IS 5311: Discrete Mathematics

Chapter 4: Graph Theory

Lecturer: MS. M.W.S. Randunu *Department of Interdisiplinary Studies,University of Ruhuna.*

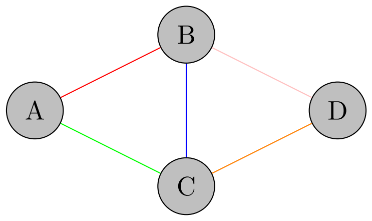
Graphs are like maps with dots (vertices) and lines (edges) connecting them. Depending on how these lines are drawn, graphs can be different types. They're useful for solving all sorts of puzzles and problems. For example, they can help you figure out if you can walk around a city without walking down the same street twice. They can also help you figure out how to color a map with as few colors as possible. Graphs are used in many other ways too, like designing computer networks or planning the shortest route between two cities. This chapter teaches you the basics of graphs and how they're used to solve problems in different areas.

4.1 Graph

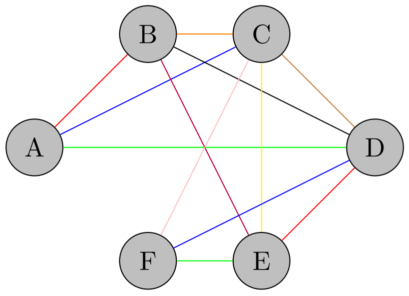
Definition 1. *A graph $G = (V, E)$ consists of V, a nonempty set of vertices (or nodes), and E, a set of edges. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints.*

The **order** of a graph refers to the number of vertices it contains, denoted by $|V|$. The **size** of a graph refers to the number of edges it contains, denoted by $|E|$.

Examples



- Order: 4 (vertices A, B, C, and D)
- Size: 5 (edges AB, AC, BC,BD and CD)



- Order: 6
- Size: 12

Note:

- The set of vertices V in a graph G can be infinite. A graph with an infinite number of vertices or edges is referred to as an infinite graph. Conversely, a graph with a finite number of vertices and edges is termed a finite graph.

- When drawing a graph, efforts are made to arrange the edges so they do not intersect. However, this is not always possible, as some graphs cannot be drawn without edges crossing.

Example: In a network consisting of data centers and communication links between computers, each data center can be represented by a point, and each communication link by a line segment.

This network can be represented using a graph, where the vertices correspond to the data centers and the edges represent the communication links. Typically, graphs are visualized by using points for vertices and line segments (which may be curved) for edges. The endpoints of a line segment representing an edge are the points representing the endpoints of the edge.

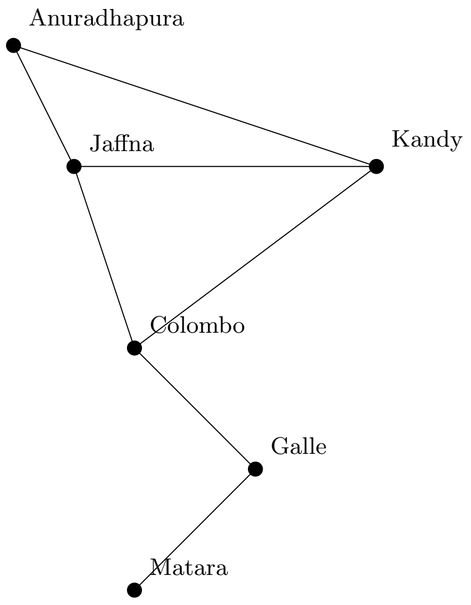


Figure 4.1: A Computer Network

Definition 2. *A **simple graph** is a graph in which each edge connects two different vertices, and no two different edges connect the same pair of vertices.*

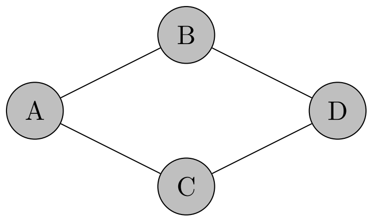


Figure 4.2: Simple graph .

Definition 3. *A **multigraph** is a graph that may have multiple edges connecting the same pair of vertices.*

When there are m different edges associated with the same unordered pair of vertices $\{u, v\}$, we also say that $\{u, v\}$ is an edge of multiplicity m. That is, we can think of this set of edges as m different copies of an edge $\{u, v\}$.

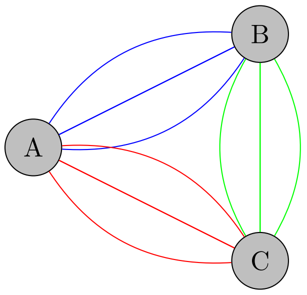


Figure 4.3: A multigraph with multiple edges between vertices A, B, and C.

In some cases, a communications link connects a data center to itself.To model this network, we need to include edges that connect a vertex to itself, known as **loops**.

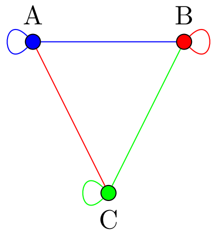


Figure 4.4: Undirected Graph with Loops

A directed graph, or digraph, has edges with directions, represented by arrows. This is useful for modeling situations where relationships are not mutual, such as one-way streets or one-way communication links in a network.

Definition 4. *A **directed graph** (or **digraph**) (V, E) consists of a nonempty set of vertices V and a set of directed edges (or arcs) E.*

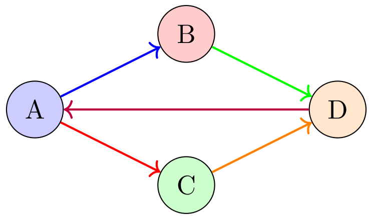


Figure 4.5: A directed graph with vertices A, B, C, and D.

In a directed graph, arrows indicate the direction of edges. Each arrow points from one vertex u to another vertex v, showing that the edge starts at u and ends at v. Directed graphs can have:

1. Loops: An edge that points back to the same vertex.
2. Multiple edges: More than one edge between the same pair of vertices.
3. Bidirectional Edges: Edges going in both directions between two vertices.

A **simple directed graph** has no loops or multiple edges between the same pair of vertices. In such graphs, each pair of vertices (u, v) has at most one edge.

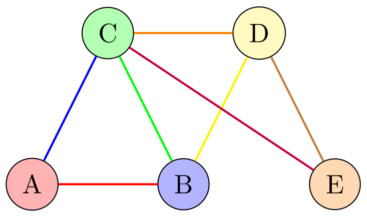
If there are m directed edges associated with the ordered pair of vertices (u, v), we say that (u, v) is an edge of multiplicity m.

Graphs are versatile tools used in various scenarios. Initially, we explored their role in modeling communication networks between data centers. Additionally, graphs are employed to represent social connections, such as friendships. In these models, each person is depicted as a vertex, with edges indicating connections between individuals who know each other. These graphs are kept simple, without multiple edges or loops, to focus on capturing social relationships clearly. This simplified representation helps visualize social networks and analyze interpersonal connections within groups or communities.

Example: Imagine we have a group of people: Alice (A), Bob (B), Charlie (C), Diana (D), and Eve (E). They know each other as follows:

- Alice knows Bob and Charlie.
- Bob knows Charlie and Diana.
- Charlie knows Diana and Eve.
- Diana knows Eve.

We will represent these relationships with a graph.

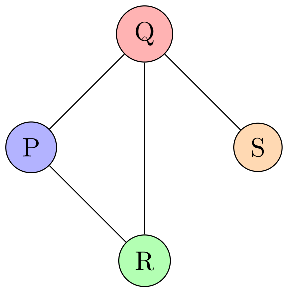


4.2 Basic Terminology

Definition 5. *The **degree** of a vertex v in an undirected graph is defined as the number of edges incident with it. However, if there is a loop at vertex v, it contributes twice to the degree of that vertex. The degree of vertex v is denoted by $deg(v)$.*

Example:

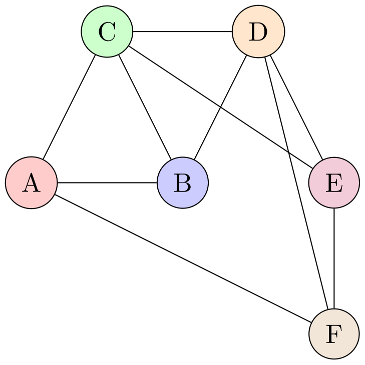
1. Consider a graph with vertices A, B, C, and D.



   In this case:

   - $\deg(P) = 2$ because P is connected to vertices Q and R.
   - $\deg(Q) = 3$ because Q is connected to vertices P, R, and S.
   - $\deg(R) = 2$ because R is connected to vertices P and Q.
   - $\deg(S) = 1$ because S is only connected to vertex Q.

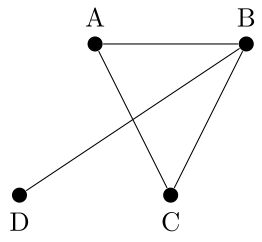
2. Consider a graph with vertices A, B, C,D, E and F.



   - $\deg(A) = 3$
   - $\deg(B) = 3$
   - $\deg(C) = 4$
   - $\deg(D) = 4$
   - $\deg(E) = 3$
   - $\deg(F) = 3$

Definition 6. *In an* ***undirected graph*** *G, two vertices u and v are considered* ***adjacent*** *or neighbors if they share an edge e. This edge e is* ***incident*** *with both vertices u and v, and it connects them.*

Example: Consider an undirected graph G with vertices A, B, C, and D.



In this graph:

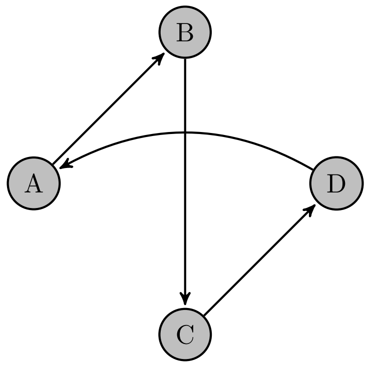
- Vertices A and B are adjacent because they share the edge AB.
- Vertices A and C are adjacent because they share the edge AC.
- Vertices B and D are adjacent because they share the edge BD.
- Vertices C and D are adjacent because they share the edge CD.

Thus, in this graph, every pair of vertices that shares an edge is considered adjacent.

Definition 7. *In a graph with* ***directed edges****, if there's an edge (u, v), then we say that u is* ***adjacent*** *to v, and v is* ***adjacent*** *from u.*

The vertex u is the initial vertex of the edge (u, v), and v is the terminal or end vertex of that edge. If there's a loop at a vertex, the initial and terminal vertices of that loop are the same.

Example: Consider a directed graph G with vertices $V = \{A, B, C, D\}$ and edges $E = \{(A, B), (B, C), (C, D), (D, A)\}$.



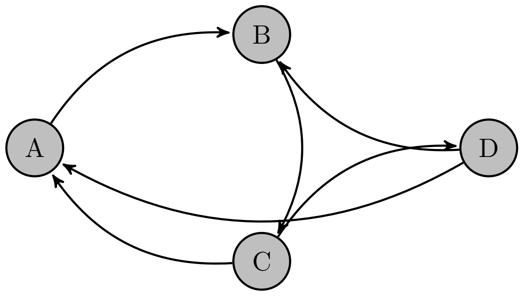
1. The edge (A, B) indicates that vertex A is adjacent to vertex B.
2. The edge (B, C) indicates that vertex B is adjacent to vertex C.

3. The edge (C, D) indicates that vertex D is adjacent from vertex C.
4. The edge (D, A) indicates that vertex A is adjacent from vertex D.

Definition 8. *In a graph with directed edges, the **in-degree** of a vertex v, denoted by $\boldsymbol{deg}^-(v)$, is the number of edges with v as their terminal vertex. The **out-degree** of v, denoted by $\boldsymbol{deg}^+(v)$, is the number of edges with v as their initial vertex.*

Note: A loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.

Exercise: Consider the directed graph shown below:

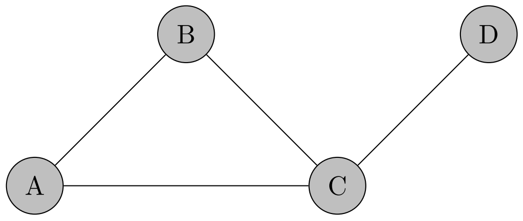


What is the out-degree of vertex C? What about the in-degree of vertex D?

Definition 9. *The **degree sequence** of a graph is a sequence of its vertex degrees in non-increasing order.*

For directed graphs, we have both in-degrees and out-degrees.

Example: Consider the graph G with vertices $V = \{A, B, C, D\}$ and edges $E = \{(A, B), (A, C), (B, C), (C, D)\}$.



1. Vertices and their degrees:
   - $\deg(A) = 2$
   - $\deg(B) = 2$
   - $\deg(C) = 3$
   - $\deg(D) = 1$
2. Degree Sequence:
   - Arranged in non-increasing order: $\{3, 2, 2, 1\}$

Note:

- Minimum Degree: The smallest degree of any vertex in the graph.
  - For the example above, the minimum degree is 1 (vertex D).
- Maximum Degree: The largest degree of any vertex in the graph.
  - For the example above, the maximum degree is 3 (vertex C).

Theorem 1:

Let $G = (V, E)$ be a graph with directed edges. Then,

$$\textstyle\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

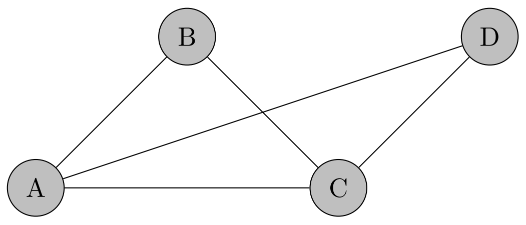
4.2.1 Handshaking Theorem

Theorem 2: Let $G = (V, E)$ be an undirected graph with m edges. Then the sum of the degrees of all vertices is equal to twice the number of edges. This can be formally stated as:

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

Note: This theorem applies even if multiple edges and loops are present.

Example: Consider a graph G with vertices $\{A, B, C, D\}$ and edges $\{(A, B), (A, C), (A, D), (B, C), (C, D)\}$.



$$\deg(A) = 3$$
$$\deg(B) = 2$$
$$\deg(C) = 3$$
$$\deg(D) = 2$$

$$\deg(A) + \deg(B) + \deg(C) + \deg(D) = 3 + 2 + 3 + 2 = 10$$

$$2m = 2 \times 5 = 10$$

Therefore:

$$\sum_{v \in V} \deg(v) = 10$$

Example: A graph has 10 edges and each vertex has a degree of 3. How many vertices does this graph have?

Solution:

Let $G = (V, E)$ be an undirected graph with $|E| = 10$ edges, and let each vertex have a degree of 3. According to the Handshaking Theorem:

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

Here, each vertex has a degree of 3, and the graph has 10 edges. Thus:

$$2 \times 10 = \sum_{v \in V} \deg(v)$$

$$20 = \sum_{v \in V} \deg(v)$$

Let n be the number of vertices in the graph. Since each vertex has a degree of 3:

$$\sum_{v \in V} \deg(v) = 3n$$

Therefore:

$$3n = 20$$

Solving for n:

$$n = \frac{20}{3}$$

Since the number of vertices must be an integer, we find that it's impossible for every vertex in a graph with 10 edges to have a degree of 3. Therefore, no such graph can exist.

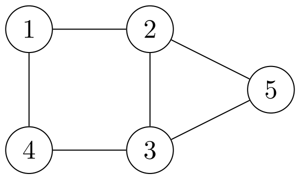
Lemma 1: An undirected graph has an even number of vertices of odd degree.

Proof: Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

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

Because $\deg(v)$ is even for $v \in V_1$, the first term on the right-hand side of the last equality is even.
Furthermore, the sum of the two terms on the right-hand side of the last equality is even, because this sum is $2m$. Hence, the second term in the sum is also even.
Because all the terms in this sum are odd, there must be an even number of such terms.
Thus, there are an even number of vertices of odd degree.

Example: Consider the following graph:



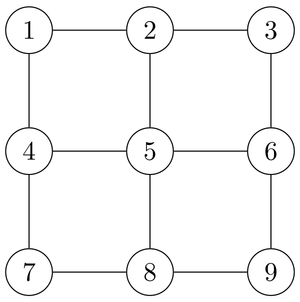
Now, let's apply the handshake lemma to this graph:

$$\begin{aligned}\deg(1) &= 2\\ \deg(2) &= 3\\ \deg(3) &= 3\\ \deg(4) &= 2\\ \deg(5) &= 2\end{aligned}$$

The sum of the degrees of all vertices is $2 + 3 + 3 + 2 + 2 = 12$.

Since each edge contributes 2 to the sum of degrees (1 for each endpoint), and there are 6 edges in total, the total sum is $2 \times 6 = 12$, which matches the sum of the degrees of all vertices.

Example: Consider the following graph:



Now, let's apply the handshake lemma to this graph:

The sum of the degrees of all vertices is $2 + 3 + 2 + 3 + 4 + 3 + 2 + 3 + 2 = 24$.

Since each edge contributes 2 to the sum of degrees (1 for each endpoint), and there are 12 edges in total, the total sum is $2 \times 12 = 24$, which matches the sum of the degrees of all vertices.

To simplify computation, graphs can be represented using matrices. Two types of matrices commonly used to represent graphs will be presented here. One is based on the adjacency of vertices, and the other is based on incidence of vertices and edges.

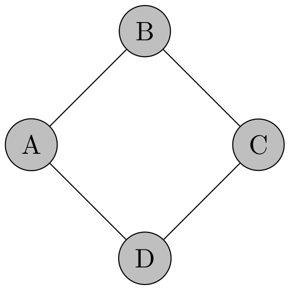
4.3 Adjacency Matrices

Definition 10. *The adjacency matrix A of a graph G with n vertices is an $n \times n$ matrix where each element a_{ij} is defined as follows:*

$$a_{ij} = \begin{cases} 1 & \textit{if there is an edge between vertices } i \textit{ and } j \\ 0 & \textit{otherwise} \end{cases}$$

In an undirected graph, the adjacency matrix is symmetric, while in a directed graph, it may not be symmetric.

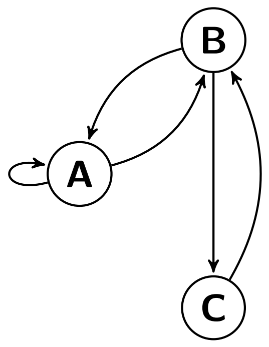
Example: Consider an undirected graph with 4 vertices labeled A, B, C, and D.



The adjacency matrix A for this graph would be:

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Exercise: For the directed graph below, what is the adjacency matrix?

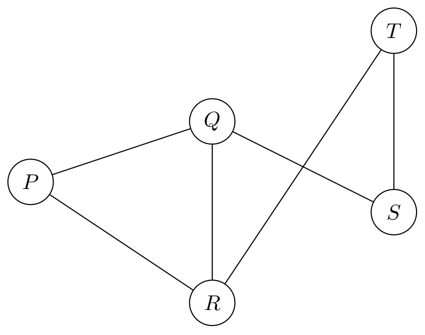


Example: Draw a graph with the adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

with respect to the ordering of vertices P, Q, R, S, T.

Solution: To draw a graph based on the given adjacency matrix, we can represent each vertex as a node and connect nodes with edges where the corresponding entry in the matrix is 1.



Note: The adjacency matrix of a simple undirected graph is symmetric, meaning $a_{ij} = a_{ji}$, because both entries are 1 when v_i and v_j are adjacent, and both are 0 otherwise. Additionally, in a simple graph with no loops, each diagonal entry a_{ii}, where $i = 1, 2, 3, \ldots, n$, is always 0.

4.4 Incidence Matrices

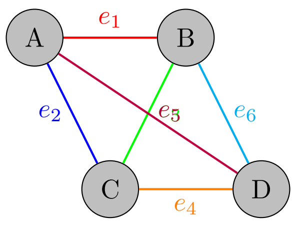
The incidence matrix of a graph is a matrix that represents the relationships between vertices and edges. It is defined as follows:

Definition 11. *Consider a graph $G = (V, E)$ with n vertices and m edges. The vertex-edge incidence matrix M of G is an $n \times m$ matrix such that:*

- *If vertex v_i is incident with edge e_j, then $M_{ij} = 1$.*
- *If vertex v_i is not incident with edge e_j, then $M_{ij} = 0$.*

In other words, each row of the matrix represents a vertex, each column represents an edge, and the entry M_{ij} is 1 if vertex v_i is incident with edge e_j, and 0 otherwise.

Example:

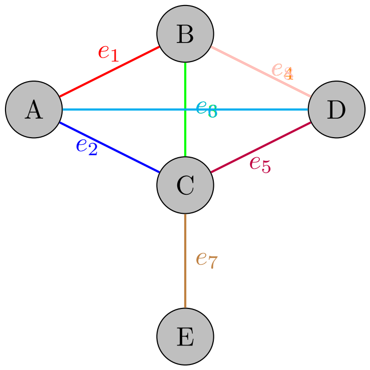


We have a graph with vertices A, B, C, D and edges $e_1, e_2, e_3, e_4, e_5, e_6$. The incidence matrix for this graph is given by:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- Each row corresponds to a vertex (in the order A, B, C, D).
- Each column corresponds to an edge (in the order $e_1, e_2, e_3, e_4, e_5, e_6$).
- An entry of 1 indicates that the vertex (row) is incident with the edge (column).

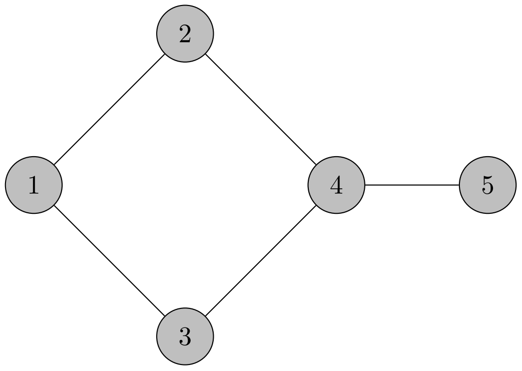
Exercise: Write down the incidence matrix for the graph with vertices A, B, C, D, E and edges $e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8$.



Definition 12. *The set of all neighbors of a vertex v in a graph $G = (V, E)$, denoted by $N(v)$, is called the **neighborhood** of v. If A is a subset of V, the set of all vertices in G that are adjacent to at least one vertex in A is denoted by $N(A)$. Formally, this is expressed as:*

$$N(A) = \bigcup_{v \in A} N(v)$$

Example: Consider a graph G with vertices $\{1, 2, 3, 4, 5\}$ and edges $\{(1, 2), (1, 3), (2, 4), (3, 4), (4, 5)\}$.



Neighborhood of a single vertex:

- $N(1) = \{2, 3\}$
- $N(2) = \{1, 4\}$
- $N(3) = \{1, 4\}$
- $N(4) = \{2, 3, 5\}$
- $N(5) = \{4\}$

Neighborhood of a subset of vertices:

- For $A = \{1, 2\}$:

$$N(A) = N(\{1, 2\}) = N(1) \cup N(2) = \{2, 3\} \cup \{1, 4\} = \{1, 2, 3, 4\}$$

4.5 Connectivity

Many problems can be modeled using paths formed by traveling along the edges of graphs. For example, determining if a message can be sent between two computers via intermediate links can be studied with a graph model. Problems such as planning efficient routes for mail delivery, garbage pickup, and diagnostics in computer networks can also be solved using graph paths.

4.5.1 Walk

Definition 13. *Given a graph $G = (V, E)$, a **walk** of length l is defined as a sequence*

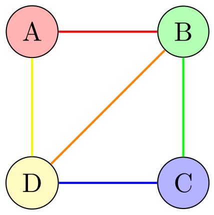
$$v_0, e_1, v_1, e_2, v_2, \ldots, v_{l-1}, e_l, v_l$$

consisting of vertices $v_0, v_1, v_2, \ldots, v_{l-1}, v_l$ and edges $e_1, e_2, e_3, \ldots, e_l$ such that each edge e_i is incident to the vertices v_{i-1} and v_i.

Generally, the e_i's and v_i's do not need to be distinct.

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

Example: Consider the following graph G:



A possible walk of length 4 in this graph could be:

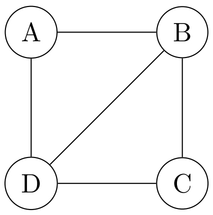
$$A, (A, B), B, (B, D), D, (D, A), A, (A, B), B$$

4.5.1.1 Types of Walks

1. Open Walk:
   - A walk where the first and last vertices are different.
   - It starts at one vertex and ends at another, allowing for the possibility of revisiting vertices and edges along the way.
2. Closed Walk:
   - A walk where the first and last vertices are the same.
   - It forms a loop, starting and ending at the same vertex, potentially revisiting other vertices and edges in between.

3. Trail:
    - A walk in which all edges are distinct.
    - It does not repeat any edges, although it may revisit vertices.
    - Trails can be open or closed.
4. Circuit:
    - A closed trail, meaning it is a trail that starts and ends at the same vertex.
    - It forms a closed loop, revisiting the starting vertex after traversing a sequence of edges and vertices.
    -

Example: Consider the graph G with vertices A, B, C, D and edges $(A, B), (B, C), (C, D), (D, A), (B, D)$.



- Open Walk: $A \to B \to D$
- Closed Walk: $A \to B \to C \to D \to A$
- Trail: $A \to B \to C \to D$
- Circuit: $A \to B \to D \to A$

4.5.2 Paths

Definition 14. *A **path** in a graph is a sequence of vertices where each adjacent pair of vertices is connected by an edge. In other words, a path is a walk in which all edges and all vertices are distinct.*

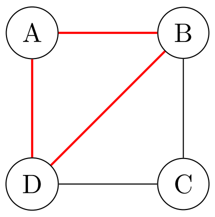
- Simple Path: A path that does not repeat any vertices (except possibly the first and last vertices if the path is a cycle).
- In an undirected graph, a path can traverse edges in any direction.
- In a directed graph (digraph), a path must follow the direction of the edges.

4.5.3 Cycle

Definition 15. *A **cycle** in a graph $G = (V, E)$ is a non-empty sequence of vertices $v_0, v_1, v_2, \ldots, v_k$ and edges $e_1, e_2, e_3, \ldots, e_k$ such that:*

1. *Each edge e_i is incident to the vertices v_{i-1} and v_i for $1 \leq i \leq k$.*
2. *The sequence forms a loop, i.e., $v_0 = v_k$, where $v_0, v_1, \ldots, v_k$ are distinct vertices (except $v_0 = v_k$).*

Example: Consider the following graph G with vertices A, B, C, D and edges $(A, B), (B, C), (C, D), (D, A), (B, D)$.

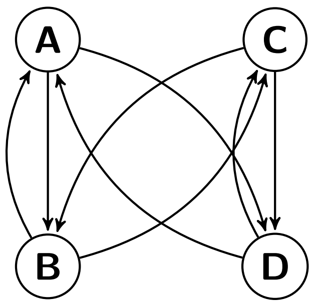


The cycle $A \to B \to D \to A$ forms a closed loop by starting and ending at vertex A.

Definition 16. *An undirected graph is called **connected** if there is a path between every pair of distinct vertices of the graph. An undirected graph that is not connected is called disconnected.*

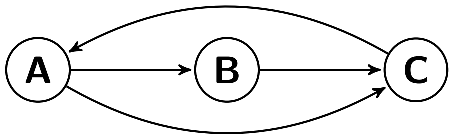
We say that we disconnect a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

Definition 17. *A directed graph is **strongly connected** if there is a path from a to b and from b to a whenever a and b are vertices in the graph.*



Definition 18. *A directed graph is **weakly connected** if there is a path between every two vertices in the underlying undirected graph.*

A directed graph is weakly connected if there's always a path between any two vertices, regardless of the direction of the edges. So, even if we ignore the directions of the edges, we can still find a path connecting any pair of vertices. It's evident that any strongly connected directed graph, where there's a directed path between every pair of vertices, also satisfies the condition for weak connectivity.



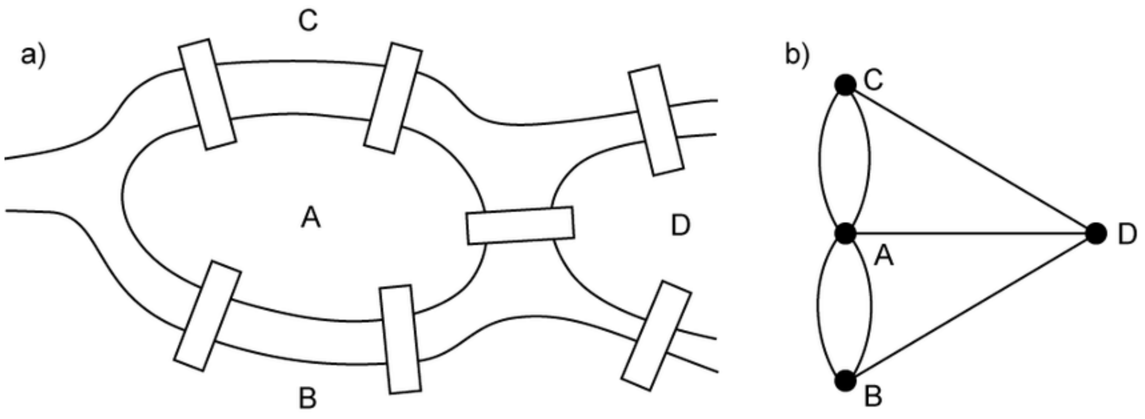
$$\begin{tikzcd} A \arrow[r] \arrow[rr, bend right] & B \arrow[r] & C \arrow[ll, bend right] \end{tikzcd}$$

4.6 Euler and Hamilton Paths

4.6.1 Euler Paths and Circuits

In Königsberg, Prussia (now Kaliningrad, Russia), the town was divided into four sections by the Pregel River branches. Seven bridges connected these areas in the 18th century. People wondered if they could walk through the town, crossing each bridge once without doubling back, and return to their starting point.

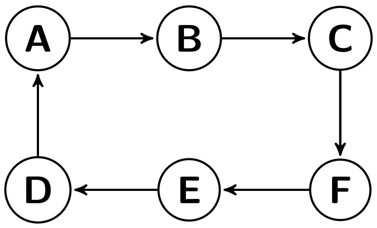
Leonhard Euler, a Swiss mathematician, tackled this problem in 1736, possibly marking the first use of graph theory. He represented the four regions as vertices and the bridges as edges in a multigraph. This approach simplified the problem, leading to Euler's solution.



The problem of traversing each bridge exactly once without doubling back can be restated using this model. It now asks: Is there a simple circuit in this multigraph that includes every edge?

Definition 19. *An Euler circuit in a graph G is a simple circuit containing every edge of G.*

Example: Consider the following graph G:

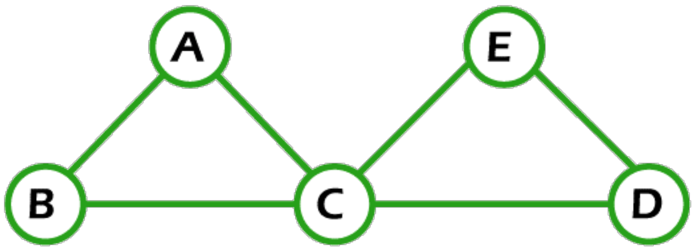


In this graph:

- An Euler circuit would be a simple circuit that starts and ends at the same vertex, visiting every edge exactly once. One example of an Euler circuit in this graph is $A \to B \to C \to F \to E \to D \to A$.
- An Euler path would be a simple path that visits every edge exactly once.

Definition 20. *An **Eulerian graph** is a graph that contains an Euler circuit, which is a simple circuit that includes every edge of the graph exactly once.*

For a graph to be Eulerian, it must be connected, and every vertex must have an even degree.



Example of Euler Graph

Theorem 3: A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has an even degree.

Example: Consider a graph $G = (V, E)$

$$V = \{1, 2, 3, 4, 5\}$$
$$E = \{(1, 2), (1, 3), (2, 3), (2, 4), (2, 5), (3, 4), (4, 5)\}$$

Determine whether G is Eulerian.

Solution: Let's calculate the degree of each vertex:

- $\deg(1) = 2$
- $\deg(2) = 4$
- $\deg(3) = 3$
- $\deg(4) = 3$
- $\deg(5) = 2$

In this graph, vertices 2, 3, and 4 have odd degrees.
Therefore, the graph G is not Eulerian because it does not satisfy the condition that all vertices must have even degree.

Exercise: Consider a graph $G = (V, E)$

$$V = \{1, 2, 3, 4, 5\}$$

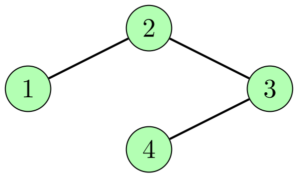
$$E = \{(1, 2), (1, 3), (2, 3), (2, 4), (2, 5), (4, 5)\}$$

Determine whether G is Eulerian.

4.6.2 Hamilton Paths and Circuits

Definition 21. *A* ***Hamilton path*** *in a graph $G = (V, E)$ is a simple path $x_0, x_1, \ldots, x_{n-1}, x_n$ such that $V = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ and $x_i \neq x_j$ for $0 \leq i < j \leq n$.*

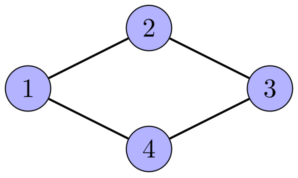
In other words, a Hamilton path is a path that visits every vertex in the graph exactly once.



In this example, the path $1 \to 2 \to 3 \to 4$ is a Hamiltonian path.

Definition 22. *A* ***Hamilton circuit*** *in a graph $G = (V, E)$ is a simple circuit $x_0, x_1, \ldots, x_{n-1}, x_n, x_0$ (with $n > 0$) such that $x_0, x_1, \ldots, x_{n-1}, x_n$ is a Hamilton path.*

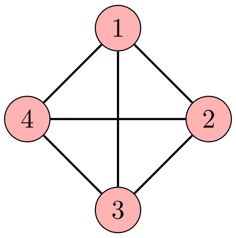
In other words, a Hamilton circuit is a circuit that starts and ends at the same vertex and visits every other vertex exactly once.



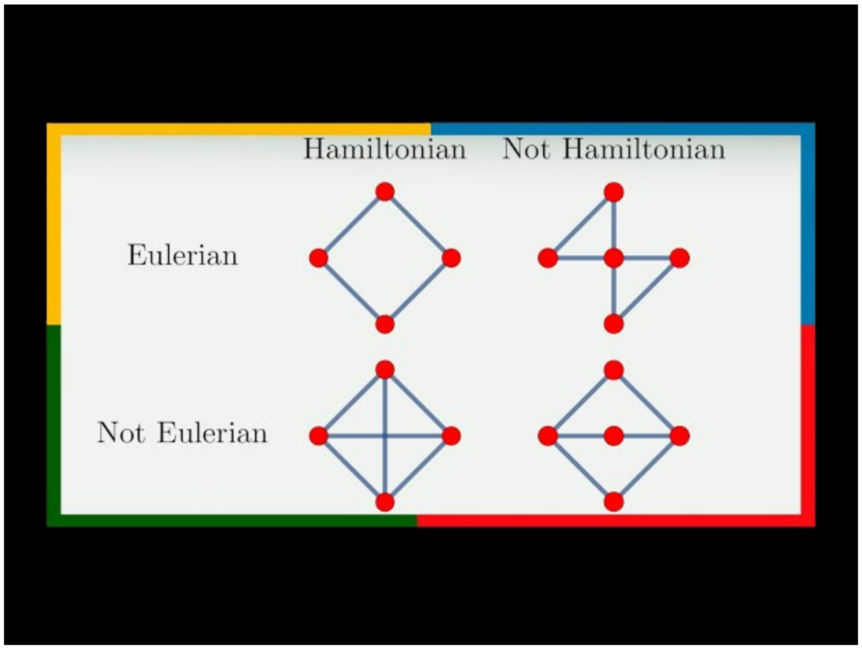
In this example, the circuit $1 \to 2 \to 3 \to 4 \to 1$ is a Hamiltonian circuit.

Definition 23. *A **Hamiltonian graph** is a graph $G = (V, E)$ in which there exists a Hamiltonian circuit, i.e., a simple circuit that passes through every vertex exactly once.*

In other words, a Hamiltonian graph is a graph that contains a Hamilton circuit.



In this example, the graph contains a Hamiltonian circuit $1 \to 2 \to 3 \to 4 \to 1$, making it a Hamiltonian graph.



Dirac's Theorem:

If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$, then G has a Hamiltonian circuit.

Ore's Theorem:

If G is a simple graph with n vertices with $n \geq 3$ such that $\deg(u) + \deg(v) \geq n$ for every pair of non-adjacent vertices u and v in G, then G has a Hamiltonian circuit.

Definition 24. *The **girth of a graph** $G = (V, E)$ is defined as the length of the smallest cycle, provided that the graph has at least one cycle. If the graph has no cycles, the girth of the graph is defined as ∞.*

Definition 25. *The **diameter of a graph** $G = (V, E)$, denoted by $diam(G)$, is defined as the length of the shortest path between the most distanced pair of vertices of G. That is,*

$$diam(G) = \max\{\ell(u, v) \mid u, v \in V\}$$

Example: consider a graph $G = (V, E)$

$$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$$
$$E = \{(1, 2), (1, 4), (1, 6), (2, 3), (2, 4), (2, 6), (2, 8), (3, 5), (3, 7), (3, 8), (4, 5), (4, 8), (5, 6), (5, 7), (6, 8)\}$$

1. Find a circuit in G which is not a cycle.

   A circuit in a graph that is not a cycle can be obtained by traversing some edges multiple times. Let's find such a circuit:

   Circuit: $1 \to 2 \to 4 \to 5 \to 6 \to 8 \to 2 \to 1$

2. Find a circuit in G which is also a cycle.

   A cycle is a circuit in which no vertex (other than the starting and ending vertices) is repeated. Let's find such a cycle:

   Cycle: $1 \to 2 \to 8 \to 6 \to 5 \to 4 \to 1$

3. What is the diameter of G?

   The diameter of a graph is the maximum shortest path distance between any pair of vertices. Let's calculate the shortest path distances between all pairs of vertices and then find the maximum:

   Shortest path distances:

	1	2	3	4	5	6	7	8
1	0	1	2	1	2	1	3	2
2	1	0	1	1	2	1	3	1
3	2	1	0	2	1	2	2	1
4	1	1	2	0	1	2	2	1
5	2	2	1	1	0	1	1	2
6	1	1	2	2	1	0	2	1
7	3	3	1	2	1	2	0	3
8	2	1	1	1	2	1	3	0

   The maximum shortest path distance is 3, so the diameter of G is 3.

4. What is the girth of the graph G?

   The girth of a graph is the length of the shortest cycle in the graph. $1 \to 2 \to 6 \to 1$, so the girth of G is 3.

5. Is the graph G Eulerian?

   For a graph to be Eulerian, all vertices must have even degree. Let's calculate the degree of each vertex:

Vertex	Degree
1	3
2	5
3	4
4	4
5	4
6	4
7	2
8	4

   Not all vertices have even degree, so the graph G is not Eulerian.

6. Find a Hamiltonian path of the graph G.

   A Hamiltonian path is a path that visits every vertex exactly once. Let's find one:

   Hamiltonian path: $6 \to 1 \to 2 \to 3 \to 8 \to 4 \to 5 \to 7$

4.7 Some Special Simple Graphs

4.7.1 Complete Graph

A **complete graph** on n vertices, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices.

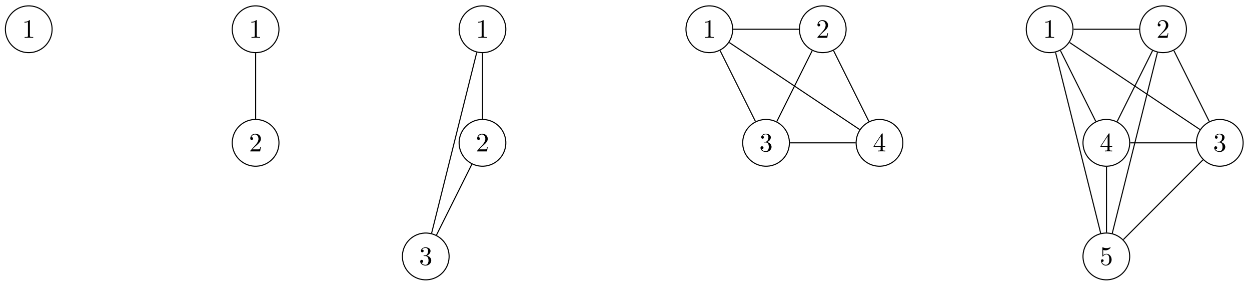


Figure 4.6: Complete graphs K_1, K_2, K_3, K_4, K_5

A simple graph for which there is at least one pair of distinct vertices not connected by an edge is called non-complete.

4.7.2 Cycle Graph

A graph is called a **cycle graph** if it consists only of a single cycle and all vertices are incident to exactly two edges of the cycle. A cycle containing n vertices is denoted by C_n.

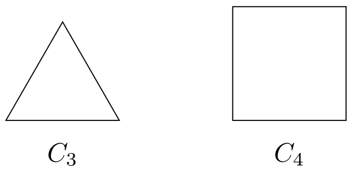
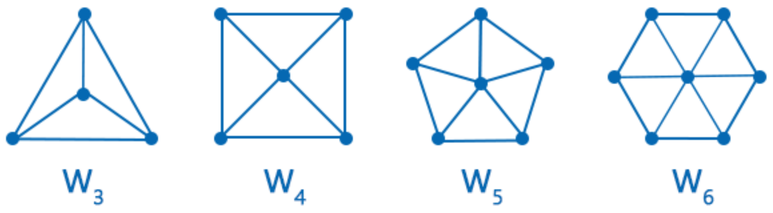


Figure 4.7: Cycle graphs C_3, and C_4

4.7.3 Wheel Graph

We obtain a **wheel graph** W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n by new edges.



4.7.4 Regular Graph

A graph is called **regular** if all of its vertices have the same degree. Specifically, a graph $G = (V, E)$ is k-regular if every vertex in G has degree k, where k is a non-negative integer.

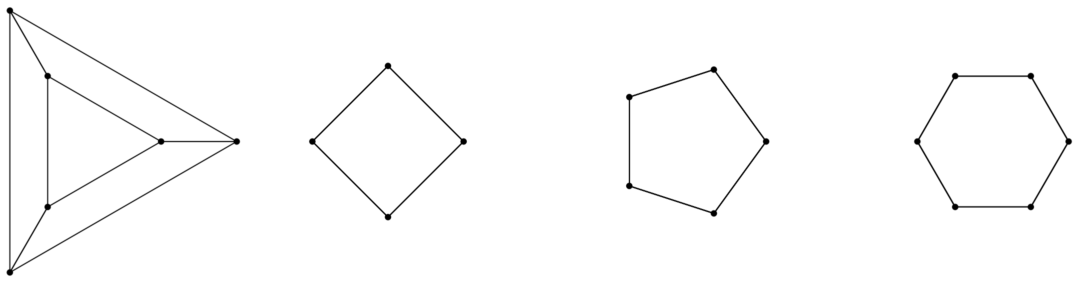


Figure 4.8: Regular graphs

4.7.5 Edgeless Graphs

An **edgeless graph**, also known as a **null graph** or an **empty graph**, is a graph with no edges and denoted by $(\overline{K_n})$, where n is the number of vertices.
It consists only of vertices and no connections between them.

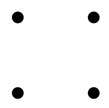


Figure 4.9: Edgeless graph with 4 vertices

4.7.6 Bipartite Graphs

A simple graph G is called **bipartite** if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2). When this condition holds, we call the pair (V_1, V_2) a **bipartition** of the vertex set V of G.

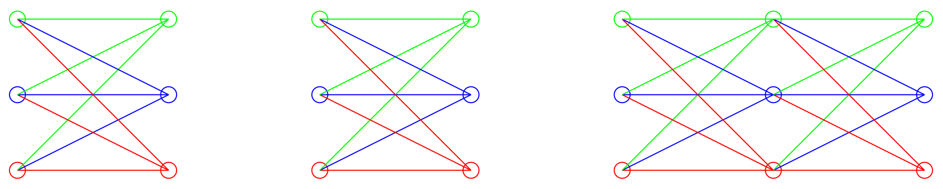
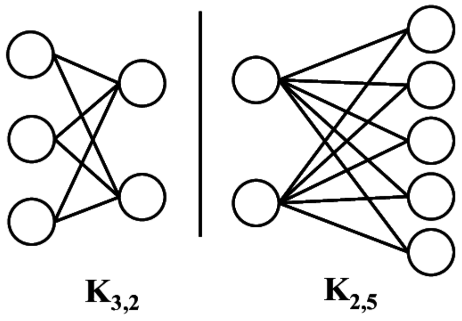


Figure 4.10: Examples of bipartite graphs

4.7.7 Complete Bipartite Graphs

A **complete bipartite graph** $K_{m,n}$ is a graph in which the vertex set is divided into two disjoint subsets containing m and n vertices, respectively, such that *every vertex in the first subset is connected to every vertex in the second subset*, and *no edges exist between vertices within the same subset.* The graph has a total of $m + n$ vertices and $m \cdot n$ edges.



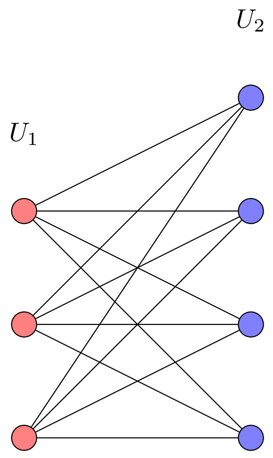


Figure 4.11: Complete bipartite graph $K_{3,4}$

Bipartite graphs can be used to model many types of applications that involve matching the elements of one set to elements of another.

Job Assignments:

Suppose there are m employees and n different jobs, where $n \geq m$. Each employee is trained to do one or more of these n jobs. We want to assign an employee to each job.

To model this, we use a bipartite graph:

- Represent each employee by a vertex in set E.
- Represent each job by a vertex in set J.
- Draw an edge from an employee to a job if the employee is trained to do that job.

Thus, the bipartite graph has a vertex set partitioned into two disjoint sets: E (employees) and J (jobs), where each edge connects a vertex in E to a vertex in J.

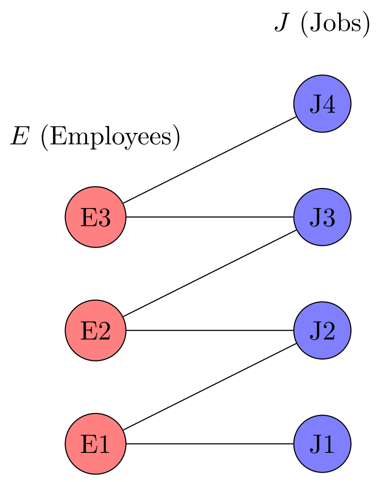


Figure 4.12: Bipartite graph for job assignments

Marriages on an Island:

Suppose there are m men and n women on an island. Each person has a list of members of the opposite gender acceptable as a spouse. We construct a bipartite graph $G = (V_1, V_2)$, where V_1 is the set of men and V_2 is the set of women. There is an edge between a man and a woman if they find each other acceptable as a spouse.

A matching in this graph consists of a set of edges, where each pair of endpoints of an edge is a husband-wife pair. A maximum matching is the largest possible set of married couples, and a complete matching of V_1 is a set of married couples where every man is married, but possibly not all women.

4.8 Trees and Forests

Definition 26. *A* ***tree*** *is a connected, undirected, and acyclic graph.*

This means that a tree has the following properties:

1. It has no cycles, and
2. There is exactly one simple path between any pair of vertices.

Since a tree must be connected and acyclic, it naturally adheres to the restrictions of a simple graph. Therefore, any tree is inherently a simple graph.

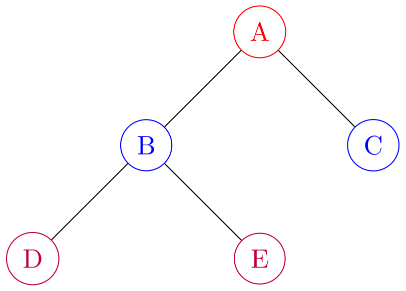


Figure 4.13: A simple tree

In a tree $G = (V, E)$, the number of edges $|E(G)|$ is one less than the number of vertices $|V(G)|$. Mathematically, this can be expressed as:

$$|E(G)| = |V(G)| - 1$$

Example:

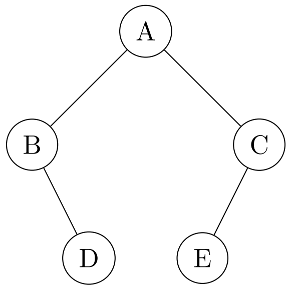
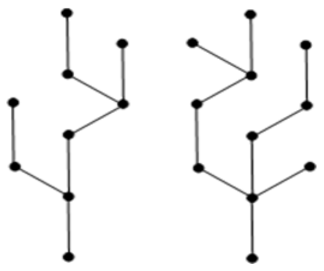


Figure 4.14: Example of a tree with 5 vertices and 4 edges

In this tree, there are $|V(G)| = 5$ vertices and $|E(G)| = 4$ edges. Following the property $|E(G)| = |V(G)| - 1$, we have $4 = 5 - 1$, which holds true.

Definition 27. *A* ***forest*** *is an undirected graph whose connected components are all trees.*

Every tree is a forest, but not every forest is a tree. A forest with only one tree is itself a tree.



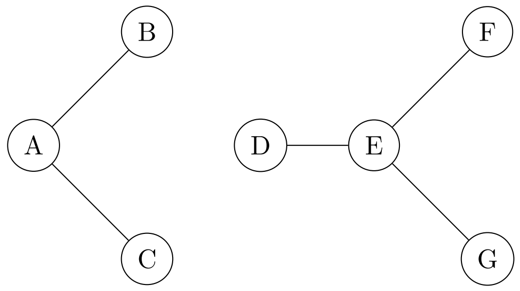


Figure 4.15: Example of a forest but not a tree

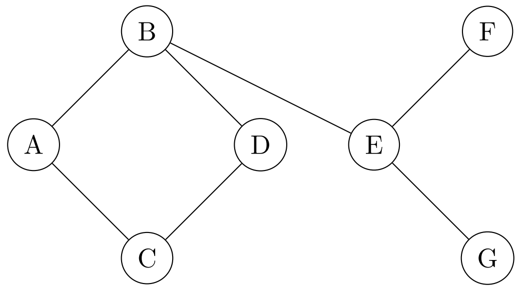


Figure 4.16: Example of not a tree and not a forest

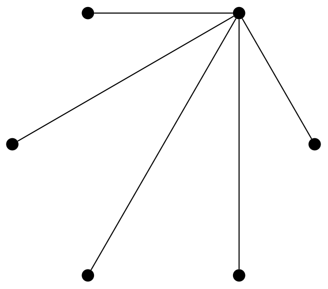


Figure 4.17: Example of both a tree and a forest

4.8.1 Binary Trees

Definition 28. *A* ***binary tree*** *is a tree with exactly one vertex of degree two, such that all the other vertices have degree one or three.*

- Root Node: The topmost node of the binary tree.
- Leaf Nodes: Nodes that have no children. They are the nodes with degree one.
- Trivalent Vertices: Nodes that have exactly two children. They are the nodes with degree three.

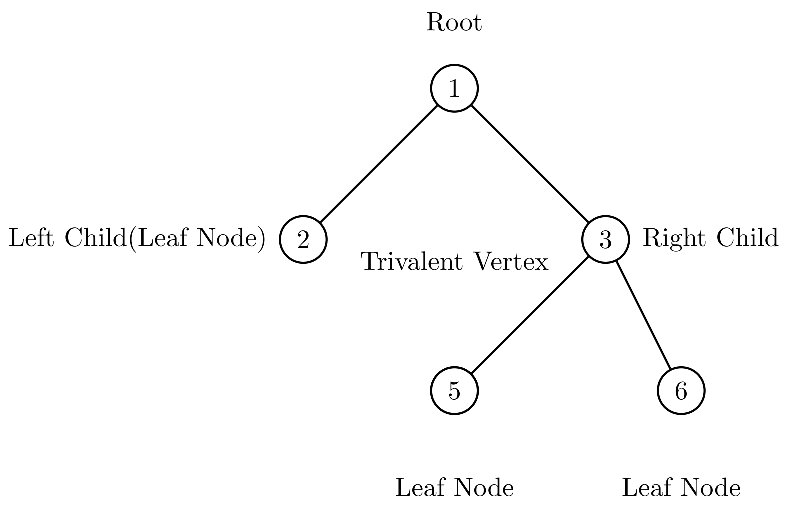


Figure 4.18: Binary tree

4.8.2 Real-world Examples of Trees and Forests

Concept	**Real-world Examples and Descriptions**
Tree	• **Computer File System:** Root directory as the top node; subfolders and files branch out hierarchically. • **Organization Hierarchy:** CEO as the root, managers as intermediate nodes, and employees as leaves. • **Family Tree:** Represents parent-child relationships between ancestors and descendants. • **Decision Tree:** Used in data science for classification and decision-making processes. • **Network Spanning Tree:** Ensures loop-free network topology in communication systems.
Forest	• **Multiple Hierarchies:** Several companies or departments, each forming its own tree structure. • **Random Forests:** Multiple independent decision trees used together in machine learning. • **Independent Networks:** Disconnected subnetworks, each forming a tree topology.

4.9 Graph Isomorphism

Two graphs are said to be equal if they have the exact same distinct elements, but sometimes two graphs can "appear equal" even if they aren't, and that is the idea behind isomorphisms.

Graph isomorphism is a fundamental concept in graph theory. It describes a scenario where two graphs are structurally identical, meaning there is a one-to-one correspondence between their vertex sets and edge sets that preserves adjacency.

Definition 29. *Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be isomorphic if there exists a bijection $f : V_1 \to V_2$ such that $\{u, v\} \in E_1$ if and only if $\{f(u), f(v)\} \in E_2$.*

In other words, the bijective function f maps vertices of G_1 to vertices of G_2 in such a way that the edge structure is preserved.

Isomorphic graphs have the **same structure** — they may appear different but are actually the same graph with renamed vertices.

Properties of Isomorphic Graphs

If two graphs G_1 and G_2 are isomorphic, then:

- They have the same number of vertices.
- They have the same number of edges.
- They have the same degree sequence.
- They have the same number of connected components.
- They have the same number of cycles of each length.

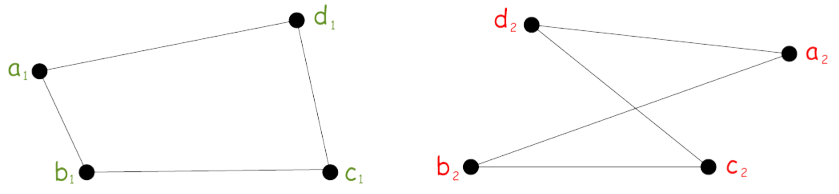
4.9.1 Method One – Checklist

To determine if two graphs G_1 and G_2 are isomorphic, follow these steps:

1. **Check Number of Vertices:** If the graphs do not have the same number of vertices, they are not isomorphic.
2. **Check Number of Edges:** If the number of edges differs, the graphs cannot be isomorphic.
3. **Compare Degree Sequences:** List and sort the degrees of all vertices in each graph. If the sequences are different, the graphs are not isomorphic.
4. **Compare Cycles of Length k:** Count cycles of length 3, 4, etc., in both graphs. If the counts differ, the graphs are not isomorphic.
5. **Find a One-to-One Correspondence of Vertices:** Attempt to map each vertex $u \in G_1$ to a vertex $f(u) \in G_2$ such that adjacency is preserved. That is, for every edge $(u, v) \in G_1$, the edge $(f(u), f(v))$ must exist in G_2.

- If such a mapping exists, the graphs are **isomorphic**.
- If no such mapping exists, the graphs are **not isomorphic**.

Example:



All we have to do is ask the following questions:

1. Are the number of vertices in both graphs the same? **Yes**, both graphs have 4 vertices.
2. Are the number of edges in both graphs the same? **Yes**, both graphs have 4 edges.
3. Is the degree sequence in both graphs the same? **Yes**, each vertex in both graphs has degree 2.
4. Do both graphs contain cycles of the same length? **Yes**, each graph contains a cycle of length 4.
5. Can we find a one-to-one correspondence of vertices that preserves adjacency? **Yes**, by mapping the vertices of G_1 to G_2 as follows:

$$a_1 \to a_2, \quad b_1 \to b_2, \quad c_1 \to c_2, \quad d_1 \to d_2$$

   all edges are preserved under this mapping.

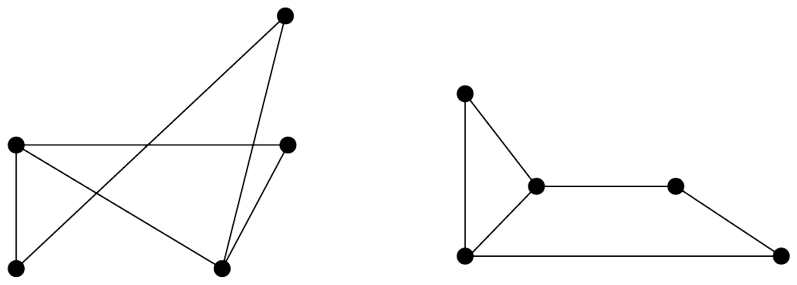
Since all necessary conditions are satisfied and a one-to-one correspondence of vertices that preserves adjacency exists, we conclude that G_1 and G_2 are **isomorphic**.

4.9.2 Method Two – Relabeling

In addition to counting vertices, edges, degrees, and cycles, there is another easy way to verify an isomorphism between two simple graphs: relabeling.

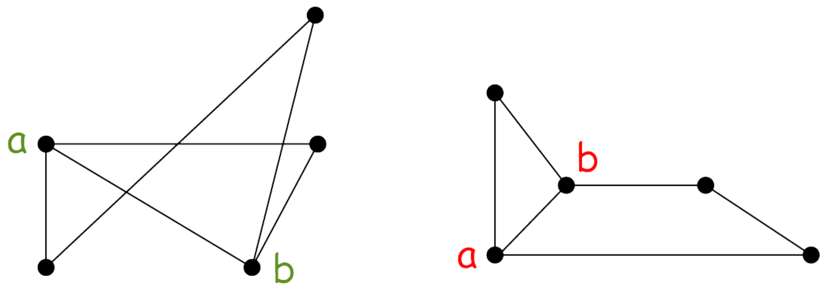
1. **Identify Vertices and Degrees:** Start with the vertices with the highest degree and label them.
2. **Label Adjacent Vertices:** Identify vertices that are adjacent to the already labeled vertices.
3. **Continue Labeling:** Continue this process until all vertices are labeled.
4. **Define Isomorphism:** Verify the one-to-one correspondence between the vertices of the two graphs.

Example:

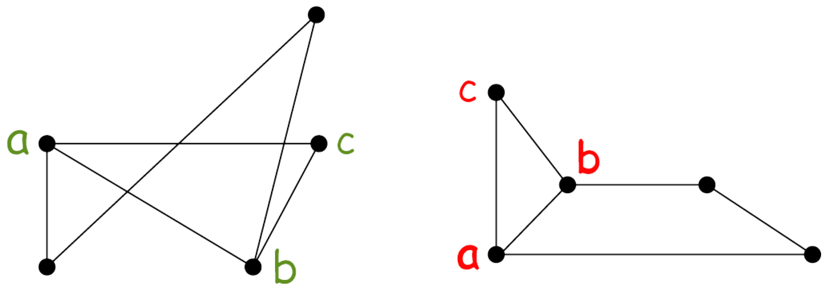


- Both graphs have 5 vertices.
- Degree sequence (in ascending order): $(2, 2, 2, 3, 3)$.

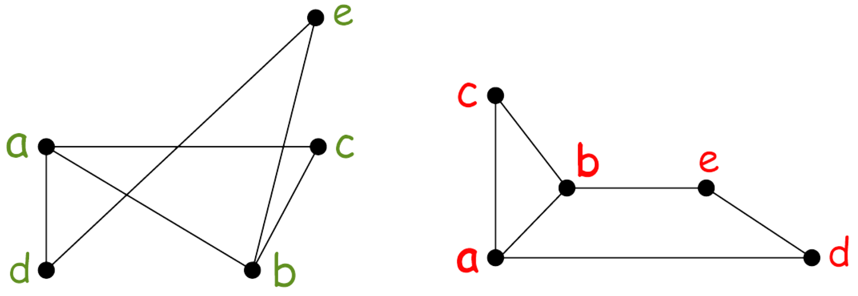
Now we methodically start labeling vertices by beginning with the vertices of degree 3 and marking a and b.



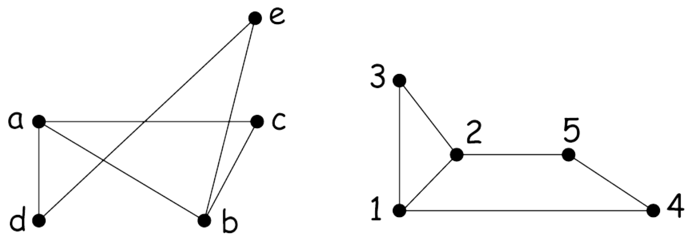
Next, we notice that in both graphs, there is a vertex that is adjacent to both a and b, so we label this vertex c in both graphs.



This now follows that there are two vertices left, and we label them according to d and e, where d is adjacent to a and e is adjacent to b.

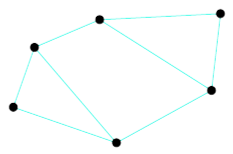
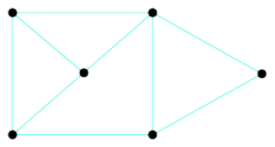


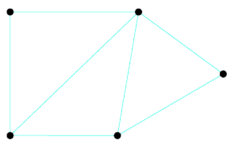
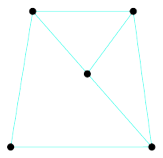
And finally, we define our isomorphism by relabeling each graph and verifying one-to-correspondence.



$$a = f(1)$$
$$b = f(2)$$
$$c = f(3)$$
$$d = f(4)$$
$$e = f(5)$$

For the following two examples, you will see that the degree sequence is the best way for us to determine if two graphs are isomorphic.

Graph G	Graph H	Degree Sequence
		G: (2,2,3,3,3,3) H: (2,3,3,3,3,3) $G \ncong H$

Graph J	Graph K	Degree Sequence
		J: (2,2,3,3,4) K: (2,3,3,3,3) $J \ncong K$

Real-world Analogies of Graph Isomorphism

Table 4.1: Examples of Graph Isomorphism in Real-world Contexts

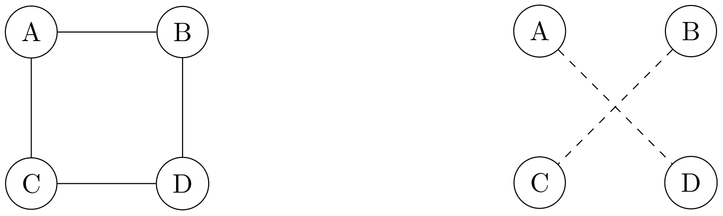
Scenario	**Description**
Network Renaming	Two computer networks have identical connections, but the devices (vertices) have different names.
Social Networks	Two friendship networks with the same pattern of friendships, but different user IDs.
Circuit Design	Two circuits with identical layouts of connections, but components are labeled differently.
Molecular Structures	Two molecules having identical bonding patterns (same topology) but atoms named differently.
Transportation Maps	Two metro maps representing the same routes and connections but with different station names.

4.10 Complement of a Graph

Definition 30. *The* ***complement of a graph*** G*, denoted as* $\overline{G}$*, is a graph that contains exactly the same set of vertices as* G*, but its edges connect pairs of vertices that are not adjacent in* G*, and vice versa.*

In other words, if there is an edge between two vertices in G, there is no edge between them in $\overline{G}$, and if there is no edge between two vertices in G, there is an edge between them in $\overline{G}$.

Formally, if $G = (V, E)$ is a graph, then its complement $\overline{G} = (V, \overline{E})$, where $\overline{E}$ consists of all possible edges between vertices in V that are not already in E.



(a) Graph G (b) Complement $\overline{G}$

Figure 4.19: Graph G and its complement $\overline{G}$

4.11 Planar Graphs

Definition 31. *A graph is called* ***planar*** *if it can be drawn in the plane without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a* ***planar representation*** *of the graph.*

A graph may be planar even if it is usually drawn with crossings, because it may be possible to draw it in a different way without crossings.

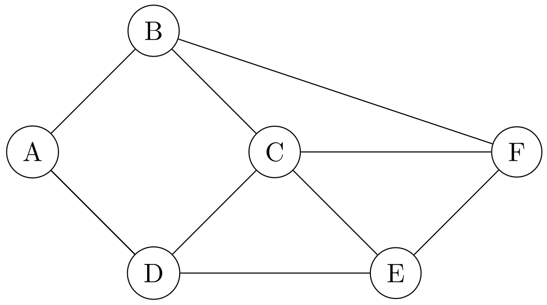


Figure 4.20: Planar Graph G

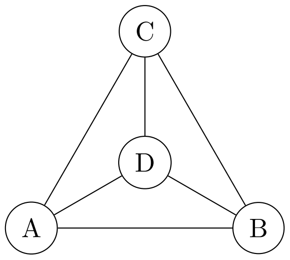


Figure 4.21: Planar Representation of K_4

4.12 Graph Coloring

The map coloring problem in graph theory involves coloring the regions of a map in such a way that no two adjacent regions have the same color while using the fewest number of colors possible.

To solve this problem, we represent each region of the map as a vertex in a graph. If two regions share a common border, we connect the corresponding vertices with an edge. This graph is called the ”dual graph” of the map.

The objective is to find the smallest number of colors needed to color the vertices of this dual graph such that no adjacent vertices have the same color. This number is known as the chromatic number.

Despite being challenging to solve in general, there are efficient algorithms for special cases, such as planar graphs. The map coloring problem has practical applications in areas like cartography, scheduling, and register allocation in computer science.

Definition 32. *A coloring of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.*

4.12.1 Steps for Graph Coloring

1. **Identify Vertices and Edges:** Begin by identifying all the vertices and edges in the graph.
2. **Choose a Color:** Start with the first vertex and assign it the first color.
3. **Color Adjacent Vertices:** Move to the next vertex. If it is adjacent to any previously colored vertices, choose a different color.
4. **Continue Coloring:** Repeat the process for all vertices, ensuring that no two adjacent vertices share the same color.

Definition 33. *The* ***chromatic number*** *of a graph is the least number of colors needed for a coloring of this graph. The chromatic number of a graph G is denoted by $\chi(G)$. (Here χ is the Greek letter chi.)*

Example:

Try coloring the following graph:

$$V = \{A, B, C, D, E\}$$
$$E = \{(A, B), (A, C), (B, C), (B, D), (C, D), (D, E)\}$$

Follow the steps outlined above to find the chromatic number and color the graph.

By using three different colors, we ensured that no two adjacent vertices share the same color. Thus, the chromatic number for this graph is 3.

The Four Color Theorem:
The chromatic number of a planar graph is no greater than four.

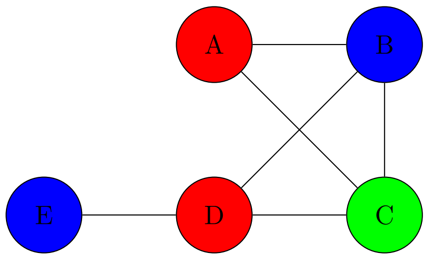


Figure 4.22: Graph Coloring Example

The Four Color Theorem states that any planar graph (a graph that can be drawn on a plane without edges crossing) can be colored with at most four colors.

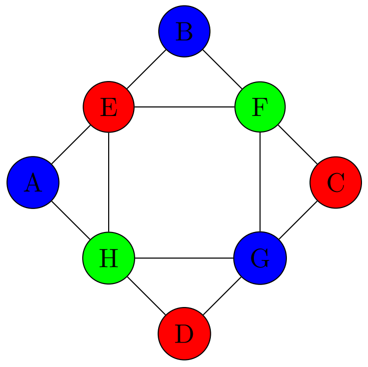


Figure 4.23: A Planar Graph Colored with Three Colors

4.12.2 Applications of Graph Coloring

- **Map Coloring:** Ensuring that no two adjacent regions on a map share the same color.
  Consider a map of a country where the regions represent different administrative divisions, such as states or provinces. The goal is to color the map in such a way that no two adjacent regions have the same color, using as few colors as possible.

  According to the Four Color Theorem, this can be achieved with no more than four colors for any such map. In other words, no matter how complex the map is, you can always color it with just four colors such that no adjacent regions share the same color.

  For instance, if we have a map with regions representing different states, and two states share a border, we must use different colors to distinguish them. By following the Four Color Theorem, we can be certain that we can color the entire map using only four colors, ensuring that no adjacent states have the same color. This theorem has practical implications in cartography, urban planning, and other fields where maps are used to represent spatial relationships. It demonstrates that, despite the potentially complex nature of maps, a relatively small number of colors is sufficient to represent them accurately.

- **Scheduling:** Assigning time slots to tasks or exams so that no two overlapping tasks share the same slot.
  As an example, imagine that your school needs to schedule eight exams. The exams can't all take place at the same time because some students need to take several of them, and those students obviously can't be in two places at once. How many time slots do you need for scheduling the exams in order to avoid clashes?

- **Register Allocation:** In compiler design, assigning variables to a limited number of CPU registers.

Definition 34. ***Edge coloring*** *involves coloring the edges of a graph so that no two edges that share the same vertex have the same color. The chromatic index* $\chi'(G)$ *is the minimum number of colors needed for edge coloring.*

Example

Consider the following bipartite graph $K_{3,3}$ with 6 vertices:

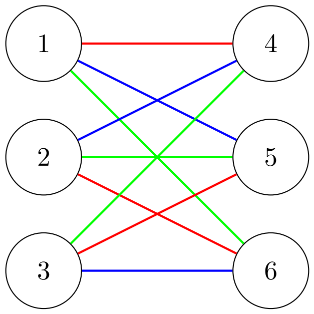


Figure 4.24: Edge Coloring Example for $K_{3,3}$

In this example, we use three colors (red, blue, and green) to color the edges of the bipartite graph $K_{3,3}$. The chromatic index $\chi'(G)$ for this graph is 3 because we need at least 3 colors to ensure that no two edges sharing the same vertex have the same color.