School of Computing, Engineering & Built Environment

Mathematics Summer School

Level 3 Entry – Computing

An Introduction to Graph Theory

Dr Calum A. Macdonald

Table of Contents

INTRODUCTION TO GRAPH THEORY

1. Introduction

In recent years graph theory has become established as an important area of mathematics and computer science. The origins of graph theory however can be traced back to Swiss mathematician Leonhard Euler and his work on the Königsberg bridges problem (1735), shown schematically in Figure 1.

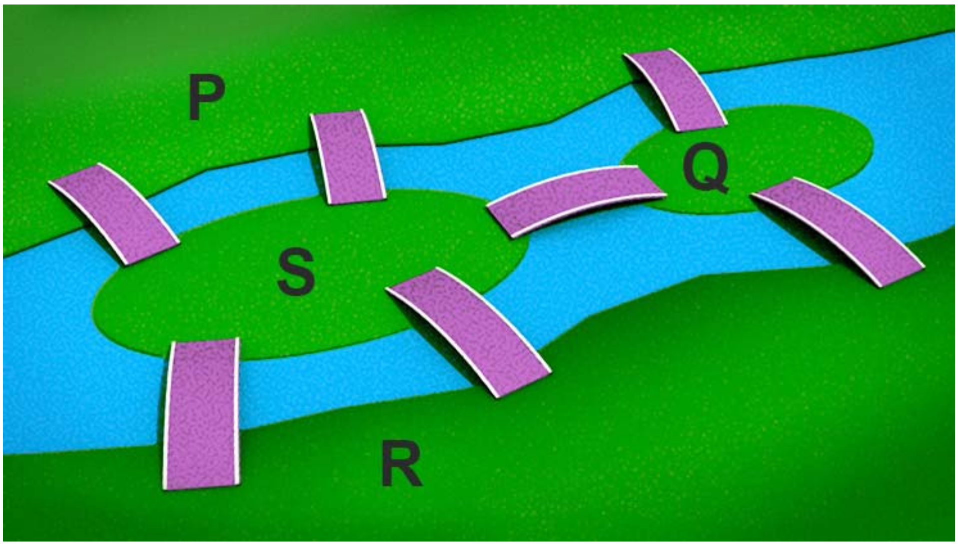


Figure 1: Bridges of Königsberg

Königsberg was a city in 18th century Germany (it is now called Kaliningrad and is in western Russia) through which the river Pregel flowed. The city was built on both banks of the river and on two large islands in the middle of the river. Seven bridges were constructed so that the city's inhabitants could travel between the four parts of the city; labelled P, Q, R and S in the diagram. The people wondered whether or not it was possible for someone to walk around the city in such a way that each bridge was crossed **exactly once** with the person ending up at their starting point. All attempts to do so ended in failure. In 1735 however Euler presented a solution to the problem by showing that it was impossible to perform such a journey. Euler reasoned that anyone standing on a land mass would need a way to get on and off. Therefore each land mass would require an even number of bridges. In Königsberg each land mass had an odd number of bridges explaining why all seven bridges could not be crossed without crossing one more than once. In formulating his solution Euler simplified the bridge problem by representing each land mass as a point and each bridge as a line as shown in Figure 2, leading to the introduction of graph theory and the concept of an Eulerian graph. A closely related problem showed that if the journey started at one land mass and ended at another, crossing each bridge exactly once, then only those two land masses could have an odd number of bridges.

Two other well-known problems from graph theory are:

- **Graph Colouring Problem**: How many colours do we need to colour a map so that every pair of countries with a shared border have different colours?

- **Travelling Salesman Problem**: Given a map of several cities and the roads between them, is it possible for a travelling salesman to visit (pass through) each of the cities exactly once?

Some of the applications of graph theory include:

- communication network design
- planning airline flight routes
- using GPS to find the shortest path between two points
- design of electrical circuits
- modelling of the Worldwide Web.

2. Definitions

The Königsberg bridge problem can be represented diagramatically by means of a set of points and lines. The points P, Q, R and S are called **vertices**, the lines are called **edges** and the whole diagram is called a **graph**.

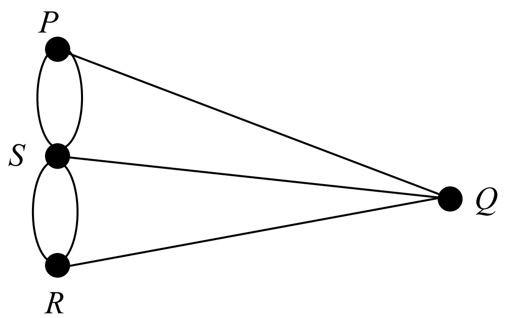


Figure 2: Graphical representation of the Königsberg bridge problem

2.1. Vertices and Edges

A graph, *G*, is a mathematical structure which consists of:

(i). a vertex set $V = V(G)$ whose elements are called vertices of *G*.

(ii). an edge set $E = E(G)$ of unordered pairs of distinct vertices called edges of *G*. Note that *E* is actually a **multiset** in that some unordered pairs can be repeated to represent more than one edge joining two vertices.

(iii). a relation that associates with each edge two vertices, which are not necessarily distinct, called its **endpoints**.

Such a graph is denoted $G = \{V(G),\ E(G)\}$, or just simply $G = \{V,\ E\}$.

Example 1

Consider the graph *G* shown in the diagram below.

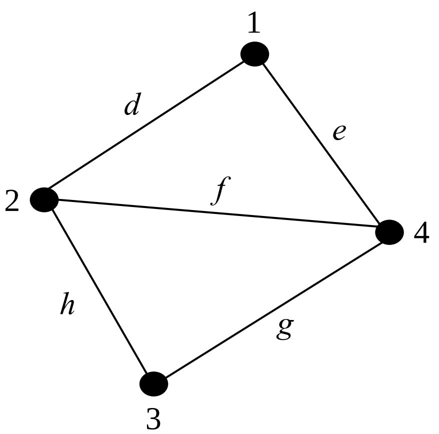


Figure 3: Graph with four vertices and five edges

The set *V* consists of the four vertices, 1, 2, 3 and 4, i.e. $V(G) = \{1,\ 2,\ 3,\ 4\}$.

The set *E* consists of the five edges, $d = \{1,\ 2\}$, $e = \{1,\ 4\}$, $f = \{2,\ 4\}$, $g = \{3,\ 4\}$ and $h = \{2,\ 3\}$, i.e. $E(G) = \{d,\ e,\ f,\ g,\ h\}$.

Hence, $G = \{V(G),\ E(G)\} = \{\{1,\ 2,\ 3,\ 4\},\ \{d,\ e,\ f,\ g,\ h\}\}$

Each edge is associated with two vertices called its **endpoints**.

For example, in Figure 3, vertices 1 and 2 are the **endpoints** of *d* and *d* is said to **connect** vertices 1 and 2.

An **edge-endpoint function** on a graph *G* defines a correspondence between edges and their endpoints.

<u>Example 2</u>

The edge-endpoint function for the graph in Figure 3 is given in the following table:

Edge	**Endpoints**
d	$\{1,\ 2\}$
e	$\{1,\ 4\}$
f	$\{2,\ 4\}$
g	$\{3,\ 4\}$
h	$\{2,\ 3\}$

An **undirected graph** is a graph in which the edges have no orientation. Hence, in an undirected graph the edge set is composed of unordered vertex pairs. In Figure 3 for example, the edge $\{1,\ 2\}$ is considered identical to the edge $\{2,\ 1\}$.

If *X* and *Y* are vertices of a graph, *G*, then *X* and *Y* are said to be **adjacent** if they are joined by an edge.

An edge in a graph that joins two vertices is said to be **incident** to both vertices.

<u>Example 3</u>

Referring to Figure 3, and the edge-endpoint table, we have the following adjacent vertices:

- vertices 1 and 2 are adjacent
- vertices 1 and 4 are adjacent
- vertices 2 and 3 are adjacent.
- vertices 2 and 4 are adjacent
- vertices 3 and 4 are adjacent

The edges that are incident with pairs of vertices as follows:

- edge d is incident to vertices 1 and 2
- edge e is incident to vertices 1 and 4.
- edge f is incident to vertices 2 and 4
- edge g is incident to vertices 3 and 4
- edge h is incident to vertices 2 and 3.

Two edges connecting the same vertices are called **multiple** or **parallel edges**.
In Figure 4 edges f and g are parallel edges.

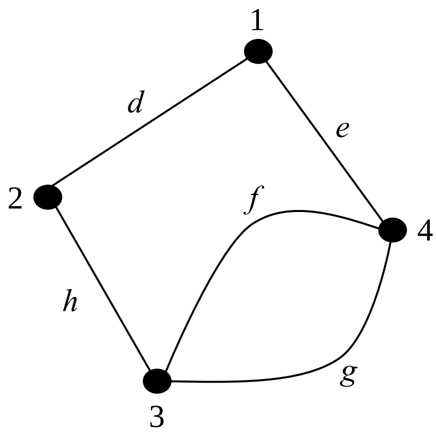


Figure 4: Graph containing parallel edges

Graphs like the one shown here, containing parallel edges, are called **multigraphs** and we shall look at these in more detail later in the unit.

The **order** of a graph, G, denoted $|V(G)|$, is the number of vertices contained in G.
In Figure 3, $|V(G)| = 4$.

The **size** of a graph, G, denoted $|E(G)|$, is the number of edges contained in G.
In Figure 3, $|E(G)| = 5$.

The **degree** of a vertex *X*, written $\deg(X)$, is the number of edges in *G* that are incident with *X*.

In Figure 3, $\deg(1) = 2$, $\deg(2) = 3$, $\deg(3) = 2$ and $\deg(4) = 3$.

Any vertex of degree zero is called an **isolated** vertex and a vertex of degree one is an **end-vertex**.

<u>Example 4</u>

In the graph below vertex 5 is an isolated vertex and vertex 3 is an end-vertex.

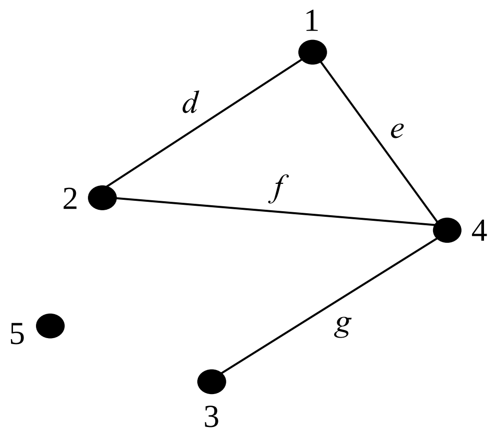


Figure 5: Graph containing an isolated vertex

A vertex is said to be **even** or **odd** according to whether its degree is an even or odd number. In Figure 3 vertices 2 and 4 are odd while vertices 1 and 3 are even.

If the degrees of all the vertices in a graph, *G*, are summed then the result is an even number. Furthermore, this value is twice the number of edges, as each edge contributes 2 to the total degree sum. We have the following lemma:

2.1.1. The Handshaking Lemma

In any undirected graph the sum of the vertex degrees is equal to twice the number of edges, i.e.

$$\sum_{X \in V(G)} \deg(X) = 2\,|\,E(G)\,|$$

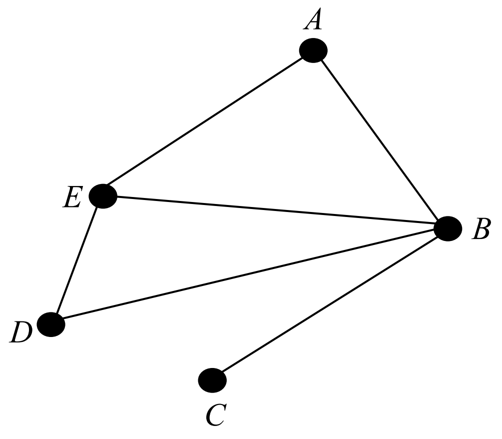
Proof: In a graph G an arbitrary edge $\{X, Y\}$ contributes 1 to $\deg(X)$ and 1 to $\deg(Y)$. Hence the degree sum for the graph is even and twice the number of edges.

Note: A corollary of the Handshaking Lemma states that the number of odd vertices in a graph must be even. So, for example, we cannot have a graph with 5 even vertices and 5 odd vertices as the degree sum would be an odd number, contradicting the Handshaking Lemma.

The **degree sequence** of an undirected graph G is a bracketed list of the degrees of all the vertices written in **ascending** order with repetition as necessary.

Example 5

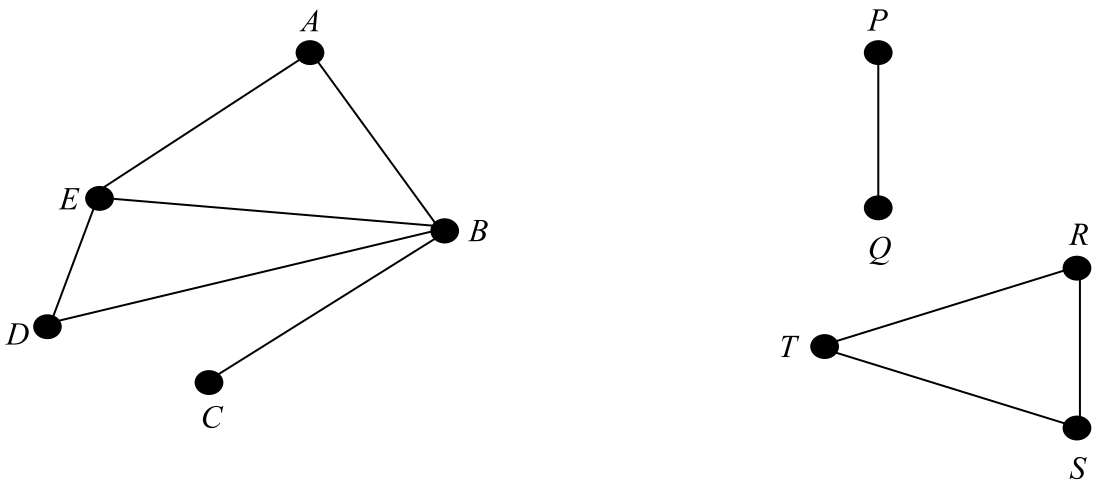
The degree sequence of the graph in the diagram below is (1, 2, 2, 3, 4).



Note that some texts define the degree sequence of a graph as the degrees of the vertices written in **descending** order with repetition as necessary. In the above case we would have (4, 3, 2, 2, 1).

2.2. Connected Graphs

A graph is said to be **connected** if it cannot be expressed as the union of two graphs. If a graph is not connected it is said to be **disconnected**. The graph on the left is connected as it is "in one piece" while the graph on the right is disconnected as it contains two distinct **components**. See Section 4 for an alternative definition of connected.

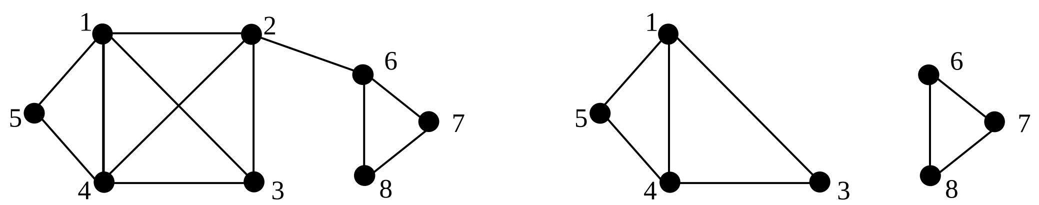


2.2.1. Cut-Points and Bridges

A vertex is a **cut-vertex**, if removal of that vertex (and the edges through it) disconnects the graph. A cut-vertex is also called a **cut-point** or an **articulation point**.

Example 6

In the graph on the left vertex 2 is a **cut-vertex** as its removal disconnects the graph. The resulting graph, on the right, has two **connected components**. Vertex 6 is also a cut-vertex.



An edge is a **bridge (or isthmus)** if removal of that edge disconnects the graph.

Here edge $\{2, 6\}$ is a bridge as its removal disconnects the graph.

3.Graph Structures

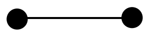
In this section we briefly look at different types of graphs.

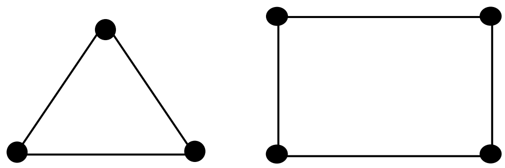
3.1. Regular Graphs

A graph G is **regular** if all vertices of G have the same degree. A regular graph where all vertices have degree k is referred to as a k-regular graph.

Example 7

0-regular: 

1-regular: 

2-regular: 

The graph on the left is called a 2-regular graph on 3 vertices and the one on the right is a 2-regular graph on 4 vertices. **Exercise**: Sketch a 2-regular graph on 5 vertices.

Notes

(i). The Handshaking Lemma tells us that the total degree of any graph is an even number, i.e. twice the number of edges. Hence, it is impossible to construct a k-regular graph, where k is odd, on an odd number of vertices. For example, we cannot have a 3-regular graph on 5 vertices as this would give a degree sum of 15, violating the Handshaking Lemma.

(ii). A 0-regular graph is called an **empty graph**.

(iii). **Cycle graphs** (see Section 3.3) are 2-regular graphs.

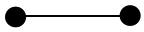
(iv). 3-regular graphs are called a **cubic graphs**.
There is only one 3-regular graph on 4 vertices. Can you sketch it?
There are two 3-regular graphs on 6 vertices. Can you sketch them?
There is no 3-regular graphs on 7 vertices. Why?

3.2. Complete Graphs

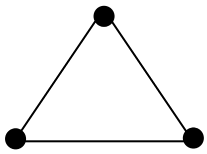
A **complete graph**, denoted K_n, is a graph with *n* vertices all of which are adjacent to each other.

K_1

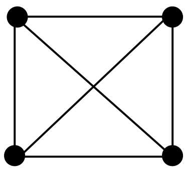
K_2



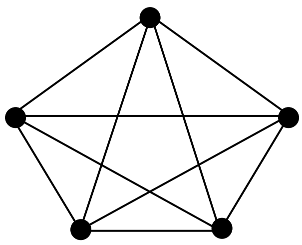
K_3



K_4



K_5



The complete graph K_n is regular and each of the *n* vertices has degree $n - 1$. Hence, the sum of the degrees is $n(n-1)$ and by the Handshaking Lemma the number of edges in K_n is, $\dfrac{n(n-1)}{2}$

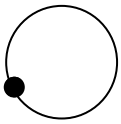
Exercise: Check that the two properties stated above hold for the complete graphs shown.

3.3. Cycle Graph

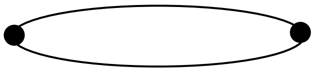
A **cycle** on a graph starts at any vertex, travelling through the graph without repeating vertices or edges before ending on the start vertex. In Example 5, *BAEB* and *AEDBA* are both cycles while *AEDBEA* is not a cycle as the vertex *E* is repeated.

A **cycle graph**, denoted C_n, is a graph on n vertices, $\{v_0, v_1, \ldots, v_{n-1}\}$, with n edges $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{n-1}, v_0\}$. Note that C_n contains a single **cycle** through all the vertices.

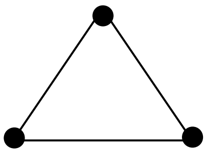
C_1



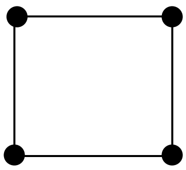
C_2



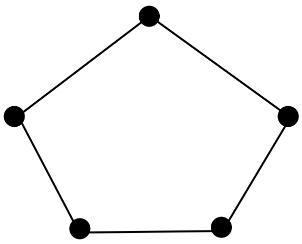
C_3



C_4



C_5



Notes

(i). In a cycle graph every vertex has degree 2.

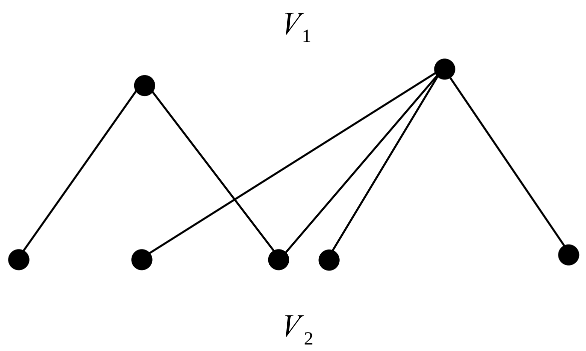
(ii). The graph C_1 contains a **self-loop** and we shall see later that a loop contributes two to the degree of the vertex. Hence, the vertex in C_1 has degree 2 ensuring that the Handshaking Lemma holds.

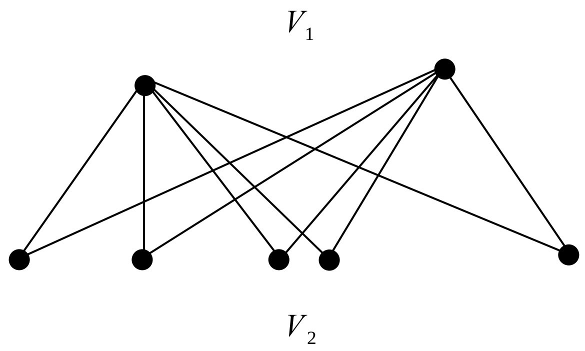
(iii). The graph C_2 contains **parallel edges**.

A graph that contains no loops or parallel edges is called a **simple graph**.

3.4. Bipartite Graphs

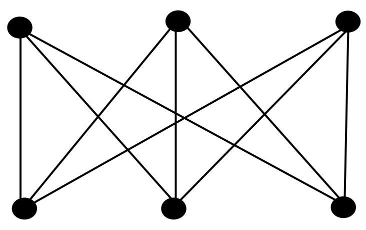
A **bipartite graph**, $G(V_1, V_2)$ is a graph whose vertices can be partitioned into two disjoint subsets V_1 and V_2, where no edge joins vertices that are in the same subset. A vertex in one of the subsets may be joined to all, some, or none of the vertices in the other subset – see the diagrams below. In the case where every vertex of V_1 is joined to every vertex of V_2 then G is called a **complete bipartite graph** and is usually denoted $K_{r,s}$. Here r and s represent the number of vertices in V_1 and V_2 respectively. A bipartite graph is usually shown with the two subsets as top and bottom rows of vertices or with the two subsets as left and right columns of vertices.



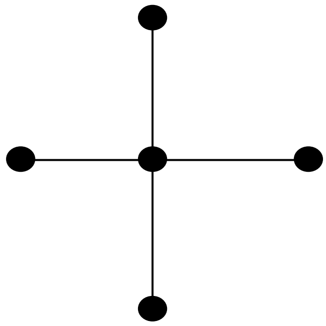


The graph on the right is the complete bipartite graph, $K_{2,5}$ with $2 + 5 = 7$ vertices and $2 \times 5 = 10$ edges. In general, a complete bipartite graph $K_{r,s}$ has $r + s$ vertices and $r \times s$ edges.

A bipartite graph $K_{r,s}$ is **regular** if and only if $r = s$. The complete bipartite graph $K_{3,3}$ shown below is regular as each vertex has degree 3.

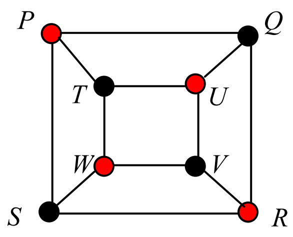


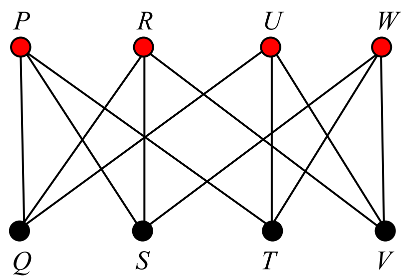
A complete bipartite graph of the form $K_{1,s}$ is called a **star graph** and $K_{1,4}$ is shown below.



Example 8

The graph on the left can be drawn as a 3-regular bipartite graph by partitioning the vertices into the two sets $V_1 = \{P, R, U, W\}$, shown in red, and $V_2 = \{Q, S, T, V\}$, shown in black. Although they look different they are in fact the same graph.





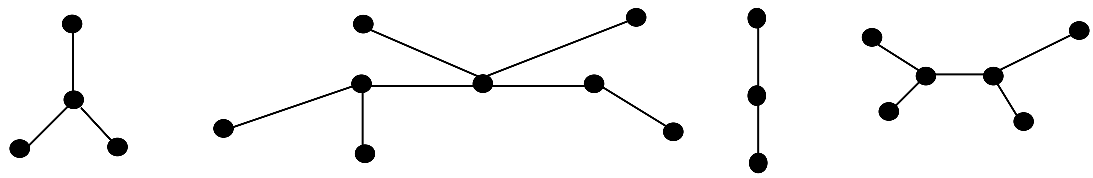
Notes

(i). Two graphs with the same number of vertices and the same number of edges, with the edges connected in the same way are said to be **isomorphic**.

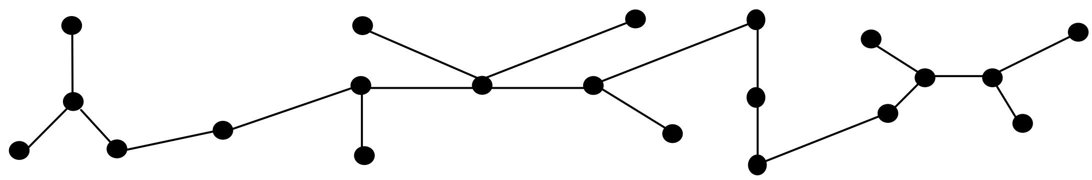
(ii). If two graphs have different degree sequences the graphs are not isomorphic.

3.5. Tree Graphs:

A **forest** is a graph containing no cycles and a connected forest is called a **tree**. Note that a graph on n vertices has fewest edges when it is a tree (as it has no cycles) and most edges when it is a complete graph. Below is a forest with four components.



If the four components in the above forest are connected we obtain the tree below.



Theorem

Let T be a graph with $n > 1$ vertices. The following statements are equivalent:

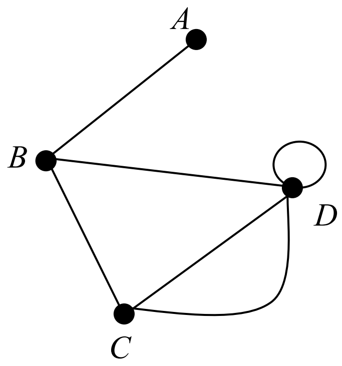
- T is a tree.
- T is cycle-free and has $n – 1$ edges.
- T is connected and has $n – 1$ edges.
- T is connected and contains no cycles.
- T is connected and each edge is a **bridge**.
- Any two vertices of T are connected by exactly one path.
- T contains no cycles, but the addition of any new edge creates exactly one cycle.

Note: From the above theorem it must be the case that a finite tree with n vertices must have $n - 1$ edges.

3.6. Multigraphs

The diagram shows the graph

$G = \{ V, E \} = \{ \{ A, B, C, D \}, \{\{A, B\}, \{B, C\}, \{B, D\}, \{C, D\}, \{C, D\}, \{D, D\}\} \}$.



This is an example of a **multigraph**. A multigraph is a graph that allows the existence of **loops** and **parallel** (**multiple**) **edges**. Note that not all texts allow multigraphs to have loops and in the case when graphs include loops they are called **pseudographs**. We shall refer to a graph with parallel edges and/or loops as a multigraph.

A **loop** is an edge that links a vertex to itself. In the figure the edge (D, D) is a loop and connects vertex D to itself.

If two vertices are joined by more than one edge then these edges are called **parallel edges**. In the figure the edges (C, D) represent parallel edges.

Notes

(i). We define a loop to contribute 2 to the degree of a vertex so that the Handshaking Lemma holds for multigraphs. In the above figure vertex D therefore has degree 5. The degree sum of the graph is 1 + 3 + 3 + 5 = 12 which is twice the number of edges (6) as required by the Handshaking Lemma.

(ii). Some texts do not allow multigraphs to have loops.

4. Walks, Trails & Paths

A **walk** of length k on a graph G is an alternating sequence of vertices (v_i) and edges (e_i):

$$v_0,\ e_1,\ v_1,\ e_2,\ v_2,\ e_3,\ .\ \ .\ \ .\ e_k,\ v_k$$

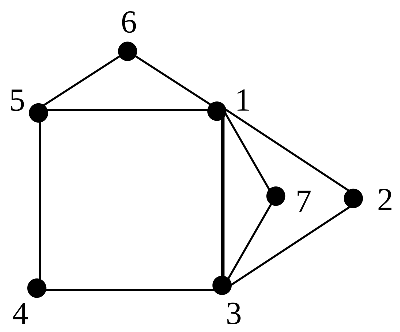
where v_i and v_{i+1} are both incident to e_{i+1}. Note that the graph has k + 1 vertices and k edges.

The **length** of a walk is the number of edges in the walk.

For convenience, and ease of reading, we omit edges and use only vertices so that the walk given above is written as $v_0,\ v_1,\ v_2,\ .\ \ .\ \ .\ v_{k-1},\ v_k$.

Example 9

(i). A **walk** on the graph below is given by: 1, 5, 4, 3, 7, 1, 6 and has length, L = 6.



Note: A walk can repeat both edges and vertices.

(ii). A walk is said to be **closed** if its first and last vertices are the same, i.e. $v_0 = v_k$.

A **closed walk**, of length 8, on the graph is given by: 1, 5, 4, 3, 7, 1, 6, 5, 1.

(iii). A **trail** is a walk where all edges are distinct but vertices may be repeated.

A **trail** on the graph is given by: 1, 5, 4, 3, 7, 1, 6, 5.

(iv). A **closed trail** is called a **circuit**.

A **circuit** on the graph is given by: 1, 2, 3, 1, 5, 4, 3, 7, 1. Note that no edges are repeated but we are allowed to repeat vertices.

(v). A **path** is a trail in which all vertices are distinct. Hence, in a path neither vertices nor edges are repeated.

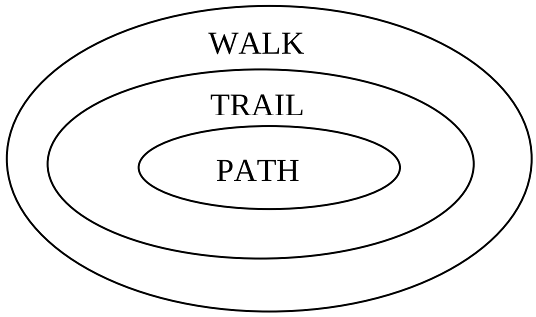
A **path** on the graph is given by: 1, 5, 4, 3, 7.

(vi). A **closed path** is called a **cycle.**

A **cycle** on the graph is given by: 1, 2, 3, 4, 5, 1. Note that no vertices or edges are repeated.

Therefore, all paths are trails and all trails are walks.

In terms of set theory, Paths $\subseteq$ Trails $\subseteq$ Walks as shown below.



The information given above can be summarised in the following table:

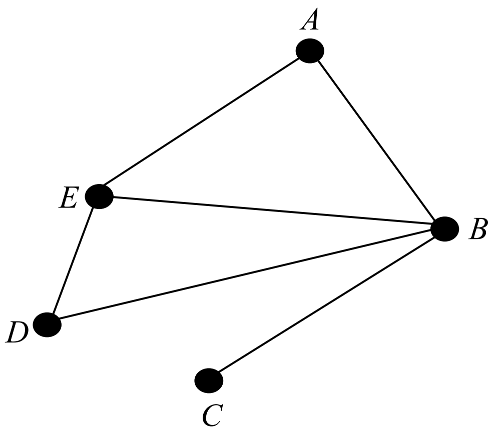
Repeated Vertex (Vertices)	**Repeated Edge(s)**	**Open**	**Closed**	**Name**
Yes	Yes	Yes		Open Walk
Yes	Yes		Yes	Closed Walk
Yes	No	Yes		Trail
Yes	No		Yes	Circuit (Closed Trail)
No	No	Yes		Path
No	No		Yes	Cycle (Closed Path)

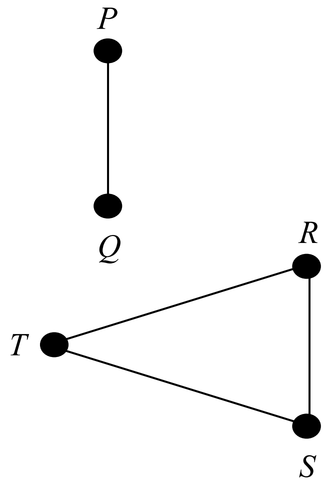
Adapted from, "Discrete and Combinatorial Mathematics" by R. P. Grimaldi.

Now that we have defined the term path we can provide an alternative definition, to that given in Section 2.2, for a graph to be **connected**.

A graph is **connected** if given any two vertices v_i and v_j there is a path from v_i to v_j.

Returning to the example in Section 2.2, reproduced below for completeness, there is clearly a path between all the vertices in the graph on the left and so it is connected. However, in the graph on the right we are unable to, for example, find a path from vertex *S* to vertex *P* and so the graph is disconnected.





5. Eulerian and Hamiltonian Graphs

This section considers special ways of traversing graphs. Examples of graph traversal problems are the Königsberg bridges and Travelling Salesman problems.

5.1. Eulerian Graphs

Definition: An **Euler circuit** on a graph, G, is a circuit (closed trail) that uses every edge of G exactly once. Note that we are allowed to use the same vertex multiple times, but we can only use each edge once. A graph is **Eulerian** if it has an Euler circuit.

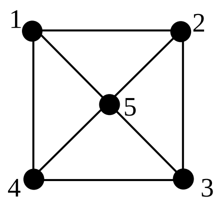
Definition: An **Euler trail** through a graph, G is an open trail that passes exactly once through each edge of G. We say that G is **semi-Eulerian** if it has an Euler trail. Note that every Eulerian graph is semi-Eulerian.

Theorem: Let G be a connected graph. Then G is Eulerian if and only if every vertex of G has even degree.

Corollary: A connected graph is semi-Eulerian if and only if there are 0 or 2 vertices of odd degree. Note that if a semi-Eulerian graph has two vertices of odd degree then any Euler trail must have one of them as its initial vertex and the other as its final vertex.

Example 10

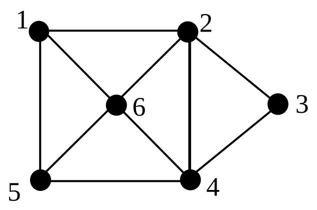
(i).



NON-EULERIAN

As there are four vertices of odd degree the graph is non-Eulerian.

(ii).

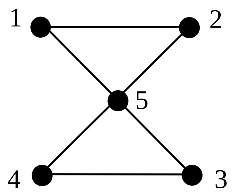


SEMI-EULERIAN

By the above corollary as there are two vertices (1 and 5) of odd degree (i.e. degree 3) the graph is semi-Eulerian.

Euler trail must start at one of the odd degree vertices and end at the other, e.g. 12342645615.

(iii).



EULERIAN

All vertices have even degree and so, by the above theorem, the graph is Eulerian.

Euler circuit: 1253451

The table below provides simple rules that count the number of odd degree vertices in a graph to decide whether or not it has an Euler circuit or Euler trail.

No. of Odd Vertices	For a Connected Graph
0	There is at least one Euler circuit.
1	Not possible
2	No Euler circuit but at least 1 Euler trail.
More than 2	No Euler circuits or Euler trails.

The following algorithm is optional but it provides a relatively simple method for finding an Euler circuit when one exists.

Fleury's Algorithm

If *G* is an Eulerian graph then using the following procedure, known as **Fleury's Algorithm**, it is always possible to construct an Euler circuit of *G*.

Starting at any vertex of *G* traverse the edges of *G* in an arbitrary manner according to the following rules:

(i). Erase edges as they are traversed and if any isolated vertices appear erase them.

(ii). At each step use a **bridge** only if there is no alternative.

Note: Since every vertex in the Königsberg graph in Figure 2 has an odd degree it is not possible to find an Euler circuit of this graph. It is therefore impossible for someone to walk around the city in such a way that each bridge is crossed exactly once and they end up at their starting point.

5.2. Hamiltonian Graphs

Definition: A **Hamiltonian cycle** on a graph, *G*, is a cycle (closed path) that uses every vertex of *G* exactly once. Note that we do not need to use all the edges. A graph is **Hamiltonian** if it has a Hamiltonian cycle.

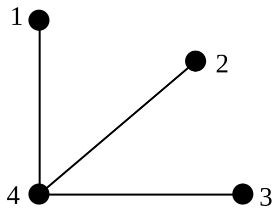
Note that some texts call a Hamiltonian cycle a **Hamiltonian circuit,** i.e. a circuit which passes exactly once through each vertex of a graph. This definition is identical to the one above because a circuit which does not repeat vertices, apart from the starting vertex, is a cycle.

Definition: A trail that passes exactly once through each vertex of *G* and is not closed is called a **Hamiltonian trail**. We say that *G* is **semi-Hamiltonian**. Note that every Hamiltonian graph is semi-Hamiltonian.

While we have a theorem that provides necessary and sufficient conditions for a connected graph to be Eulerian (i.e. '*G* is Eulerian if and only if every vertex of *G* has even degree') there is no similar characterisation for Hamiltonian graphs – this is one of the unsolved problems in graph theory. In general, it is much harder to find a Hamiltonian cycle than it is to find an Eulerian circuit.

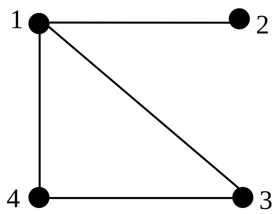
Example 11

(i).



NON-HAMILTONIAN

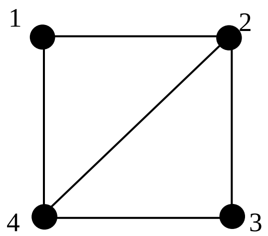
(ii).



SEMI-HAMILTONIAN

Hamiltonian trail: 2143

(iii).



HAMILTONIAN

Hamiltonian cycle: 12341

Note that we do not need to use all edges.

Notes

(i). The Travelling Salesman problem (TSP) searches for the most efficient (least total distance) Hamiltonian cycle a salesman can take so that each of *n* cities is visited. To date, no solution to the TSP has been found.

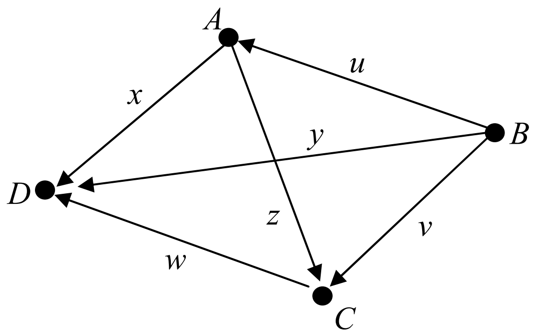
(ii). An Eulerian circuit traverses every edge in a graph exactly once, and may repeat vertices. A Hamiltonian cycle, on the other hand, visits each vertex in a graph exactly once but does not need to use every edge.

6. Digraphs (Directed Graphs)

The graphs that we have met up to now have all been **undirected graphs** in the sense that the edges have no orientation. In this section we extend the notion of a graph to include graphs in which "edges have a direction". These kind of graphs are known as **directed graphs**, or **digraphs** for short. As shown in the diagram below the direction of an edge is defined so that movement between two vertices is only possible in the specified direction. The terminology for digraphs is essentially the same as for undirected graphs except that it is commonplace to use the term **arc** instead of **edge**. Digraphs can be used to model real-life situations such as flow in pipes, traffic on roads, route maps for airlines and hyperlinks connecting web-pages. We have actually encountered the concept of a digraph before in an earlier unit when we looked at relations on sets. In Section 3.3 of that unit, which was optional, we described how a relation R could be represented diagrammatically by a digraph as an alternative to using an arrow diagram or a matrix.

Example 12

The figure below shows a digraph on four vertices with six arcs.



Considering the arc labelled x, we say that x **goes from** A **to** D with A being the **initial vertex** and D the **terminal vertex** of x.

6.1. In-degree and Out-degree

- The **in-degree** of a vertex is the number of arcs that terminate at that vertex.
  For example, the in-degree of vertex C in Example 12 is 2.

- The **out-degree** of a vertex is the number of arcs that originate at that vertex.
  For example, the out-degree of vertex B in Example 12 is 3.

6.1.1. The Handshaking (Di)Lemma

In any digraph the sum of the out-degrees, equals the sum of the in-degrees, equals the number of arcs.

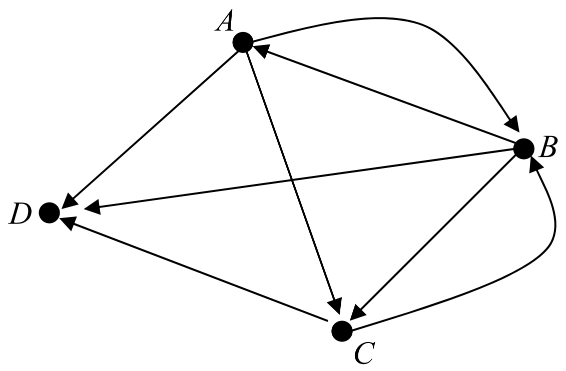
Proof: Every arc contributes exactly once to the out-degree total and exactly once to the in-degree total.

The **in-degree sequence** of a digraph is a bracketed list of the in-degrees of all the vertices in **ascending** order with repetition as necessary.

The **out-degree sequence** of a digraph is a bracketed list of the out-degrees of all the vertices in **ascending** order with repetition as necessary.

Example 13

Consider the following digraph.



(i). Determine the in-degree and out-degree of the vertices and show that the Handshaking (Di)Lemma holds.

(ii). Write down the in-degree and out-degree sequences.

Solution

(i). Create a table of in-degrees and out-degrees.

	A	***B***	***C***	***D***	**Total**
Out-degree	3	3	2	0	**8**
In-degree	1	2	2	3	**8**

The sum of the out-degrees (8) equals the sum of the in-degrees (8) and these values both equal the number of edges (8). The Handshaking (Di)Lemma therefore holds.

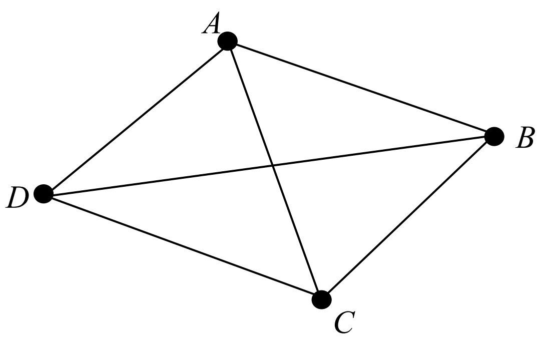
(ii). From the table in part (i), the in-degree sequence is,{ 1, 2, 2, 3 } and the out-degree sequence is,{ 0, 2, 3, 3 }.

6.2. Underlying Graph

The **underlying graph** of a digraph is the undirected graph obtained when the arrows are removed from the digraph.

Example 14

The graph underlying the digraph in Example 12 is the undirected graph shown below. Note that arcs have been replaced by edges.

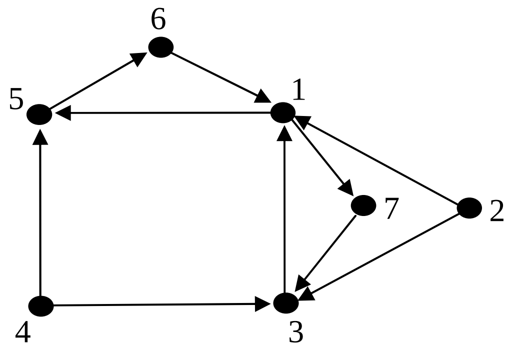


6.3. Walks, Trails and Paths on Digraphs

The concept of walks, trails and paths carries over from undirected graphs to digraphs – see Section 4. However, we must remember that on a digraph we can only move along an edge in a single direction, i.e. the direction in which the arrow is pointing.

Example 15

Find a walk, trail and path on the digraph shown below.



Solution

A **walk** is any route from one vertex to another along the edges of the graph. A walk can repeat edges and vertices any number of times and can end on any vertex.

One example of a **walk** is given by: 1, 5, 6, 1, 7, 3, 1, 5, 6.

A **trail** is a walk where all edges are distinct but vertices may be repeated.

One example of a **trail** is given by: 1, 5, 6, 1, 7, 3.

A **path** is a trail in which all vertices are distinct.

One example of a **path** is given by: 1, 5, 6.

7. Adjacency Matrices

Up to now we have only considered graphs where the number of edges and vertices is relatively small so that they can be easily be shown in diagram form. However, as graphs become large it is no longer feasible to display them visually. When storing a graph on a computer it is useful to represent it in matrix form, as the calculation of paths, trails and circuits, for example, can easily be performed. In this section we look at adjacency matrices for both undirected and directed graphs.

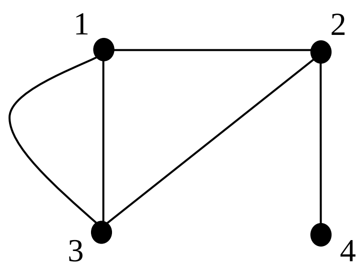
7.1. Adjacency Matrix of an Undirected Graph

In Section 2 we defined an **undirected graph** to be a graph in which the edges have no orientation. Hence, all edges are bidirectional. For example, in the graph shown in Example 14 the edge $\{A,\ B\}$ is considered identical to the edge $\{B,\ A\}$. We now look at how to generate adjacency matrices for undirected graphs.

If G is a graph with n vertices its **adjacency matrix**, A is defined as the $n \times n$ matrix whose ij-th entry is the number of edges joining vertex i and vertex j.

Example 16

Determine an adjacency matrix for following graph.



Solution

The graph has 4 vertices and so the adjacency matrix will have dimension, 4×4.

The entries of the matrix are determined as follows:

- 0 edges connect vertex 1 to vertex 1, so the entry in Row1/Column1 is a '0'.
- 1 edge connects vertex 1 to vertex 2, so the entry in Row1/Column2 is a '1'.
- 2 edges connect vertex 1 to vertex 3, so the entry in Row1/Column3 is a '2'.
- 0 edges connect vertex 1 to vertex 4, so the entry in Row1/Column4 is a '0'.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 0 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{pmatrix} \end{array}$$

- 1 edge connects vertex 2 to vertex 1, so the entry in Row2/Column1 is a '1'.
- 0 edges connect vertex 2 to vertex 2, so the entry in Row2/Column2 is a '0'.
- 1 edge connects vertex 2 to vertex 3, so the entry in Row2/Column3 is a '1'.
- 1 edge connects vertex 2 to vertex 4, so the entry in Row2/Column4 is a '1'.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ . & . & . & . \\ . & . & . & . \end{pmatrix} \end{array}$$

- 2 edges connect vertex 3 to vertex 1, so the entry in Row3/Column1 is a '2'.
- 1 edge connects vertex 3 to vertex 2, so the entry in Row3/Column2 is a '1'.
- 0 edges connect vertex 3 to vertex 3, so the entry in Row3/Column3 is a '0'.
- 0 edges connect vertex 3 to vertex 4, so the entry in Row3/Column4 is a '0'.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ . & . & . & . \end{pmatrix} \end{array}$$

- 0 edges connect vertex 4 to vertex 1, so the entry in Row4/Column1 is a ‘0’.
- 1 edge connects vertex 4 to vertex 2, so the entry in Row4/Column2 is a ‘1’.
- 0 edges connect vertex 4 to vertex 3, so the entry in Row4/Column3 is a ‘0’.
- 0 edges connect vertex 4 to vertex 4, so the entry in Row4/Column4 is a ‘0’.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

This is the adjacency matrix for the graph.

Notes

(i). A graph can be represented by several adjacency matrices as different labelling of the vertices produces different matrices.

(ii). In the matrix *A*, the entry a_{ij} records the number of edges joining vertices *i* and *j*.

(iii). For an undirected simple graph: Sum of Row j = Sum of Column j = Degree of vertex j .

(iv). The adjacency matrix for an undirected graph is **symmetric**, i.e. $A = A^T$.

(v). The entries on the main diagonal are all 0 unless the graph has loops.

Example 17

Given an adjacency matrix we can construct the associated graph, *G*.

Determine the graph corresponding to the adjacency matrix below,

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 1 \\ 2 & 2 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

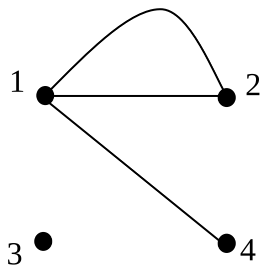
Solution

The matrix has dimension 4×4 and so the graph has 4 vertices.

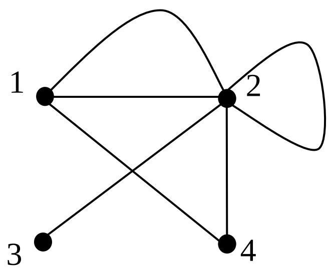
Note that a **loop** is defined to contribute **2** to the degree of a vertex. This approach ensures that the Handshaking Lemma holds for multigraphs.

We proceed as follows processing one row of the matrix A at a time:

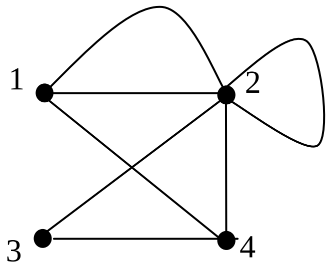
- Entry in Row1/Column1 is a ‘0’so 0 edges connect vertex 1 to vertex 1.
- Entry in Row1/Column2 is a ‘2’so 2 edges connect vertex 1 to vertex 2.
- Entry in Row1/Column3 is a ‘0’so 0 edges connect vertex 1 to vertex 3.
- Entry in Row1/Column4 is a ‘1’so 1 edge connects vertex 1 to vertex 4.



- Entry in Row2/Column1 is a ‘2’so 2 edges connect vertex 2 to vertex 1.
- Entry in Row2/Column2 is a ‘2’so vertex 2 has a self-loop.
- Entry in Row2/Column3 is a ‘1’so 1 edge connects vertex 2 to vertex 3.
- Entry in Row2/Column4 is a ‘1’so 1 edge connects vertex 2 to vertex 4.

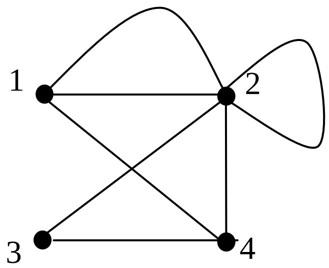


- Entry in Row3/Column1 is a ‘0’so 0 edges connect vertex 3 to vertex 1.
- Entry in Row3/Column2 is a ‘1’so 1 edge connects vertex 3 to vertex 2.
- Entry in Row3/Column3 is a ‘0’so 0 edges connect vertex 3 to vertex 3.
- Entry in Row3/Column4 is a ‘1’so 1 edge connects vertex 3 to vertex 4.



- Entry in Row4/Column1 is a ‘1’so 1 edge connects vertex 4 to vertex 1.
- Entry in Row4/Column2 is a ‘1’so 1 edge connects vertex 4 to vertex 2.
- Entry in Row4/Column3 is a ‘1’so 1 edge connects vertex 4 to vertex 3.
- Entry in Row4/Column4 is a ‘0’so 0 edges connect vertex 4 to vertex 4.

The graph corresponding to the adjacency matrix is therefore:

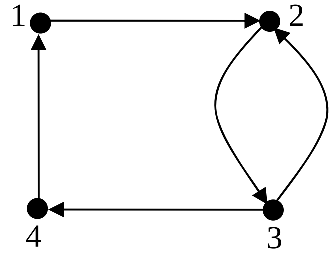


7.2. Adjacency Matrix of a Digraph

The adjacency matrix of a digraph having n vertices is a $n \times n$ matrix. For each directed edge $\{v_i, v_j\}$, i.e. arrow from vertex v_i to vertex v_j, we place a ‘1’ at the i^{th} row, j^{th} column position. Otherwise we place a ‘0’ at the appropriate position in the matrix.

Example 18

Determine an adjacency matrix for the digraph shown below,



Solution

- The digraph has 4 vertices and so the adjacency matrix will have dimension 4×4.

- There is an arc from vertex 1 to vertex 2, so the entry in Row1/Column2 is a '1'
- There is an arc from vertex 2 to vertex 3, so the entry in Row2/Column3 is a '1'
- There is an arc from vertex 3 to vertex 2, so the entry in Row3/Column2 is a '1'
- There is an arc from vertex 3 to vertex 4, so the entry in Row3/Column4 is a '1'
- There is an arc from vertex 4 to vertex 1, so the entry in Row4/Column1 is a '1'

- All other entries in the adjacency matrix will be zero

From the calculations above an adjacency matrix for the digraph is therefore:

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Notes

- The total number of 1's in an adjacency matrix equals the number of arcs in the digraph.
- In general, the adjacency matrix is not symmetric for a digraph.
- The number of 1's in **row** i of an adjacency matrix corresponds to the **out-degree** of vertex i.
- The number of 1's in **column** j of an adjacency matrix corresponds to the **in-degree** of vertex j.

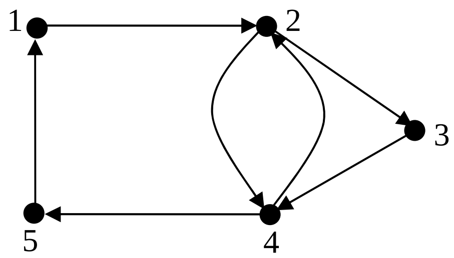
7.3. Eulerian Digraphs

A digraph is **Eulerian** if it is connected and the in-degree of each vertex equals its out-degree.

Equivalently a digraph is Eulerian if it is connected and there exists a **closed trail** (circuit) which uses each arc exactly once. Vertices however, can be repeated. This definition is essentially the same as for undirected graphs, see Section 5.1, except that we can only traverse the graph in the direction of the arrows.

Example 19

Consider the following digraph, *D*.



(i). Determine an adjacency matrix for *D*.

(ii). Is *D* Eulerian? Either state an Euler circuit or explain why the *D* is not Eulerian.

Solution

(i). The digraph has 5 vertices and so the adjacency matrix will have dimension 5×5.

- There is an arc from vertex 1 to vertex 2, so the entry in Row1/Column2 is a '1'
- There is an arc from vertex 2 to vertex 3, so the entry in Row2/Column3 is a '1'
- There is an arc from vertex 2 to vertex 4, , so the entry in Row2/Column4 is a '1'
- There is an arc from vertex 3 to vertex 4, , so the entry in Row3/Column4 is a '1'
- There is an arc from vertex 4 to vertex 2, , so the entry in Row4/Column2 is a '1'
- There is an arc from vertex 4 to vertex 5, so the entry in Row4/Column5 is a '1'
- There is an arc from vertex 5 to vertex 1, so the entry in Row5/Column1 is a '1'
- All other entries in the adjacency matrix will be zero.

The adjacency matrix is therefore,

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

(ii). Recall that the row sums of A give the out-degrees while the column sums provide the in-degrees of the vertices. We construct the following table:

Vertex	**Out-degree**	**In-degree**
1	1	1
2	2	2
3	1	1
4	2	2
5	1	1

This digraph is Eulerian as the out-degree of each vertex is the same as its in-degree.

An Euler circuit is given by: 1, 2, 3, 4, 2, 4, 5, 1.

7.4. Hamiltonian Digraphs

For a digraph to be **Hamiltonian** it must be connected and include a cycle (closed path) that uses every vertex of G exactly once. Such a cycle is called a **Hamiltonian cycle** and need not use every arc of the graph.

Example 20

A Hamiltonian cycle for the digraph in Example 19 is, 1, 2, 3, 4, 5, 1. We have been able to visit each vertex exactly once and return to the start vertex.

8. Adjacency Matrices & Paths

Adjacency matrices can be used to determine the number of paths of different lengths between vertices.

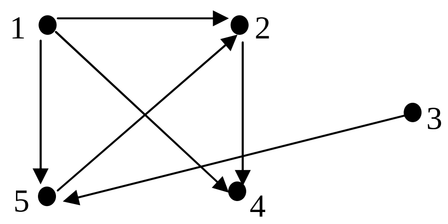
In an adjacency matrix the entry at position $(i,\ j)$ corresponds to the number of paths of length 1 between vertex v_i and vertex v_j. It is also possible to construct matrices that provide information on paths of length other than 1 between vertices.

For example, to calculate the matrix for paths of length 2 we must square the matrix *A*, i.e. calculate $A^2 = A \times A$.

In general, if we calculate *k*-th power of the adjacency matrix *A* the entry at position $(i,\ j)$ of the matrix A^k indicates the number of paths of length *k* between vertex v_i and vertex v_j.

Example 21

Let *D* be a digraph with 5 vertices as shown:



An adjacency matrix is given by

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

If a path of length 1 exists between two vertices (i.e. vertices are adjacent) then there is a 1 in the corresponding position in the adjacency matrix, *A*. Here, for example, inspection of *A* reveals the following paths of length 1:

- from vertex 1 to vertices 2, 4 and 5
- from vertex 2 to vertex 4
- from vertex 3 to vertex 5
- from vertex 5 to vertex 2.

There are no paths of length 1 from vertex 4 to any of the other vertices.

To calculate paths of length 2 the adjacency matrix, A, is multiplied by itself to give A^2, i.e.

$$A^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}.$$

The matrix shows that there are only four paths of length 2 in the digraph:

- from vertex 1 to vertex 2,
- from vertex 1 to vertex 4,
- from vertex 3 to vertex 2
- from vertex 5 to vertex 4.

In general, the matrix of path length k is generated by multiplying the matrix of path length $k - 1$ by the matrix of path length 1, i.e. the adjacency matrix, A.

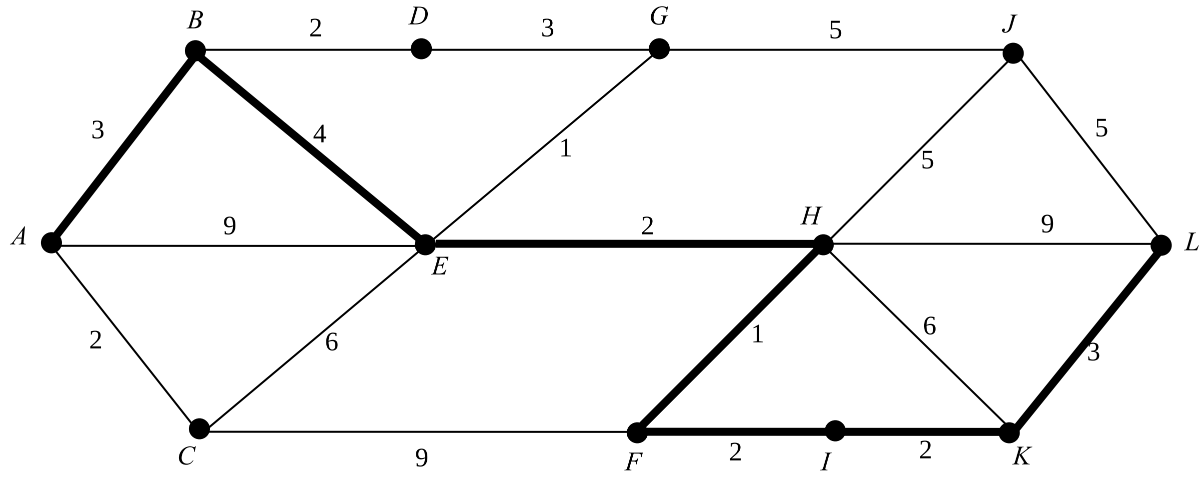
We say that a digraph is **strongly connected** if there is a path from every vertex to every other vertex.

9. Weighted Graphs

The edges in a graph can be weighted or unweighted. In a weighted graph a non-negative real number is assigned to each edge, e, and is called the **weight** of e, denoted $w(e)$. These weights may correspond to the lengths of roads (edges) between towns (vertices) in a graphical representation of a map and we may be required to find the length of the shortest path from town A to town L, say. The problem is then to find the path from A to L with minimum weight. An example of a shortest path problem is the well-known Travelling Salesman Problem.

<u>Example 22</u>

The shortest path from A to L has length 17 and is shown in bold in the figure.

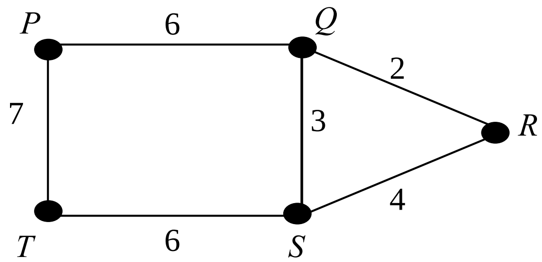


(from Introduction to Graph Theory, Fourth Edition, Wilson R.J., 1996)

<u>9.1. Adjacency Matrix of an Undirected Weighted Graph</u>

The adjacency matrix is calculated in the same way as for the previous examples except that instead of placing a 1 in the i^{th} row and j^{th} column when vertices v_i and v_j are adjacent we enter the weight.

<u>Example 26</u>



$$A = \begin{pmatrix} 0 & 6 & 0 & 0 & 7 \\ 6 & 0 & 2 & 3 & 0 \\ 0 & 2 & 0 & 4 & 0 \\ 0 & 3 & 4 & 0 & 6 \\ 7 & 0 & 0 & 6 & 0 \end{pmatrix}$$

10. Isomorphisms between Graphs

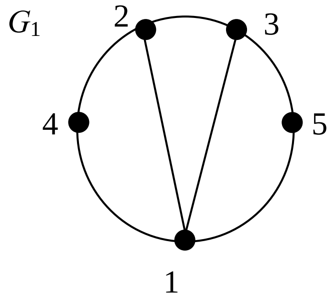
Graphs G and H are said to be **isomorphic** (essentially the same graph) if there is a one-one and onto map,

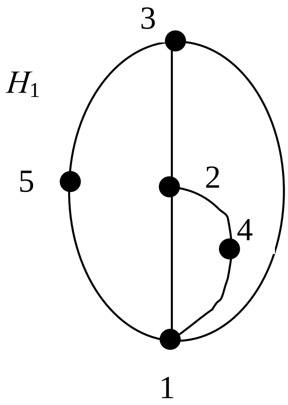
$$\phi: V(G) \to V(H) \text{ such that edge } \{A, B\} \in E(G) \Leftrightarrow \text{edge } \{\phi(A)\, \phi(B)\} \in E(H).$$

In other words there is a one-one correspondence between the vertices of G and the vertices of H with the property that the number of edges joining any two vertices of G is equal to the number of edges joining the corresponding vertices of H.

Example 27

The graphs G_1 and H_1 below are isomorphic.





In graph G_1: vertex 1 has degree 4 and is joined to vertices 2, 3, 4 and 5.
In graph G_1: vertex 2 has degree 3 and is joined to vertices 1, 3, and 4.
In graph G_1: vertex 3 has degree 3 and is joined to vertices 1, 2, and 5.
In graph G_1: vertex 4 has degree 2 and is joined to vertices 1 and 2.
In graph G_1: vertex 5 has degree 2 and is joined to vertices 1 and 3.

It is easily checked that this is the same for graph H_1 and so the graphs are isomorphic.

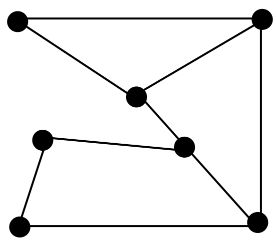
Hence, the **adjacency list** is the same for both graphs,

Vertex	Adjacent vertices
1	2, 3, 4, 5
2	1, 3, 4
3	1, 2, 5
4	1, 2
5	1, 3

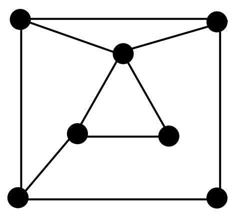
Example 28

The graphs G_2 and H_2 below are not isomorphic as they have different degree sequences.

G_2



H_2



Both graphs have the same number of vertices, i.e. 7. However, Graph G_2 has degree sequence (2, 2, 2, 3, 3, 3,3) while Graph H_2 has degree sequence (2, 2, 3, 3, 3, 3, 4). Alternatively you could show that the two graphs have different adjacency lists.

11. Vertex (Graph) Colouring

The most well-known graph colouring problem is the Four Colour Problem which was first proposed in 1852 when Francis Guthrie noticed that four colours were sufficient to colour a map of the counties of England so that no two counties with a border in common had the same colour. Guthrie conjectured that any map, no matter how complicated, could be coloured using at most four colours so that adjacent regions (regions sharing a common boundary segment, not just a point) are not the same colour. Despite many attempts at a proof it took until 1976 when two American scientists, Appel and Haken, using graph theory produced a computer-based proof to what had become known as the Four Colour Theorem.

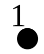
In graph theory terms vertex (graph) colouring problems require the assignment of colours (usually represented by integers) to the vertices of the graph so that no two adjacent vertices are assigned the same colour (integer).

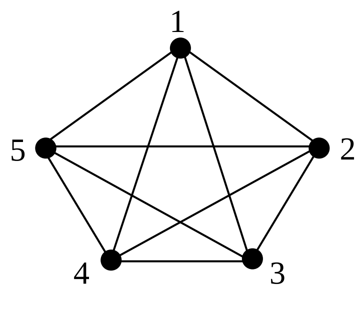
Definition

A k-colouring of a graph is a colouring in which only k colours (numbers) are used. The **chromatic number** for a graph is the **minimum** number of colours (numbers) required to produce a vertex colouring of the graph. The chromatic number of a graph G is denoted by $\chi(G)$.

Example 29

A graph with no edges has chromatic number 1 while the complete graph K_n has chromatic number n. In the figures below we assign a '1' to the graph with no edges on the left and say that it is 1-colourable while we assign the numbers 1, 2, 3, 4, 5 to the complete graph K_5 on the right and say that it is 5-colourable. We have that $\chi(K_5) = 5$.





Identifying the chromatic number in the two cases shown above is straightforward. In general, however determining the exact chromatic number of a graph is a hard problem and no efficient method exists. The only approach that would identify the chromatic number of a graph G with absolute certainty would involve investigating all possible colourings. Clearly as graphs become larger this method becomes impractical, even using the most powerful computers that are available. The best that can be done is to determine lower and upper bounds on the chromatic number and techniques such as looking for the largest complete subgraph in G (for a lower bound) and the Greedy algorithm (for an upper bound) enables us to do so. The Greedy algorithm however is very inefficient but is adequate for ‘small’ graphs with the aid of a computer.

Summary

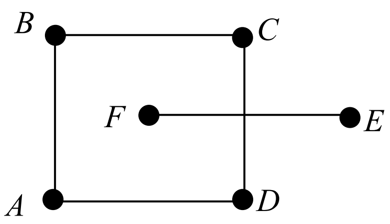
This unit has provided an introduction to the important topic of graph theory and you should now be able to:

- identify different types of general graphs including: undirected and directed graphs; simple graphs and multigraphs.
- understand basic terminology associated with graphs, including: connected, vertices, edges, arcs, adjacent, incident, degree sequence, in-degree, out-degree, etc.
- identify different types of specific graphs: regular graphs, complete graphs, cycle graphs, bipartite graphs, tree graphs and weighted graphs.
- state the Handshaking Lemmas for both undirected graphs and digraphs.
- identify walks trails and paths on undirected graphs and digraphs.
- determine whether or not a graph (undirected or digraph) is Eulerian and identify an Euler circuit if one exists.
- determine whether or not a graph (undirected or digraph) is Hamiltonian and, for “small” graphs, identify a Hamiltonian cycle if one exists.
- construct adjacency matrices for undirected graphs and digraphs.
- construct an undirected graph or digraph given an adjacency matrix.
- understand what is meant by isomorphic graphs.
- understand what is meant by a graph colouring and the chromatic number of a graph.

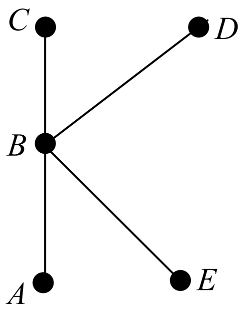
INTRODUCTION TO GRAPH THEORY - TUTORIAL

Q1. **(i).** Which of the following graphs are connected?

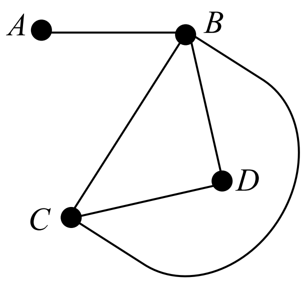
(a).



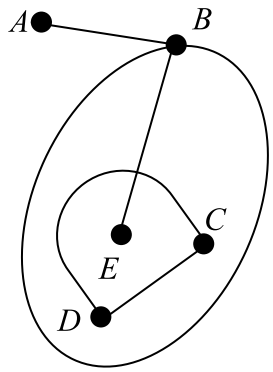
(b).



c).



(d).



(ii). If a graph is not connected state its connected components.

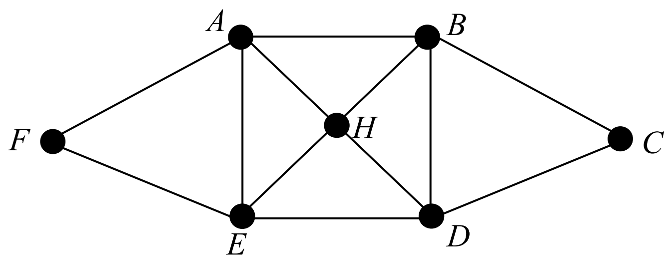
(iii). Which are simple graphs and which are multigraphs?

Q2. Sketch the following graphs:

(i).	4-regular on 6 vertices	**(ii).**	K_5	**(iii).**	C_6
(iv).	K_6	**(v).**	$K_{2,3}$	**(vi).**	$K_{4,4}$.

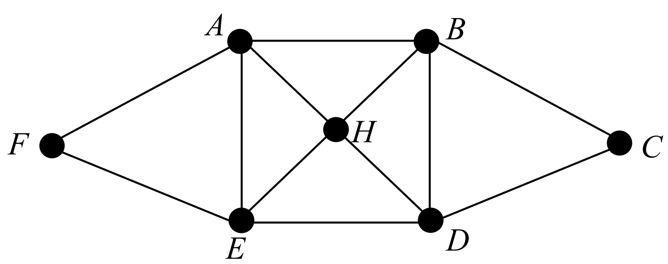
Q3. **(i).** Define the terms walk, trail and path on a graph.

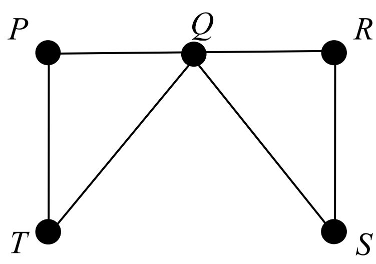
(ii). Find a walk, closed walk, trail, closed trail (circuit), path and a closed path (cycle) on the graph below.



Q4. **(i).** Define the term **Euler circuit** on a graph and find an Euler circuit on each of the graphs below if one exists. If none exist explain why not.

(ii). Define the term **Hamiltonian cycle** on a graph and find a Hamiltonian cycle on each of the graphs below if one exists. If none exist explain why not.





Q**5.** Sketch the undirected graph G defined below and construct an adjacency matrix for G.

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

Q6. Consider the adjacency matrix

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}.$$

(i). Sketch the associated undirected graph, *G*, clearly labelling all the vertices.

(ii). Write down the degree sequence for *G*.

(iii). Show that the Handshaking Lemma holds for *G*.

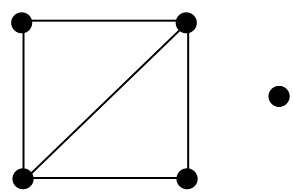
(iv). Is *G* Eulerian? Justify your answer and give an Euler circuit if appropriate.

(v). Is *G* Hamiltonian? Justify your answer and give a Hamiltonian cycle if appropriate.

(vi). Removal of an edge from *G* results in a bipartite graph. Identify which edge should be removed and sketch the resulting graph.

(vii). How many edges need to be added to *G* to obtain a complete graph? Identify which edges need to be added and sketch the resulting graph.

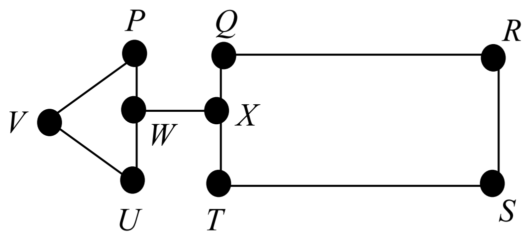
Q7. Given a graph, *G*, its **complementary graph** denoted $\overline{G}$, is obtained from *G* by replacing edges with non-edges and non-edges by edges. If *G* is given by the graph below sketch its complementary graph, $\overline{G}$.



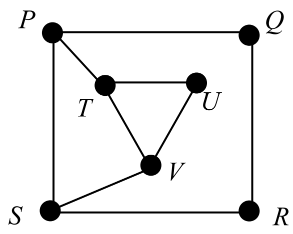
Q8. A graph, *G*, is *k*-regular if all vertices have degree *k*. Calculate the degree sum for a *k*-regular graph with *n* vertices and the number of edges in *G*.

Q9. In a simple graph, with at least two vertices, there are at least two vertices of the same degree. This result is not true for multigraphs. Sketch a three vertex multigraph with all vertices of different degree.

Q10. Consider the graph, G below. Explain why G does not have a Hamiltonian cycle.



Q11. Consider the graph, G, below,

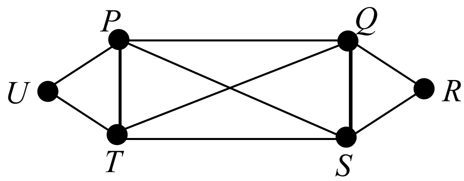


(i). Is G Eulerian? Either state an Euler circuit on G or explain why G is not Eulerian.

(ii). Is G Hamiltonian? Either state a Hamiltonian cycle on G or explain why G is not Hamiltonian.

Q12. Sketch a simple graph G whose vertices all have even degree but G is not Eulerian.

Q13. Consider the graph G below,

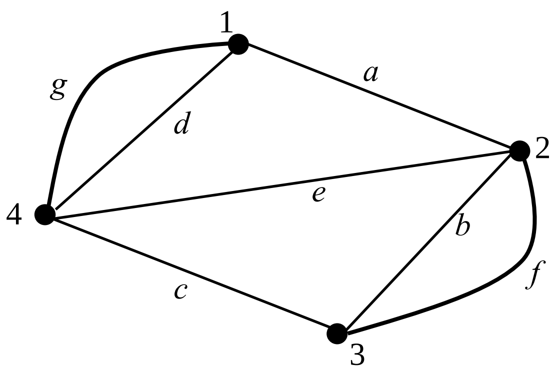


(i). Is G Eulerian? Either state an Euler circuit on G or explain why G is not Eulerian..

(ii). Is G Hamiltonian? Either state a Hamiltonian cycle on G or explain why G is not Hamiltonian.

Q14. Determine whether the complete graphs K_{77} and K_{32} are Eulerian.

Q15. Determine an adjacency matrix and an incidence matrix for the graph shown below,



Q16. An adjacency matrix for an undirected graph, G is given by,

$$A = \begin{pmatrix} 2 & 1 & 1 & 3 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 3 & 1 & 1 & 0 \end{pmatrix}.$$

Without drawing G, and using only the matrix A, answer the following:

(i). How many edges does G have?

(ii). How many paths of length 2 join Vertices 1 and 4.

Q17. How many edges does a tree, T, with 5000 vertices have?

Q18. Determine which complete bipartite graphs, $K_{m,n}$ are trees.

Q19. **(i).** Determine the conditions on r and s that will guarantee that the complete bipartite graph, $K_{r,s}$ will have an Euler circuit.

(ii). How many edges and vertices does the complete bipartite graph $K_{r,s}$ have? Give you answer in terms of r and s.

Q20. Explaining your answer state whether a graph on 7 vertices can have each vertex of degree 5.

Q21. Consider a graph G on 12 vertices where each vertex has degree 7. How many edges does G have? Explain your answer.

Q22. **(i)** Sketch the digraph

$$D = \{ \ \{ 1, 2, 3, 4 \}, \{ \{1, 2\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 2\}, \{3, 4\}, \{4, 1\} \} \ \}.$$

(ii) Determine an adjacency matrix for *D*.

(iii). Calculate the in-degree and out-degree of each vertex.

(iv). State the Handshaking (Di)Lemma and show that it holds for *D*.

(v). State what it means for a digraph to be Eulerian.

(vi). Is the digraph, *D*, Eulerian? Explain your answer.

(vii) Calculate the matrix A^3 and explain the meaning of the entry at position (1, 2) in A^3.

Q23. Consider the following adjacency matrix, *A*, for a digraph, *D*

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

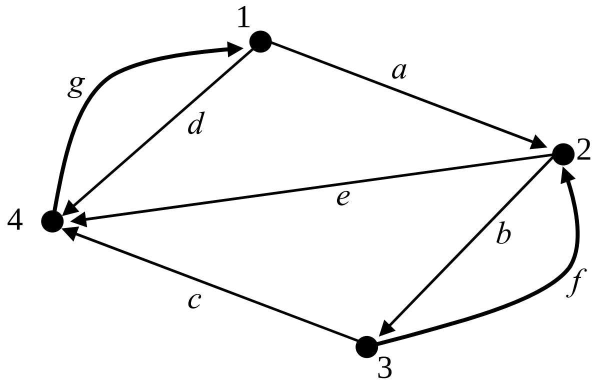
Without drawing *D*, and using only the matrix *A* answer the following:

(i). Calculate the in-degree and out-degree of each vertex.

(ii). Determine whether *D* is Eulerian. Explain your answer.

(iii). How many arcs (edges) are there in *D*? Explain your answer.

Q24. Determine an adjacency matrix for the digraph below.



Q25. Consider the following adjacency matrix

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}.$$

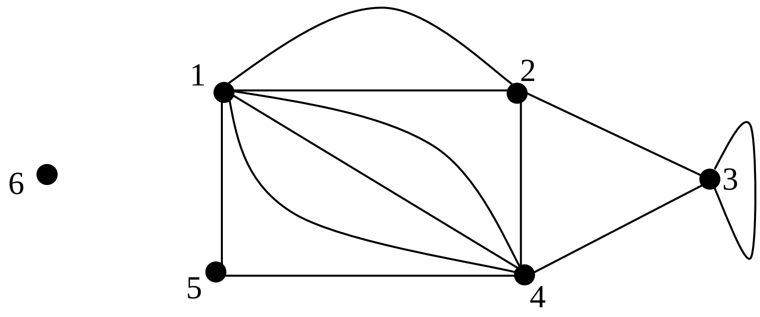
(i). Sketch the associated digraph, *D*, clearly labelling the vertices.

(ii). Determine whether the digraph is Eulerian and state an Euler circuit if one exists.

Q26. **(i).** In a cycle graph, C_n, state how the number of vertices is related to the number of edges.

(ii). Sketch the cycle graphs C_5 and C_6.

Q27. Consider the following multigraph, G.



(i). Write down an adjacency matrix for G. (Note that for an undirected graph we define a loop to contribute 2 to the degree of a vertex).

(ii). Interpret the row sum of the entries in row j of the adjacency matrix.

(iii). What is the degree of Vertex 3? Explain your answer.

Q28. **(i)** Sketch the complete graph K_5 and label the vertices P, Q, R, S and T.

(ii). Construct an adjacency matrix for K_5.

(iii). Describe an adjacency matrix for the general complete graph, K_n.

(iv). Interpret the row sum of the entries in row j of the adjacency matrix for K_n.

(v). How many 1's are contained in the adjacency matrix for a general K_n?

(vi). Interpret the result in part (v).

Q29. Let $A = \{1, 2, 3, 4, 5, 6\}$ and define the relation R as follows,

$R = \{(1, 1), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5),$
$(3, 2), (3, 3), (3, 4), (4, 1), (4, 2), (4, 3), (4, 4), (5, 1), (5, 2), (5, 5)\}$ on A.

(i). Sketch the **digraph**, D, that represents R.

(ii). Determine an adjacency matrix for R.

Q30. (i). Sketch the directed (digraph) and undirected graphs corresponding to the following adjacency matrix.

$$A = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

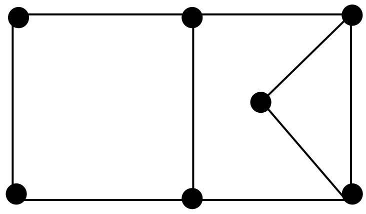
(ii). For the undirected graph determine the degree of each vertex. Then for the digraph determine the in-degree and out-degree of each vertex.

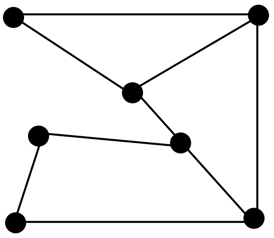
(iii). For both undirected and directed graphs determine whether they are Eulerian and/or Hamiltonian.

Q31. Explain why it is not possible to have the following adjacency matrix for a **simple** graph, (a simple graph is undirected, unweighted and has no loops or parallel edges),

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

Q32. Determine whether the two graphs below are isomorphic.





Q33. What is the chromatic number of a cycle graph, C_n?

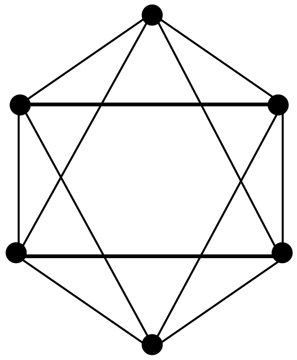
SOLUTIONS

S1. **(i).** Graphs (b) and (c) are connected as there is a path between any two of their vertices.

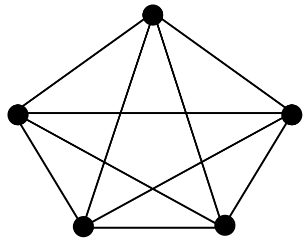
(ii). Graph (a) is disconnected and its disconnected components are $\{ABCD\}$ and $\{EF\}$. Graph (d) is disconnected and its disconnected components are $\{ABE\}$ and $\{CD\}$.

(iii). Graphs (a) and (b) are simple graphs.
Graph (c) is a multigraph with multiple (parallel) edges $\{B, C\}$ and $\{B, C\}$.
Graph (d) is a multigraph with multiple (parallel) edges $\{C, D\}$ and $\{C, D\}$ and a self-loop $\{B, B\}$.

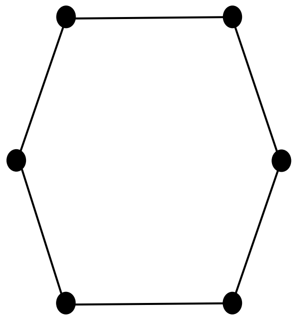
S2.(i).



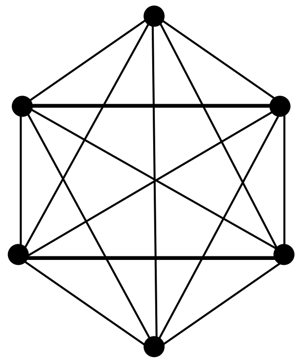
(ii).



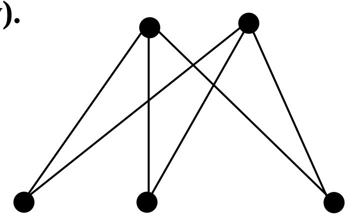
(iii).



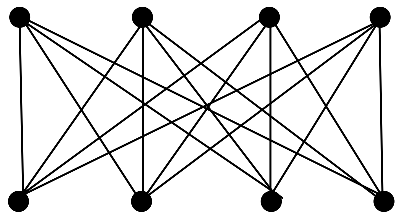
(iv).



(v).



(vi).



S3. **(i).** A **walk** on a graph is any route from one vertex to another along the edges of the graph. A walk can repeat edges and vertices any number of times and can end on any vertex.

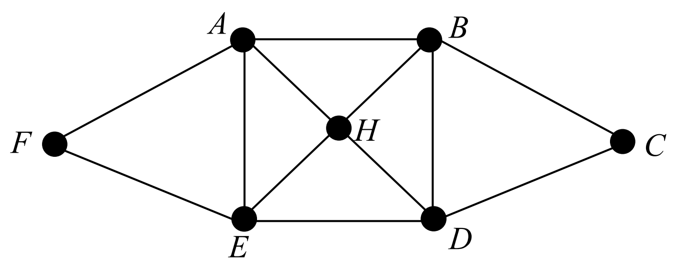
If a walk ends on the vertex it started from it is called a **closed walk**.

A **trail** is a walk where all edges are distinct but vertices may be repeated.

If a trail ends on its starting vertex it is called a **closed trail** or a **circuit**.

A **path** is a trail in which all vertices are distinct. Hence, in a path neither vertices nor edges are repeated.

If a path ends on its starting vertex it is called a **closed path** or a **cycle**.



(ii). An example of a **walk** is given by: *EHBDEHABAF*.

An example of a **closed walk** is given by: *EHBDEHABAFE*.

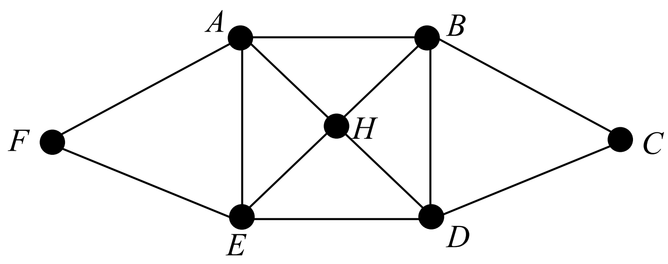
An example of a **trail** is given by: *EHBDEFA*.

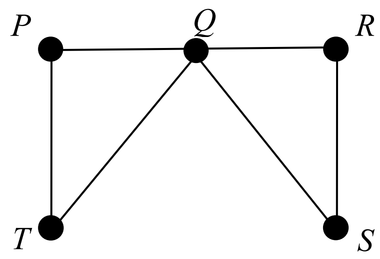
An example of a **closed trail,** or **circuit,** is given by: *EHBDEFAE*.

An example of a **path** is given by: *EHBD*.

An example of a **closed path**, or **cycle**, is given by: *EHBDE*

S4.





(i). An **Euler circuit** on a graph is a circuit (closed trail) that uses every edge exactly once. Note that we are allowed to use the same vertex multiple times but we can only use each edge once.
A graph is **Eulerian** if it has an Euler circuit.

The graph on the left is Eulerian, and has an Euler circuit, as all vertices are of even degree. An **Euler circuit** is given by: *EFABHEDCBDHAE.*

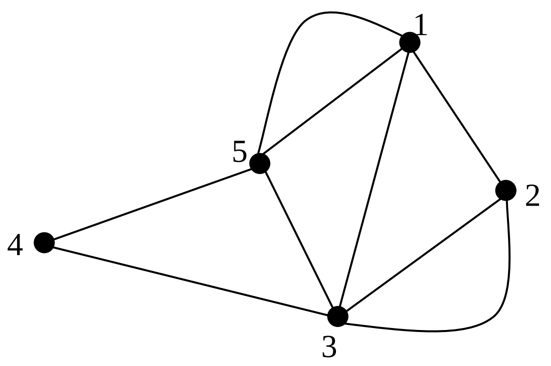
The graph on the right is Eulerian, and has an Euler circuit, as all vertices are of even degree. An **Euler circuit** is given by: *PQRSQTP.*

(ii). A **Hamiltonian cycle**, also called a **Hamiltonian circuit**, is a circuit (closed trail) which passes exactly once through every vertex of a graph G and G is called a **Hamiltonian graph**. We do not need to use all the edges.

The graph on the left is Hamiltonian.
A **Hamiltonian cycle** is given by: *ABCDHEFA.*

The graph on the right is not Hamiltonian as, no matter where we start, we need to visit vertex Q twice to get back to the start vertex. Starting at vertex Q will not help as in this case we would visit Q three times!

S5. The graph, G, has 5 vertices and so the adjacency matrix, A, will be 5×5.

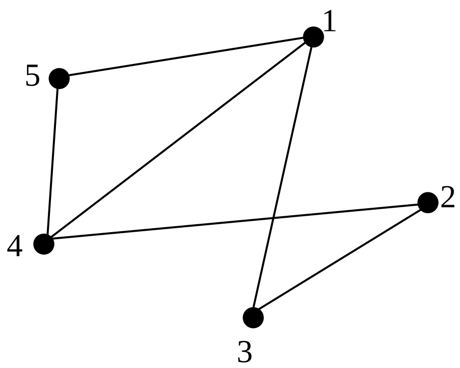


$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 2 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

S6. (i). The adjacency matrix is 5×5 and so G has 5 vertices.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Vertex 1 is adjacent to vertices 3, 4, and 5 so join vertex 1 to each of these three vertices. Continue in this manner to obtain the graph below.



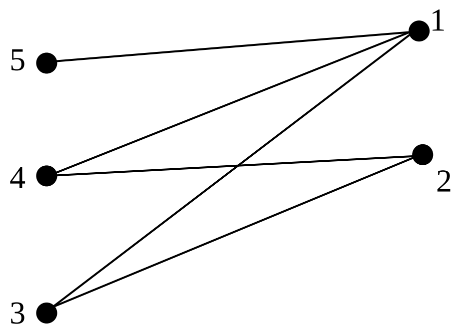
(ii). Degree sequence, (2, 2, 2, 3, 3).

(iii). By the Handshaking Lemma $\sum_{j=1}^{n} \deg(v_j) = 2|E(G)|$ where $|E(G)|$ is the number of edges in G. We therefore have $\sum_{j=1}^{5} \deg(v_j) = 2 + 2 + 2 + 3 + 3 = 12$ and $2|E(G)| = 2 \times 6 = 12$. Hence, the Handshaking Lemma holds for G.

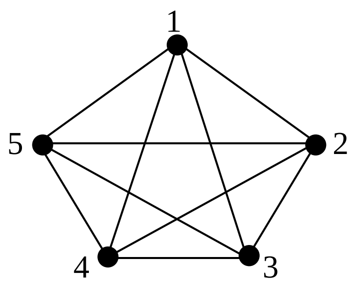
(iv). G is not Eulerian as not all the vertices have even degree.

(v). G is Hamiltonian as we can visit every vertex of G exactly once and return to the start vertex. A Hamiltonian cycle is: 154231.

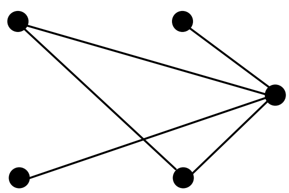
(vi). Removal of edge (4, 5) results in the bipartite graph below.



(vii). Adding the five edges (1, 2), (2, 5). (3, 4), (3, 5), (4, 5) results in the complete graph K_5.

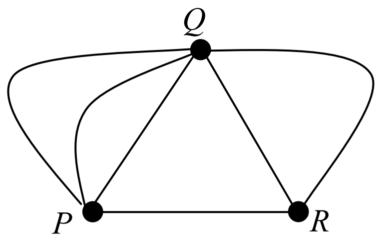


S7. The complementary graph, $\overline{G}$ is



S8. The regular graph G has n vertices all of degree k and so the sum of all the degrees is nk. By the Handshaking Lemma $\sum_{j=1}^{n} \deg(v_j) = 2|E(G)|$ where $|E(G)|$ is the number of edges in G. We therefore have $nk = 2|E(G)| \Rightarrow |E(G)| = \frac{nk}{2}$.

S9. In the graph below; deg(P) = 4, deg(Q) = 5, deg(R) = 3



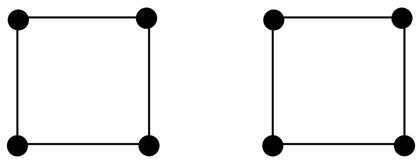
S10. A Hamiltonian cycle visits each vertex of a connected graph exactly once and returns to the starting vertex. Note that G consists of two subgraphs PUV and $QRST$ connected by a bridge WX. If we start on the left-hand-side (PUV) we must cross the bridge (WX) in order to visit every vertex on the right-hand-side, but to get back to our starting vertex we must cross the bridge again thereby visiting the vertices X and W for a second time. Therefore G does not have a Hamiltonian cycle.

Note: No graph with a bridge has a Hamiltonian cycle.

S11. **(i).** The graph is not Eulerian as it contains vertices of odd degree, i.e. vertices P, S, T and V all have degree 3.

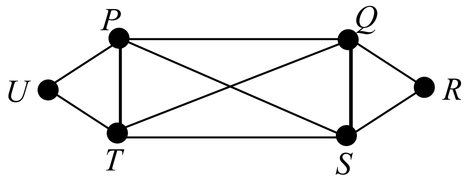
(ii). The graph is Hamiltonian and a Hamiltonian cycle is, $PTUVSRQP$.

S12. For example, the graph below has every vertex of n degree 2 but it is not Eulerian as it is not connected.



S13. **(i).** Eulerian: Yes as all vertices have even degree. Euler circuit: *PSRQSTUPTQP*.

(ii). Hamiltonian: Yes. Hamiltonian cycle: *PQRSTUP*.



S14. The graph K_{77} is 76-regular and so all vertices therefore have even degree, i.e. all vertices have degree 76. Hence, K_{77} is Eulerian.

The graph K_{32} is 31-regular and so all vertices therefore have odd degree, i.e. all vertices have degree 31. Hence, K_{32} is not Eulerian.

S15 (i). $A = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 2 & 1 \\ 3 & 0 & 2 & 0 & 1 \\ 4 & 2 & 1 & 1 & 0 \end{array}$

(ii). $M = \begin{array}{c|ccccccc} & a & b & c & d & e & f & g \\ \hline 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}$.

S16. **(i).** Number of edges in G is, $\left|E(G)\right| = \frac{1}{2}\sum_{X \in V(G)} \deg(X) = \frac{1}{2} \times 20 = 10$.

(The degree sum (20) is obtained by adding the entries in the adjacency matrix).

(ii) For the number of paths of length 2 joining Vertices 1 and 4 we must calculate the matrix A^2.

$$A^2 = \begin{pmatrix} 2 & 1 & 1 & 3 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 3 & 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 & 1 & 3 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 3 & 1 & 1 & 0 \end{pmatrix} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 15 & 7 & 7 & 8 \\ 7 & 6 & 2 & 5 \\ 7 & 2 & 6 & 5 \\ 8 & 5 & 5 & 11 \end{pmatrix} \end{array}.$$

Defining Row 1 to correspond to Vertex 1 and Column 4 to correspond to Vertex 4, the matrix shows that there are 8 paths of length 2 joining Vertices 1 and 4.

S17. As T is a tree by definition, T is cycle-free and has $n - 1$ edges.
As $|V| = 5000$ then $|E| = 5000 - 1 = 4999$. So T has 4999 edges.

S18. If $m = 1$ and/or $n = 1$ then $K_{m,n}$ is a tree.

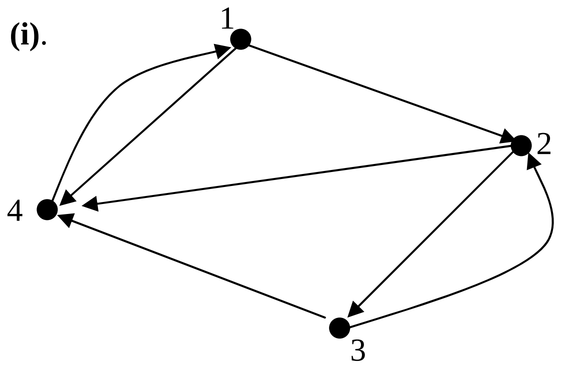
S19. **(i).** If r and s are both even the complete bipartite graph, $K_{r,s}$ will have an Euler circuit as each vertex will have even degree.

(ii). The complete bipartite graph, $K_{r,s}$, has $r + s$ vertices and $r \times s$ edges.

S20. By the Handshaking Lemma it is not possible to construct a graph on 7 vertices where each vertex has degree 5 as the sum of the degrees of the vertices will be, $7 \times 5 = 35$ which is an odd number.

S21. By the Handshaking Lemma the degree sum is twice the number of edges. Hence, since degree sum is $12 \times 7 = 84$ we have that $2|E| = 84$ and so the number of edges is 42.

S22. **(i)**.



(ii).

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

(iii). The table shows the in-degrees and out-degrees of each vertex.

	1	**2**	**3**	**4**	**Total**
In-degree	1	2	1	3	**7**
Out-degree	2	2	2	1	**7**

(iv). The **Handshaking (Di)Lemma** states that in any digraph the sum of the in-degrees is equal to the sum of the out-degrees and both are equal to the number of arcs. This is because every arc counts exactly once to the out-degree total and exactly once to the in-degree total.

For the digraph D: the sum of the in-degrees (7), equals the sum of the out-degrees (7) and both equal the number of arcs (7). Hence, the Handshaking (Di)Lemma holds for D. .

(v). A digraph is **Eulerian** if and only if it is connected and the in-degree of each vertex equals its out-degree. (Equivalently, a digraph is **Eulerian** if it is connected and there exists a **closed trail** (circuit) which uses each arc exactly once.)

(vi). No, D is not Eulerian. The table in part (iii) shows there are vertices where the in-degree of the vertex does not equal its out-degree. We can also obtain this result from inspection of the graph or the adjacency matrix.

(vii). The entry at position (1, 2) in A^3 below indicates that there are exactly two paths of length 3 from Vertex 1 to Vertex 2, i.e. 1412 and 1232.

$$A^3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 2 & 0 & 2 \\ 1 & 1 & 1 & 2 \\ 1 & 2 & 0 & 2 \\ 1 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

S23. **(i).** Label the rows and columns of the matrix as shown:

$$\begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} P & Q & R & S & T \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

The sum of the entries in row j corresponds to the out-degree of vertex j.
The sum of the entries in column j corresponds to the in-degree of vertex j.

	P	***Q***	***R***	***S***	***T***
Out-degree	1	2	3	2	2
In-degree	2	2	2	2	2

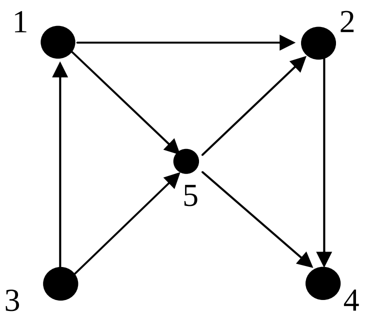
(ii). No, D is not Eulerian as the in-degree does not equal the out-degree for each vertex.

(iii). The graph D contains 10 arcs (edges) as each 1 in the adjacency matrix corresponds to an arc.

S24.(i). Adjacency matrix:

$$A = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

S25. (i).

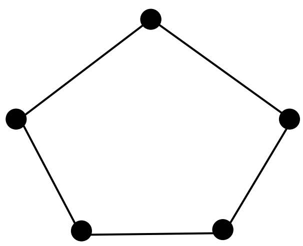


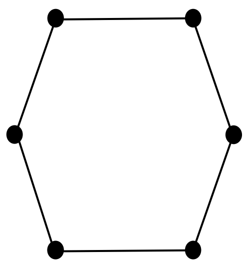
$$A = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

(ii) No, *D* is not Eulerian as the in-degree does not equal the out-degree for each vertex. We can determine this either from inspection of *D* or from the adjacency matrix.

S26. **(i).** The number of vertices in C_n equals the number of edges, and every vertex has degree 2.

(ii). The cycle graphs C_5 and C_6 are shown below.





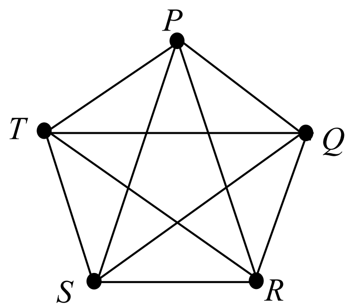
S27. **(i).** An adjacency matrix is:

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 3 & 1 & 0 \\ 2 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 \\ 3 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

(ii). The row sum for row j corresponds to the degree of vertex j.

(iii). Vertex 3 has degree 4 as the loop contributes 2 to the degree.

S28. **(i)** The complete graph K_5 is shown below:



(ii). An adjacency matrix for K_5 is:

$$A = \begin{array}{c} \\ P \\ Q \\ R \\ S \\ T \end{array}\begin{array}{c} \begin{array}{ccccc} P & Q & R & S & T \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

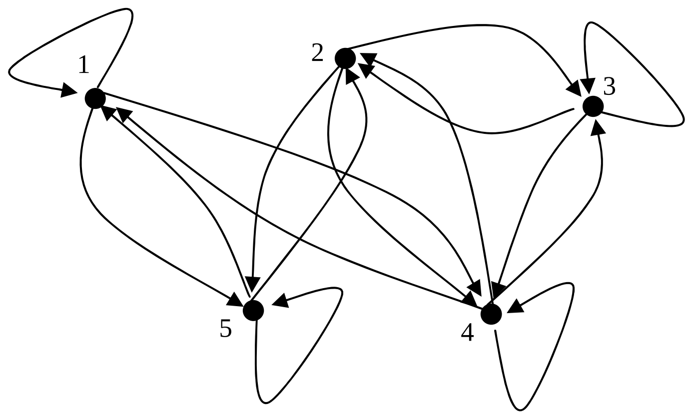
(iii). An adjacency matrix for K_n will have 0's on the leading diagonal and 1's elsewhere.

(iv). The sum of the entries in row j corresponds to the degree of vertex j.

(v). The complete graph K_n has n vertices and so an adjacency matrix will be an $(n \times n)$ matrix. Each row of the adjacency matrix will contain $n - 1$, 1's, giving a total of $n(n - 1)$, 1's.

(vi). If the $n(n - 1)$, 1's are summed this gives the degree sum of all the vertices which, by the Handshaking Lemma, is even and twice the number of edges.

S29. **(i).** The digraph is:

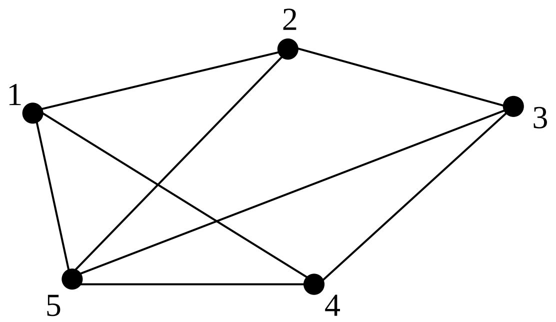


(ii).

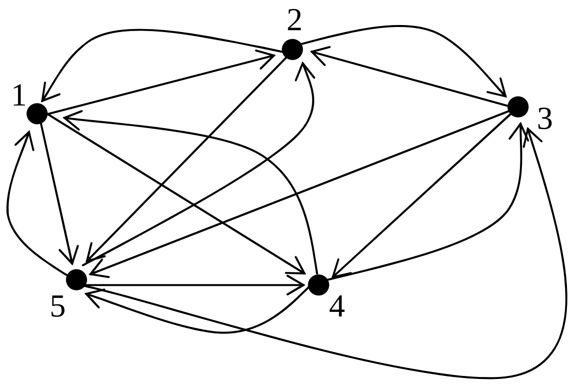
$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

S30. **(i).**

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$



Undirected



Directed

Undirected Graph

Vertex	Degree
1	3
2	3
3	3
4	3
5	4

Directed Graph

Vertex	Indegree	Outdegree
1	3	3
2	3	3
3	3	3
4	3	3
5	4	4

(iii). **Undirected**

No, the graph is not Eulerian as it contains vertices of odd degree.

Yes, the graph is Hamiltonian as it is possible to start at a vertex, visit each vertex exactly once and return to the starting vertex. Hamiltonian cycle: *PQRSTP*.

Directed

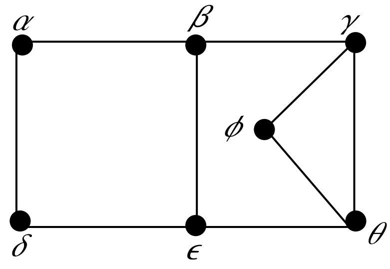
Yes, the graph is Eulerian as the in-degree equals the out-degree at each vertex.

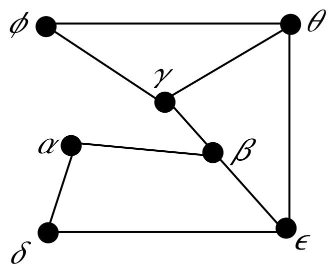
Can you find an Euler circuit?

Yes, the graph is Hamiltonian as it is possible to start at a vertex, visit each vertex exactly once, and return to the starting vertex. Hamiltonian cycle: *PQRSTP*.

S31. The adjacency matrix has dimension, 5×5 so that the graph will have 5 vertices. The rows of the adjacency matrix show that each vertex has degree 3. Hence, the sum of the degrees will be $5 \times 3 = 15$, i.e. an odd number. However, this is impossible as the Handshaking Lemma states that if the degrees of all the vertices in a graph are summed the result must be an even number.

S32. The graphs are isomorphic under the correspondence shown:





The adjacency list is the same for both graphs:

α	β, δ
β	$\alpha, \gamma, \varepsilon$
γ	β, φ, θ
φ	γ, θ
θ	$\varepsilon, \gamma, \varphi$
ε	δ, θ, β
δ	α, ε

S33. The chromatic number of a cycle graph, C_n, is 2 if n is even and 3 if n is odd.