Combinatorics

Mark Wildon

June 18, 2007

Small print: These notes are intended to give the logical structure of the course; proofs and further remarks will be given in lectures. Further installments will be issued as they are ready. I would very much appreciate being told of any corrections or possible improvements. These notes are available via my home page,

http://www-maths.swan.ac.uk/staff/mjw/

My email address is `m.j.wildon@swansea.ac.uk`.

Contents

1 Introduction

Combinatorial arguments may be found lurking in all branches of mathematics. Many people first become interested in mathematics by a combinatorial problem. But, strangely enough, until quite recently, many mathematicians tended rather to sneer at combinatorics. Thus one finds:

> "*Combinatorics is the slums of topology.*"
>
> J. H. C. Whitehead (early 1900s, attr.)

Fortunately attitudes have changed, and the importance of combinatorial arguments is now widely recognised:

> "*The older I get, the more I believe that at the bottom of most deep mathematical problems there is a combinatorial problem.*"
>
> I. M. Gelfand (1990)

Combinatorics is a very broad subject. Often the techniques used to prove theorems are more important than the theorems themselves. Algorithmic constructions (rather than mere existence proofs) are particularly valuable. There is no shortage of interesting and easily understood motivating problems.

This course aims to give a straightforward introduction to three related areas of combinatorics:

(1) Graph theory.
(2) Enumeration under group actions.
(3) Generating functions and partitions.

Part (3) will be quite brief[1], and lectured after Easter.

With the exception of some of the material in the first lecture, I do not intend things to be very difficult: if they are, please object.

Recommended Reading

(1) Ralph P. Grimaldi. *Discrete and Combinatorial Mathematics: an applied introduction* (5th edition). Academic Press, 2003. QA39.
(2) Béla Bollobás. *Graph Theory.* Springer, Graduate Texts in Mathematics 63, 1979. QA166.
(3) Peter Cameron. *Combinatorics.* CUP, 1996.
(4) Robin J. Wilson. *Introduction to graph theory.* CUP, 1996. QA166.

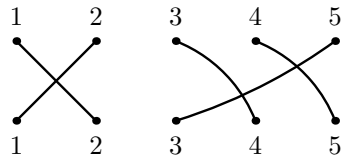
[1] In fact it occupied all of one lecture.

Derangements*

(The star means this section is non-examinable.) Recall that a *permutation* of the set $\{1, 2, \ldots, n\}$ is a bijective function

$$\sigma : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}.$$

It is often useful to represent permutations by diagrams; the diagram below shows the permutation $\sigma : \{1, 2, 3, 4, 5\} \to \{1, 2, 3, 4, 5\}$ defined by $\sigma(1) = 2$, $\sigma(2) = 1$, $\sigma(3) = 4$, $\sigma(4) = 5$, $\sigma(5) = 3$.



Particularly in part (2) of the course we shall use shorter notation for permutations. Recall that in disjoint-cycle form, $\sigma = (12)(345)$ and that in one-line form, $\sigma = 21453$.

Problem 1.1 (Derangements)**.** One says that a permutation of $\{1, 2, \ldots, n\}$ is a *derangement* if $\sigma(i) \neq i$ for any $i \in \{1, 2, \ldots, n\}$. How many permutations of $\{1, 2, \ldots, n\}$ are derangements?

Let $d(n)$ be the number of permutations of $\{1, 2, \ldots, n\}$ that are derangements. *Exercise:* check by going through all $n!$ permutations of $\{1, 2, \ldots, n\}$ for $n = 0, 1, 2, 3, 4$ that $d(0) = 1$, $d(1) = 0$, $d(2) = 1$, $d(3) = 2$, $d(4) = 9$.

Lemma 1.2. *If $n \geq 2$ then $n! = (n-1)\big((n-1)! + (n-2)!\big)$*

We use the same combinatorial idea to prove the next theorem.

Theorem 1.3. *If $n \geq 2$ then $d(n) = (n-1)\big(d(n-1) + d(n-2)\big)$*

There are many methods for solving such recurrence relations. Often the quickest method is to somehow guess the answer[2], and then to prove it by induction. A better method will be seen in part (3) of the course.

Corollary 1.4. *For all $n \in \mathbf{N}$,*

$$d(n) = n!\Big(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \ldots + \frac{(-1)^n}{n!}\Big).$$

[2]A useful aid is N. J. A. Sloane's Online Encyclopedia of Integer Sequences: see `www.research.att.com/~njas/sequences/`

Exercise: check directly that the right-hand-side is an integer.

Theorem 1.5. *Two results with a probabilistic flavour.*

(i) *The proportion of permutations of* $\{1, 2, \ldots, n\}$ *that are derangements tends to* $1/\mathrm{e}$ *as* $n \to \infty$.

(ii) *The average number of fixed points of a permutation of* $\{1, 2, \ldots, n\}$ *is* 1.

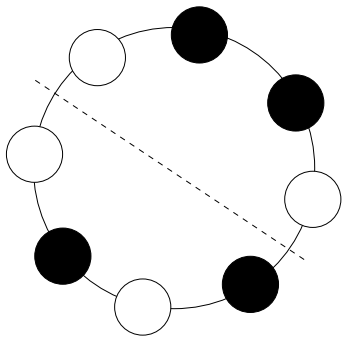
It will be seen later that part (ii) of this theorem is a special case of a much more general result: see Theorem 7.4.

Challenge problems*

Here are three hopefully interesting problems embodying combinatorial ideas related to the three parts of the course. I will happily give £5 to anyone who solves one of these problems and explains their solution at the board[3].

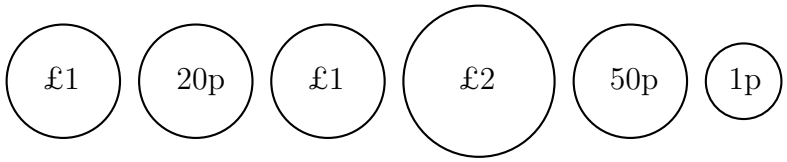
Problem 1.6 (Knights and spies)**.** In a room there are 100 people. An absolute majority are *knights*, who have sworn to always tell the truth. The rest are *spies*, who may lie or tell the truth as they see fit. Everyone knows the true identity of everyone else. Asking only questions of the form 'Person i, what is person j?', can you be sure of correctly identifying all the spies in 200 questions or fewer?

Problem 1.7 (Beads)**.** Given a necklace with 168 beads, 84 black and 84 white, can it be cut, and the new ends retied, so that two necklaces each with 42 beads of either colour are obtained? (See below for one solution with an 8 bead necklace.)



[3]My money was far safer than I had expected.

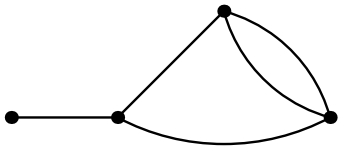
Problem 1.8 (Coins)**.** The hare and the tortoise — both of them capable mathematicians — play a game. The hare first places evenly many coins in a row. The players then alternately take coins from either end, until none are left. The winner is the person who ends up with the most money. Tradition dictates that the tortoise starts. Show that the tortoise never loses.



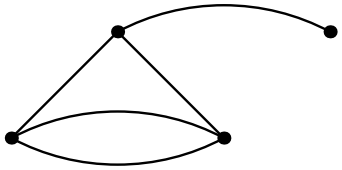
A possible starting position for a six coin game.

2 GRAPHS: Basic definitions

Suppose we are interested in travelling around a number of towns; some pairs of towns are connected by a direct road, others are not. (Some may be directly connected by several roads.) We don't care about the exact geographical location of the towns or roads, only whether two towns are connected or not. We can describe this situation by a *graph* such as:



which we regard as the same graph as:



In either representation, this graph has 4 vertices (the towns) and 5 edges (the roads).

It is well worth giving a more formal definition of a graph.

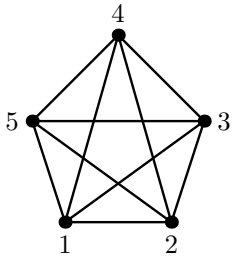
Definition 2.1. A *graph* is a set V of *vertices* together with a list E of 2-subsets of V. If $\{x, y\}$ appears r times in E then we say there are r *edges* between the vertices x and y. A graph is *simple* if there is at most one edge between any two vertices.

Example 2.2. Will give a formal definition of the graph above.

Sometimes one also allows a graph to have loops, that is, edges which connect a vertex directly to itself. This would complicate the formal definition, and isn't helpful for the applications in this course.

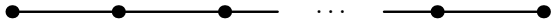
All our graphs will have finitely many vertices and edges unless it is specifically stated otherwise.

Example 2.3. (a) The *complete graph* on a set V has vertex set V and edge set all 2-subsets of V. For example, the complete graph on $\{1, 2, 3, 4, 5\}$ is:



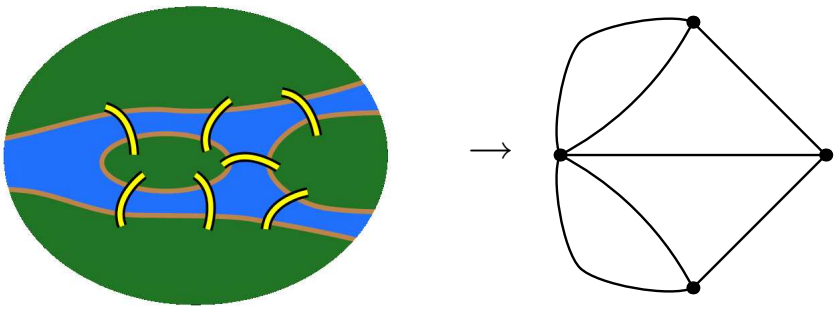
(b) The *null graph* on a set V has vertex set V and empty edge set.

(c) The *chain of length* n is the graph:



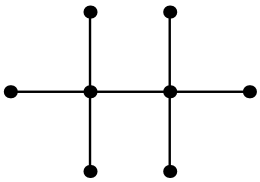
where there are n vertices.

(d) A non-simple graph: graphs were discovered[4] by Euler in 1736 in his solution of the Königsberg bridge problem: is it possible to walk around Königsberg, crossing each of its seven bridges bridge exactly once? (It is not required that you finish the walk at your starting point.)



The corresponding graph has multiple edges between two pairs of its vertices, so is not simple.

(e) Chemical structures are naturally represented by graphs. For example, we might define an *alkane* to be a simple graph each of whose vertices has either 1 or 4 edges leading off it. For example, ethane:



Notation 2.4. If G is a graph, we shall write $V(G)$ for the vertex set of G and $E(G)$ for the list of edges of G.

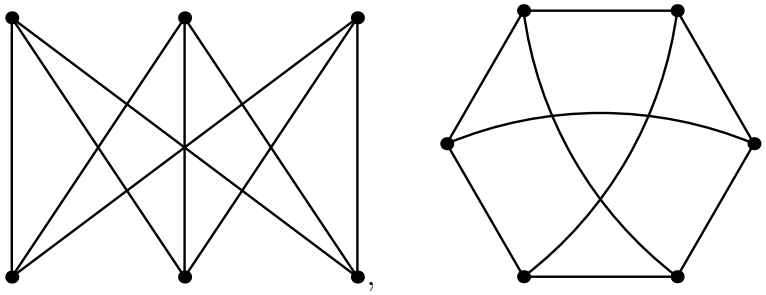
We now define graph isomorphims.

Definition 2.5. Two graphs G and G' are *isomorphic* if there is a bijection $f : V(G) \to V(G')$ such that for any two vertices $x, y \in V(G)$ the number of edges between x and y is equal to the number of edges between $f(x)$ and $f(y)$.

Graphs that look 'the same' are isomorphic. For example, it is easy to write down an isomorphism between the two graphs at the top of page 5. Sometimes though isomorphisms can be a little harder to spot.

[4]or possibly invented

Example 2.6. (a) The two graphs below are isomorphic.



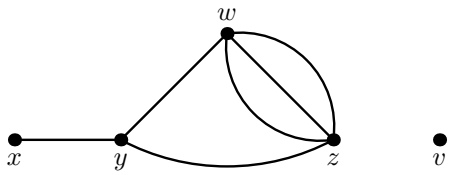
(b) Will give an example of two non-isomorphic graphs.

Definition 2.7. Let G be a graph and let x be a vertex of G. We say that x is *adjacent* to (or is a *neighbour* of) a vertex $y \in V(G)$ if $\{x, y\}$ is an edge of G.

The *degree* of x, written $\delta(x)$, is the number of vertices to which it is adjacent. (Or equivalently, the number of edges containing x.)

If G has vertices of degrees $d_1, d_2, \ldots, d_n$ where $d_1 \geq d_2 \geq \ldots \geq d_n$ then we say that G has *degree sequence* $(d_1, d_2, \ldots, d_n)$.

Example 2.8. The graph below has degree sequence $(4, 4, 3, 1, 0)$.



Lemma 2.9. *Isomorphic graphs have the same degree sequence.*

This gives one easy way to show that two graphs are not isomorphic. WARNING: it is possible for two graphs to have the same degree sequence, yet not be isomorphic — see sheet 1, question 1.

Theorem 2.10 (Handshaking Theorem)**.** *Let G be a graph with n vertices and e edges. Suppose that G has degree sequence $(d_1, d_2, \ldots, d_n)$. Then*

$$d_1 + d_2 + \ldots + d_n = 2e.$$

In particular, the sum of the degrees is even.

[4]Dangerous bend ahead.

Hence a necessary condition for $(d_1, \ldots, d_n)$ to be the sequence of degrees of a graph is that $d_1 + \ldots + d_n$ is even. However this condition is not sufficient: see sheet 1, question 2.

Definition 2.11. Let G be a graph. A *walk* in G is a sequence of vertices $(x_0, x_1, \ldots, x_m)$ such that $\{x_i, x_{i+1}\}$ is an edge of G for $0 \leq i < m$. We say x_0 is the *initial vertex* and x_m is the *final vertex*. The number of edges, m, is the *length* of the walk.

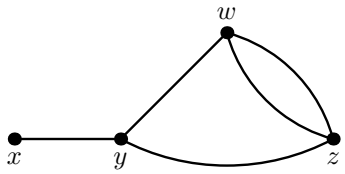
A *trail* is a walk in which the edges may be chosen to be distinct. (Thus if there are r edges between vertices x and y, a trail may go directly between x and y at most r times.)

A *path* in G is a trail $(x_0, x_1, \ldots, x_m)$ in which the vertices are all distinct, *except* we allow $x_0 = x_m$; i.e. the initial vertex may equal the final vertex.

A walk / trail / path is *closed* if its initial vertex is equal to its final vertex.

One helpful mnemonic is that terms which occur earlier in the dictionary are more specialised. Thus a trail is a special kind of walk and a path is a special kind of trail.

Example 2.12. In the graph shown below



(a) (x, y, z, y, w) is a walk, but not a trail;

(b) (x, y, z, w, z) is a trail (since we can use different edges for the two trips between z and w and so avoid repeating an edge), but not a path;

(c) (x, y, z, w) is a path;

(d) (y, z, w, y) is a closed path, as is (w, z, w).

Theorem 2.13. *Let G be a graph and let x, y be vertices of G. There is a walk between x and y if and only if there is a path between x and y.*

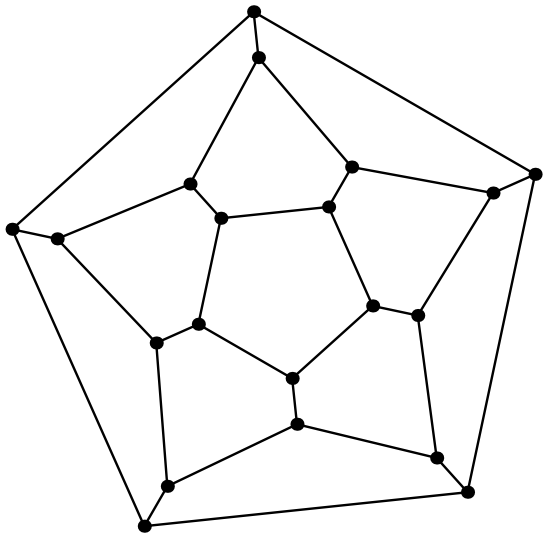
Definition 2.14. Let G be a graph. We say that G is *connected* if there is a walk between any two of its vertices. If G is a connected graph then we say that the *distance* between vertices x and y of G is m if the shortest walk between x and y has length m.

We end this section by giving a simple algorithm for finding distances in a connected graph.

Algorithm 2.15. *Given a connected graph G one may determine the distance between two vertices $x, y \in V(G)$ by proceeding as follows:*

1. *Mark x with the label 0. Set $d = 0$.*
2. *Look at all vertices labelled d; if any of their adjacent vertices are unlabelled, label them with $d + 1$.*
3. *If y is now labelled, its label is its distance from x. Otherwise increment d by 1 and return to step 2.*

Example 2.16. Will determine all distances in the dodecahedral graph.



3 Eulerian graphs, Planar graphs

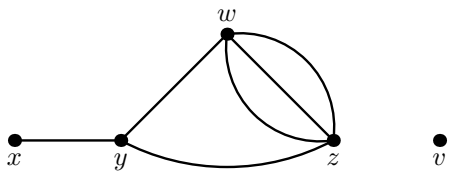
We first solve the general version of the Königsberg bridges problem.

Definition 3.1. An *Eulerian trail* in a graph is a trail which uses every edge.

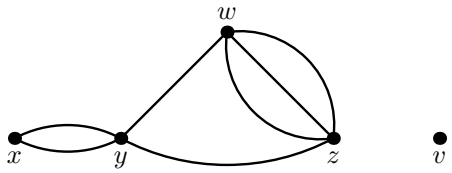
By definition, a trail uses each edge at most once, so an Eulerian trail in a graph uses every edge exactly once. Vertices may be visited multiple times (or not at all, if they are isolated).

Example 3.2. (a) Some experimentation will quickly convince you that the Königsberg bridges graph does not have an Eulerian trail.

(b) The graph below has an Eulerian trail.



(c) If we add another edge between vertices x and y then it has a closed Eulerian trail.



We first prove a necessary condition for a graph to have a closed Eulerian trail.

Lemma 3.3. *Let G be a connected graph. If G has a closed Eulerian trail then every vertex of G has even degree.*

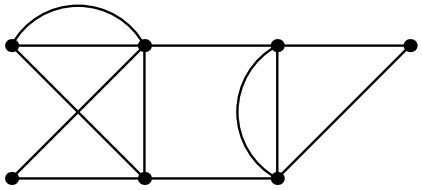
Euler's main contribution was to show that the converse also holds.

Theorem 3.4. *Let G be a connected graph. If every vertex of G has even degree then G has a closed Eulerian trail.*

Theorem 3.5. *A connected graph has a non-closed Eulerian trail if and only if it has exactly two vertices of odd degree.*

The easier 'only if' part of this theorem tells us that the Königsberg bridges graph does not have an Eulerian trail.

Example 3.6. The proofs of the last two theorems are essentially constructive. To illustrate this will use them to find an Eulerian trail in the graph below.

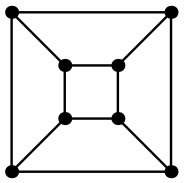


Hamiltonian graphs

Definition 3.7. A *Hamiltonian path* in a graph is a path which visits every vertex.

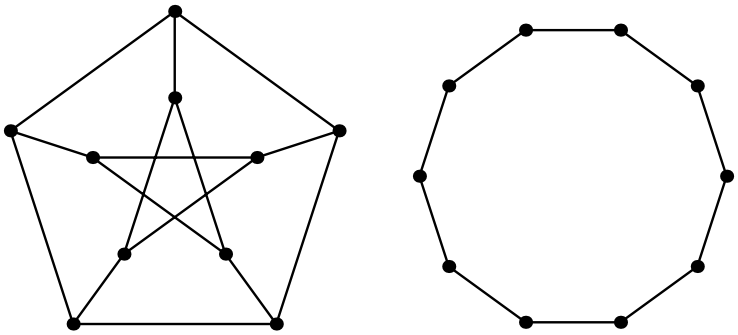
Thus a non-closed Hamiltonian path in a graph G visits each vertex exactly once. A closed Hamiltonian path with initial vertex $x \in V(G)$ visits all vertices other than x exactly once, and vertex x twice. Edges are used either once or not at all.

Example 3.8. (a) The graph of the cube has a closed Hamiltonian path



(b) *Exercise:* Find a closed Hamiltonian path in the graph of the dodecahedron[6].

(c) The *Petersen graph* (shown below left) does not have a closed Hamiltonian path. If it did, it would be isomorphic to the graph obtained from a closed path of length 10 (shown below right), by drawing 5 chords connecting pairs of its vertices ...



[6]Apparently, William Hamilton (more famous as the discoverer of quaternions) patented an 'icosian game' in which the aim was to find such paths, and in 1859 even managed to sell the idea to a London game dealer for £25.

A major unsolved problem in graph theory is to find a reasonable necessary and sufficient condition for a graph to have a closed Hamiltonian path. Almost the strongest known sufficient condition is due to O. Ore (1960).

Theorem 3.9 (Ore)**.** *Let G be a simple graph on n vertices. If $n \geq 3$, and*

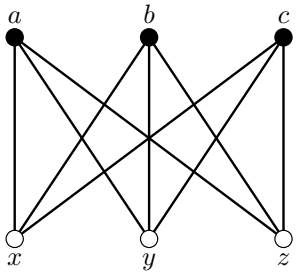
$$\delta(x) + \delta(y) \geq n$$

for each pair of non-adjacent vertices x and y, then G has a closed Hamiltonian path.

(See the Appendix for a proof of Ore's Theorem.) We now define a family of graphs for which a fairly strong necessary condition is known.

Definition 3.10. A graph G is said to be *bipartite* if its vertex set $V(G)$ can be partitioned into disjoint subsets A and B so that every edge of G joins a vertex of A to a vertex of B. We say $\{A, B\}$ is a *bipartition* of G. A bipartite graph is *complete* if it is simple and every vertex in A is joined to every vertex in B.

Example 3.11. (a) Let $K_{m,n}$ denote the complete bipartite graph on disjoint sets of sizes m and n. For example, $K_{3,3}$.

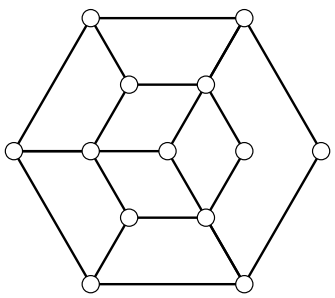


(b) The graph of the cube is bipartite.

Exercise (to appear with hints on sheet 2): Show that a connected bipartite graph has a unique bipartition.

Theorem 3.12. *If G is a bipartite graph with bipartition $\{A, B\}$ then G has a Hamiltonian path only if $|A| = |B|$.*

Example 3.13. The graph below does not have a Hamiltonian path.



Planar graphs

Definition 3.14. A graph is *planar* it can be drawn in the plane with no two edges crossing.

Planar graphs arise naturally in road networks, printed circuit boards, geographical maps etc. Projecting a convex polyhedron onto a plane gives a planar graph: for example, the graphs of the cube and dodecahedron.

A well-known problem asks whether it is possible to connect up three houses to three utilities (say gas, waterworks, electricity) by non-crossing pipes. If tunnelling is banned, the answer is given by the following theorem.

Theorem 3.15. *The complete bipartite graph $K_{3,3}$ is not planar.*

This does almost all the work needed to prove:

Theorem 3.16. *Let $m, n \in \mathbf{N}$ with $m \leq n$. The complete bipartite graph $K_{m,n}$ is planar if and only if $m \leq 2$.*

The proof suggests we should make the following definition.

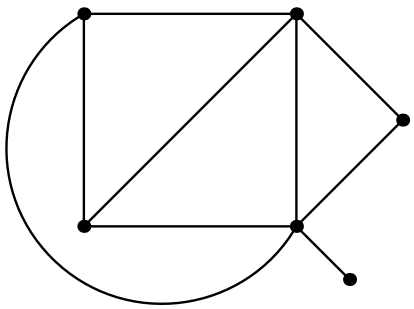
Definition 3.17. Let G be a graph (not necessarily planar). A subgraph of G is a graph all of whose vertices belong to $V(G)$, and all of whose edges appear in the list $E(G)$.

For example, a graph has the chain of length n as a subgraph if and only if has a non-closed path of length n.

Definition 3.18. Let G be a planar graph. The regions formed when we draw G in the plane without crossing edges are called *faces*.

As our graphs are finite, there is always one unbounded face.

Example 3.19. The graph below has 6 vertices, 9 edges and 5 faces.



Theorem 3.20 (Euler)**.** *Let G be a connected planar graph with n vertices and e edges. Suppose G is drawn in the plane without crossing edges. If f is the number of faces in this drawing then*

$$n - e + f = 2.$$

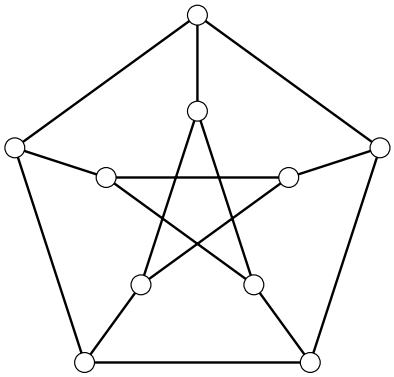
One can use Euler's Theorem to prove that some graphs are not planar.

Theorem 3.21. *The complete graph on 5 vertices, K_5, is non-planar.*

A nice application of this theorem occurs in graph colouring: see question 4 on problem sheet 2. We introduce the definition here.

Definition 3.22. A graph G is *k-colourable* if it possible to assign one of k colours to each vertex so that adjacent vertices have different colours.

Example 3.23. (a) The Petersen graph is 3-colourable.



(b) A graph is 2-colourable if and only if it is bipartite.

(c) The famous Four Colour Theorem states that any planar graph is 4-colourable.

Another interesting application occurs in elementary geometry. (This will be omitted if time is pressing.)

Theorem 3.24. *Suppose that there is a convex polyhedron whose faces are congruent r-gons, and such that exactly s faces meet at every vertex. Then*

$$0 < (r-2)(s-2) < 4$$

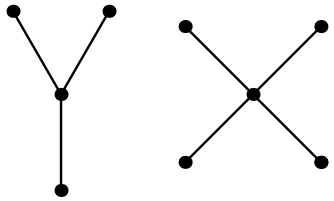
and the polyhedron appears in the table below.

r	s	
3	3	tetrahedron
3	4	octahedron
3	5	icosahedron
4	3	cube
5	3	dodecahedron

4 Trees

Definition 4.1. A *forest* is a simple graph with no closed paths. A *tree* is a connected forest. A *leaf* is a vertex of a forest of degree 1.

For example, the graph below is a forest with 2 connected components (each a tree).



Practical applications of graph theory often involve trees. For example, the minimal connector problem: given n cities separated by various distances, what is the shortest road network that will connect them all? (See Cameron §11.3.)

Theorem 4.2. *A tree on n vertices has exactly $n-1$ edges.*

This was used at the end of the proof of Theorem 3.20 (Euler's formula).

Theorem 4.3. *Let T be a graph on $n \geq 1$ vertices. The following are equivalent:*

(i) *T is a tree;*

(ii) *T is a forest and has $n-1$ edges;*

(iii) *T is connected and has $n-1$ edges;*

(iv) *T is connected, but becomes disconnected if we remove any single edge.*

See problem sheet 3 for another characterisation of trees.

Definition 4.4. Let G be a connected graph. A *spanning tree* is a subgraph of G which is a tree, and which contains all of the vertices of G.

It should be fairly clear that any connected graph has a spanning tree: just remove edges involved in closed paths until no closed paths are left.

Example 4.5. Will give an example of this process.

We use this to prove the following theorem.

Theorem 4.6. *If G is a graph with exactly k connected components then G has at least $n-k$ edges. Equality holds if and only if G is a forest.*

We now consider an enumeration problem concerning trees: how many spanning trees are there in K_n, the complete graph on $\{1, 2, \ldots, n\}$? We regard two spanning trees in K_n as different if they involve different edges of K_n. (As abstract graphs, they could still be isomorphic.)

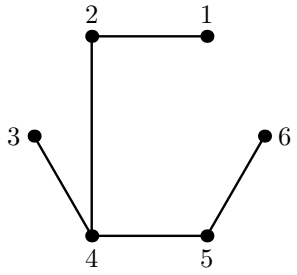
Example 4.7. There is a unique spanning tree in K_2. There are 3 spanning trees in K_3. There are 16 spanning trees in K_4.

We solve this problem by using a clever way to encode spanning trees in K_n as sequences of length $n-2$ in the numbers $\{1, 2, \ldots, n\}$.

Algorithm 4.8. *(Prüfer). Let $n \geq 2$. Let T be a spanning tree in K_n.*

1. *Set $k = 1$.*
2. *Find the leaf of T with the smallest label and delete it from T. Let s_k be the label of its parent vertex.*
3. *If $k = n - 2$ then the algorithm is finished, with resulting sequence $(s_1, \ldots, s_{n-2})$. Otherwise increment k by 1 and return to step 2.*

Example 4.9. (a) Will apply the algorithm to the spanning tree in K_6 shown below.



(b) Will show how to construct the unique spanning tree in K_6 associated to the sequence $(2, 2, 1, 5)$.

Theorem 4.10. *For all $n \geq 2$, Prüfer's algorithm gives a bijection between spanning trees in K_n and sequences of length $n-2$ in the numbers $\{1, 2, \ldots, n\}$. In particular, the number of spanning trees in K_n is n^{n-2}.*

5 Ford–Fulkerson Algorithm

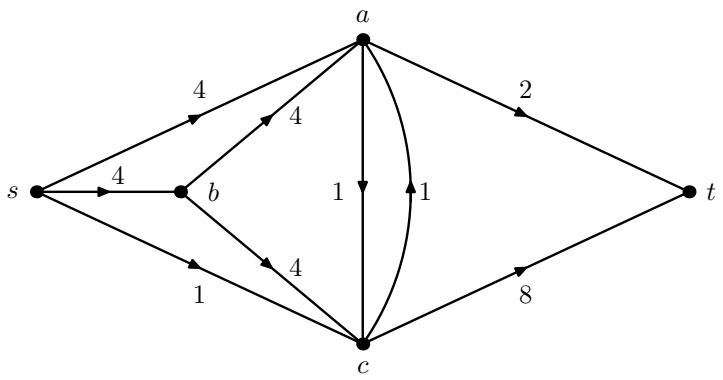
Definition 5.1. A *digraph*, or *directed graph*, is a set V of *vertices* together with a list E of ordered pairs of elements of V, not containing any pair of the form (x, x). If $(x, y) \in E$ then we say that (x, y) is an *edge* with *initial vertex* x and *terminal vertex* y.

The only digraphs we will consider in this course have some further structure.

Definition 5.2. A *network* is a digraph with no repeated edges and with two distinguished vertices, a *source*, which is never a terminal vertex of an edge, and a *target*, which is never an initial vertex of an edge. Each edge (x, y) has an assigned *capacity* $c(x, y) \in \mathbf{N}$.

Note that we allow both (x, y) and (y, x) to appear as edges of a network.

Example 5.3. The diagram below shows a network with source s and target t. The capacities are indicated by numbers attached to the edges.



The underlying digraph has vertex set $\{s, t, a, b, c\}$ and edge list (s, a), (s, b), (s, c), (b, a), (b, c), (a, c), (c, a), (a, t), (c, t). Note that no edge goes into s or comes out of t.

One useful model to bear in mind is a network of water pipes and tanks. Water enters at the source, and out at target. An edge (x, y) corresponds to a pipe from tank x to tank y: without pumping water flows downhill, so it is natural to deal with directed edges. The capacity of an edge gives the maximum amount of water that can flow through the pipe in an hour.

We aim to maximise the amount of water that can flow from the source to the target in one hour.

Definition 5.4. Let N be a network with source s and target t and edge list E. A *flow* in N is assignment of a non-negative real number $f(x, y)$ to each edge (x, y) of N such that

(i) $f(x, y) \leq c(x, y)$ for all edges (x, y) of N.

(ii) if x is a vertex of N other than s and t then

$$\sum_{y:(x,y)\in E} f(x, y) = \sum_{z:(z,x)\in E} f(z, x).$$

Condition (ii) says that flow is conserved at vertices other than the source and target: the first sum is the flow leaving x, the second the flow coming into x.

Example 5.5. Will give an example of a flow in the network above.

Definition 5.6. Let f be a flow in a network with source s and edge list E. The *value* of f is the total flow leaving s,

$$\operatorname{val} f = \sum_{y:(s,y)\in E} f(s, y).$$

A flow is said to be *maximal* if its value is as large as possible.

As one might hope, this definition is not as asymmetric as it appears: see Sheet 3, Question 7.

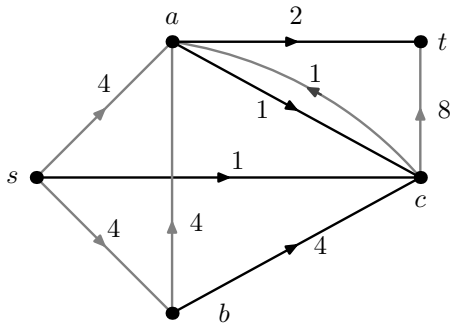
Definition 5.7. Let N be a network. A *cut* (S, T) of N is a partition of its vertices into subsets S, T such that $s \in S$ and $t \in T$.

The *capacity* of the cut (S, T) is

$$\operatorname{cap}(S, T) = \sum c(x, y)$$

where the sum is over all edges (x, y) of N such that $x \in S$ and $y \in T$.

Example 5.8. Let N be the network shown in Example 5.3. Let $S = \{s, a, b\}$ and let $T = \{c, d\}$. The capacity of the cut (S, T) is 8.



Theorem 5.9. *Let N be a network. If f is a flow in N and (S,T) is a cut of N then* $\operatorname{val} f \leq \operatorname{cap}(S,T)$. *In particular, if equality holds then f is a maximal flow.*

See Appendix A for two different proofs of this theorem.

Theorem 5.9 gives a practical way to prove a flow is maximal: it is enough to exhibit a cut whose capacity is equal to the given flow.

Example 5.10. Take the network from Example 5.3. In Example 5.5 we saw that it had a flow with value 8, and in Example 5.8 we found a cut with capacity 8. This shows that the maximum value of any flow is 8, so the flow we found is maximal.

In fact this approach always works: in any network there is always a cut with capacity equal to the maximum flow value. We give a constructive proof of this by using the following algorithm.

Algorithm 5.11 (Ford–Fulkerson algorithm)**.** *Let N be a network with source s and target t and vertex set V and let f be an flow in N taking integral values.*

1. *Set $S = \{s\}$.*
2. *If $x \in S$ and there is an edge (x,y) of N with $f(x,y) < c(x,y)$ then add y to S.*

   If $y \in S$ and there is an edge (x,y) of N with $f(x,y) > 0$ then add x to S.
3. *Repeat step 2 until no new vertices are added to S. There are now two possibilities.*

   (A) *If $t \in S$ then, by construction of S, there is a path $(x_0, x_1, \ldots, x_m)$ in the underlying graph of N such that $x_0 = s$, $x_m = t$ and for each i,* either

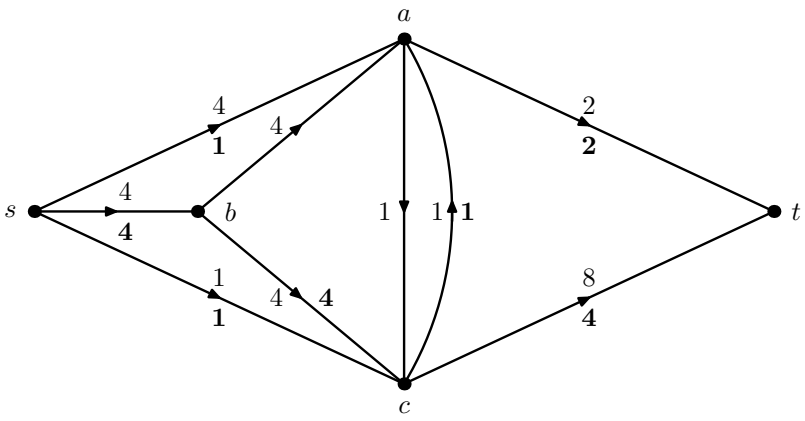
   (x_i, x_{i+1}) *is an edge of N and $f(x_i, x_{i+1}) < c(x_i, x_{i+1})$* or
   (x_{i+1}, x_i) *is an edge of N and $f(x_{i+1}, x_i) > 0$.*

   Increase the flow in edges of the first type by 1 and decrease the flow in edges of the second type by 1. This gives a new flow f^+ with $\operatorname{val} f^+ = \operatorname{val} f + 1$.

   (B) *If $t \notin S$ then $(S, V \backslash S)$ is a cut of N with capacity equal to* $\operatorname{val} f$.

So, assuming we believe the assertions made in 3(A) and 3(B), either we manage to increase the flow, or we find a cut with capacity equal to the value of the existing flow. The Appendix contains a proof that the Ford–Fulkerson algorithm works as claimed.

Example 5.12. Will apply the Ford–Fulkerson algorithm to find a maximal flow in the network of Examples 5.3, 5.5, 5.8. We start with the flow with value 6 shown in bold on the diagram below.



Theorem 5.13 (Max-flow-min-cut Theorem)**.** *Let N be a network. There is an integer-valued flow f in N and a cut (S,T) of N such that* $\operatorname{val} f = \operatorname{cap}(S,T)$.

6 GROUPS: Permutation groups

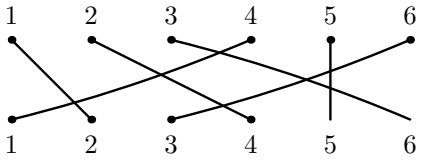
Recall that if X is a set then the *symmetric group* on X is the set of all bijective functions from X to itself, with the group operation being composition of functions. We denote this group by $\mathrm{Sym}(X)$ and call its elements *permutations*.

If h, k are permutations then we shall write hk rather than $h \circ k$ for their composition. Similarly h^2 means $h \circ h$, $h^{-2} = h^{-1} \circ h^{-1}$, and so on. We write e for the permutation sending each $x \in X$ to itself. (*Exercise*: check that $\mathrm{Sym}(X)$ reallly is a group and that e is its identity element.)

There are various notations for permutations. For example, if $X = \{1, 2, 3, 4, 5, 6\}$ and $h \in \mathrm{Sym}(X)$ is the permutation defined by

$$h(1) = 2, h(2) = 4, h(3) = 6, h(4) = 1, h(3) = 6, h(5) = 5, h(6) = 3$$

and represented by the diagram below



then in *cycle-notation*, $h = (124)(36)$ and in *one-line notation*, $h = 246153$. See page 3 for another example.

It is fairly easy to multiply permutations written in cycle form. For example, if $k = (135) \in \mathrm{Sym}(X)$ then we find that

$$hk(1) = h(3) = 6,\ hk(6) = h(6) = 3,\ hk(3) = h(5) = 5, \quad \text{etc}$$

hence $hk = (163524)$.

Revision exercise:

(1) With h, k as defined above, check that $h^2 = (142)$ and $h^{-1} = (142)(36)$ and that h has order 6. Show that $kh \neq hk$.

(2) Show that if $g \in \mathrm{Sym}(X)$ is a permutation of a set X and $x_1, \ldots, x_r$ are any elements of X then

$$g(x_1\ x_2\ \ldots\ x_r)g^{-1} = (g(x_1)\ g(x_2)\ \ldots\ g(x_r)).$$

This is often useful when calculating with permutations.

Definition 6.1. Let X be a set. A *permutation group* on X is a subgroup of the symmetric group on X.

Recall that that a subset $H \subseteq \mathrm{Sym}(X)$ is a subgroup of $\mathrm{Sym}(X)$ if and only if

(1) $e \in H$;
(2) if $h \in H$ then $h^{-1} \in H$;
(3) if $h, k \in H$ then $hk \in H$.

Subgroups are groups in their own right. Often the most efficient way to show that a given object is a group is to show that it is a subgroup of some known group, for example, a symmetric group.

Example 6.2. (1) Let $X = \{1, 2, 3, 4, 5, 6\}$ and let H be the subgroup of $\mathrm{Sym}(X)$ generated by the permutations $(123)(56)$ and (14):

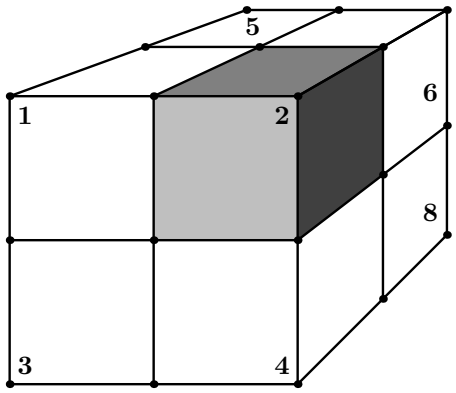
$$H = \langle (123)(56), (14) \rangle .$$

By taking suitable products of the generators one can show that $(12), (23), (34) \in H$ and that H is the set of all permutations of X which permute within themselves elements of the sets $\{1, 2, 3, 4\}$ and $\{5, 6\}$. That is, H is the direct product

$$H = \mathrm{Sym}(\{1, 2, 3, 4\}) \times \mathrm{Sym}(\{5, 6\}).$$

For example, $(234)(56) \in H$ but $(45) \notin H$.

(2) The *simplified Rubik's cube* is a $2 \times 2 \times 2$ cube with faces coloured with 6 different colours. We label the 8 positions as shown below.



In this diagram the shaded cube occupies position 2. Position 7 is the far-left-bottom of the cube.

Any face may be rotated 90° anticlockwise, giving 6 permutations:

$$f = (1342),\ p = (5786),\ r = (2486),\ l = (1375),\ t = (1562),\ b = (3784).$$

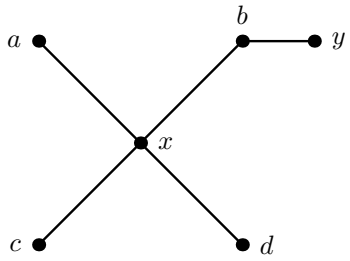
The (position) group of the cube is the subgroup of $\mathrm{Sym}(\{1, 2, \ldots, 8\})$ generated by these permutations. For example, first rotating the front face, then the right face, gives the permutation

$$rf = (2486)(1342) = (13862).$$

This permutation sends the cube in position 1 to position 3, the cube in position 3 to position 8, and so on.

Exercise: show that $r^{-1}f^{-1}rf = (12)(48)$. (With this and similar sequences you can now swap any two adjacent cubes, at the cost of disturbing just two others. Of course further work may be required to put these cubes in the right orientation.)

(3) Let G be the graph shown below.

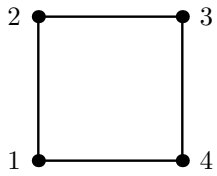


Let $X = \{a, b, c, d, x, y\}$ be the set of vertices. Let H be the subgroup of $\mathrm{Sym}(X)$ consisting of the permutations which preserve edges of G. That is,

$$H = \big\{h \in \mathrm{Sym}(X) : \{u, v\} \text{ is an edge of } G \iff \{h(u), h(v)\} \text{ is an edge of } G\big\}.$$

For example, $(ac) \in H$. But $(ab) \notin H$ as $\{b, y\}$ is an edge, but $\{a, y\}$ is not. (In lectures I gave this definition with $\implies$ in place of $\iff$. This defines the same group, as if $\{h(u), h(v)\}$ is an edge then, applying the definition from lectures with h^{-1}, we find that $\{u, v\}$ is an edge. The more symmetric definition is probably preferable.)

(4) Let G be the closed path of length 4 shown below



and let

$$H = \big\{h \in \mathrm{Sym}(X) : \{u, v\} \text{ is an edge of } G \iff \{h(u), h(v)\} \text{ is an edge of } G\big\}.$$

Exercise: Show that H is isomorphic to the dihedral group of order 8. *Outline solution:* first show that $(1234) \in H$. Now show that if $h \in H$ then $(1234)^a h(1) = 1$ for some suitable power a. Finally show that the only permutations in H which fix 1 are the identity, e, and (24). Deduce that $h \in \langle (1234), (24) \rangle$, and so $H = \langle (1234), (24) \rangle$.

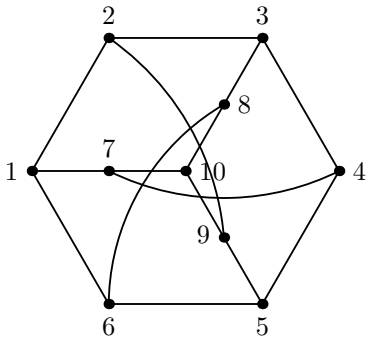
Another way to think about examples (3) and (4) uses the idea of a graph isomorphism, as given in Definition 2.5. Recall if G is a simple graph with vertex set X, then $h : X \to X$ is an isomorphism from G to itself if and only if

(1) h is a bijection;

(2) for all $x, y \in X$, $\{x, y\}$ is an edge of G if and only $\{h(x), h(y)\}$ is an edge of G.

The subgroups H therefore consist of all isomorphisms from G to itself.

Example 6.3. Let G be the Petersen graph (shown below) and let H be the group of isomorphisms from G to itself.



Will use the symmetry of the graph to find some elements of H.

Definition 6.4 (Orbits and stabilisers)**.** Let X be a set and let $H \leq \mathrm{Sym}(X)$ be a permutation group. Let $x \in X$. The *orbit of* x is the set

$$\mathrm{Orb}(x) = \{h(x) : h \in H\}.$$

The *stabiliser of* x is the subgroup

$$\mathrm{Stab}(x) = \{h \in H : h(x) = x\}.$$

We write *the orbit of* H *on* x if the permutation group needs to be emphasised.

Example 6.5. (1) Let $X = \{1, 2, 3, 4, 5, 6\}$. Let

$$H = \langle (123)(56) \rangle \leq \mathrm{Sym}(X)$$

Will determine the orbits of H on elements of X and find the stabiliser of each element.

(2) Let $X = \{1, 2, 3, 4\}$ and let $H \leq \mathrm{Sym}(X)$ be the group generated by $(12)(34)$ and (13). The orbit of any $x \in X$ is X. There are two possible stabilising subgroups.

(3) (A geometric example). Let $X = \mathbf{R}^2$ and let H be the group of all rotations of the plane about the origin.

Lemma 6.6. *Let X be a set and let $H \leq \mathrm{Sym}(X)$ be a permutation group. Define a relation on X by $x \sim y$ if y is in the orbit of x. This is an equivalence relation. The equivalence class of $x \in X$ under $\sim$ is the orbit of x.*

Hence if H is a permutation group on a set X then X splits up into disjoint orbits of H.

Theorem 6.7 (Orbit-Stabiliser Theorem)**.** *Let X be a set and let H be a finite permutation group on X. If $x \in X$ then*

$$|\operatorname{Orb}(x)| = \frac{|H|}{|\operatorname{Stab} x|}.$$

In particular, the size of the orbit divides the size of the group.

Here is one nice application of the Orbit-Stabiliser Theorem. (Our main application occurs in the next section.)

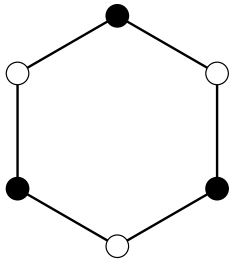
Theorem 6.8. *The group of all rotational and reflectional symmetries of the regular n-gon has order $2n$. It consists of n rotations (counting the identity element e as a rotation of 0°) and n reflections.*

The group in the last theorem is known as the *dihedral group* of order $2n$.

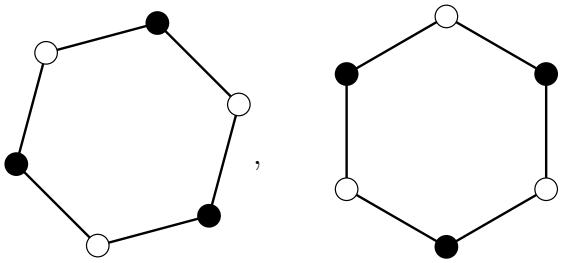
7 Orbit counting and necklaces

Definition 7.1. Let $c \in \mathbf{N}$. A *necklace* on c colours is a regular polygon with vertices coloured with any of c different colours. (It is not necessary that every colour is used.) The *length* of a necklace is its number of vertices.

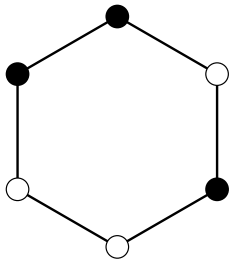
The diagram below shows a necklace on 2 colours of length 6.



We regard two necklaces as the same if one can be rotated into the other. (Later we will also allow turning the necklace over.) So the necklaces below should be regarded as the same as the necklace above.



But this necklace is different to those seen so far.



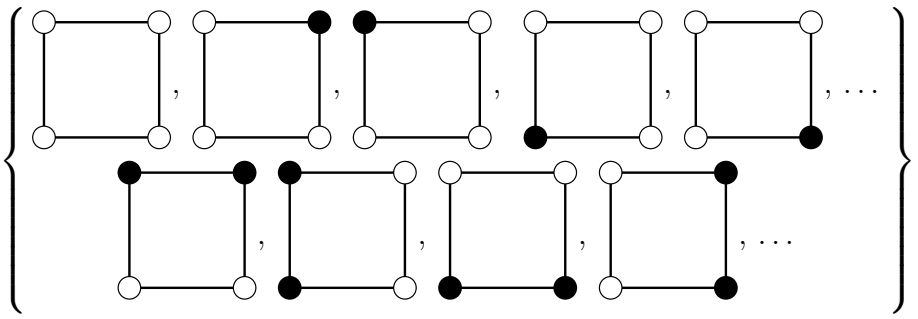
Example 7.2. (1) There are 6 necklaces on 2 colours of length 4.

(2) There are 4 necklaces of length 6 with 3 black vertices and 3 white vertices.

Clearly naïve enumeration will not be a practical method if we want to allow more colours or consider longer necklaces.

Instead we convert the problem of counting necklaces into a problem about counting orbits of a permutation group.

Example 7.3. Let X be the set of all 16 ways to colour the vertices of a square using the colours black and white:

$X =$ 

Let h be the permutation of X induced by rotating diagrams by 90°. Let $H = \langle h \rangle \leq \mathrm{Sym}(X)$. There is a bijection

$$\left\{\text{orbits of } H \text{ on } X\right\} \leftrightarrow \left\{\begin{matrix}\text{necklaces of length 4} \\ \text{on 2 colours}\end{matrix}\right\}$$

The next theorem gives a practical way to count the number of orbits of a permutation group on a set[7]. The proof is a nice exercise in double-counting.

Theorem 7.4 (Orbit-Counting Theorem)**.** *Let X be a set and let $H \leq \mathrm{Sym}(X)$ be a permutation group. For $h \in H$, let*

$$\mathrm{Fix}\, h = \{x \in X : h(x) = x\}.$$

The number of orbits of H on X is given by

$$\frac{1}{|H|} \sum_{h \in H} |\,\mathrm{Fix}\, h|.$$

Example 7.5. Will check that the Orbit-Counting Theorem gives the expected answer in the case of necklaces of length 4 on 2 colours.

The next example is sufficiently general to show the ideas needed to enumerate necklaces of any length.

[7]Most books on permutation groups will call this result 'Burnside's Lemma'. As is often the case, this is a misattribution. An increasingly popular renaming is 'not-Burnside's Lemma'.

Example 7.6. The number of different necklaces on c colours of length 6 is

$$\frac{1}{6}(c^6 + c^3 + 2c^2 + 2c)$$

If $c = 2$ there are 14 necklaces in all. As found in Example 7.2, exactly 4 of these necklaces have 3 beads of each colour.

If we also regard two necklaces as the same if one is a reflection of the other, then the number of different necklaces on c colours is

$$\frac{1}{12}(c^6 + 3c^4 + 4c^3 + 2c^2 + 2c).$$

We end with an unexpected corollary of counting necklaces of prime length.

Theorem 7.7 (Fermat's Little Theorem)**.** *Let p be a prime number and let $c \in \mathbf{N}$. The number of necklaces on c colours of length p is*

$$\frac{c^p + (p-1)c}{p}.$$

Hence p divides $c^p - c$.

8 GENERATING FUNCTIONS: Partitions

Definition 8.1. A partition of a number $n \in \mathbf{N}_0$ is a sequence of natural numbers $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ such that

(i) $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq 1$

(ii) $\lambda_1 + \lambda_2 + \ldots + \lambda_k = n$.

Let $p(n)$ be the number of partitions of n. The entries in a partition are called *parts*.

By this definition, $\varnothing$ is the unique partition of 0.

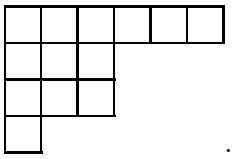
Example 8.2. (1) The number of ways to make change for 30p using 5p and 2p coins is the number of partitions of 30 into parts of size 5 and 2.

(2) The number of ways to put n unlabelled balls into 3 identical urns is the number of partitions of n with at most 3 parts.

It is often useful to represent partitions by *Young diagrams*[8].

Definition 8.3. The *Young diagram* of the partition $(\lambda_1, \ldots, \lambda_k)$ has k rows of boxes. The ith row (from the top) has λ_i boxes, aligned from the left.

For example the Young diagram of $(6, 3, 3, 1)$ is



.

When studying partitions it is often useful to use generating functions.

Definition 8.4. Let $f : \mathbf{N}_0 \to \mathbf{N}_0$. The *generating function* of f is the power series

$$f(0) + f(1)x + f(2)x^2 + f(3)x^3 + \ldots.$$

Example 8.5. Let $f(n)$ be the number of partitions of n into parts of size 5 and 2. The generating function for f is

$$\frac{1}{(1-x^2)(1-x^5)}.$$

Theorem 8.6. *The generating function for $p(n)$ is*

$$\sum_{n=0}^{\infty} p(n)x^n = \frac{1}{(1-x)(1-x^2)(1-x^3)\ldots}$$

[8]Named after the Reverend Alfred Young: see `www-groups.dcs.st-and.ac.uk/~history/Biographies/Young_Alfred.html`

We end with a nice application of generating functions. Say that a partition has *distinct parts* if it has at most one part of any given size.

Theorem 8.7. *The number of partition of n into parts of odd size is equal to the number of partition of n with distinct parts.*

van Lint's argument*

Here is an elegant way to obtain an upper bound on $p(n)$, due to the Dutch mathematician van Lint. Let $P(x)$ be the generating function for $p(n)$.

Standard power series manipulations give that

$$\log P(x) = -\sum_{m=1}^{\infty} \log(1-x^m) = \sum_{m=1}^{\infty}\sum_{r=1}^{\infty} \frac{x^{mr}}{r} = \sum_{r=1}^{\infty} \frac{x^r}{r(1-x^r)}$$

Replace x with e^{-y} to get

$$\log P(\mathrm{e}^{-y}) = \sum_{r=1}^{\infty} \frac{1}{r(e^{ry}-1)} \leq \sum_{r=1}^{\infty} \frac{1}{y}\frac{1}{r^2} = \frac{1}{y}\frac{\pi^2}{6}.$$

In the last line we used that $e^{ry} \geq 1 + ry$ (take the power series) and that

$$\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \ldots = \frac{\pi^2}{6}.$$

Now $P(\mathrm{e}^{-y}) \geq p(n)e^{-ny}$ since the right-hand-side is just one term in the sum defining P. Hence

$$\log(p(n)\mathrm{e}^{-ny}) \leq \frac{1}{y}\frac{\pi^2}{6},$$

or equivalently,

$$\log p(n) \leq ny + \frac{1}{y}\frac{\pi^2}{6}.$$

The right-hand-side is minimised by taking

$$y = \sqrt{\frac{\pi^2}{6n}}.$$

With this choice we obtain a fairly strong upper bound on $p(n)$:

$$\log p(n) \leq 2\sqrt{\frac{\pi^2}{6}}\sqrt{n}.$$

9 Appendix: Selected proofs

9.1 Ore's Theorem

Here is a more carefully explained proof of Ore's Theorem than the one given in lectures. The first two steps are illustrated by the attached example. *This proof may be considered non-examinable.*

Theorem 3.9 (Ore)**.** *Let G be a simple graph on n vertices. If $n \geq 3$, and*

$$\delta(x) + \delta(y) \geq n$$

for each pair of non-adjacent vertices x and y, then G has a closed Hamiltonian path.

Proof. Suppose, for a contradiction, that G does not have a closed Hamiltonian path.

1. Pick any any two vertices of G which aren't already joined by an edge, and add a new edge between them. Keep on doing this until we reach a graph G_{last} which *does* have a closed Hamiltonian path. (The process must stop because eventually we will reach the complete graph on n vertices, which obviously has a closed Hamiltonian path.)

2. Let $\bar{G}$ be the graph obtained immediately before G_{last}, and suppose that $\{x, y\}$ is the edge added to $\bar{G}$ to obtain G_{last}.

Let $(z_1, \ldots, z_n, z_1)$ be a closed Hamiltonian path in G_{last}. This must use the edge $\{x, y\}$ at some point (otherwise $\bar{G}$ would have a closed Hamiltonian path, and there would have been no need to consider G_{last}). If $\{z_n, z_1\} = \{x, y\}$ then $(z_1, \ldots, z_n)$ is a non-closed Hamiltonian path in $\bar{G}$. Otherwise there is some r such that $1 \leq r < n$ and $z_r = x$ and $z_{r+1} = y$; now

$$(z_{r+1}, \ldots, z_n, z_1, \ldots, z_r)$$

is a non-closed Hamiltonian path in $\bar{G}$. Note that either way, all the edges used in this path appear in $\bar{G}$: it is only $\{x, y\}$ that appears in G_{last} but not in $\bar{G}$. Relabel the vertices so that this path is $(x_1, \ldots, x_n)$.

3. Suppose we could find a vertex x_i such that x is adjacent to x_i, and y is adjacent to x_{i-1}. Then

$$(x, x_i, \ldots, x_{n-1}, y, x_{i-1}, \ldots, x)$$

would be a closed Hamiltonian path in $\bar{G}$, a contradiction.

Aside: It is at this point that we need $n \geq 3$: if $n = 2$ then the first step is (x, y), and the second is (y, x), which means we have used an edge twice.

Paths are, in particular, trails, so they aren't allowed to repeat edges. As long as $n \geq 3$ this problem doesn't arise.

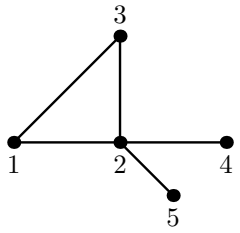
4. It remains to show that there must be such a vertex x_i. This is where we need the hypothesis on degrees. Since $\bar{G}$ is obtained from G by adding edges, it still satisfies this hypothesis. Let

$$A = \{i : 2 \leq i \leq n \text{ and } x_i \text{ is adjacent to } x\},$$
$$B = \{i : 2 \leq i \leq n \text{ and } x_{i-1} \text{ is adjacent to } y\}.$$

As our graphs have no loops, $|A| = \delta(x)$ and $|B| = \delta(y)$. As x and y are not adjacent in $\bar{G}$ (recall that $\{x, y\}$ was added to $\bar{G}$ to obtain G_{last}), our hypothesis tells us that $\delta(x) + \delta(y) \geq n$.

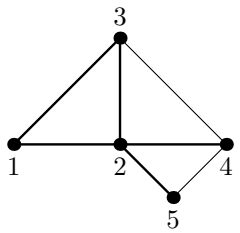
Hence A and B are subsets of $\{2, \ldots, n\}$ containing at least n elements between them. It follows that they must intersect non-trivially. If $i \in A \cap B$ then x_i is a suitable vertex for step 3. □

Example: Let $n = 5$. The graph below has vertex set $\{1, 2, 3, 4, 5\}$ and edges $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$, $\{2, 4\}$, $\{2, 5\}$.



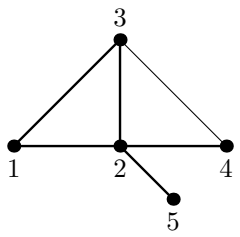
(This graph doesn't satisfy the hypothesis on the degrees, but we don't use this until step 4. This saves drawing a large number of edges which would be irrelevant in steps 1 and 2.)

1. We might first add the edge $\{3, 4\}$. The resulting graph still doesn't have a closed Hamiltonian path, so we add another edge, say $\{4, 5\}$. This gives the graph



which has $(1, 2, 5, 4, 3, 1)$ as a closed Hamiltonian path. (So $z_1 = 1$, $z_2 = 2$, $z_3 = 5$, $z_4 = 4$, $z_5 = 3$.)

2. The last edge added is $\{x, y\} = \{4, 5\}$ so $\bar{G}$ is as shown below.



Starting with the closed path $(1, 2, 5, 4, 3, 1)$ in G_{final} we find that $r = 3$, $x = 5$, $y = 4$. The resulting non-closed Hamiltonian path in $\bar{G}$ is $(4, 3, 1, 2, 5)$. So in the relabelling step we take $x_1 = 4$, $x_2 = 3$, $x_3 = 1$, $x_4 = 2$, $x_5 = 5$.

9.2 Proof of Theorem 5.9

Theorem 5.9. *Let N be a network. If f is a flow in N and (S, T) is a cut of N then* $\operatorname{val} f \leq \operatorname{cap}(S, T)$. *In particular, if equality holds then f is a maximal flow.*

In lectures I proposed the following proof.

Proof 1. Think of S and T as countries separated by a sea, and the edges contributing to $\operatorname{cap}(S, T)$ as under-sea pipes flowing from S to T. It is obvious that the maximum amount of water that can flow from s, the capital city of S, to t, the capital city of T, is bounded by the total capacity of these pipes.

For the last part, suppose that $\operatorname{val} f = \operatorname{cap}(S, T)$. If g is any flow then, by what we have just shown, $\operatorname{val} g \leq \operatorname{cap}(S, T)$, i.e. $\operatorname{val} g \leq \operatorname{val} f$. Hence f is a maximal flow. □

Here is an alternative proof of the first part. (Probably the only thing that can be said for it is that it makes it makes explicit use of flow conservation.)

Proof 2. Let E be the list of edges of N. It is sufficient to prove that

$$\operatorname{val} f = \sum_{\substack{x \in S, y \in T \\ (x,y) \in E}} f(x, y) - \sum_{\substack{x \in S, y \in T \\ (y,x) \in E}} f(y, x)$$

since the first term is $\operatorname{cap}(S, T)$. For each vertex $x \in S$ other than s we have

$$\sum_{w:(x,w) \in E} f(x, w) - \sum_{z:(z,x) \in E} f(z, x) = 0$$

while for s we have

$$\sum_{w:(s,w) \in E} f(x, w) = \operatorname{val} f.$$

Adding up these equations over all $x \in S$ gives

$$\operatorname{val} f = \sum_{\substack{x \in S \\ w:(x,w) \in E}} f(x,w) - \sum_{\substack{x \in S \\ z:(z,x) \in E}} f(z,x).$$

We now split each sum up according to whether $w \in S$ or $w \in T$. This gives

$$\operatorname{val} f = \sum_{\substack{x \in S, w \in S \\ (x,w) \in E}} f(x,w) - \sum_{\substack{x \in S, z \in S \\ (z,x) \in E}} f(z,x) + \sum_{\substack{x \in S, w \in T \\ (x,w) \in E}} f(x,w) - \sum_{\substack{x \in S, z \in T \\ (z,x) \in E}} f(z,x).$$

Now note that the first two terms cancel (each is the total sum of all flows over edges lying entirely within S). □

9.3 Ford–Fulkerson Algorithm*

You may be asked to apply the Ford–Fulkerson Algorithm, but you will not be asked to provide a general proof that it works as claimed.

Algorithm 5.11 (Ford–Fulkerson algorithm)**.** *Let N be a network with source s and vertex set V and let f be an flow in N taking integral values.*

1. *Set $S = \{s\}$.*
2. *If $x \in S$ and there is an edge (x, y) of N with $f(x, y) < c(x, y)$ then add y to S.*

   If $y \in S$ and there is an edge (x, y) of N with $f(x, y) > 0$ then add x to S.
3. *Repeat step 2 until no new vertices are added to S. There are now two possibilities.*

   (A) *If $t \in S$ then, by construction of S, there is a path $(x_0, x_1, \ldots, x_m)$ in the underlying graph of N such that $x_0 = s$, $x_m = t$ and for each i,* either

   (x_i, x_{i+1}) *is an edge of N and $f(x_i, x_{i+1}) < c(x_i, x_{i+1})$* or

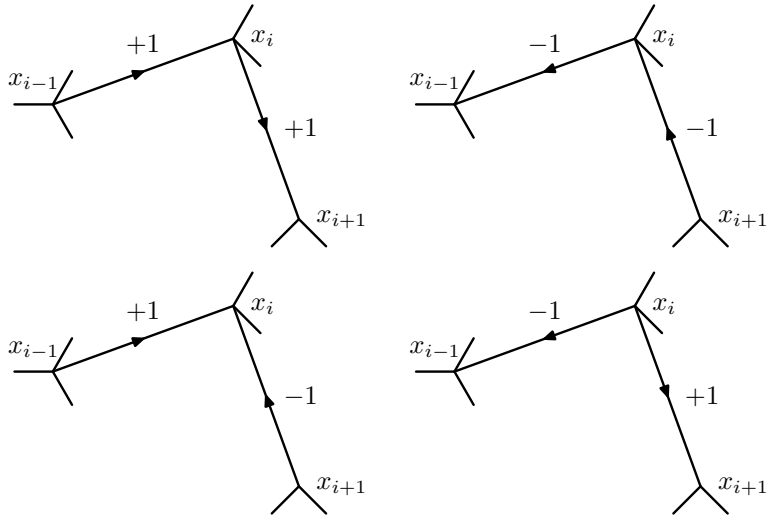
   (x_{i+1}, x_i) *is an edge of N and $f(x_{i+1}, x_i) > 0$.*

   Increase the flow in edges of the first type by 1 and decrease the flow in edges of the second type by 1. This gives a new flow f^+ with $\operatorname{val} f^+ = \operatorname{val} f + 1$.

   (B) *If $t \notin S$ then $(S, V \backslash S)$ is a cut of N with capacity equal to* $\operatorname{val} f$.

We claim that this algorithm either ends by increasing the flow in f or by giving a cut with capacity equal to $\operatorname{val} f$, thereby proving that f is a maximal flow. To show this we need to verify the assertions made in 3(A) and 3(B).

Case (A). Suppose we end in case (A). We must check that f^+ is a valid flow and that its value is 1 more than f. If we increase the flow on the edge (x_i, x_{i+1}) then this can only be because $f(x_i, x_{i+1}) < c(x_i, x_{i+1})$, so, as we work with integral flows, the capacity constraints are still satisfied. To show that flow conservation still holds requires some case-by-case checking. We only change the flow in edges meeting the vertices x_i. The diagrams below below show the four cases that can arise: in each case, one can easily see that the total flow into x_i equals the total flow out of x_i.



Finally, the value of f^+ is one more than the value of f because in f^+ we have an extra unit leaving s on the edge (s, x_1) and all other flows out of s are unchanged.

Case (B). Now suppose that we end in case (B). Let E be the set of edges of N. Let $(x, y) \in E$ with $x \in S$ and $y \in T$. If $f(x, y) < c(x, y)$ then we would have put y into the set S in Step 2 of the algorithm. Therefore $f(x, y) = c(x, y)$. Similarly, if $(y, x) \in E$ with $y \in T$ and $x \in S$ then $f(y, x) = 0$. Substituting in

$$\operatorname{val} f = \sum_{\substack{x \in S, y \in T \\ (x,y) \in E}} f(x, y) - \sum_{\substack{x \in S, y \in T \\ (y,x) \in E}} f(y, x)$$

we get that

$$\operatorname{val} f = \sum_{\substack{x \in S, y \in T \\ (x,y) \in E}} c(x, y) = \operatorname{cap}(S, T).$$

as required. (The first equation for val f was proved in the last handout.)

Exercise: For each of the networks below, find a good flow by inspection, and then use the Ford–Fulkerson algorithm to improve it until it is maximal.