MAS2203/3203 Graph Theory

Semester 1, 2006/2007

Lecturer: Dr A Duncan

This module is an introduction to Combinatorial Mathematics an important part of which is Graph Theory. Graph does not refer to the familiar notion of the graph of a function. Instead it is a construct that can be visualised as a finite set of points (vertices) in 3 dimensional space, together with a finite set of lines (edges) joining them. The vertices and edges of a cube are an example. Often a graph has an additional structure. Its edges can be assigned lengths or capacities, and the graph can be interpreted as a network of roads or pipes. Typical problems are: in a system of roads joining given towns find the shortest network joining all the towns; find the number of different molecules that have a given chemical formula; determine the maximal flow from source to sink through a network of pipes. Any of these problems can be solved by trying all the possibilities; graph theory looks for economical methods of solution.

Books

1. Introduction to Graph Theory, R J Wilson (Longman Scientific & Technical): probably the book closest in level and content to this course.
2. Graphs, an introductory approach, R J Wilson & J J Watkins (Wiley): similar to the book above with more waffle
3. Discrete Mathematics, N L Biggs (Oxford Science Publ.): contains alot of material besides Graph Theory, some of which will arise in other Pure Maths courses
4. Graph Theory with applications, J A Bondy & U S R Murty (MacMillan): goes further than this course
5. Graph Theory "1736–1936", N L Biggs, E K Lloyd & R J Wilson (Clarendon Press): a readable account of the history of Graph Theory
6. **Library §511.5**

Notes

The printed notes consist of lecture notes, intended to supplement the notes you make during the lectures, exercises and a mock exam with solutions. Material given on slides in the lectures is covered in the printed notes, what is written on the blackboard during lectures may not be. There should be enough space in the printed notes for you to write down the notes you take in lectures. The notes, exercises and other course information can be found on the web at

`www.mas.ncl.ac.uk/~najd2/teaching/mas2203/`

from where they can be viewed or printed out.

AJ Duncan August 2006

Contents

0 Prerequisites

This section does not form part of the course but is included to show you what you will need to know. You should refer to it as and when necessary.

Functions revisited

Let X and Y be sets.

Definition 0.1. A **function** (or **map**) from X to Y is a rule which assigns to each element of X a unique element of Y. A function f from X to Y is denoted $f : X \longrightarrow Y$, with $f(x)$ denoting the element of Y assigned to $x \in X$.

1. $f : \mathbb{R} \longrightarrow \mathbb{R}$, with $f(x) = x + 1$ is a function (or map).
2. $g : \mathbb{R} \longrightarrow \mathbb{R}$, with $g(x) = x^2$ is a function (or map).
3. $g' : \mathbb{R} \longrightarrow \mathbb{R}^+$, with $g'(x) = x^2$ is a function (or map), where $\mathbb{R}^+$ denotes the set of non-negative real numbers.
4. $h : \mathbb{Z} \longrightarrow \mathbb{Z}$, with $h(n) = 2n$ is a function (or map).

Definition 0.2. A map (or function) f satisfying the property that whenever $f(a) = f(b)$ then $a = b$, is called **injective**.

1. f above is injective because $x + 1 = x' + 1$ implies $x = x'$.
2. g above is not injective because $g(-1) = 1 = g(1)$ but $-1 \neq 1$.
3. g' above, like g is not injective.
4. h above is injective because $2n = 2m$ implies $n = m$.

Definition 0.3. A map (or function) $\phi : X \longrightarrow Y$ satisfying the property that there exists $x \in X$ such that $\phi(x) = y$, for all $y \in Y$ is called **surjective**.

1. f above is surjective as if $y \in \mathbb{R}$ then $y - 1 \in \mathbb{R}$ and $f(y - 1) = y$.
2. g above is not surjective as $-1 \in \mathbb{R}$ but $-1 \neq x^2$, for all $x \in \mathbb{R}$.
3. g' above is surjective as if $y \in \mathbb{R}$ and $y \geq 0$ then $\sqrt{y} \in \mathbb{R}$ and $g'(\sqrt{y}) = y$.
4. h above is not surjective as $1 \in \mathbb{Z}$ but $1 \neq 2n$, for all $n \in \mathbb{Z}$.

Note that we now have

1. f is both injective and surjective.

2. g is neither injective nor surjective.
3. g' is not injective but is surjective.
4. h is not surjective but is injective.

Definition 0.4. A function which is both injective and surjective is called **bijective**.

Of the examples above only f is bijective. In set theory two sets are defined to have the same **size** if there exists a bijective map from one to the other.

Notation.

- *function* $\equiv$ *map*
- *injective* $\equiv$ *one-one*
- *surjective* $\equiv$ *onto*
- *bijective map* $\equiv$ *one-one correspondence*

Language. Two things are **distinct** if they are not the same. Sets A and B are distinct if there is an element of A which is not in B or vice-versa (which means the other way round). Three numbers are distinct if no two of them are equal. Two things are **disjoint** if they share nothing in common. Sets A and B are disjoint if their intersection is empty.

1 Introduction

Example 1.1.

Definition 1.2. A **graph** G consists of

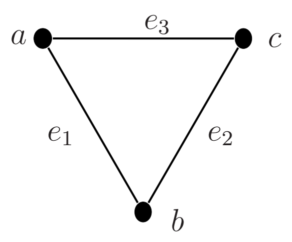
(i) a finite non-empty set $V(G)$ of **vertices** and

(ii) a set $E(G)$ of **edges**

such that every edge $e \in E(G)$ is an unordered pair $\{a, b\}$ of vertices $a, b \in V(G)$.

Example 1.3.

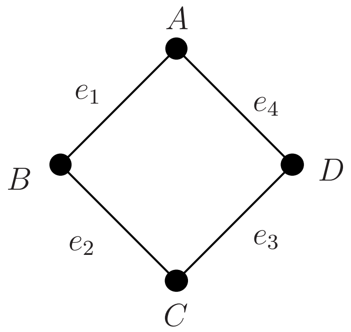
1. $G_1 = (V_1, E_1)$



$$V_1 = \{a, b, c\} \text{ and } E_1 = \{e_1, e_2, e_3\} \text{ with}$$

$$e_1 = \{a, b\}, e_2 = \{b, c\}, e_3 = \{c, a\}.$$

2. $G_2 = (V_2, E_2)$

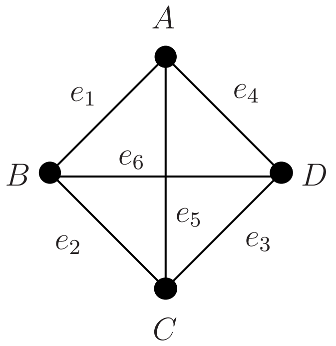


$V_2 = \{A, B, C, D\}$,

$E_2 = \{e_1, \ldots, e_4\}$, and

$e_1 = \{A, B\}, e_2 = \{B, C\}, e_3 = \{C, D\}$,
$e_4 = \{D, A\}$.

3. $G_3 = (V_3, E_3)$

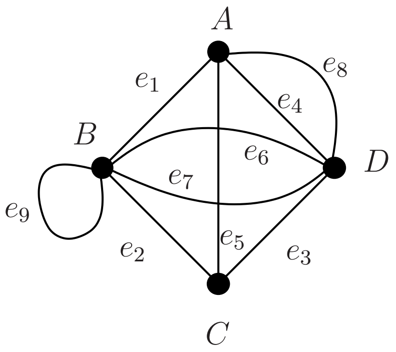


$V_3 = \{A, B, C, D\}$,

$E_3 = \{e_1, \ldots, e_6\}$, and

$e_1 = \{A, B\}, e_2 = \{B, C\}, e_3 = \{C, D\}$,
$e_4 = \{D, A\}, e_5 = \{A, C\}, e_6 = \{B, D\}$.

4. $G_4 = (V_4, E_4)$



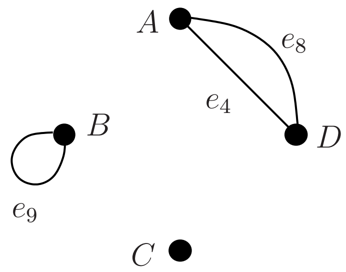
$V_4 = \{A, B, C, D\}$

$E_4 = \{e_1, \ldots, e_9\}$, and

$e_1 = \{A, B\}$, $e_2 = \{B, C\}$, $e_3 = \{C, D\}$,
$e_4 = \{D, A\}$, $e_5 = \{A, C\}$, $e_6 = \{B, D\}$,
$e_7 = \{B, D\}$, $e_8 = \{D, A\}$, $e_9 = \{B, B\}$.

---

5. $G_5 = (V_5, E_5)$



$V_5 = \{A, B, C, D\}$

$E_5 = \{e_4, e_8, e_9\}$, and

$e_4 = \{D, A\}$, $e_8 = \{D, A\}$, $e_9 = \{B, B\}$.

---

6. $G_6 = (V_6, E_6)$

$V_6 = \{A, B\}$,
$E_6 = \emptyset$.

---

7. $G_7 = (V_7, E_7)$

$V_7 = \{A\}$,
$E_7 = \emptyset$.

A graph must have at least one vertex but need not have any edges.

2 First steps

Terminology

Definition 2.1. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are **adjacent** if there exists an edge $e \in E$ with $e = \{a, b\}$.

(ii) Edges e and f are **adjacent** if there exists a vertex $v \in V$ with $e = \{v, a\}$ and $f = \{v, b\}$, for some $a, b \in V$.

(iii) If $e \in E$ and $e = \{c, d\}$ then e is said to be **incident** to c and to d and to **join** c and d.

(iv) If a and b are vertices joined by edges $e_1, \dots e_k$, where $k > 1$, then $e_1, \dots e_k$ are called **multiple** edges.

(v) An edge of the form $\{a, a\}$ is called a **loop**.

Definition 2.2. A graph which has no multiple edges and no loops is called a **simple** graph.

(In Examples (1.3) the graphs in 1–3 and 6 and 7 are simple whereas those in 4 and 5 are not.)

Drawing graphs

From the definition we can easily prove that any graph can be represented by a diagram in $\mathbb{R}^3$. We need the following lemma.

Lemma 2.3. *Let n be a positive integer. Then we can choose n points in $\mathbb{R}^3$ so that no four of them lie on the same plane.*

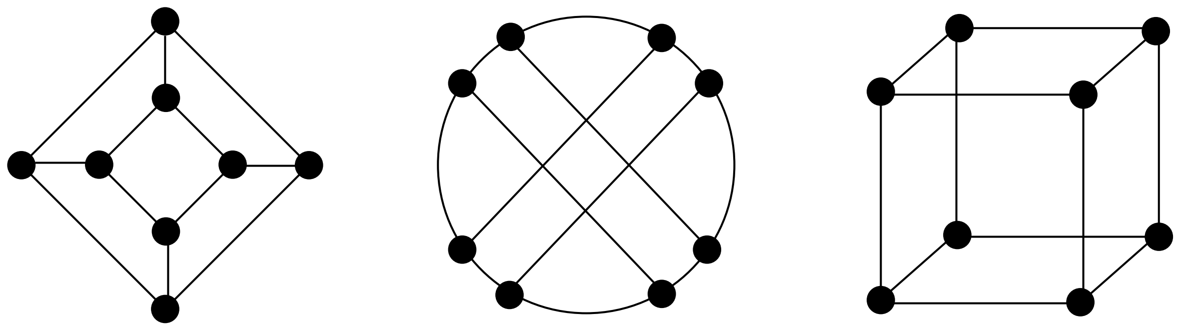
Proof. It is true for $n \leq 3$. Suppose that it is true for $n - 1$. Choose $n - 1$ points $P_1, P_2, \dots, P_{n-1}$ so that no four are coplanar. There are only finitely many planes containing three of these points. Choose P_n so that it does not lie in any of these planes. Then no four of $P_1, P_2, \dots, P_n$ are coplanar. □

To construct the diagram of a graph G, choose a set of points in $\mathbb{R}^3$ having as many members as there are vertices in G, and so that no four of the points are coplanar. Label the points by the names of the vertices and join a, b by an interval $[a, b]$ if and only if $\{a, b\}$ is an edge. Two intervals $[a, b]$ and $[c, d]$ cannot intersect if a, b, c, d are all distinct, since that would force the four points to be coplanar.

Drawings of graphs are projections of diagrams in $\mathbb{R}^3$ into the plane $\mathbb{R}^2$. In some cases it is impossible to draw a graph without edges crossing.

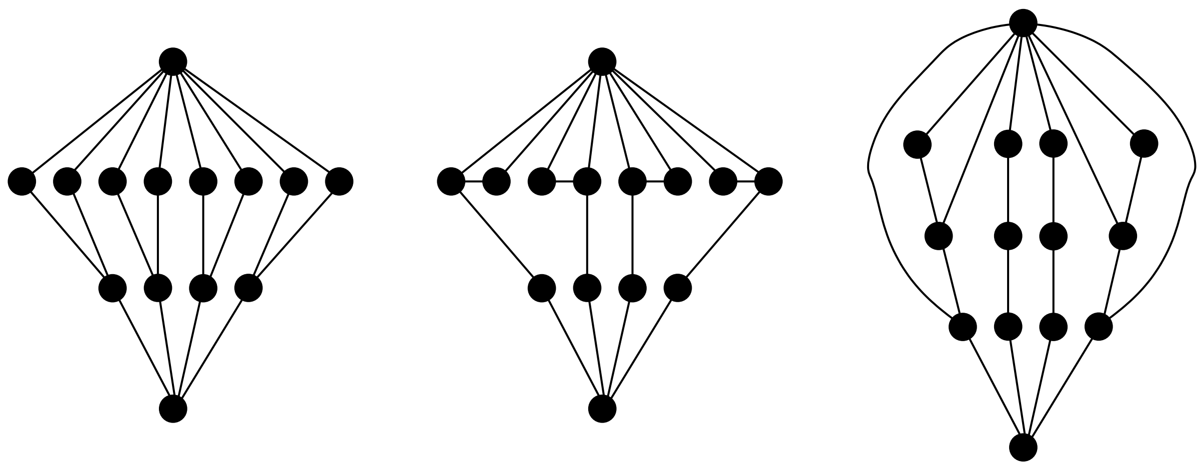
$\boxed{\textbf{A graph may have many different drawings}}$

Example 2.4. Are these three graphs the same?



(a) (b) (c)

Example 2.5. What about these three?



(a) (b) (c)

Definition 2.6. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic** if there exist bijections

$$\phi : V_1 \longrightarrow V_2 \quad \text{and} \quad \theta : E_1 \longrightarrow E_2$$

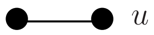
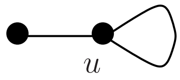
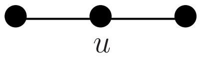
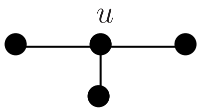
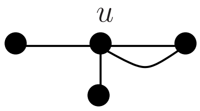
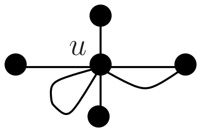
which preserve incidence. That is, such that if

$$e = \{a, b\} \in E_1 \quad \text{then} \quad \theta(e) = \{\phi(a), \phi(b)\} \in E_2.$$

The pair (ϕ, θ) is called an **isomorphism** from G_1 to G_2.

Definition 2.7. The **degree** of a vertex u is the number of ends of edges incident to u and is denoted $\deg(u)$ or $\text{degree}(u)$.

Example 2.8.

Graph	**degree of u**	**degree sequence of G**
	0	$\langle 0 \rangle$
	1	$\langle 1, 1 \rangle$
	3	$\langle 1, 3 \rangle$
	2	$\langle 1, 1, 2 \rangle$
	3	$\langle 1, 1, 1, 3 \rangle$
	4	$\langle 1, 1, 2, 4 \rangle$
	7	$\langle 1, 1, 1, 2, 7 \rangle$

Definition 2.9. Let G be a graph with n vertices. Order the vertices $v_1, \ldots, v_n$ so that $\deg(v_i) \leq \deg(v_{i+1})$. Then G has **degree sequence**

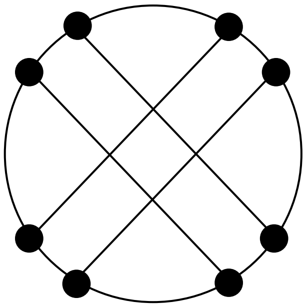
$$\langle \deg(v_1), \ldots, \deg(v_n) \rangle.$$

All the graphs of Example 2.4 are isomorphic – so all have the same degree sequence, $\langle 3, 3, 3, 3, 3, 3, 3, 3 \rangle$.

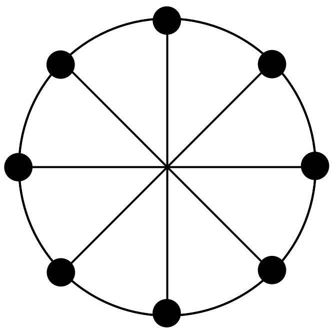
However, all the graphs of Example 2.5 have degree sequence $\langle 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 8 \rangle$ although no two are isomorphic.

Definition 2.10. A graph is **regular** if every vertex has degree d, for some fixed $d \in \mathbb{Z}$. In this case we say the graph is regular of **degree** d.

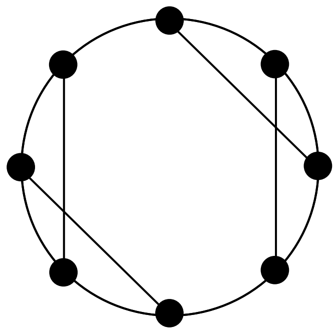
Example 2.11. Graphs which are simple, have 8 vertices, 12 edges and are regular of degree 3. Are any two of these isomorphic? What are their degree sequences?



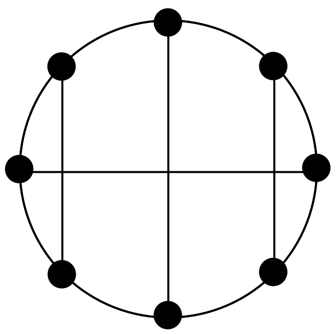
(a)



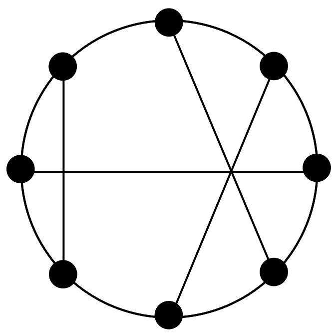
(b)



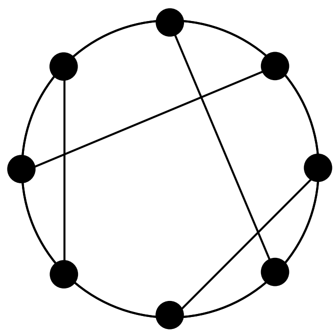
(c)



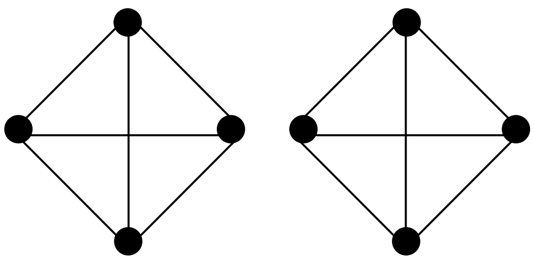
(d)



(e)



(f)



(h)

3 Basic results and particular examples

Counting edges and vertices

For this subsection G is a graph with vertices V and edges E, that is $G = (V, E)$. Recall that, for a set X, $|X|$ = the number of elements of X.

Lemma 3.1 (The Handshaking Lemma)**.**

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

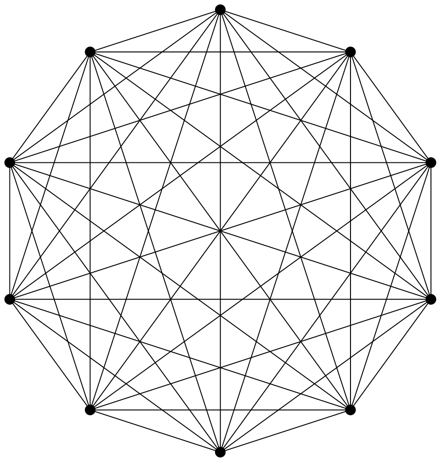
Lemma 3.2. *Suppose that G has q vertices of odd degree. Then q is even.*

Corollary 3.3. *If G has n vertices and is regular of degree d then G has $nd/2$ edges.*

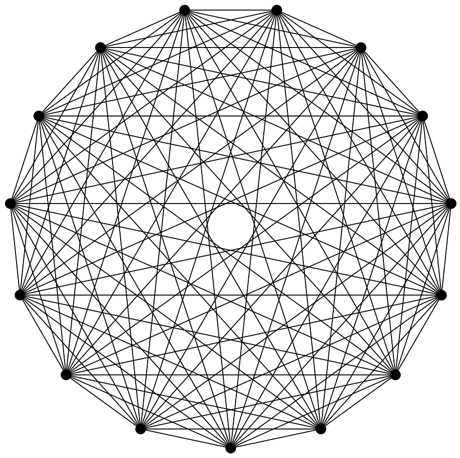
Examples of graphs

Example 3.4. The **Null** graph N_d, for $d \geq 1$.

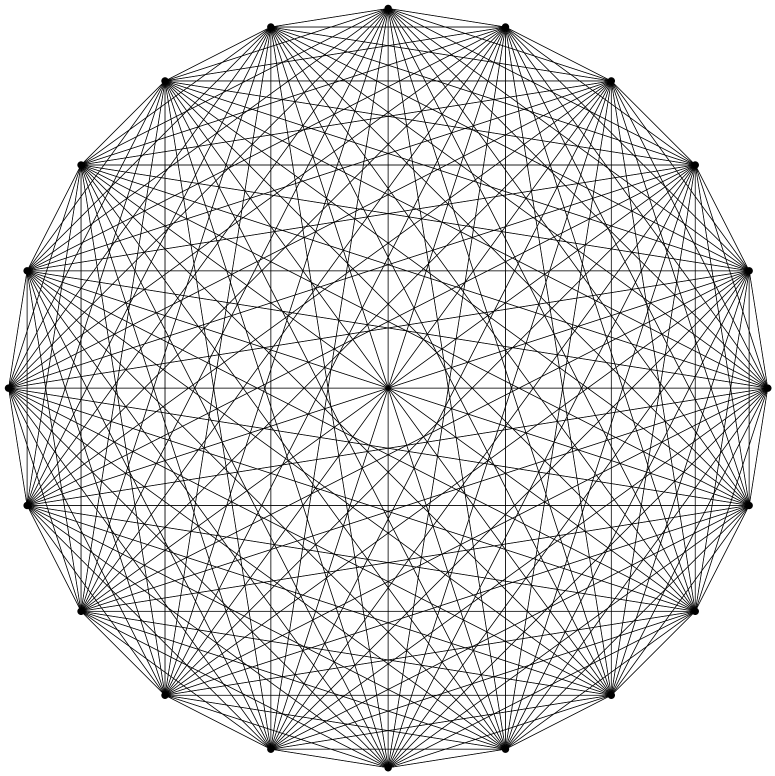
Example 3.5. The **Complete** graph K_d, for $d \geq 1$.



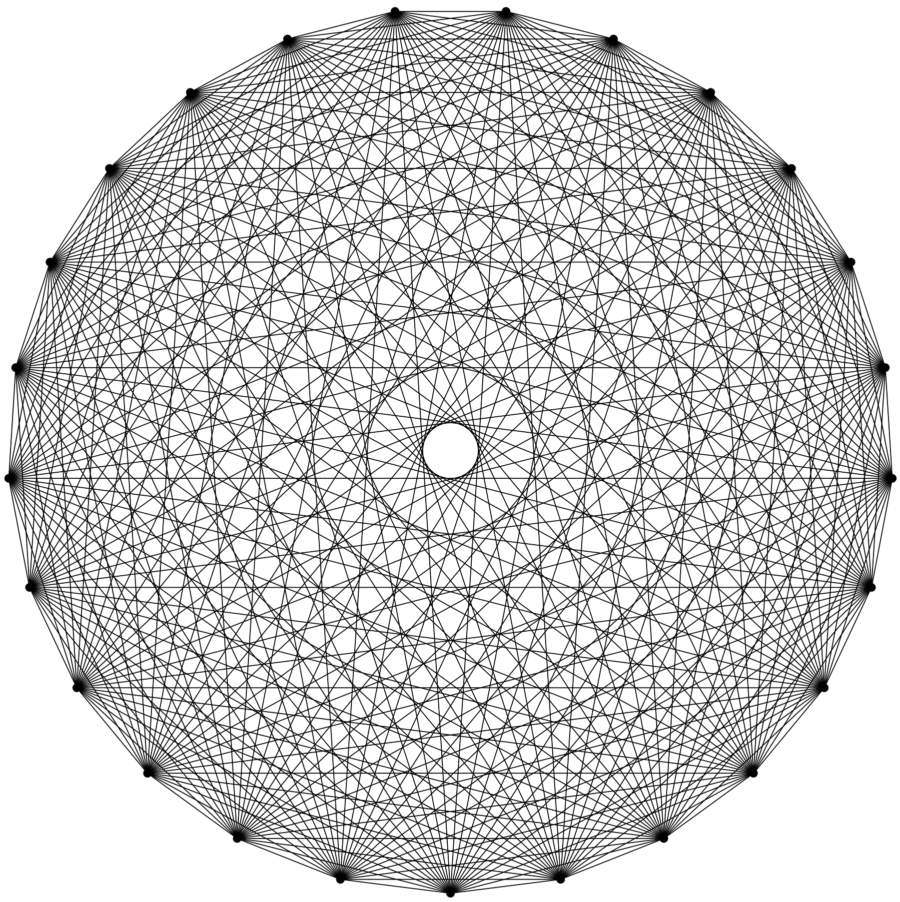
K_{10}



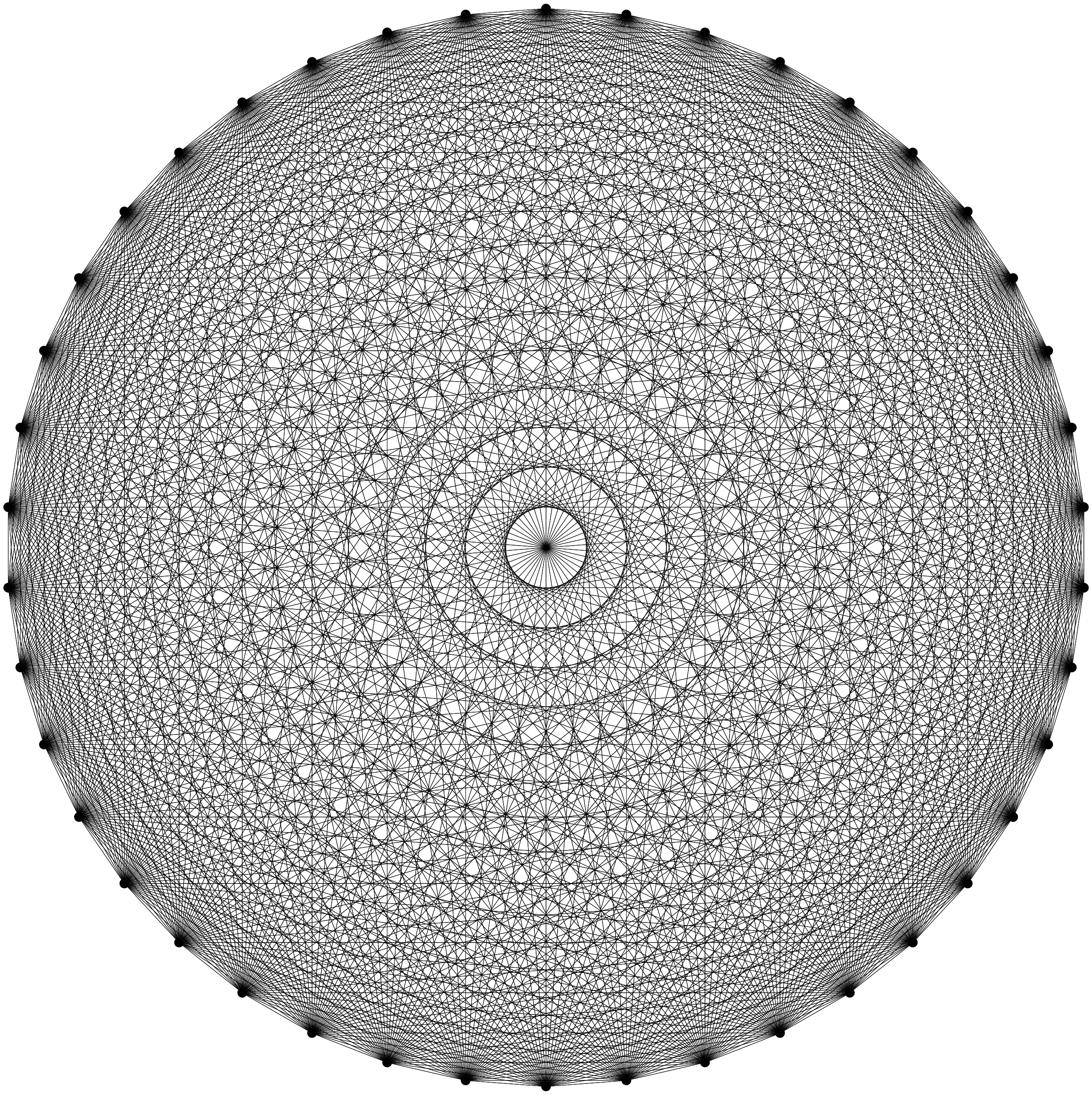
K_{15}



K_{20}



K_{25}

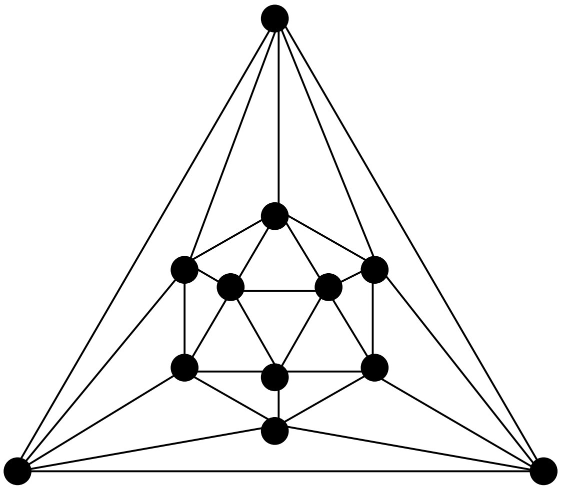


K_{42}

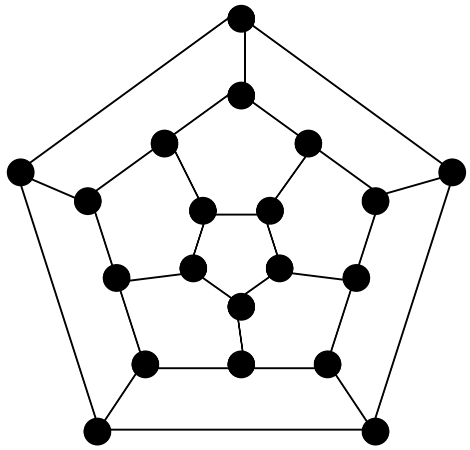
Lemma 3.6. *The complete graph K_d is regular of degree $d-1$ and has $d(d-1)/2$ edges.*

Example 3.7. The **Petersen** graph.

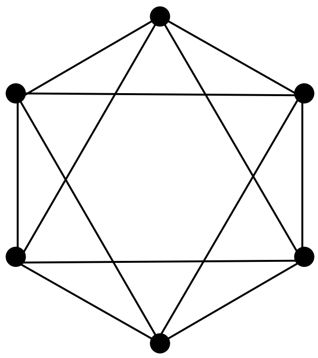
Example 3.8. The **Platonic** graphs. There are five of these graphs: called the Tetrahedron, Cube, Octahedron, Dodecahedron and Icosahedron. Each graph is based on the corresponding regular solid. Corners and edges of the regular solid correspond to vertices and edges of the graph, respectively.



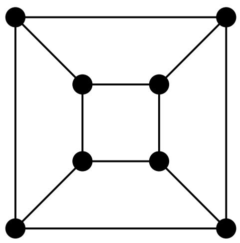
Icosahedron



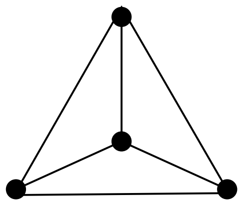
Dodecahedron



Octahedron



Cube



Tetrahedron

4 Bipartite graphs

Definition 4.1. Let $G = (V, E)$ be a graph such that

(i) $V = V_1 \cup V_2$, where V_1 and V_2 are non-empty subsets of V and

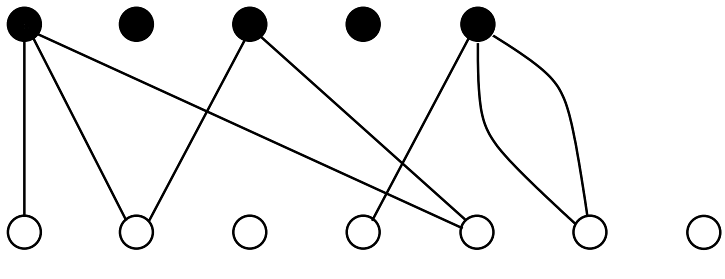
(ii) $V_1 \cap V_2 = \emptyset$ and

(iii) for $i = 1$ and 2, no edge of G joins a vertex of V_i to a vertex of V_i.

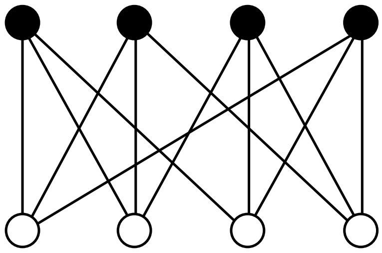
Then G is called a **bipartite** graph with **bipartition** (V_1, V_2).

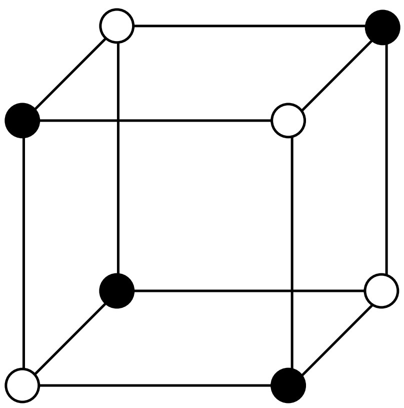
Example 4.2. 1. The Null graph N_d, where $d \geq 2$, is bipartite. (Colour one vertex blue, one vertex red and all the rest red or blue as you please.)

2. Here is a drawing of a bipartite graph with bipartition (V_1, V_2), where V_1 is white and V_2 is black. (Note that not every vertex of V_1 need be incident to a vertex of V_2.)



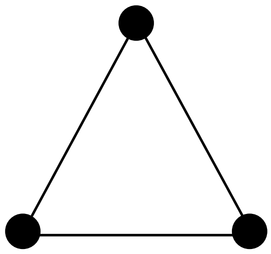
3. The cube is bipartite.





4. The octahedron is not bipartite.

5. Any graph which contains the following configuration is not bipartite.



The **Gray code** of length n

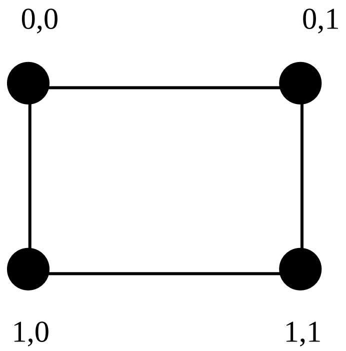
Example 4.3. Let k be any integer greater than 0. The **k-cube** Q_k is a graph whose vertex set is the set of sequences of length k of the symbols 1 and 0. That is

$$V(Q_k) = \{\langle a_1, \ldots, a_k \rangle : a_i = 0 \text{ or } 1\}.$$

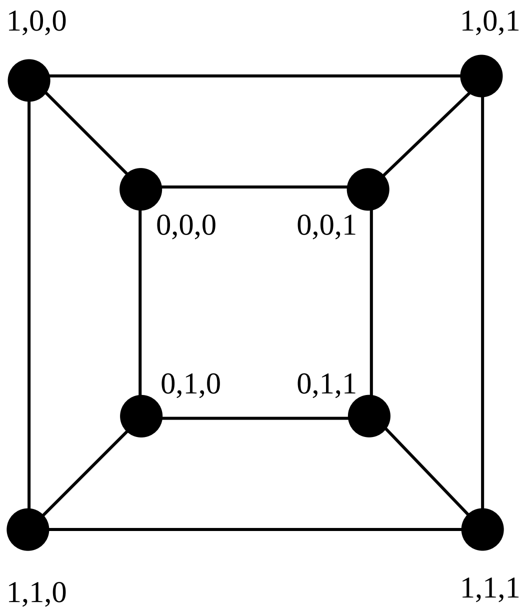
Two vertices $\langle a_1, \ldots, a_k \rangle$ and $\langle b_1, \ldots, b_k \rangle$ are joined by and edge if and only if these sequences differ in exactly one term.

$\mathbf{Q_1}$**:** $V(Q_1) = \{\langle 0 \rangle, \langle 1 \rangle\}$.

$\mathbf{Q_2}$: $V(Q_2) = \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 1,1\rangle\}$.



$\mathbf{Q_3}$: $V(Q_3) = \{000, 001, 010, 011, 100, 101, 110, 111\}$, (where the angle brackets are left out for readability).



Lemma 4.4. *1. Q_k is regular of degree k.*

2. $|E(Q_k)| = k2^{k-1}$.

3. Q_k is bipartite with bipartition (V_1, V_2) where

$$V_1 = \{\langle a_1, \ldots, a_k\rangle : \sum a_i \equiv 0 \mod (2)\}$$

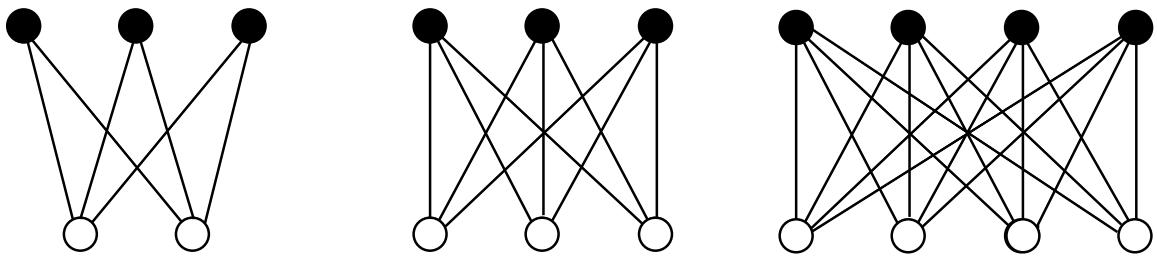
and

$$V_2 = \{\langle a_1, \ldots, a_k\rangle : \sum a_i \equiv 1 \mod (2)\}.$$

Definition 4.5. Let $r, s \in \mathbb{Z}$ with $r, s \geq 1$. The **complete bipartite** graph $K_{r,s}$ is the simple graph with bipartition (V_1, V_2), where

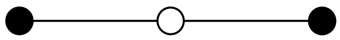
1. $|V_1| = r$ and $|V_2| = s$ and
2. every vertex of V_1 is joined to every vertex of V_2.

Example 4.6. 1. Some complete bipartite graphs:

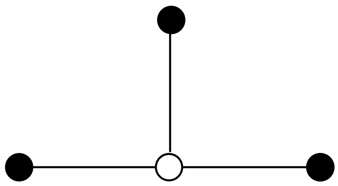


$K_{2,3}$ $K_{3,3}$ $K_{4,4}$

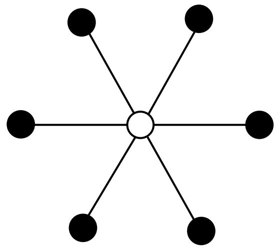
2. As a special case of the complete bipartite graphs we have the family of **star** graphs which are the graphs $K_{1,s}$, $s \geq 1$.



$K_{1,2}$



$K_{1,3}$



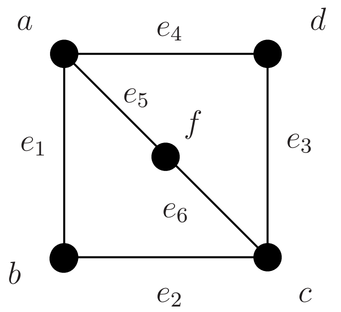
$K_{1,6}$

5 Subgraphs

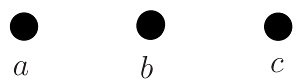
Definition 5.1. A **subgraph** of a graph $G = (V, E)$ is a graph $H = (V', E')$ such that $V' \subset V$ and $E' \subset E$.

Example 5.2. Let $G = (V, E)$, where $V = \{a, b, c, d, f\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ and

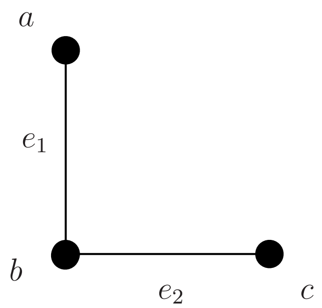
$$e_1 = \{a, b\}, e_2 = \{b, c\}, e_3 = \{c, d\}, e_4 = \{d, a\}, e_5 = \{a, f\}, e_6 = \{f, c\}.$$



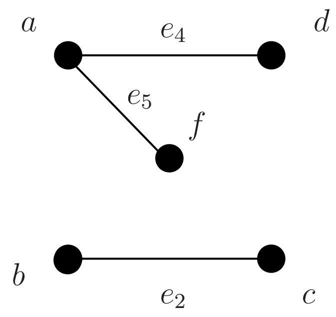
1. $H_1 = (V_1, E_1)$, where $V_1 = \{a, b, c\}$ and $E_1 = \emptyset$ is a subgraph of G.



2. $H_2 = (V_2, E_2)$, where $V_2 = \{a, b, c\}$ and $E_1 = \{e_1, e_2\}$ is a subgraph of G.



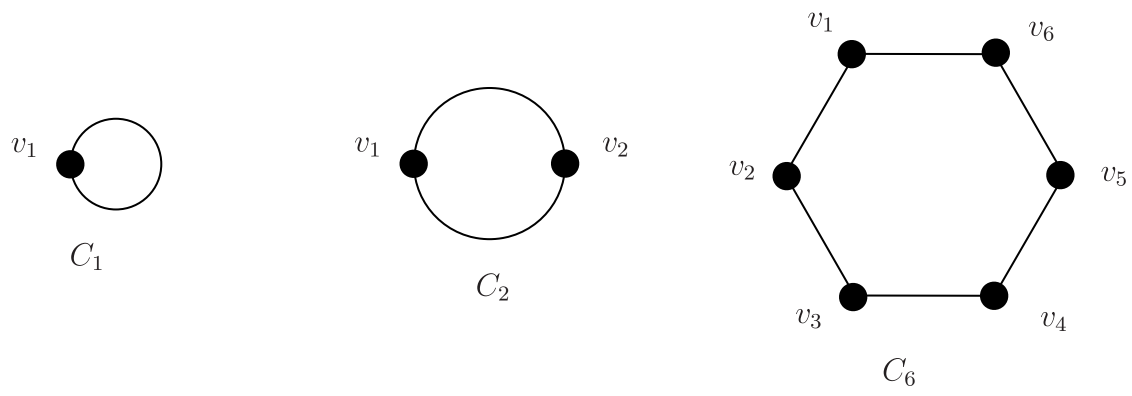
3. $H_3 = (V_3, E_3)$, where $V_3 = \{a, b, c, d, f\}$ and $E_1 = \{e_2, e_4, e_5\}$ is a subgraph of G.



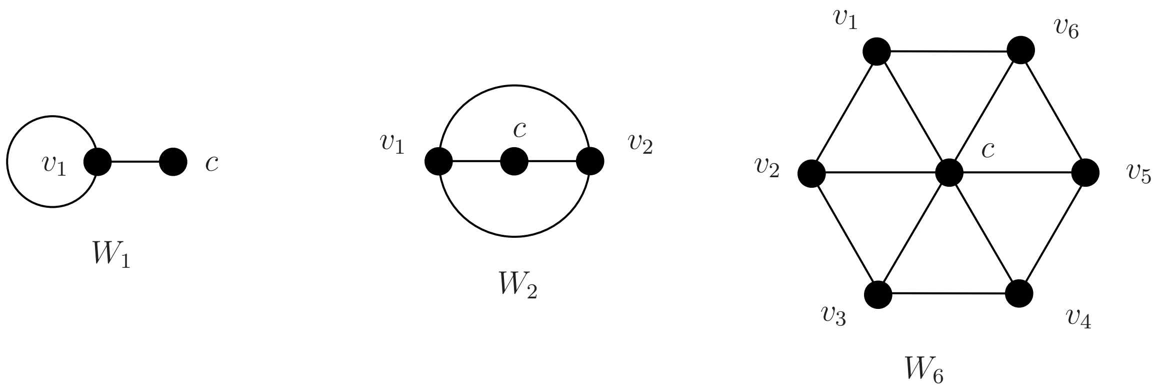
4. G is a subgraph of G.
5. $H_4 = (V_4, E_4)$, where $V_4 = \{a, b, c, x, y\}$ and $E_1 = \{e_2, e_4, e_5\}$ is not a subgraph of G, because V_4 is not a subset of V.
6. $H_5 = (V_5, E_5)$, where $V_5 = \{a, b, c, d\}$ and $E_5 = \{e_7, e_8\}$ with $e_7 = \{b, f\}, e_8 = \{a, c\}$ is not a subgraph of G, because E_5 is not a subset of E.
7. $H_6 = (V_6, E_6)$, where $V_6 = \{a, b, c, d\}$ and $E_6 = \{e_5\}$ is not a subgraph of G, because it is not a graph ($e_5 = \{a, f\}$ is not a pair of elements of V_6).

The last 3 examples illustrate the 3 possible ways in which H may fail to be a subgraph of G.

Example 5.3. 1. For $d \geq 1$ we define the **cycle graph** C_d to be the graph with d vertices $v_1, \ldots, v_d$ and d edges $\{v_1, v_2\}, \ldots, \{v_{d-1}, v_d\}, \{v_d, v_1\}$. ($C_1$ has one vertex v_1 and one edge $\{v_1, v_1\}$.) The cycle graph is regular of degree 2 and simple if $d \geq 3$.



2. For $d \geq 1$ we define the **wheel graph** W_d to be the graph with $d+1$ vertices $c, v_1, \ldots, v_d$ and $2d$ edges $\{v_1, v_2\}, \ldots, \{v_{d-1}, v_d\}, \{v_d, v_1\}, \{c, v_1\}, \ldots \{c, v_d\}$.



The wheel graph has a subgraph isomorphic to the cycle graph C_d and a subgraph isomorphic to the star graph $K_{1,d}$. For $d \geq 3$ it is simple.

6 Walks, trails, paths, circuits and cycles

Throughout this section let $G = (V, E)$ be a graph.

Definition 6.1. A sequence

$$v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n,$$

where

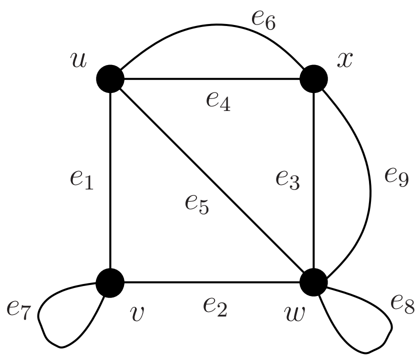
(i) $n \geq 0$ and

(ii) $v_i \in V$ and $e_i \in E$ and

(iii) $e_i = \{v_{i-1}, v_i\}$, for $i = 1, \ldots, n$,

is called a **walk** of **length** n. The walk is **from** its **initial** vertex v_0 and **to** its **terminal** vertex v_n.

Example 6.2. *G is the graph drawn below.*



Definition 6.3. Let $W = v_0, e_1, v_1, \dots, e_n, v_n$ be a walk in a graph.

(i) If no edges of the walk are repeated (that is $e_i \neq e_j$ when $i \neq j$) then W is called a **trail**.

(ii) If no vertices of the walk are repeated (that is $v_i \neq v_j$ when $i \neq j$) then W is called an **open path**.

(iii) If $v_0 = v_n$ then W is a **closed** walk.

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

(v) If W is a closed trail and no two of the vertices $v_0, \dots, v_{n-1}$ are the same then W is called a **closed path**. [Note that from the definition it follows that $v_n \neq v_i$, for $1 \leq i \leq n-1$.]

(vi) We refer to both open and closed paths as **paths**.

(vii) A closed path of length at least 1 is called a **cycle**.

Example 6.4. Consider the following walks in the graph of Example 6.2.

7. the walk $v, e_1, u, e_4, x, e_6, u, e_5, w$.
8. the walk $w, e_3, x, e_4, u, e_1, v$.
9. the walk $u, e_4, x, e_9, w, e_3, x, e_6, u$.

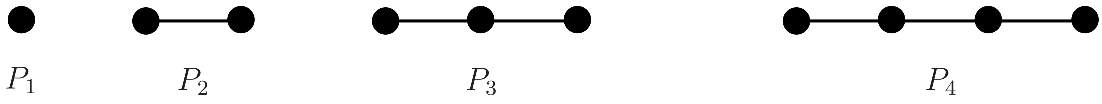
The table below shows which of the definitions applies to each of the sequences of Examples 6.2 and 6.4.

	walk	**trail**	**open path**	**closed walk**	**circuit**	**closed path**	**cycle**	**length**
1								
2								
3								
4								
5								
6								
7								

Note that an open path may be a closed walk, as in 3 above, but only when it has length 0.

Example 6.5. 1. The cycle graph C_d consists of a cycle of length d.

2. The **path graph** P_n, for $n \geq 1$, is the graph with n vertices $v_1, \ldots, v_n$ and $n-1$ edges $e_2, \ldots, e_n$, with $e_i = \{v_{i-1}, v_i\}$, for $i = 2, \ldots, n$. The path graph P_n consists of an open path of length n.



Walks in simple graphs

If G is a simple graph then, to simplify notation, we may describe a walk by writing only the subsequence of vertices, which we call the **vertex sequence** of the walk. For example the sequence

$$v_1, c, v_5, v_4, c, v_2$$

is the vertex sequence of a unique walk in the wheel graph W_6 shown above.

7 Connectedness

Definition 7.1. A graph is **connected** if, for any two vertices a and b there is an open path from a to b. A graph which is not connected is called **disconnected**.

Lemma 7.2. *There is an open path from a to b if and only if there is a walk from a to b.*

Example 7.3.

Definition 7.4. A **connected component** of a graph G is a subgraph H of G such that

1. H is a connected subgraph of G and
2. H is not contained in any larger connected subgraph of G.

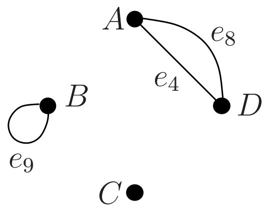
Example 7.5. 1. A connected graph has only one connected component – itself.

2. The graph G_3 of Example 1.3 (5) has 3 connected components:

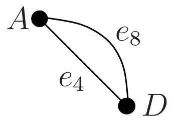
   H_1**:** the graph with vertices A, D and edges e_4, e_8;

   H_2**:** the graph with vertex B and edge e_9;

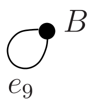
   H_3**:** the graph with vertex C and no edges.



G_3



H_1

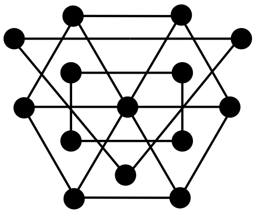


H_2

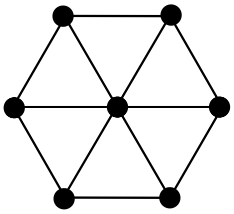
H_3

3. The null graph N_d has d connected components, each with 1 vertex.

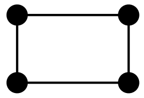
4. The graph G on the left has 3 connected components A, B and C, as shown.



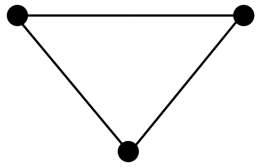
G



A



B

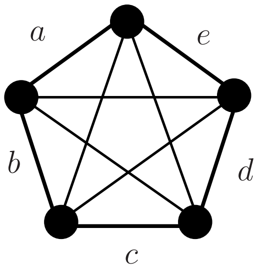


C

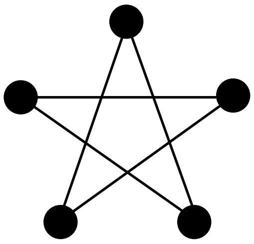
Definition 7.6. Let $G = (V, E)$ be a graph and let E' be a subset of E. The graph with vertex set V and edge set $E - E'$ is called the graph obtained from G by **deleting** E', denoted $G - E'$. When E' consists of only one element we write $G - e$ instead of $G - \{e\}$.

Example 7.7. In this example we use letters $a, b, c, \ldots, o, p, q$ to denote edges of the graphs shown.

1.

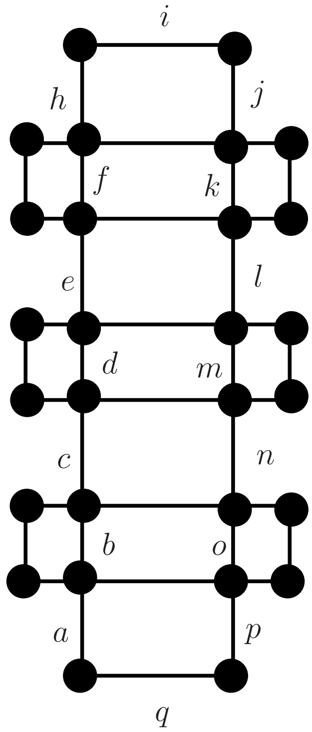


G_1

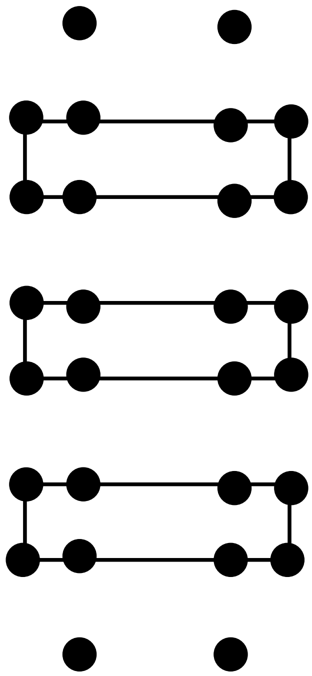


$G_1 - \{a, b, c, d, e\}$

2.

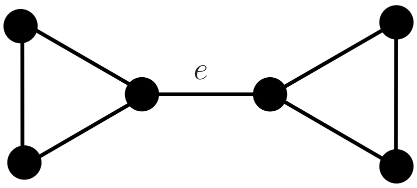


G_2

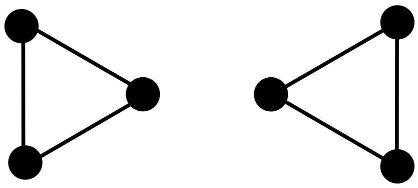


$G_2 - \{a, \ldots, q\}$

3.



G_3



$G_3 - e$

Lemma 7.8. *Let G be a connected graph, C a circuit in G and e an edge of C. Then $G - e$ is connected.*

8 Eulerian graphs

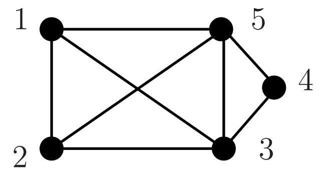
Definition 8.1. 1. A trail containing every edge of a graph is called an **Eulerian trail**.

2. A circuit containing every edge of a graph is called an **Eulerian circuit**.

3. A graph is called **semi-Eulerian** if it is connected and has an Eulerian trail.

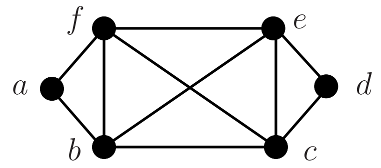
4. A graph is called **Eulerian** if it is connected and has an Eulerian circuit.

Example 8.2. 1. The walk $1, 2, 3, 1, 5, 4, 3, 5, 2$ is a semi-Eulerian trail in the graph G_1 below. Therefore G_1 is semi-Eulerian. Does G_1 have an Eulerian circuit?



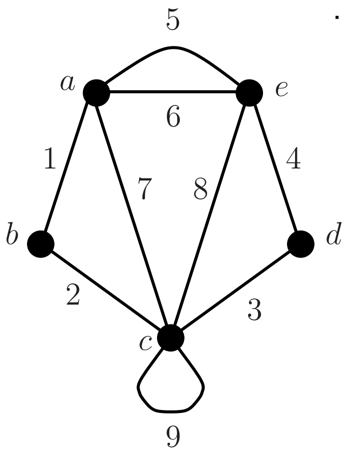
G_1

2. The walk $a, b, c, d, e, f, b, e, c, f, a$ is an Eulerian circuit in the graph G_2 below. Therefore G_2 is Eulerian.



G_2

3. The walk $a, 1, b, 2, c, 3, d, 4, e, 5, a, 7, c, 9, c, 8, e, 6, a$ is an Eulerian circuit in the graph G_3 below.



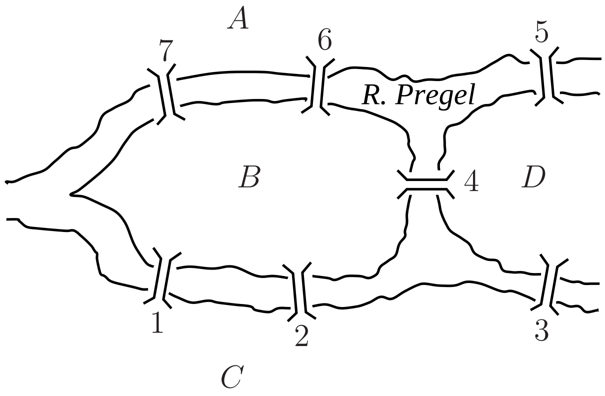
G_3

Consider the vertex c in Example 8.2.3. Each occurrence of c appears between two edges. As all edges of the graph appear in the Eulerian circuit we can compute the degree of c as twice the number of times it occurs in the sequence.

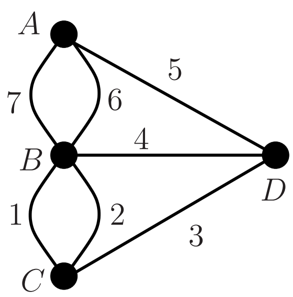
Theorem 8.3 (Euler, 1736)**.** *If G is an Eulerian graph then every vertex of G has even degree.*

Example 8.4. 1. There is no Eulerian circuit for the graph of Example 8.2.1 above: this graph has vertices of odd degree.

2. The Königsberg bridge problem.



The River Pregel in Königsberg



A graph of the Königsberg bridges

3. Of the Platonic graphs only the Octahedron can be Eulerian. Can you find an Eulerian circuit for the Octahedron?

4. The graph K_d is not Eulerian if d is even.

Lemma 8.5. *Let G be a graph such that every vertex of G has even degree. If $v \in V(G)$ with $\deg(v) > 0$ then v lies in a circuit of positive length.*

Theorem 8.6. *Let G be a connected graph. Then G is Eulerian if and only if every vertex of G has even degree.*

Proof. We have already seen (Theorem 8.3) that if G is Eulerian then every vertex of G is of even degree.

Suppose then that every vertex of G has even degree. Let C be a circuit of maximal length in G. If C contains every edge of G then it's an Eulerian circuit and so G is Eulerian as required. We assume that C does not contain all edges of G and derive a contradiction.

Assume that E' is the set of edges of C and that there are some edges of G not in E'. Consider the graph $G - E'$. First of all, every vertex of $G - E'$ has even degree (as C is a circuit). Also, although $G - E'$ may be a disconnected graph it does have a connected component, H say, with at least one edge. Furthermore, as G is connected it follows that C and H have a vertex, a say, in common. Now a has positive degree and so, by Lemma 8.5, is contained in a circuit D, of positive length, in H.

Let C be

$$v_0, e'_1, \dots, e'_i, a, e'_{i+1}, \dots, v_m$$

and let D be

$$u_0, e_1, \dots, e_j, a, e_{j+1}, \dots, u_n.$$

Then

$$v_0, e'_1, \dots, e'_i, a, e_{j+1}, \dots, u_n = u_0, e_1, \dots, e_j, a, e'_{i+1}, \dots, v_m$$

is a circuit in G of length greater than that of C (see Figure 8.1). However this contradicts the choice of C as a circuit of maximal length in G. Therefore C must contain all edges of G as required. □

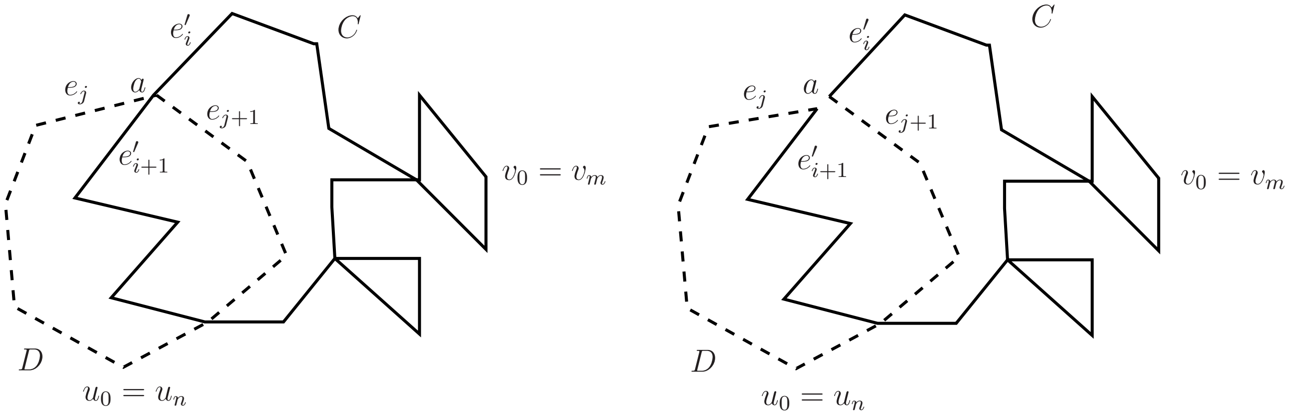
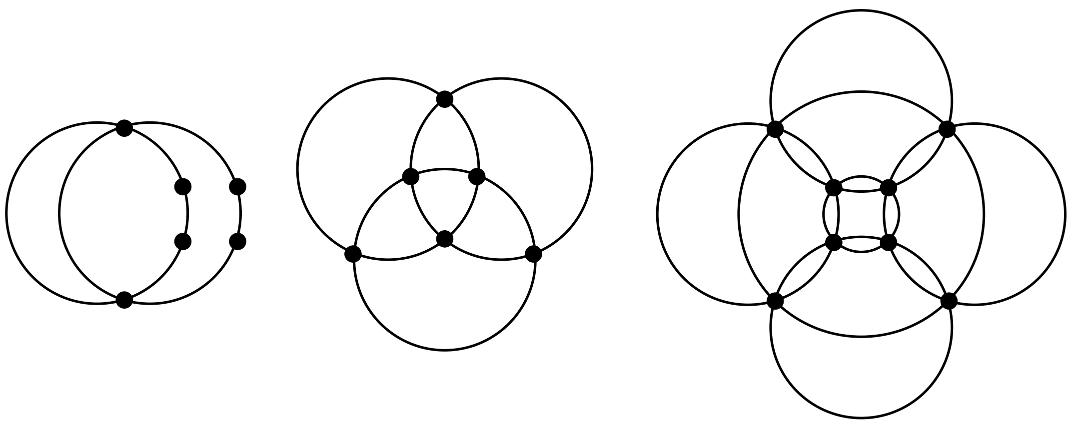


Figure 8.1: Construction of a circuit in Theorem 8.6

Example 8.7.



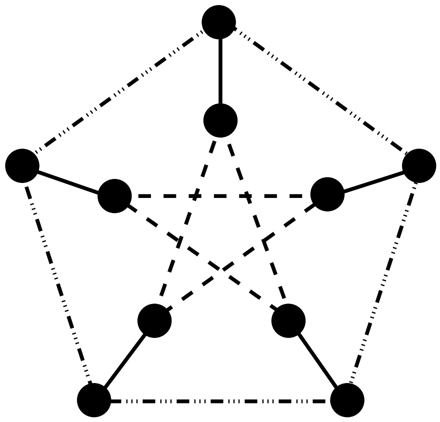
Definition 8.8. A graph $G = (V, E)$ has a **decomposition into subgraphs** $H_1 = (V_1, E_1), \dots, H_n = (V_n, E_n)$ if

1. $E = E_1 \cup \dots \cup E_n$ and
2. $E_i \cap E_j = \emptyset$, when $i \neq j$.

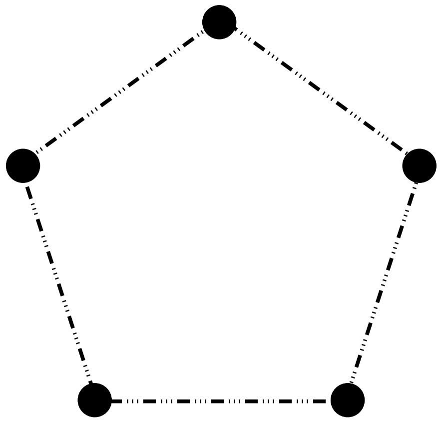
If G has a decomposition into subgraphs $H_1, \dots, H_n$, where H_i is a cycle graph or a null graph, for all i, we say G has a **decomposition into closed paths**. (See Example 5.3.1, Notes page 28, for the definition of a cycle graph.)

Example 8.9.

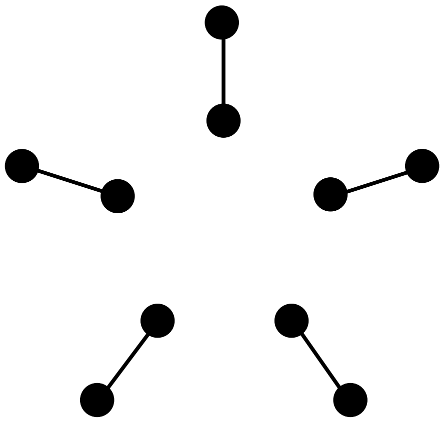
1. The Octahedron, the 4-cube Q_4, the graphs of Example 7.7.1 and 7.7.2 and all the graphs of Example 8.7.5 have decompositions into closed paths. Isolated vertices may occur in a decomposition into closed paths. In particular the Null graph N_d has a decomposition into closed paths.
2. A decomposition of the Petersen graph into subgraphs H_1, H_2 and H_3.



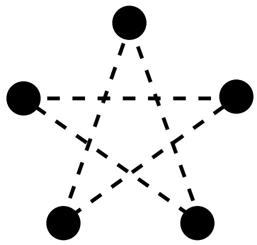
The Petersen graph



H_1

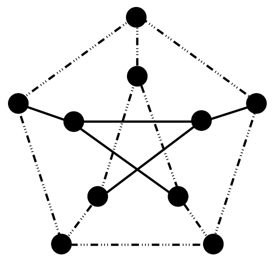


H_2

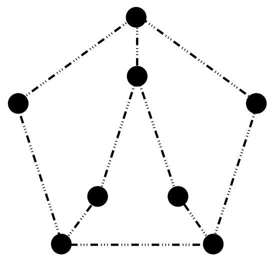


H_3

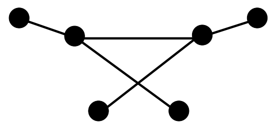
3. A different decomposition of the Petersen graph into subgraphs A_1 and A_2.



The Petersen graph

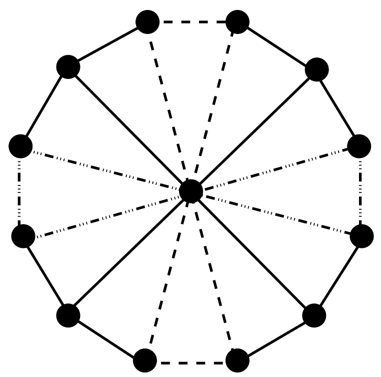


A_1

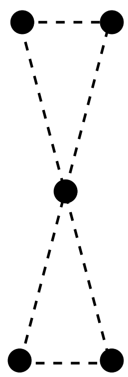


A_2

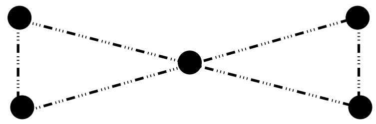
4. Another example of a decomposition into 3 subgraphs B_1, B_2 and B_3.



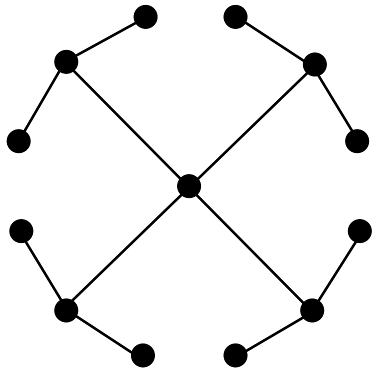
The wheel graph W_{12}



B_1



B_2



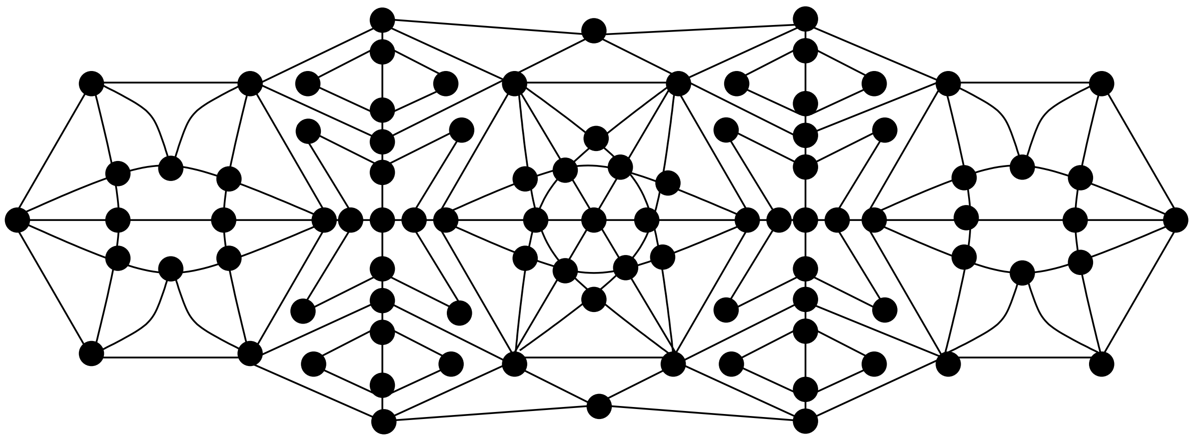
B_3

Theorem 8.10. *A graph G has a decomposition into closed paths if and only if every vertex of G has even degree.*

Corollary 8.11. *A connected graph is Eulerian if and only if it has a decomposition into closed paths.*

Theorem 8.12. *A connected graph is semi-Eulerian but not Eulerian if and only if precisely* 2 *of its vertices have odd degree.*

Example 8.13. The following graph has exactly 2 vertices of odd degree and is therefore semi-Eulerian.



9 Hamiltonian graphs

Definition 9.1.

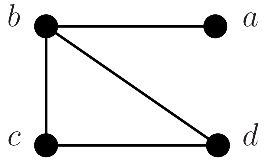
1. A path containing every vertex of a graph is called a **Hamiltonian path**.

2. A closed path containing every vertex of a graph is called a **Hamiltonian closed path**.

3. A graph is called **semi-Hamiltonian** if it has a Hamiltonian path and **Hamiltonian** if it has a Hamiltonian closed path.

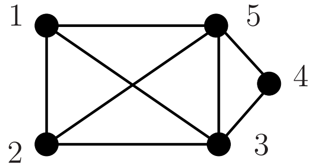
Example 9.2.

1. The walk a, b, c, d is a Hamiltonian path in the graph G_1 below. Therefore G_1 is semi-Hamiltonian.



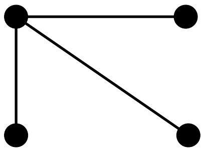
G_1

2. The walk $1, 2, 3, 4, 5, 1$ is a Hamiltonian closed path in the graph G_2 below. Therefore G_2 is Hamiltonian.



G_2

3. The graph G_3 below is not semi-Hamiltonian (and therefore not Hamiltonian).



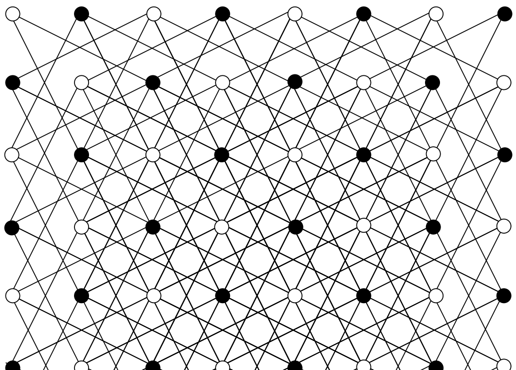
G_3

4. The complete graph K_2 is semi-Hamiltonian but not Hamiltonian. For $d \neq 2$ the graphs K_d are Hamiltonian.

5. The cycle graphs are Hamiltonian for $d \geq 1$.

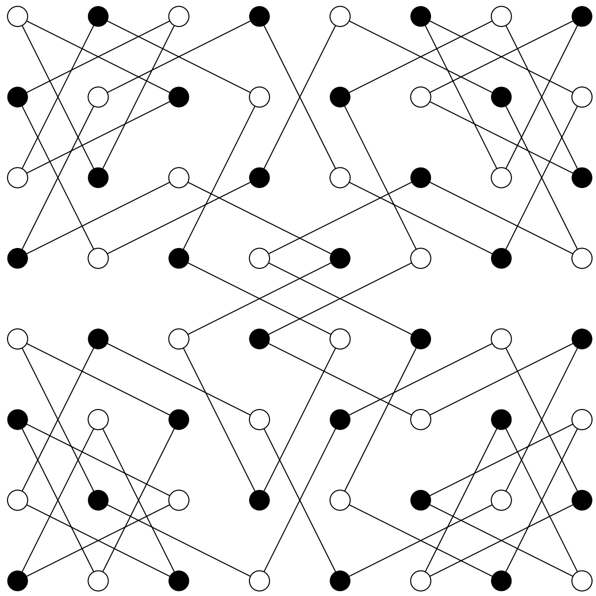
6. The wheel graph W_d is Hamiltonian for $d \geq 2$. The wheel graph W_1 is semi-Hamiltonian but not Hamiltonian.

7. Construct a graph with one vertex corresponding to each square of a chess-board and an edge joining two vertices if a knight can move from one to the other. We call this the **knight's move graph**.



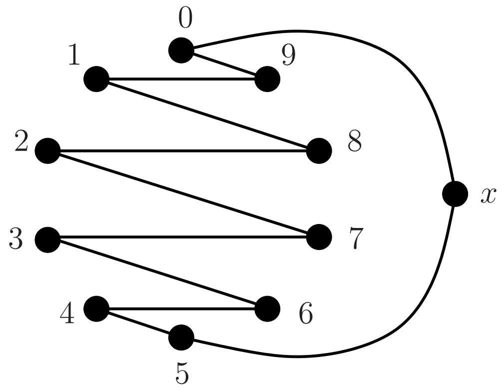
The knight's move graph

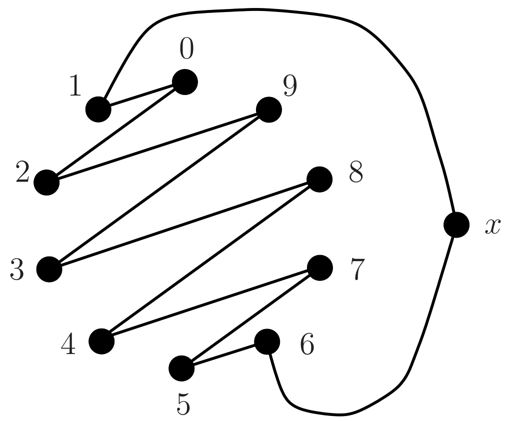
Here is a Hamiltonian closed path for the knight's move graph.



We say a graph G has a **decomposition into Hamiltonian closed paths** if it has a decomposition into subgraphs each of which forms a Hamiltonian closed path in G.

Example 9.3. Consider the complete graph K_{11}.


0
1
9
2
8
x
3
7
4
6
5



We continue this process, turning the zig-zag one position anti-clockwise each time to obtain 3 further closed paths

$$C_3 = 2, 1, 3, 0, 4, 9, 5, 8, 6, 7, x, 2,$$
$$C_4 = 3, 2, 4, 1, 5, 0, 6, 9, 7, 8, x, 3 \text{ and}$$
$$C_5 = 4, 3, 5, 2, 6, 1, 7, 0, 8, 9, x, 4.$$

We now have the required decomposition of K_{11} into 5 Hamiltonian closed paths.

Theorem 9.4. *The complete graph K_{2d+1} has a decomposition into d Hamiltonian closed paths.*

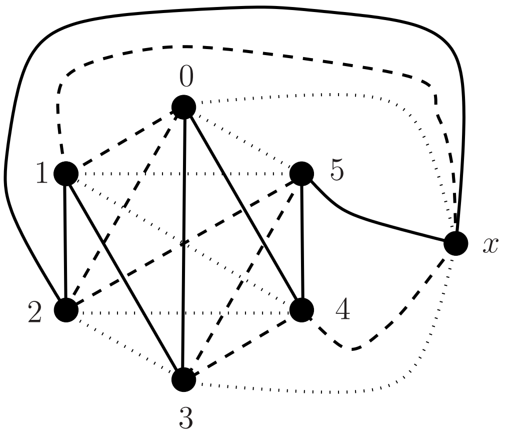
Proof. The method of the above example (which is called *the turning trick*) can be used to prove the general case. Note that K_{2d+1} has $(2d+1)2d/2 = (2d+1)d$ edges. A Hamiltonian decomposition, if it exists, must therefore involve d Hamiltonian closed paths. We label the vertices $0, 1, \ldots, 2d-1, x$. Our closed paths are

$$\begin{array}{lcccccccccc}
C_1 & 0, & 2d-1, & 1, & 2d-2, & 2 & \dots & d-1, & d, & x, & 0 \\
C_2 & 1, & 0, & 2, & 2d-1, & 3 & \dots & d, & d+1, & x, & 1 \\
C_3 & 2, & 1, & 3, & 0, & 4 & \dots & d+1, & d+2, & x, & 2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\
C_{d-1} & d-2, & d-3, & d-1, & d-4, & d & \dots & 2d-3, & 2d-2, & x, & d-2 \\
C_d & d-1, & d-2, & d, & d-3, & d+1 & \dots & 2d-2, & 2d-1, & x, & d-1
\end{array}$$

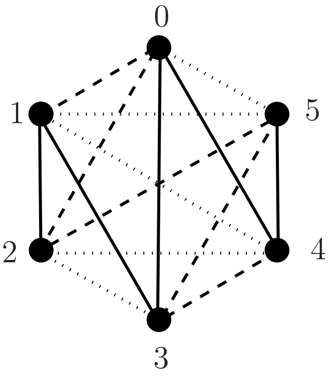
There are d of these closed paths and they have no edges in common. Therefore we have the required decomposition into d Hamiltonian closed paths. □

Corollary 9.5. *The complete graph K_{2d} has a decomposition into d Hamiltonian paths.*

Example 9.6. The drawings below show the decomposition of K_7 into 3 Hamiltonian closed paths, given by Theorem 9.4 and the corresponding decomposition of K_6 into 3 Hamiltonian paths.



A decomposition of K_7 into
Hamiltonian closed paths



A decomposition of K_6 into
Hamiltonian paths

Example 9.7.

This example generalises to give the following theorem.

Theorem 9.8. *The complete graph K_{2d} has a decomposition into $2d-1$ paths of lengths $1, 2, \ldots, 2d-1$.*

Theorem 9.9. *Let G be a simple graph with $n \geq 3$ vertices. Suppose*

$$\deg(u) + \deg(v) \geq n,$$

whenever u and v are vertices of G which are not adjacent. Then G is Hamiltonian.

Proof. (Non-examinable) The proof is by contradiction. Assume there is a graph G which satisfies the given conditions but is non-Hamiltonian. Note that G cannot be a complete graph as these are all Hamiltonian. We can therefore add an edge to G to form a new simple graph. If we find that adding this edge gives a Hamiltonian graph then we know that G is semi-Hamiltonian. Otherwise we replace G with the non-Hamiltonian graph with the new edge added. This graph still satisfies all the given conditions and is non-Hamiltonian. We carry on this process of adding an edge until the stage where the addition of one more edge would make the graph Hamiltonian. We now have a semi-Hamiltonian graph G which satisfies all the given conditions but is not Hamiltonian.

Suppose P is a Hamiltonian path for G, say P has vertex sequence

$$v_1, \ldots, v_n.$$

If v_1 and v_n are adjacent then (as $n \geq 3$) we can extend this path to a Hamiltonian closed path. As G is non-Hamiltonian it follows therefore that v_1 and v_n are not adjacent. Thus $\deg(v_1) + \deg(v_n) \geq n$, by assumption. Suppose $\deg(v_1) = r$ so

$$\deg(v_n) \geq n - r. \tag{9.1}$$

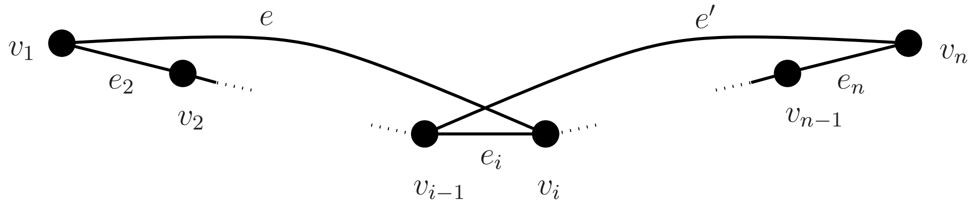
Let $i_1 < i_2 < \cdots < i_r$ be integers such that the elements $v_{i_1}, v_{i_2}, \ldots, v_{i_r}$ of the sequence $P = v_1, \ldots, v_n$ are the r vertices incident to v_1. Let

$$S = \{v_{i_1-1}, v_{i_2-1}, \ldots, v_{i_r-1}\}.$$

Note that $v_n \notin S$, $v_1 = v_{i_1-1}$ and $|S| = r$. Now assume that v_n is not adjacent to any of the vertices of S. As G has n vertices and $v_n \notin S$, it follows that v_n is adjacent to at most $n - r - 1$ vertices. Hence $\deg(v_n) \leq n - r - 1$, contradicting equation (9.1). Therefore v_n must be adjacent to a vertex of S. That is, there is some integer i such that P is the sequence

$$v_1, \ldots, v_{i-1}, v_i, \ldots, v_n,$$

with v_1 adjacent to v_i and v_n adjacent to v_{i-1}, as illustrated in the diagram below.

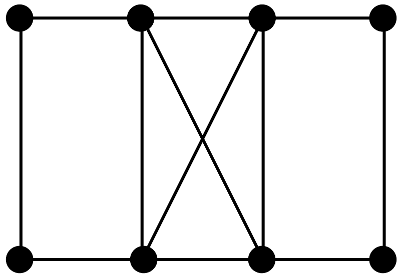


Now, using the notation of the diagram

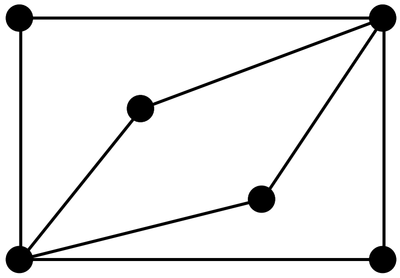
$$v_1, e_2, v_2, \ldots, v_{i-1}, e', v_n, e_n, v_{n-1}, \ldots, v_i, e, v_1$$

is a Hamiltonian closed path for G. This contradicts the assumption that G is non-Hamiltonian. Hence no such G exists and the proof is complete. □

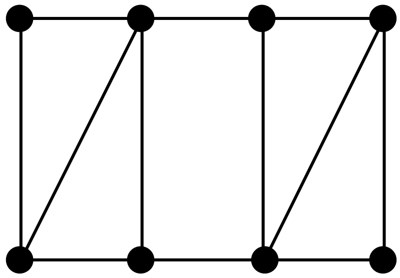
Example 9.10. Note that although there are theorems relating Eulerian and Hamiltonian graphs there do exist graphs with any combination of these properties:



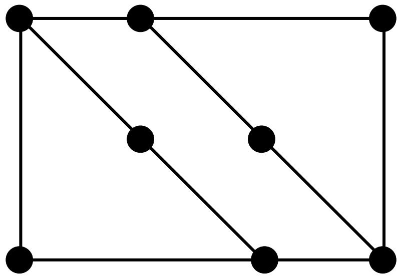
A graph which is Hamiltonian and Eulerian



A graph which is Eulerian and non-Hamiltonian



A graph which is Hamiltonian and non-Eulerian



A graph which is non-Eulerian and non-Hamiltonian

10 Trees

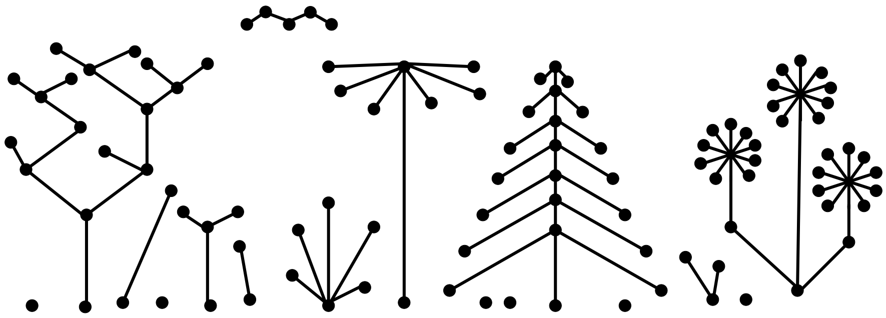
Recall that a cycle is a closed path of length at least 1.

Definition 10.1.

1. A **forest** is a graph with no cycle.
2. A **tree** is a connected graph with no cycle.

Example 10.2.

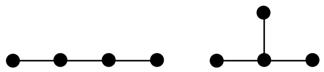
1. A forest:



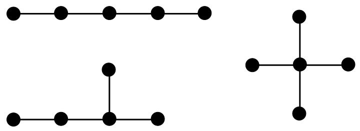
2. The graphs of Example 4.6.2 are all trees.
3. The path graph is a tree for all $n \geq 1$.

Example 10.3.

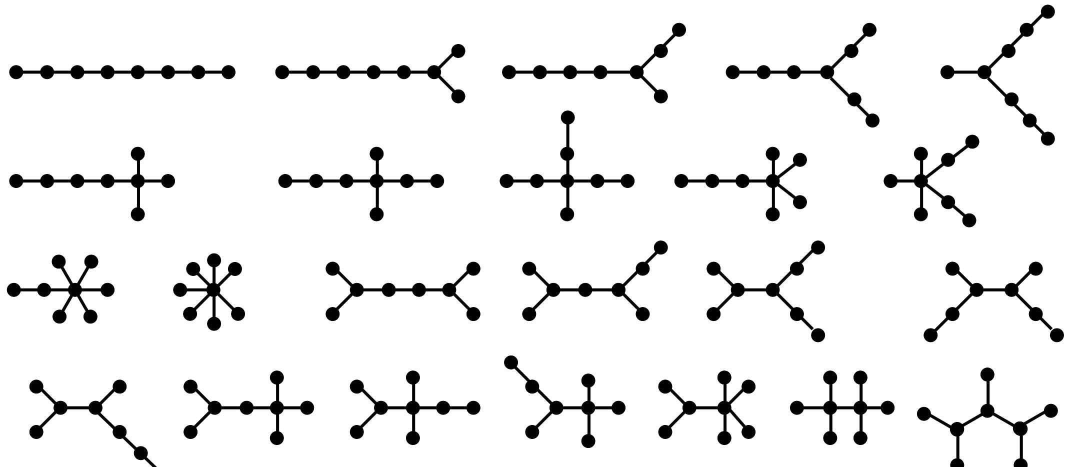
1. There is only one tree with one vertex, $N_1 = P_1$. There is only one tree with 2 vertices, $K_2 = P_2$. There is only one tree with 3 vertices, namely P_3.
2. There are 2 trees with 4 vertices:



3. There are 3 trees with 5 vertices.



4. There are 6 trees with 6 vertices and 11 trees with 7 vertices (see the Exercises).
5. There are 23 trees with 8 vertices:



There are several possible ways of formulating the definition of a tree. Starting from the above definition we can prove the next theorem, which could have been used as the definition. First we recall the result of Exercise 2.20.

Lemma 10.4. *If a graph G contains two distinct paths from vertices u to v then G contains a cycle.*

Proof. Amongst all pairs of distinct paths with the same initial and terminal vertices choose a pair such that the sum of their lengths is minimal. Suppose this is the pair of paths

$$p = u_0, \dots, u_m \text{ and } q = v_0, \dots v_n,$$

where $u_0 = v_0$ and $u_m = v_n$. Suppose that $u_i = v_j$ for some i, j, with $0 < i < m$ and $0 \le j \le n$. Then either there is a pair of distinct paths from u_0 to u_i or from u_i to u_m which have smaller lengths than p and q. This contradicts our choice of p and q, so cannot occur. It follows that $u_0, u_1, \dots, u_m = v_n, v_{n-1}, \dots, v_1, v_0 = u_0$ is a cycle. □

Theorem 10.5. *A graph G is a tree if and only if*

(i) G has no loops and

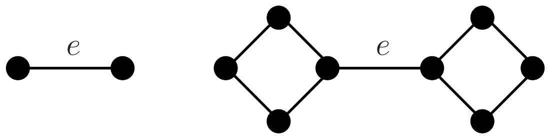
(ii) there is exactly one open path from u to v, for all pairs u, v of vertices of G.

Another characterising property of trees can be obtained with the help of the following definition.

Definition 10.6. An edge e of a graph G is a **bridge** if $G - e$ has more connected components than G.

Example 10.7.

1. In these diagrams the edge e is a bridge:



2. If e is an edge of a cycle then e is not a bridge (Lemma 7.8).

Theorem 10.8. *A graph G is a tree if and only if G is connected and every edge of G is a bridge.*

Lemma 10.9. *Let G be a connected graph with m edges and n vertices. Then $n \leq m + 1$. Furthermore G contains a cycle if and only if $n < m + 1$.*

Theorem 10.10. *Let G be a connected graph with n vertices and m edges. Then G is a tree if and only if $n = m + 1$.*

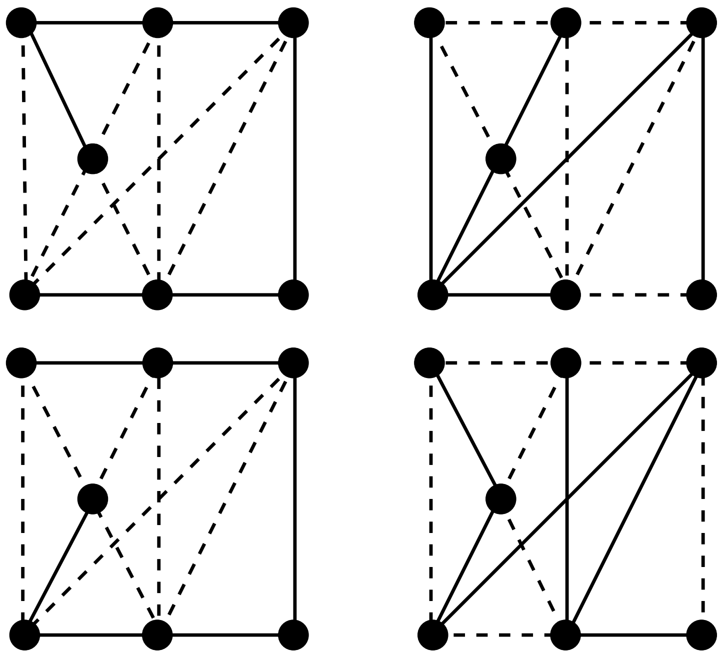
This follows immediately from Lemma 10.9

Definition 10.11. Let G be a graph. A **spanning tree** for G is a subgraph of G which

1. is a tree and
2. contains every vertex of G.

A graph which has a spanning tree must be connected. A graph may have many different spanning trees.

Example 10.12. In the diagrams below the solid lines indicate some of the spanning trees of the graph shown: there are many more.



Theorem 10.13. *Every connected graph has a spanning tree.*

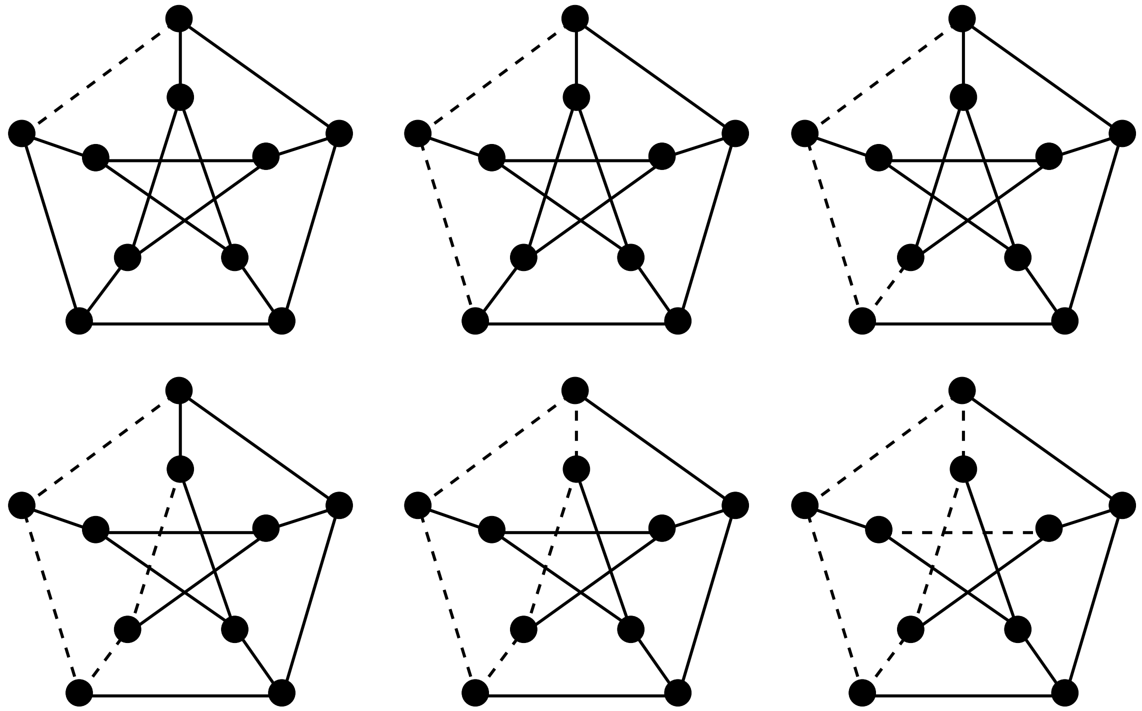
The above proof suggests an algorithm for construction of a spanning tree of a graph.

The cut-down algorithm
Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 10.14. Starting with the Petersen graph:



Proof that this process results in a spanning tree is contained in the proof of Theorem 10.13. Another approach is the following.

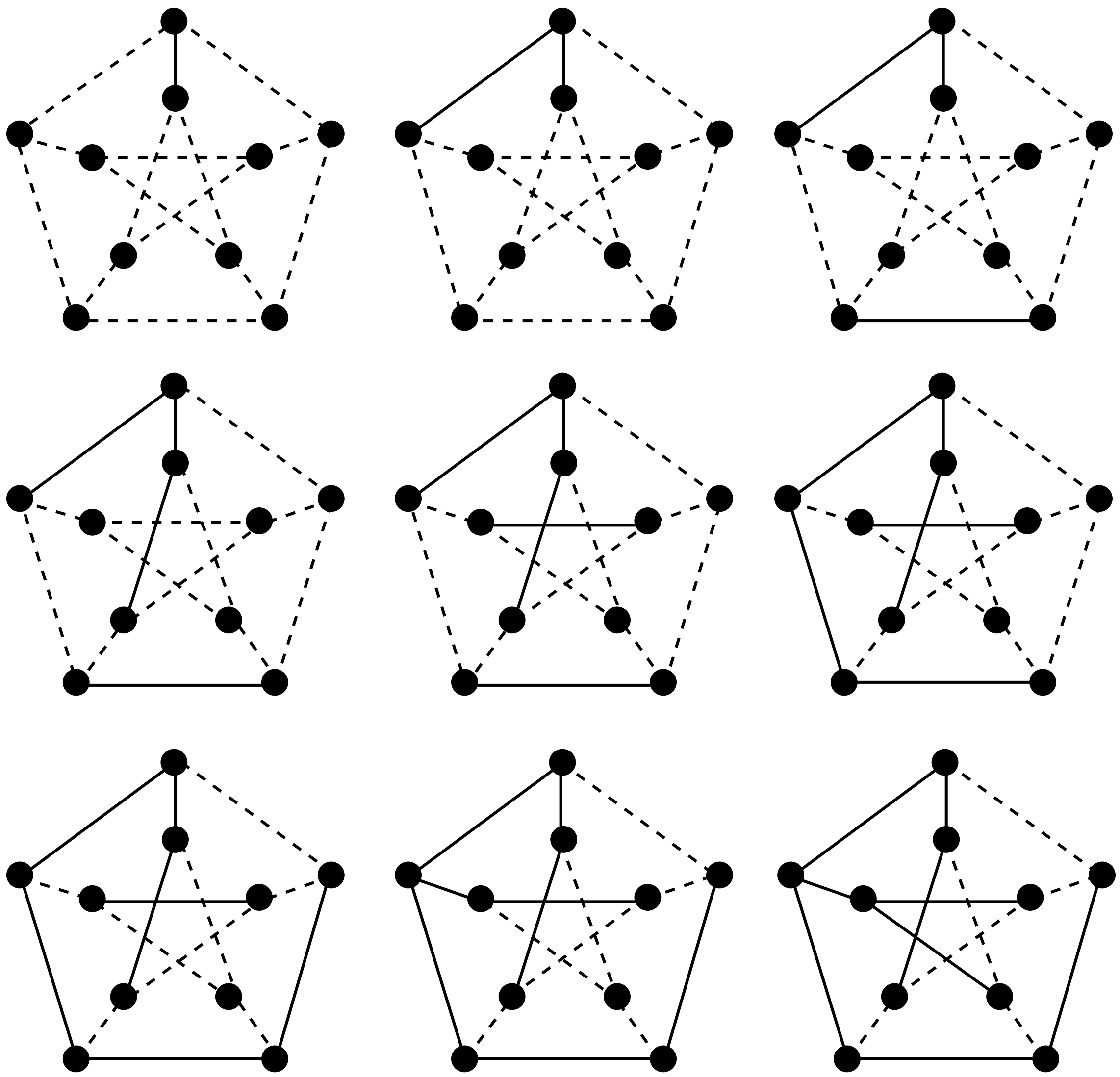
The build-up algorithm
Given a connected graph G to construct a spanning tree:

1. Start with a graph T consisting of the vertices of G and no edges.

2. If T is connected stop.

3. Add an edge e of G to T which does not form a cycle in T. Repeat from 2.

Example 10.15. This time starting with the Petersen graph:



Proof that the build-up algorithm stops when T is a spanning tree for G is straightforward. Strictly speaking neither of these is an algorithm. Recall that a graph consists of a set V of vertices and a set E of edges. The input to our algorithm is a list of the elements of V and a list of the elements of E. Given such data we have described no way of testing whether the graph $G = (V, E)$ contains a cycle or whether the graph is connected. We shall see how to remedy this defect in the next section.

11 Weighted graphs

In Section 10 it was pointed out that the cut-down and build-up algorithms described there are incomplete as they do not address the questions of finding cycles or testing for connectedness.

In fact, as all the graphs we consider are finite, it is clear that algorithms to find a cycle or a connected component do exist. The difficulty is to find an algorithm that works fast enough to be practical. Generally speaking, in a very large graph it may take a very long time to find a cycle, even with the best known algorithms. It is somewhat easier to test for connectedness. In fact to determine the connected component of a given vertex v we might colour v and all vertices incident to v red. Next colour red all uncoloured vertices which are incident to a red vertex. Continue until there are no uncoloured vertices incident to red vertices. The vertices of the connected component of v are now red.

Next we describe an adaption of the build-up algorithm which, by choosing carefully which edge to add at each stage avoids asking the questions "does adding this edge create a cycle?". The algorithm also tests for connectedness, using something similar to the "colour red" algorithm described above.

A programmable spanning tree algorithm

A graph with one vertex has a spanning tree consisting of one vertex and no edges. We eliminate these before we begin. Assume that we have a graph $G = (V, E)$ with at least 2 vertices. If the graph has no edge then it is not connected (and we are finished) so we shall also assume G has an edge

Step 1 Choose an element $e \in E$, say $e = \{a, b\}$, with $a \neq b$.
Set $v_1 = a$, $v_2 = b$ and $t_1 = e$.
Start building a tree T with vertices $V(T) = \{v_1, v_2\}$ and edges $E(T) = \{t_1\}$.
Set $i = 1$ and $j = 2$. (i is the number of the "base vertex", j is the number of the last vertex added.)

Step 2 If there is a vertex u of G which is adjacent to v_i and not in the subgraph T then

add 1 to j;
set $v_j = u$ and $t_{j-1} = \{v_i, v_j\}$ (an edge of G joining v_i to u);
continue to build up T by adding v_j to $V(T)$ and t_{j-1} to $E(T)$.

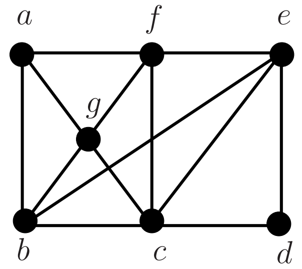
Step 3 If j is equal to the number of vertices of G then output the tree T and **stop**.

Step 4 If all vertices of G which are adjacent to v_i are in T then
add 1 to i.

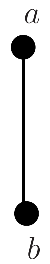
Step 5 If $i > j$ then output the message "G is not connected" and **stop**. Otherwise repeat from Step 2.

We illustrate the algorithm using the following example.

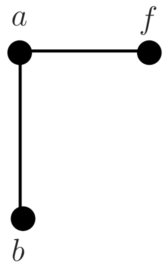
Example 11.1.



Step 1

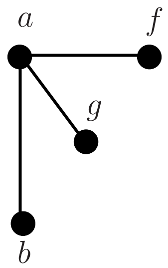


Step 2



Step 3

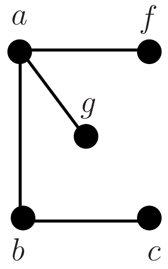
Step 2

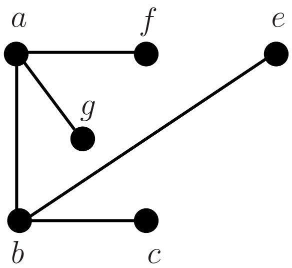


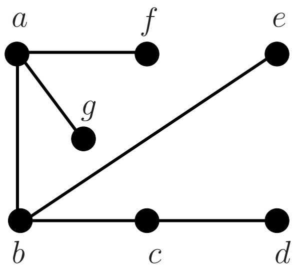
Step 3

Step 4

Step 2







To see that the above procedure always outputs a spanning tree of a connected input graph G note that it produces a sequence $T_1, \dots, T_k, T_{k+1}, \dots$ of subgraphs of G. For each k the subgraph T_{k+1} is constructed from T_k by adding a new edge; one end of which is in T_k while the other is not. Therefore each subgraph in the sequence is connected. As T_1 has 2 vertices and 1 edge it follows that the number of vertices of T_k is one more than the number of edges of T_k, for all k. From Theorem 10.9 it follows that T_k is a tree, for all k. Therefore, the only way the process could fail is if some vertex of G never enters any of the T_k. Suppose that this happens for some graph G and that v is a vertex of G which is never included in the vertex set of the T_k's. If G is connected then there is a path in G from v to a vertex of T_k, for some k. Let u be the last vertex in this path which is never added to any of the T_k's. Then the next vertex in the path x, say, is a vertex of T_k, for some k. Hence, at some stage the algorithm sets $x = v_i$. In the subsequent steps of the algorithm u becomes a vertex of one of the T_k's, a contradiction. Hence no such v exists. Therefore we conclude that if G is connected then the algorithm always halts and outputs a spanning tree T of G. On the other hand, if G is not connected the algorithm cannot stop in Step 3, because j is always less than the number of vertices of G. Hence it must stop in Step 5, saying that G is not connected.

Weighted graphs

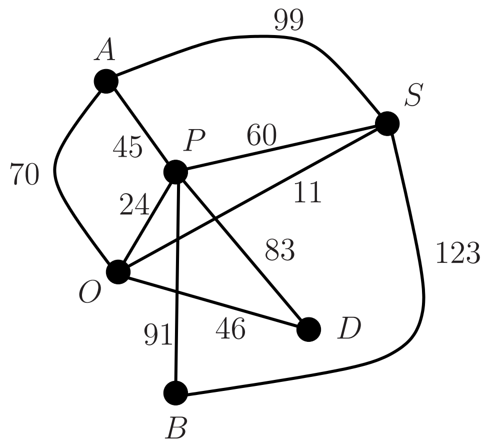
It is often useful to associate further information to the edges and vertices of a graph. For example the edges of a graph may represent roads, in which case we may wish to associate a distance, cost of travel or speed restriction to each edge. If the vertices represent places we may require them to carry additional information about population, temperature or cost of living. We concentrate here on graphs in which additional information is associated to edges. We assume that the required information is encoded as a number.

Definition 11.2. Let G be a connected graph with edge set E. To each edge $e \in E$ assign a non-negative real number $w(e)$. Then G is called a **weighted** graph and the number $w(e)$ is called the **weight** of e. The sum

$$W(G) = \sum_{e\in E} w(e)$$

is called the **weight** of G.

Example 11.3. The following drawing shows a weighted graph G. The weight of edge $\{A, S\}$ is $w(\{A, S\}) = 99$ and the weight of edge $\{O, P\}$ is $w(\{O, P\}) = 24$. The graph has weight $W(G) = 652$.



The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a **spanning subgraph**. We have seen several examples of spanning trees and obviously every spanning graph must contain a spanning tree.

In a connected, weighted graph the problem of finding a spanning subgraph of minimal weight is called the **minimal connector** problem. A spanning subgraph of minimal weight is always a spanning tree, so the problem is to find a spanning tree of minimal weight. The following algorithm does so. Again we leave aside the problem of testing for a cycle.

The Greedy Algorithm (also known as Kruskal's Algorithm)

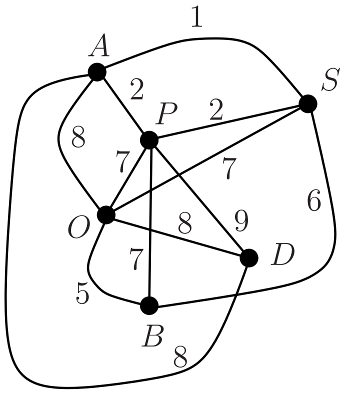
Let G be a connected weighted graph. To find a spanning tree T for G of minimal weight:

Step 1

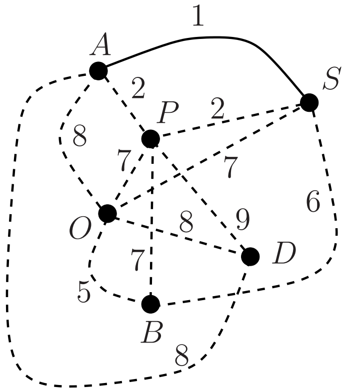
Step 2

Step 3

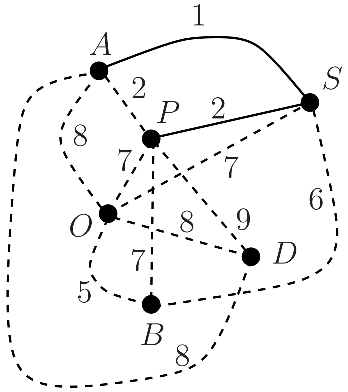
Example 11.4. The algorithm proceeds as shown on the weighted graph G below, producing forests $T_1, \ldots, T_5$ the last of which, T_5, is a minimal weight spanning tree. Note that there are some choices that have to be made in the running of the algorithm on this graph. For instance, either of the edges of weight 2 could have been included in T. A different choice results in a different minimal weight spanning tree, of which there may be many.



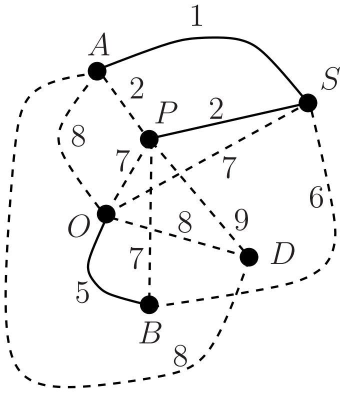
G



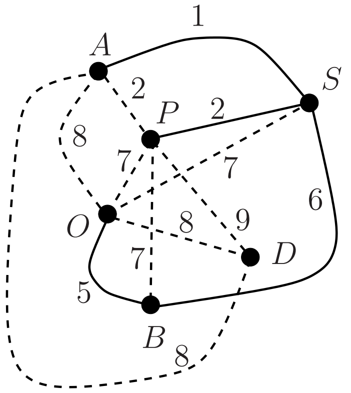
T_1



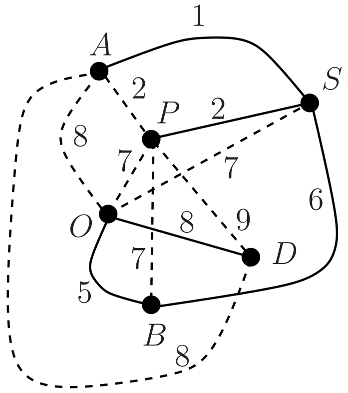
T_2



T_3



T_4



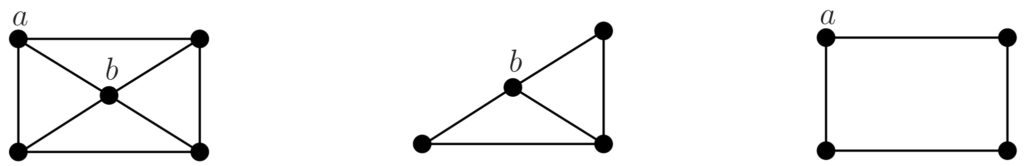
T_5

The Travelling Salesman Problem

A problem which arises in many applications is: "Given a connected weighted graph G, find a closed walk in G containing all vertices of G and of minimal weight amongst all such closed walks." This problem proves to be very difficult to solve in general. An easier problem, which we shall call the **Travelling Salesman** problem is: "Given a connected weighted graph G, find a minimal weight Hamiltonian closed path in G." The Travelling Salesman problem is easier in the sense that there are fewer possible solutions, so the search has fewer items to consider. However it is still very difficult to solve. We show here how the algorithm for the Minimum Connector problem can be used to find a lower bound for the Travelling Salesman problem. First however we establish some useful notation.

Definition 11.5. Let G be a graph and let v be a vertex of G. The graph $G - v$ obtained from G by **deleting** v is defined to be the graph formed by removing v and all its incident edges from G.

Example 11.6.



G $\qquad$ $G - a$ $\qquad$ $G - b$

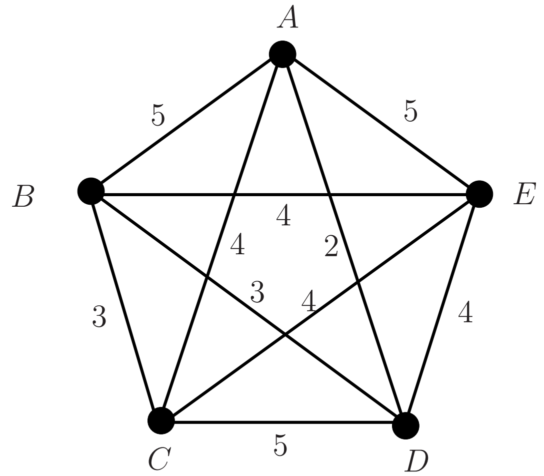
Theorem 11.7. *If G is a weighted graph, C is a minimal weight Hamiltonian closed path in G and v is a vertex of G then*

$$w(C) \geq M + m_1 + m_2,$$

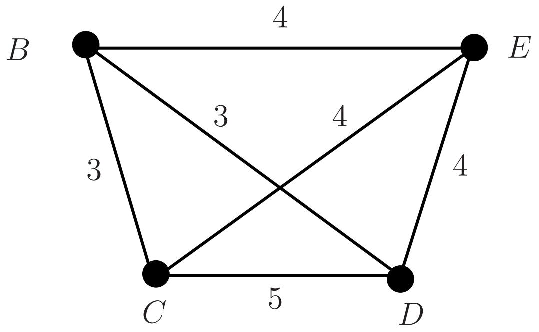
where M is the weight of a minimal weight spanning tree for $G - v$ and m_1 and m_2 are the weights of two edges of least weight incident to v.

As pointed out above the inequality in this Theorem may be strict. Thus, what we have is a lower bound for the Travelling Salesman problem, which in some cases may be smaller than the weight of minimal weight Hamiltonian closed path.

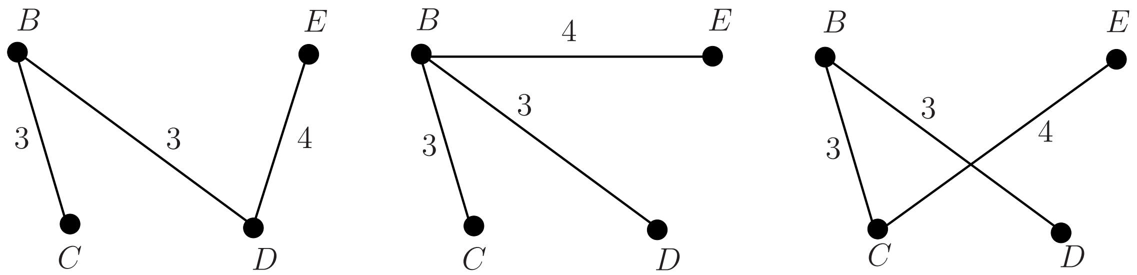
Example 11.8. We shall find a lower bound for the Travelling salesman problem in the weighted graph G below by removing vertex A.



Removing A we obtain the weighted graph $G - A$:

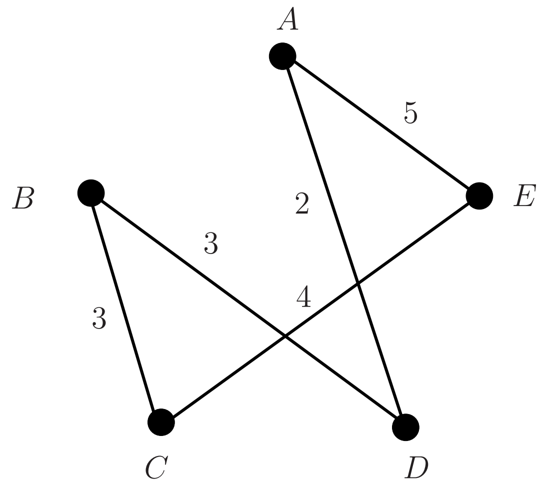


Running the Greedy Algorithm on $G - A$ we might obtain any of the minimal weight spanning trees below. We show all three only for purposes of illustration: any one will suffice. In this case we have $M = 10$.



Spanning Tree 1 Spanning Tree 2 Spanning Tree 3

The edges of minimal weight incident to A are $\{A, C\}$ and $\{A, D\}$ which have weights $m_1 = 2$ and $m_2 = 4$. Combining this information we have a lower bound of $10+2+4 = 16$.

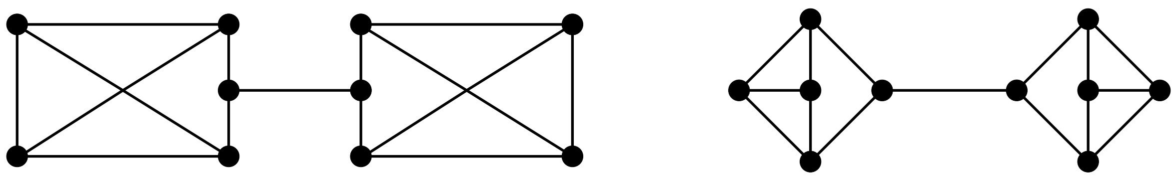

A
5
B
2
E
3
4
3
C
D

12 Planar Graphs

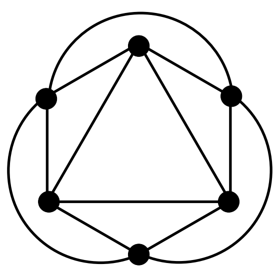
Definition 12.1. A graph is **planar** if it can be drawn in the plane without edges crossing. A **plane drawing** of a graph is a drawing of a graph in the plane which has no edge crossings.

We shall abuse notation and refer to a plane drawing of a graph as a **plane graph**. Note however that a drawing is merely a representation of a graph: as always a graph consists merely of a pair of sets.

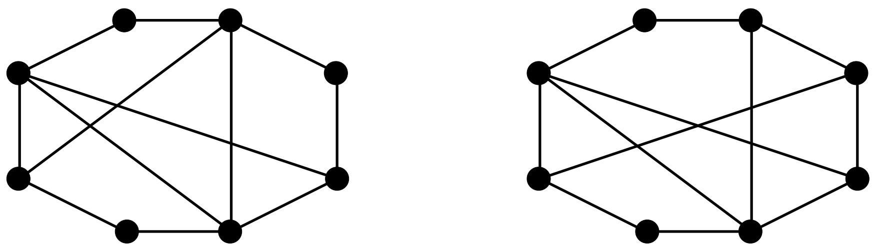
Example 12.2. 1.



2.



3.



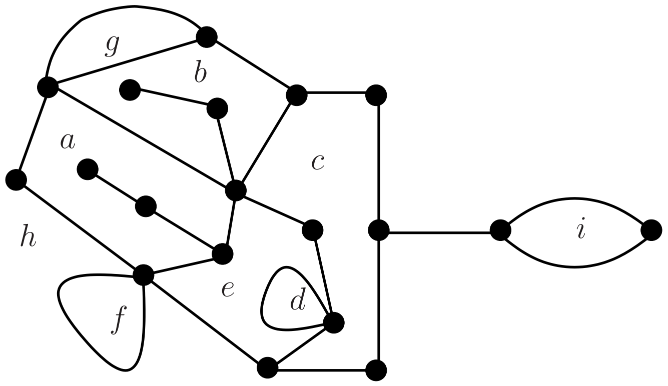
A plane drawing of a graph divides the plane up into polygonal regions which we call **faces**. In the Example 12.2 the first plane graph divides the plane into 7 regions and the Octahedron divides the plane into 8 regions. (Note that in each case one of the regions is unbounded.) The following definition attempts formalise this idea.

Definition 12.3. Let D be a plane drawing of a graph. If x is a point of the plane not lying on D then the set of all points of the plane that can be reached from x without crossing D is called a **face** of D. One face is always unbounded and is called the **exterior face**.

(To make a rigorous definition of *face* requires the Jordan Curve theorem, which says that a simple closed curve in the plane divides the plane into two parts, one inside and one outside the curve. This theorem is beyond the scope of this course.)

Example 12.4. 1. A plane drawing of a tree has one face (which is exterior).

2. The graph below has 9 faces labelled $a, \ldots, i$. Face h is the exterior face.



Euler noticed that for a plane drawing of a platonic graph with n vertices m edges and r faces the sum $n - m + r = 2$. He went on to prove the following theorem.

Theorem 12.5 (Euler's Formula)**.** *Let G be a connected plane graph (i.e. a plane drawing of a connected graph) with n vertices, m edges and r faces. Then $n - m + r = 2$.*

Proof.

□

Definition 12.6. Let F be a face of a plane graph. The **degree** of F, denoted $\deg(F)$ is the number of edges in the boundary of F, where edges lying in no face except F count twice. (To compute $\deg(F)$ walk once round the boundary of F, counting each edge on the way.)

In Example 12.4.2 above we have

face	a	b	c	d	e	f	g	h	i
degree									

The degree of a face has much in common with the degree of a vertex. Compare the following to Lemma 3.1.

Lemma 12.7. *If G is a plane graph with m edges and r faces $F_1, \ldots, F_r$ then*

$$\sum_{i=1}^{r} \deg(F_i) = 2m.$$

Proof. Every edge meets either one or two faces. Edges meeting only one face contribute 2 to the degree of their face. Edges meeting two faces contribute 1 to the degree of each of their faces. The result follows. □

We can use Euler's formula to find graphs which are non-planar.

Corollary 12.8. *If G is a simple connected planar graph with $n \geq 3$ vertices and m edges then $m \leq 3n - 6$.*

Corollary 12.9. *If G is a connected simple planar graph with $n \geq 3$ vertices, m edges and no cycle of length 3 then $m \leq 2n - 4$.*

Proof. A plane drawing of G can have no face of degree less than 4. The proof proceeds as that of Corollary 12.8, except that this time $2m \geq 4r$. □

We can now prove the following.

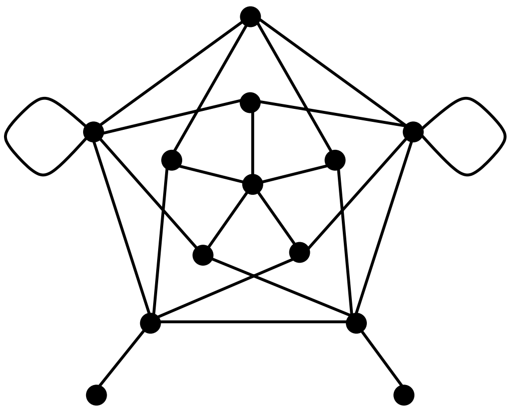
Theorem 12.10. *The complete graph K_5 and the complete bipartite graph $K_{3,3}$ are both non-planar.*

If a graph G is non-planar then any graph which contains G as a subgraph is also non-planar. It follows that if a graph contains K_5 or $K_{3,3}$ as a subgraph it must be non-planar. We can however prove a stronger result. First some terminology.

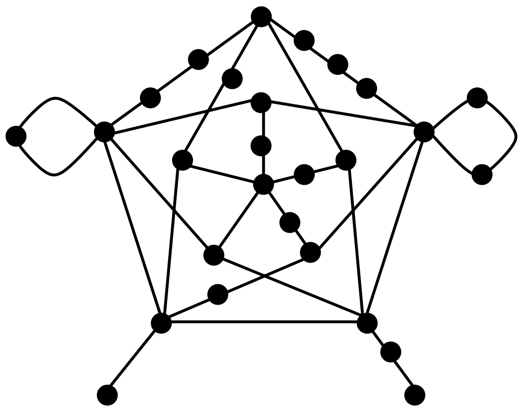
Definition 12.11. A graph H is a **subdivision** of a graph G if H is obtained from G by the addition of a finite number of vertices of degree 2 to edges of G.

Note that in this definition it is possible to add no vertices and so a graph is a subdivision of itself.

Example 12.12. The graph H below right is a subdivision of the graph G below left.



G



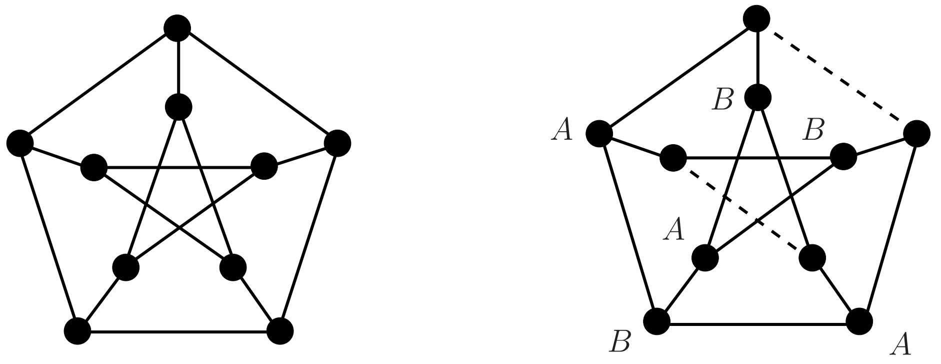
H

The following theorem is an easy consequence of Theorem 12.10.

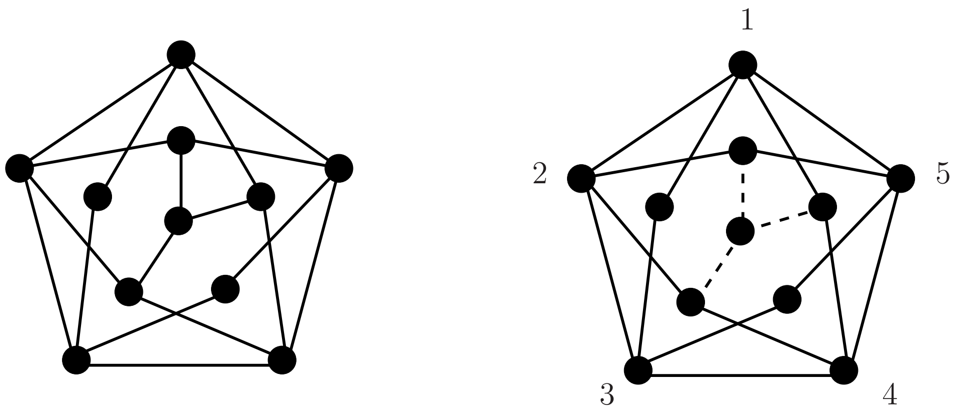
Theorem 12.13. *If G is a graph containing a subgraph which is a subdivision of K_5 or $K_{3,3}$ then G is non-planar.*

Example 12.14. Neither Corollary 12.8 nor Corollary 12.9 are sufficient to show that the graphs of this example are non-planar.

1. The Petersen graph shown below has 10 vertices and 15 edges. The diagram on the right shows a subgraph which is a subdivision of $K_{3,3}$. Therefore the graph is non-planar. (Vertices which are not labelled A or B are those added in the subdivision.



2. The graph shown below has 11 vertices and 18 edges. The diagram on the right shows a subgraph which is a subdivision of K_5. Therefore the graph is non-planar.



A more surprising theorem, which we shall not prove here, is known as Kuratowski's theorem:

Theorem 12.15 (Kuratowski)**.** *If G is a non-planar graph then G contains a subgraph which is a subdivision of K_5 or $K_{3,3}$.*

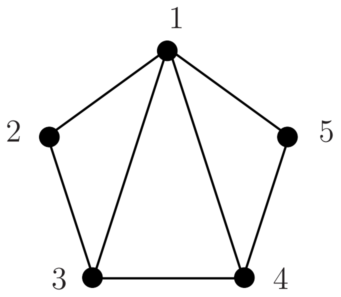
13 Colourings of graphs

Vertex colouring

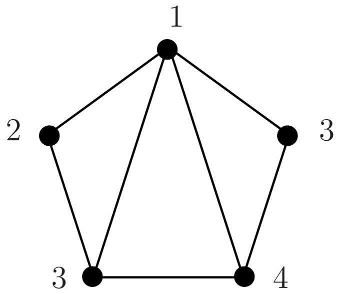
Definition 13.1. Let G be a graph without loops. A k**-colouring** of G is an assignment of k colours to the vertices of G such that no two adjacent vertices are assigned the same colour. If G has a k-colouring it is said to be k**-colourable**.

Example 13.2. We use colours $1, 2, 3, 4$ and 5.

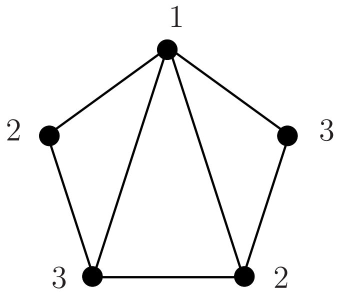
1. a 5-colouring:



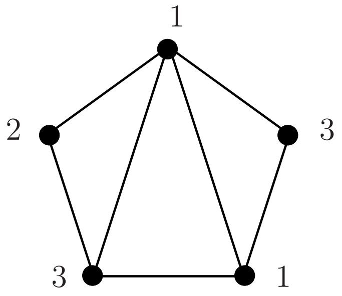
2. a 4-colouring:



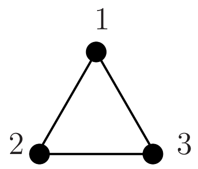
3. a 3-colouring:



4. not a colouring:



Example 13.3. 1.



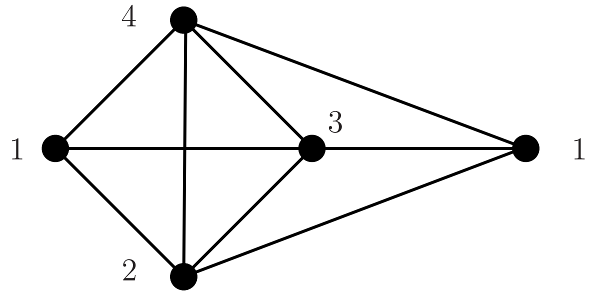
2.

Definition 13.4. The **chromatic number** $\chi(G)$ of a graph G is the least positive integer k such that G has a k-colouring.

Example 13.5. 1. From Example 13.3 it follows that $\chi(K_d) = d$, for all $d \geq 2$.

2.

3.



We state, without proof, the following strengthening of the above result. This theorem was proved in 1941 by Brooks.

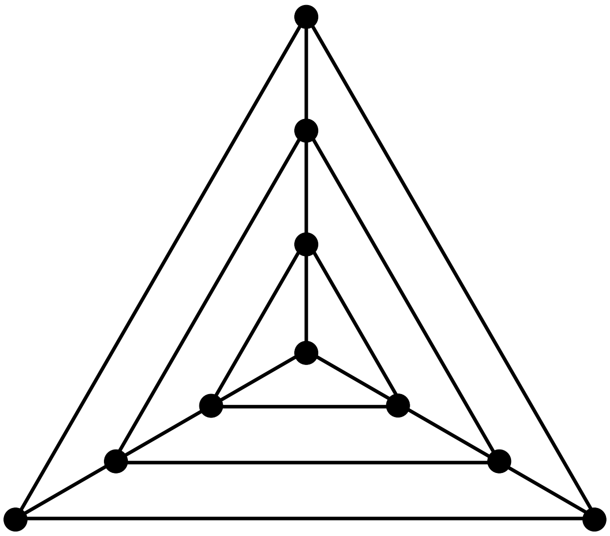
Theorem 13.6 (Brooks)**.** *Let G be a connected simple graph and d a non-negative number such that* $\deg(v) \leq d$*, for all vertices v of G. If*

1. *G is not a cycle graph C_n with n odd and*

2. *G is not a complete graph K_n*

then $\chi(G) \leq d$.

Example 13.7. The graph G below has a subgraph isomorphic to K_4, so $\chi(G) \geq 4$. Using Brooks' theorem $\chi(G) \leq 4$. Hence $\chi(G) = 4$. The graph must therefore be 4-colourable. Can you find a 4-colouring?

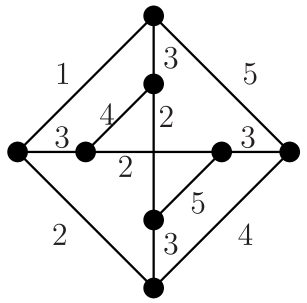


G

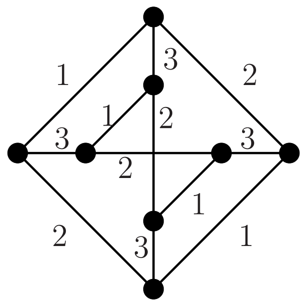
Edge colouring

Definition 13.8. An **edge-colouring using k colours** of a graph G is an assignment of one of k colours to each edge of G. A **proper** edge-colouring is one with the additional property that no two adjacent edges are assigned the same colour. The **edge-chromatic number** $\chi^e(G)$ of G is the smallest integer k such that G has a proper edge-colouring using k colours.

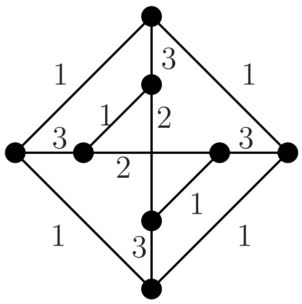
Example 13.9. 1. a proper edge-colouring using 5 colours:



2. a proper edge-colouring using 3 colours:



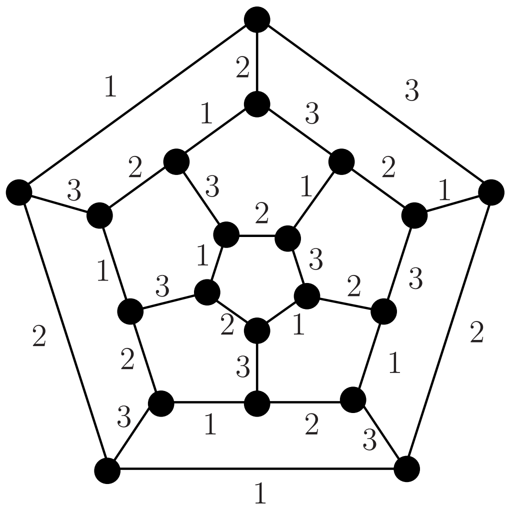
3. an edge-colouring using 3 colours which is not proper:



Lemma 13.10. *Let G be a graph and let d be the largest degree of a vertex of G. Then any proper edge-colouring of G uses at least d colours. That is $\chi^e(G) \geq d$.*

Example 13.11. 1.

2.



Theorem 13.12. *The graph K_{2d} has edge-chromatic number $\chi^e(K_{2d}) = 2d - 1$, for all $d \geq 1$.*

Proof. We use the "turning trick" (see Example 9.3 and Theorem 9.4). Label the vertices of K_{2d} with $0, 1, \ldots, 2d - 2$ and x. Arrange the numbered vertices as the corners of a regular $2d-1$-gon and place the vertex x outside. Assume we have colours $C_1, \ldots, C_{2d-1}$. Colour the edges

$$\{0, x\}, \{1, 2d - 2\}, \{2, 2d - 3\}, \ldots, \{d, d - 1\}$$

with colour C_1. Note that all these edges except $\{0, x\}$ are parallel, as shown in Figure 13.1, for the case $d = 5$. Now turn the $2d - 1$-gon one position anticlockwise. This gives a new set of parallel edges which, in addition to $\{1, x\}$, are coloured with C_2; see Figure 13.2 for the case $d = 5$. A list of the new edges is obtained by adding 1, modulo $2d-1$, to the label of each vertex in the list of edges coloured C_1. Continuing in this way edges are coloured as follows.

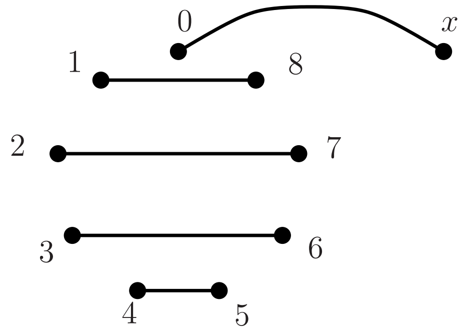


Figure 13.1: Solid edges coloured C_1 in K_{10}

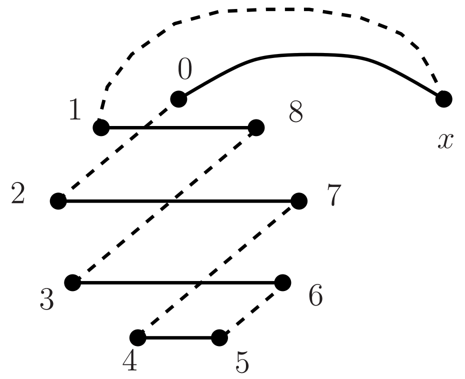


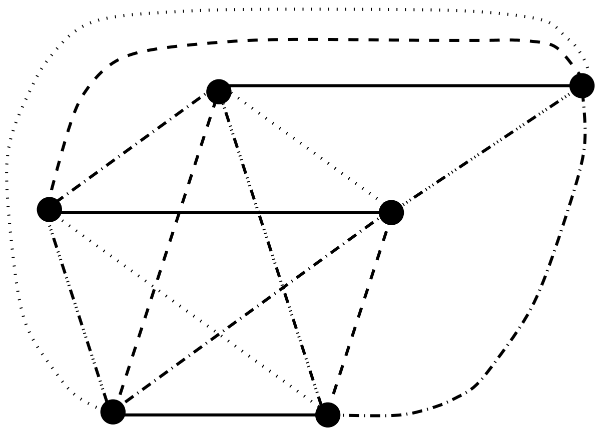
Figure 13.2: Dashed edges coloured C_2 in K_{10}

C_1	$\{0, x\}$	$\{1, 2d-2\}$	$\{2, 2d-3\}$	$\dots$	$\{d, d-1\}$
C_2	$\{1, x\}$	$\{2, 0\}$	$\{3, 2d-2\}$	$\dots$	$\{d+1, d\}$
C_3	$\{2, x\}$	$\{3, 1\}$	$\{4, 0\}$	$\dots$	$\{d+2, d+1\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
C_{2d-2}	$\{2d-3, x\}$	$\{2d-2, 2d-4\}$	$\{0, 2d-5\}$	$\dots$	$\{d-2, d-3\}$
C_{2d-1}	$\{2d-2, x\}$	$\{0, 2d-3\}$	$\{1, 2d-4\}$	$\dots$	$\{d-1, d-2\}$

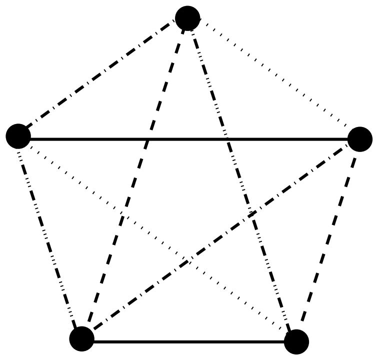
Each edge of K_{2d} appears exactly once in this table and each vertex appears exactly once on each row. Therefore we have a proper edge-colouring using $2d-1$ colours. As K_{2d} is regular of degree $2d-1$ at least $2d-1$ colours are required in any proper edge-colouring. Hence $\chi^e(K_{2d}) = 2d-1$. □

Corollary 13.13. *The graph K_{2d-1} has edge-chromatic number $\chi^e(K_{2d-1}) = 2d-1$, for all $d \geq 1$.*

Example 13.14. The following diagrams show the proper edge-colourings of K_6 and K_5 obtained using the methods of Theorem 13.12 and Corollary 13.13, respectively.



Proper edge-colouring of K_6 using 5 colours



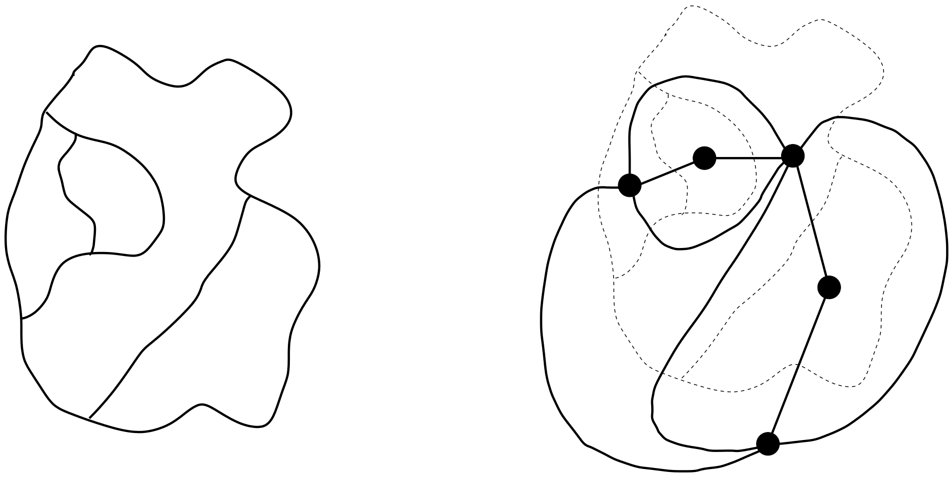
Proper edge-colouring of K_5 using 5 colours

The Four-colour Problem

In 1852 De Morgan made the conjecture that any map of countries could be coloured using only 4 colours, in such a way that countries with a common border would have different colours.

We can interpret this question in terms of graph theory: given a map of countries we construct a plane drawing of a graph as follows. Place one vertex in each country (the "capital" of the country). Join two vertices with an edge whenever their countries have a common border.

Example 13.15. The map of countries on the left gives rise to the plane graph on the right.



Now, if it could be shown that any planar graph without loops is 4–colourable then it would follow that every map of countries can be coloured as required by De Morgan. The graph theoretic version of the conjecture is therefore:

Conjecture 13.16 (The 4–colour problem)**.**
Every simple planar graph is 4*–colourable.*

The problem has a long and chequered history.

1852 De Morgan proposes the 4–colour conjecture.

1873 Cayley presents a proof to the London Mathematical Society. The proof is fatally flawed.

1879 Kempe publishes a proof; which collapses.

1880 Tait gives a proof which turns out to be incomplete.

1976 Appel & Haken at the University of Illinois prove the 4–colour conjecture: using thousands of hours of CPU time on a Cray computer.

A problem with Appel & Haken's proof is that the program runs for so long that it is impossible to verify manually. We cannot even to be sure that the hardware performed well enough, over such an extended period, to give a reliable result.

By contrast a 6–colour theorem is easy to prove.

Theorem 13.17.
Every simple planar graph G is 6–colourable.

A proof of a 5–colour theorem, although somewhat harder, can be found in most introductory texts on graph theory.

We finish with a result which links vertex and edge colouring. The map of countries shown above does itself constitute a graph: put a vertex at each point where two borders meet. The resulting graph is plane, connected, regular of degree three and has no bridges or loops. Furthermore any "reasonable" map of countries constitutes a plane drawing of a graph with all these properties. The 4–colour conjecture states that the faces of such a plane graph can be coloured using 4 colours, where **colouring** means that no edge meets two faces of the same colour. In 1880 Tait made the following connection between 4–colouring of faces and edge–colouring.

Theorem 13.18. *Let G be a plane drawing of a graph which is connected, regular of degree three and has no bridges or loops. Then the faces of G can be coloured using 4 colours if and only if G has a proper edge–colouring using 3 colours.*