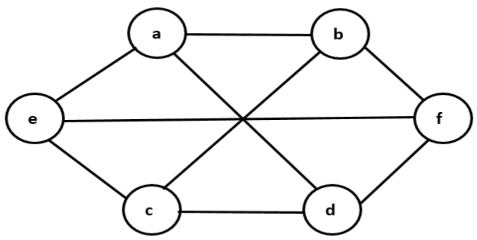


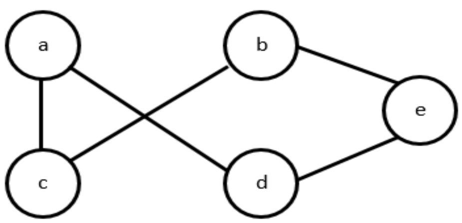
6. What is graph colouring? Explain using an example.

7. What is graph covering?

8. Differentiate between vertex covering and edge covering.

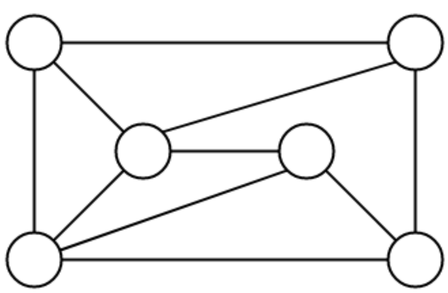
9. Find chromatic number of the following graphs





(a) (b)

10. Find chromatic number of the following graph.



Space for learners:

5.11 REFERENCES AND SUGGESTED READINGS

Space for learners:

- Graph Theory with Applications to Engineering and Computer Science by Narsingh Deo, Published by Prentice Hall India Learning Private Limited.
- Introduction to Graph Theory by Richard J Trudeau, Published by Courier Corporation.
- A First Course in Graph Theory by Gary Chartrand and Ping Zhang, Published by Courier Corporation.
- Graph theory with applications by John Adrian Bondy, Published by Elsevier Publishing Company