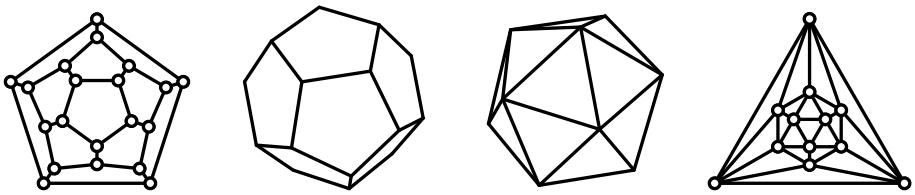
Graph Theory – v 1.00

Paul Buckingham



About these notes

These notes provide the core material for a course in graph theory taught at the University of Alberta. The topics are connectivity, trees, planarity, colouring, and flows. With the presentation I chose for the course, time did not allow for much treatment of flows, but I do provide here at least a discussion of the Max-Flow Min-Cut Theorem.

A mention of 'Wilson' in this document is a reference to the fifth edition of Robin J. Wilson's *Introduction to Graph Theory*.

I welcome being told about errors.

Contents

1 Basic definitions and concepts

Informally speaking, a graph is a collection of points with lines or sets of lines between them. Graphs are valuable mathematical objects in their own right, but a graph could also represent, for example,

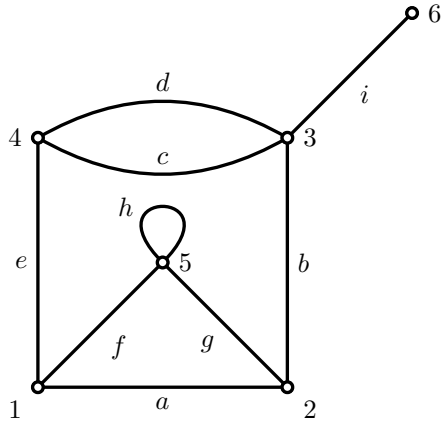
- a roadmap,
- a social network,
- a computer circuitboard,
- a factory production process,
- the file structure in a computer.

In the case of a roadmap, a practical application of graph theory is an algorithm to find the shortest route by road between two places. We will see how to do that later.

1.1 Definition of a graph

Formally speaking, a graph is an ordered triple $G = (V, E, \psi)$ where V is a non-empty set of *vertices*, E is a set of *edges* (disjoint from V), and ψ is a function that assigns to each edge a subset of V of cardinality 1 or 2. The vertex or vertices in the set assigned to e are called the *ends* of e, and e is said to *join* its ends. Note that the ends of an edge need not be distinct. An edge whose ends are the same is called a *loop*. The function ψ is called the *incidence function*.

Let us look at an example. In the graph



the vertices are 1, 2, 3, 4, 5, 6, and the edges are

$$\begin{aligned} &a, && \psi(a) = \{1, 2\} \\ &b, && \psi(b) = \{2, 3\} \end{aligned}$$

$$
\begin{aligned}
c, &\qquad \psi(c) = \{3,4\} \\
d, &\qquad \psi(d) = \{3,4\} \\
e, &\qquad \psi(e) = \{1,4\} \\
f, &\qquad \psi(f) = \{1,5\} \\
g, &\qquad \psi(g) = \{2,5\} \\
h, &\qquad \psi(h) = \{5\} \\
i, &\qquad \psi(i) = \{3,6\}
\end{aligned}
$$

In this course, we'll be interested only in *finite* graphs, those in which the vertex and edge sets are finite. But note that there can be infinite graphs as well.

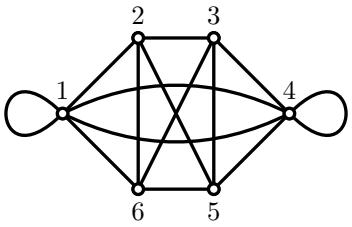
1.2 Adjacency and incidence

Two vertices v and w are said to be *adjacent* if there is an edge between them, i.e., if there is an edge e such that $\psi(e) = \{v, w\}$ where ψ is the incidence function. This includes the possibility that $v = w$, in which case the edge e is called a *loop*. Thus, a vertex may be adjacent to itself.

Two edges e and e' are said to be *adjacent* if $e \neq e'$ and $\psi(e) \cap \psi(e') \neq \emptyset$.

A vertex v is said to be *incident* with an edge e if v is an end of e, i.e., $v \in \psi(e)$. We also say in this situation that e is *incident* with v.

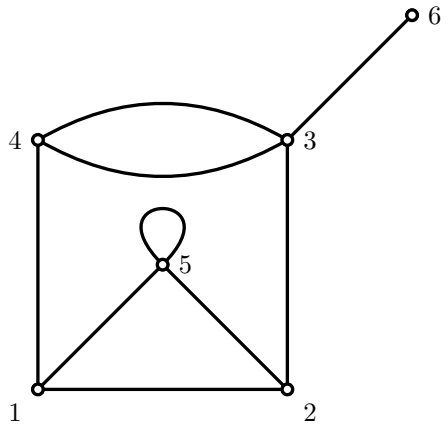
Example 1.1 Consider the graph G below having vertices $1, \ldots, 6$:



Here, there are 5 edges incident with vertex 1. The vertices adjacent to vertex 1 are $1, 2, 4, 6$. There are 4 edges incident with vertex 2. The vertices adjacent to vertex 2 are $1, 3, 5, 6$.

Let G be a graph with vertices $v_1, \ldots, v_n$. The *adjacency matrix* of G is the $n \times n$ matrix $A = (a_{i,j})_{i,j}$ such that $a_{i,j}$ is the number of edges of G having v_i and v_j as its ends, i.e., the number of edges e such that $\psi(e) = \{v_i, v_j\}$. Note that a loop counts only once, not twice. The adjacency matrix is symmetric.

For example, the graph



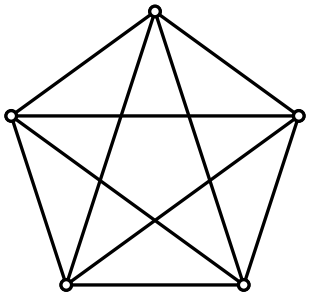
has adjacency matrix

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

1.3 Special examples and types of graph

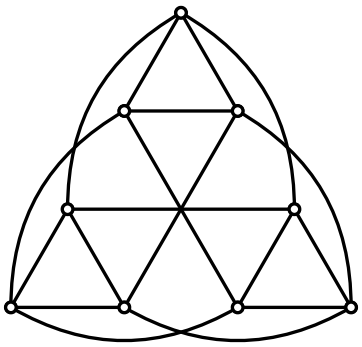
Null graphs. A *null graph* is a graph with no edges. They often form the base case of induction proofs concerning graphs.

Complete graphs. A graph is *complete* if it has no loops, and between every pair of distinct vertices there is one and only one edge. One denotes by K_n a complete graph on n vertices. It has $\binom{n}{2} = \frac{1}{2}n(n-1)$ edges. Here is K_5:

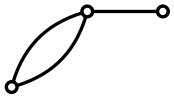


Simple graphs. A *simple* graph is a graph with no loops and no multiple edges. (Multiple edges are two or more edges with the same pair of ends.) That is, in the adjacency matrix, all entries lie in the set $\{0, 1\}$, and all entries along the main diagonal are zero.

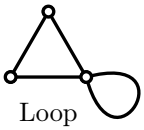
For example, every complete graph is simple, as is this one:



These graphs are not simple:



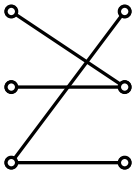
Multiple edges



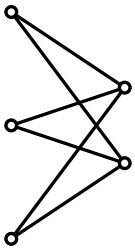
Loop

Bipartite graphs. A graph G is said to be *bipartite* if its vertex set can be partitioned into two non-empty sets A and B such that no two vertices of A are adjacent and no two vertices of B are adjacent. Equivalently, every edge of G has a vertex of A at one end and a vertex of B at the other.

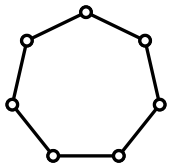
For example, the graph below is bipartite, with the vertices in the set A on the left and those in B on the right:



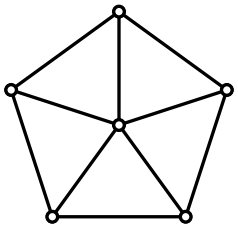
A *complete* bipartite graph is one in which v and v' are adjacent for every pair of vertices $v \in A$ and $v' \in B$. If A has m vertices and B has n, we denote the resulting complete bipartite graph by $K_{m,n}$. Here is $K_{3,2}$:



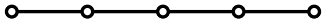
Cycle graphs. For each $n \geq 3$, the cycle graph C_n is the simple graph on n distinct vertices $v_1, \ldots, v_n$ with edges between v_i and v_{i+1} for $i \in \{1, \ldots, n-1\}$ and between v_n and v_1. For example, C_7 is the graph



Wheel graphs. For each $n \geq 4$, the wheel graph W_n is the simple graph on n distinct vertices $v_1, \ldots, v_n$ with edges between v_1 and v_i for all $i > 1$, between v_i and v_{i+1} for $i \in \{2, 3, \ldots, n-1\}$, and between v_n and v_2. For example, W_6 is the graph



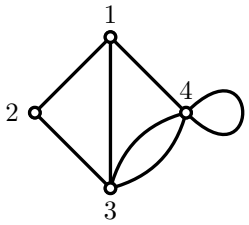
Path graphs. For each $n \in \mathbb{Z}_{\geq 1}$, we will let P_n denote the *path graph* on n vertices, the simple graph consisting of n vertices $v_1, \ldots, v_n$ such that v_i is adjacent to v_{i+1} for every $i \in \{1, \ldots, n-1\}$, with no other adjacencies. For example, this is P_5:



1.4 Vertex degrees and the handshaking lemma

If v is a vertex of a graph, its *degree* is the number of edges incident with it, loops counting twice. The degree of v will be denoted $d(v)$ in this course.

Example 1.2 In the graph below, $d(1) = 3$, $d(2) = 2$, $d(3) = 4$, and $d(4) = 5$.



Lemma 1.3 (Handshaking lemma) *If G is a graph whose number of edges is m, then*

$$\sum_{v \in V(G)} d(v) = 2m.$$

Proof. Each edge has two ends (even a loop), and each end contributes one to the sum of the vertex degrees. ■

Example 1.4 It follows from the handshaking lemma that the number of vertices of odd degree in any graph is even. Indeed, if V_0 is the set of vertices of even degree and V_1 the set of odd degree, then

$$2m = \sum_{v \in V(G)} d(v) = \sum_{v \in V_0} d(v) + \sum_{v \in V_1} d(v),$$

so

$$\sum_{v \in V_1} d(v) = 2m - \sum_{v \in V_0} d(v),$$

which is even. Since each term in the sum $\sum_{v \in V_1} d(v)$ is odd, there must be an even number of terms for the sum to have an even value.

Example 1.5 Let G be a graph whose vertex and edge sets have cardinalities n and m respectively. If δ is the least vertex degree in G and Δ the greatest, then

$$\delta \leq \frac{2m}{n} \leq \Delta.$$

Indeed,

$$2m = \sum_{v \in V(G)} d(v) \geq \sum_{v \in V(G)} \delta = n\delta,$$

and similarly $2m \leq n\Delta$.

Example 1.6 Let G be a graph on 5 vertices $v_1, \ldots, v_5$, and suppose that v_i is incident with i edges for each $i \in \{1, \ldots, 5\}$. Show that G must have a loop. Find such a graph.

Solution: Each non-loop incident with vertex i counts 1 towards $d(v_i)$, and each loop counts 2, so if l_i is the number of loops incident with vertex i, then using the assumption given that vertex i has i edges incident with it, we obtain

$$d(v_i) = (i - l_i) + 2l_i = i + l_i.$$

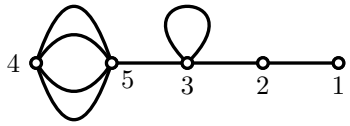
Hence, by the handshaking lemma,

$$2m = \sum_{i=1}^{5} (i + l_i) = \sum_{i=1}^{5} i + \sum_{i=1}^{5} l_i,$$

so

$$\sum_{i=1}^{5} l_i = 2m - \sum_{i=1}^{5} i,$$

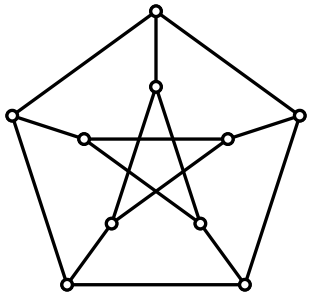
which is odd (it is equal to $2m - 15$, in fact). Therefore, $\sum_{i=1}^{5} l_i \neq 0$, so $l_i > 0$ for some i. An example is



1.5 Regular graphs

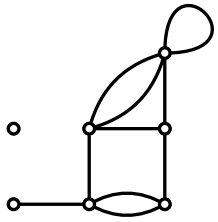
A *regular graph* is a graph in which all vertices have the same degree. If all degrees are k, then we say that the graph is *k-regular*. Any cycle graph is 2-regular, and K_n is $(n-1)$-regular.

A 3-regular graph is also referred to as a *cubic graph*. The Petersen graph is an example:



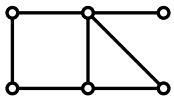
1.6 Degree sequences

The *degree sequence* of a graph is the sequence of vertex degrees in non-ascending order. For example, the degree sequence of the following graph is $(5, 4, 4, 3, 3, 1, 0)$:



It is a straightforward exercise to show that every non-ascending sequence of non-negative integers whose sum is even is the degree sequence of some graph. (Why must the sum of the integers in the sequence be even?)

A sequence $(d_1, d_2, \ldots, d_n)$ of non-negative integers is called *graphic* if it is the degree sequence of a simple graph. For example, the sequence $(4, 3, 2, 2, 2, 1)$ is a graphic sequence, being the degree sequence of the following simple graph:



There is an algorithm, due to Havel and Hakimi, that allows us to decide whether a given sequence is graphic and, if so, to construct a simple graph with that degree sequence.

Preparation for the Havel–Hakimi algorithm

For brevity, we say that a sequence $(d_1, \ldots, d_n)$ is *permissible* if $d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$ and $\sum_{k=1}^{n} d_k$ is even.

Suppose that $\alpha = (d_1, \ldots, d_n)$ is permissible, and let $\Delta = d_1$. We will say that α is *reducible* if $n \geq \max(\Delta + 1, 2)$ and $d_2, \ldots, d_{\Delta+1}$ are all positive, and say that α is *irreducible* otherwise. We describe two processes for producing a permissible sequence from a reducible one.

1. Remove the left-most entry, subtract 1 from each of the next $\Delta = d_1$ entries, and reorder the entries if necessary so that they are in non-ascending order again.

2. Remove the left-most entry, and then subtract 1 from each of Δ entries in the remaining sequence $(d_2, \ldots, d_n)$ in the following way: Find the greatest remaining entry (d_2 initially), find the right-most occurrence of it, and subtract 1 from that, then 1 from the one to its left, and so on until either you have made Δ subtractions or you have reached the left-most greatest entry. Then, as needed, find the next greatest remaining entry, go to its right-most occurrence, and repeat. Do this until you have subtracted 1 from Δ entries.

It is a simple matter to verify that these two processes yield the same permissible sequence. We will call it the *reduction* of $(d_1, \ldots, d_n)$. The second process, while appearing more complicated when written down, is actually simpler because there is no need to put the entries in numerical order at the end: they are still in numerical order even once the subtractions have been made.

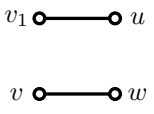
We define a map ρ from the set of reducible sequences to the set of permissible sequences by sending a reducible sequence α to its reduction. Note that ρ is not surjective. For example, if d is an even positive integer, then the permissible sequence (d) of length 1 is not in the image.

Proposition 1.7

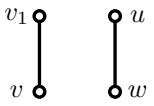
(i) If a permissible sequence $(d_1, \ldots, d_n)$ is graphic, and if $\Delta = d_1$, then there is a simple graph with this degree sequence that possesses a vertex v_1 of degree Δ whose neighbours have degrees $d_2, \ldots, d_{\Delta+1}$.

(ii) A reducible sequence is graphic if and only if its reduction is.

Proof. We begin with the first assertion. Let $(d_1, \ldots, d_n)$ be a graphic sequence, and take a simple graph G on vertices $v_1, \ldots, v_n$ with this degree sequence. Suppose that v_1 has degree $\Delta = d_1$, and let S be the set of neighbours of v_1. We compare the sequence $(d_2, \ldots, d_{\Delta+1})$ with the degree sequence $(e_2, \ldots, e_{\Delta+1})$ of the vertices in S. Necessarily, $e_i \leq d_i$ for all $i \in \{2, \ldots, \Delta + 1\}$, so if the degree sequence of the vertices in S is not $(d_2, \ldots, d_{\Delta+1})$, we must have $e_i < d_i$ for some i. In this case, there are $u \in S$ with degree e_i and $v \in V(G) \setminus S$ with degree d_i, so the fact that $e_i < d_i$ means there is a neighbour w of v that is not adjacent to u, so we may form a new simple graph by swapping the edges



with



This swap leaves all vertex degrees unchanged, so the new graph still has degree sequence $(d_1, \ldots, d_n)$. However, the sum of the degrees of the neighbours of v_1 has increased by $d_i - e_i \geq 1$. Therefore, because of the inequalities $e_i \leq d_i$, repeating this process will eventually yield a simple graph G' on vertices $v_1, \ldots, v_n$ such that the neighbours of v_1 in G' have degrees $d_2, \ldots, d_{\Delta+1}$, and of course G' has the same degree sequence as G, namely, $(d_1, \ldots, d_n)$.

Now we prove the second assertion. Let $\alpha = (d_1, \ldots, d_n)$ be reducible. If α is graphic, then we may take a graph such as that posited in the first assertion and number its vertices so that the neighbours of v_1 are $v_2, \ldots, v_{\Delta+1}$ and $d(v_i) = d_i$ for $i = 2, \ldots, \Delta + 1$. The simple graph obtained from this by removing v_1 has vertex degrees $d_2 - 1, \ldots, d_{\Delta+1} - 1, d_{\Delta+2}, \ldots, d_n$, and putting these in non-ascending order yields $\rho(\alpha)$, so $\rho(\alpha)$ is graphic.

Conversely, suppose that $\rho(\alpha)$ is graphic, and let G be a simple graph with degree sequence $\rho(\alpha)$. Consider the second process for obtaining the reduction of α, where no reordering is necessary. To undo that process, i.e., to return to α, one adds 1 back to each entry that was decreased during reduction and then inserts d_1 at the beginning. But if we numbered the vertices $v_1, \ldots, v_n$ of G such that $d(v_i) = d_i$, then the simple graph obtained from G by joining a new vertex of degree d_1 to the vertices v_i corresponding to the changed entries would have degree sequence α, so α is graphic. ∎

The Havel–Hakimi algorithm

We now describe the Havel–Hakimi algorithm. Take a permissible sequence α, and produce permissible sequences α_k by letting $\alpha_0 = \alpha$, letting $\alpha_k = \rho(\alpha_{k-1})$ if $k \geq 1$ and α_{k-1} is reducible, and stopping once you arrive at a sequence that is irreducible or that consists only of zeroes. Let α_t be the last sequence, i.e., irreducible or all zeroes. By Proposition 1.7, α is graphic if and only if α_t is graphic, if and only if α_t has no non-zero entries. (Every sequence comprising only zeroes is graphic, and if a sequence is irreducible and has at least one non-zero entry, it is not graphic.)

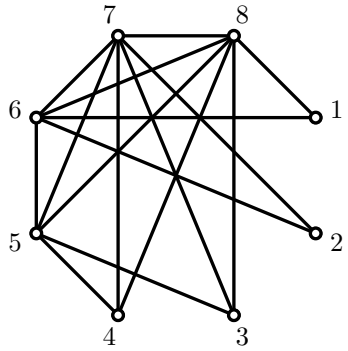
If α is graphic, a simple graph with degree sequence α can be found as follows. Draw n vertices, where n is the length of α. Begin with the last non-zero sequence, i.e., α_{t-1} in the notation above. Number its vertices right to left with $1, \ldots, l$ (where l is its number of entries). Join vertex l to each vertex corresponding to an entry in which α_t and α_{t-1} differ. Then move to α_{t-2}, of length $l+1$, and join vertex $l+1$ to each vertex corresponding to an entry in which α_{t-1} and α_{t-2} differ, and so on. The graph obtained after completing this process for the top-most sequence $\alpha_0 = \alpha$ will have the desired property.

Example 1.8 Decide whether the sequence $(6, 6, 5, 5, 3, 3, 2, 2)$ is graphic, and find a simple graph with this degree sequence if so.

Solution: Stage 1 of the algorithm proceeds as follows:

6	6	5	5	3	3	2	2
	5	4	4	2	2	2	1
		3	3	1	1	1	1
			2	1	1	0	0
				0	0	0	0

We have obtained a sequence of zeroes, so the original sequence is graphic, and Stage 2 yields this simple graph with the given degree sequence:



Example 1.9 Repeat for the sequence $(6, 6, 5, 4, 2, 1, 1, 1)$.

Solution:

6	6	5	4	2	1	1	1
	5	4	3	1	1	0	0

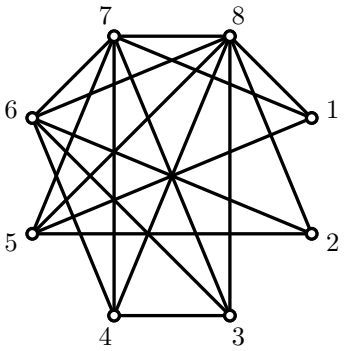
This non-zero sequence is irreducible, so the original sequence is not graphic.

Example 1.10 Repeat for the sequence $(7, 6, 5, 4, 4, 4, 3, 3)$.

Solution: Stage 1 of the algorithm proceeds as follows:

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

Having arrived at a sequence of zeroes, we see that the original sequence is graphic. Stage 2 yields this simple graph with the given degree sequence:



Example 1.11 One final example: Repeat for the sequence $(7, 7, 6, 6, 5, 4, 3, 2)$.

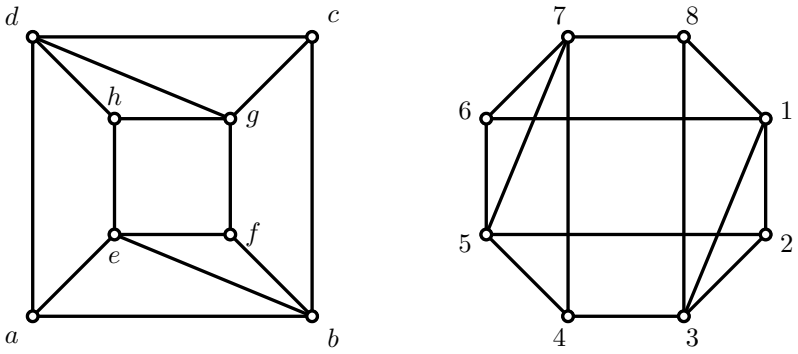
Solution:

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

The last sequence is a non-zero irreducible sequence, so the original sequence is not graphic.

1.7 Isomorphisms

Consider these two graphs:



G H

These graphs, while not being literally equal, can still be 'deformed' into one another in such a way that adjacency between vertices is preserved. For example, starting with H, move vertices $3, 4, 7, 8$ inwards until they are inside the rectangle with vertices $1, 2, 5, 6$, and then apply anticlockwise rotation by a quarter of a turn. The resulting graph will resemble G.

More formally, define a map $\theta : V(G) \to V(H)$ as follows:

$$\begin{aligned} a &\mapsto 6 \\ b &\mapsto 5 \\ c &\mapsto 2 \\ d &\mapsto 1 \\ e &\mapsto 7 \\ f &\mapsto 4 \\ g &\mapsto 3 \\ h &\mapsto 8 \end{aligned}$$

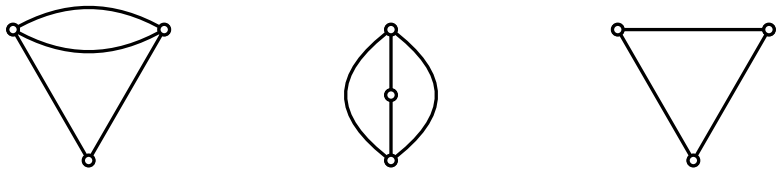
Having done that, we find that for any two vertices v and w of G, v is adjacent to w in G if and only if $\theta(v)$ is adjacent to $\theta(w)$ in H.

Formal definition of isomorphism

Let G and H be graphs. An *isomorphism* from G to H is a bijective map $\theta : V(G) \to V(H)$ such that for all $v, w \in V(G)$, the number of edges in G between v and w equals the number of edges in H between $\theta(v)$ and $\theta(w)$. If there is an isomorphism $G \to H$, then we say that G is *isomorphic* to H. In this situation, the inverse of the isomorphism is again an isomorphism, so that H is isomorphic to G, so we may say simply that G and H are isomorphic. In the situation where graphs G and H are isomorphic, we write $G \cong H$. Otherwise, we may write $G \ncong H$.

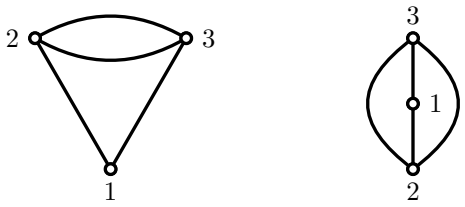
In practice, a convenient way to specify an isomorphism between two graphs is to label their vertices using the same set of labels so that vertices that correspond under the isomorphism have the same label.

Example 1.12 Consider the following graphs G_1, G_2, and G_3:



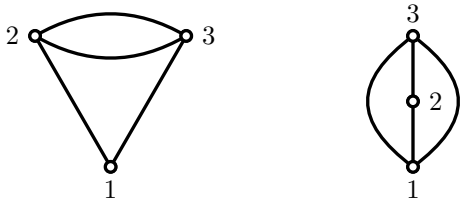
G_1 G_2 G_3

The graphs G_1 and G_2 are isomorphic, as we may see via the following vertex labellings:



G_1 G_2

The following, however, would not describe an isomorphism:



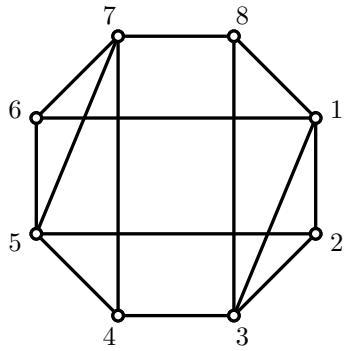
G_1 G_2

Indeed, the vertices with label 1 and 3 in G_1 are joined by only one edge, but in G_2 there are two edges joining them.

There is no isomorphism from G_1 to G_3, because in G_1 there are vertices of degree 3, whereas there are no such vertices in G_3. Because $G_2 \cong G_1$, it follows that $G_2 \ncong G_3$.

An *automorphism* is an isomorphism from a graph to itself.

Example 1.13 Consider again this graph:

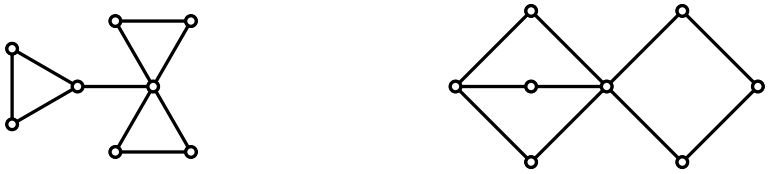


An automorphism of it is

$$\begin{aligned} 1 &\mapsto 5 \\ 2 &\mapsto 6 \\ 3 &\mapsto 7 \\ 4 &\mapsto 8 \\ 5 &\mapsto 1 \\ 6 &\mapsto 2 \\ 7 &\mapsto 3 \\ 8 &\mapsto 4 \end{aligned}$$

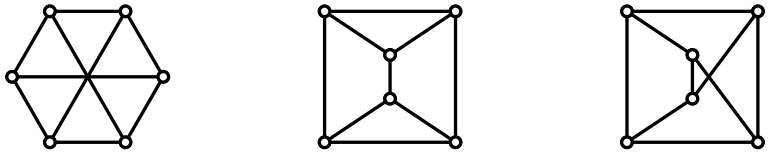
Graphs with the same degree sequence are not necessarily isomorphic.

Example 1.14 The two graphs below have the same degree sequence, namely, $(5, 3, 2, 2, 2, 2, 2, 2)$.



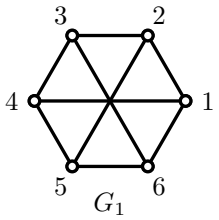
However, they are not isomorphic. Indeed, in the left-hand graph, the vertex of degree 3 is adjacent to the vertex of degree 5, but that is not the case in the right-hand graph.

Example 1.15 Decide which pairs of the following graphs are isomorphic. For each isomorphic pair, provide an isomorphism by labelling the vertices appropriately. For each non-isomorphic pair, explain why they are not isomorphic.

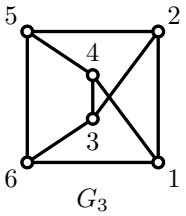


G_1 G_2 G_3

Solution: The graphs G_1 and G_3 are isomorphic:



G_1



G_3

However, G_2 is isomorphic to neither of the others, because G_2 contains copies of K_3 while G_1 and G_3 do not.

1.8 Subgraphs

A *subgraph* of a graph G is a graph H such that

(i) $V(H) \subseteq V(G)$,

(ii) $E(H) \subseteq E(G)$, and

(iii) $\psi_H(e) = \psi_G(e)$ for all $e \in E(H)$.

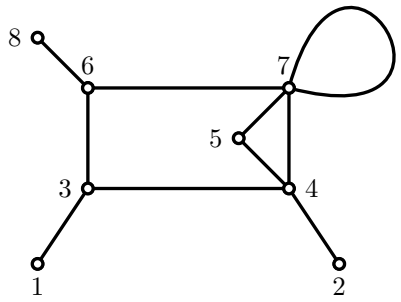
In practice, one obtains subgraphs of G by

(a) removing edges from G,

(b) removing vertices from G together with all the edges incident with those vertices, or

(c) some combination of the two.

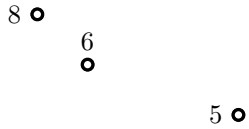
Example 1.16 Consider the following graph G:



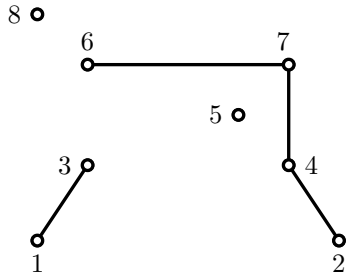
Here are some subgraphs of G:

(i) G itself

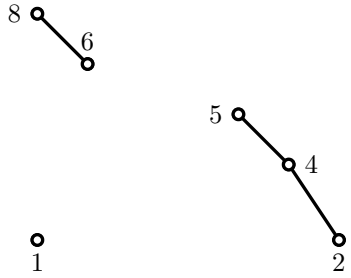
(ii) The null graph on any non-empty subset of $V(G)$, such as



(iii)



(iv)



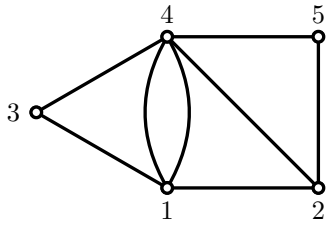
We have some common ways of specifying subgraphs:

- If G has more than one vertex and $v \in V(G)$, then $G - v$ is the graph obtained by removing from G the vertex v and all its incident edges.
- If S is a proper subset of $V(G)$, then $G - S$ is the graph obtained by removing from G all vertices in S and all their incident edges.
- If $e \in E(G)$, then $G - e$ is the graph obtained by removing from G the edge e.
- If $T \subseteq E(G)$, then $G - T$ is the graph obtained by removing from G all the edges in T.

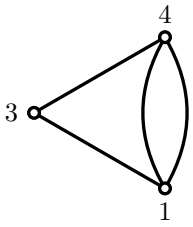
Induced subgraphs

If G is a graph and S a non-empty subset of $V(G)$, then the *induced subgraph* on S is the subgraph H of G whose vertices are those in S, such that all edges in G between vertices in S remain in H.

Example 1.17 Consider the following graph G:



The induced subgraph of G on $\{1, 3, 4\}$ is

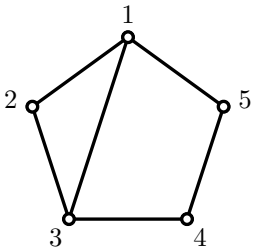


1.9 Graphs associated with simple graphs

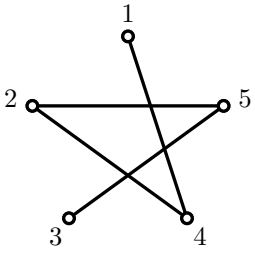
Complement of a simple graph

If G is a simple graph, then its *complement*, denoted $\overline{G}$, is the simple graph with the same vertex set as G but in which vertices v and w are adjacent in $\overline{G}$ if and only if they are *not* adjacent in G.

Example 1.18 If G is the graph



then $\overline{G}$ is



If G is a simple graph on n vertices and has m edges, then $\overline{G}$ has $\binom{n}{2} - m$ edges. Indeed, when the two edge sets are combined, the resuling graph is K_n, which has $\binom{n}{2}$ edges.

A simple graph is said to be *self-complementary* if it is isomorphic to its own complement. For example, the path graph P_4 on 4 vertices is self-complementary. Here is it with its complement, which is easily seen to be a path graph on 4 vertices again:

Proposition 1.19 *If a simple graph on n vertices is self-complementary, then 4 divides either n or $n-1$, i.e., n is congruent to 0 or 1 mod 4.*

Proof. Suppose that G is simple on n vertices and has m edges. Then $\overline{G}$ has $\binom{n}{2} - m$ edges, so if G is self-complementary,

$$\begin{aligned} \binom{n}{2} - m &= m, \\ \text{i.e.,} \quad \frac{1}{2}n(n-1) &= 2m, \\ \text{i.e.,} \quad n(n-1) &= 4m. \end{aligned}$$

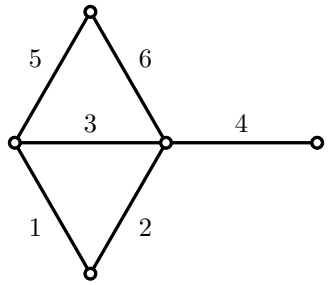
If n is odd, then 4 and n are coprime so 4 divides $n-1$, and if n is even, then 4 and $n-1$ are coprime so 4 divides n. ■

For example, there is no self-complementary simple graph on 11 vertices, beacause 4 divides neither 11 nor $11 - 1 = 10$.

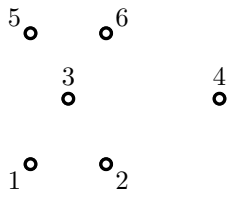
Line graph of a simple graph

Let G be a simple graph. Then the *line graph* of G, denoted $L(G)$, is the simple graph that has a vertex v_e for every edge e of G and in which v_e is adjacent to $v_{e'}$ if and only if e is adjacent to e' as edges in G.

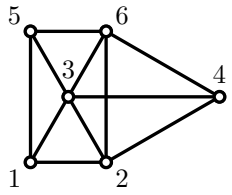
Example 1.20 Consider this simple graph G:



To find its line graph, let us first create a vertex for each edge of G:



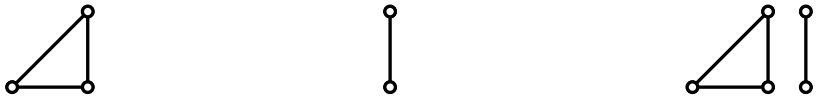
In G, edge 1 is adjacent to edge 2, for example, so in $L(G)$ vertex 1 is adjacent to vertex 2, and so on. Thus, we obtain $L(G)$:



2 Connectivity

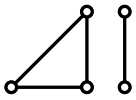
2.1 Unions and components

Let G and H be graphs. Their *union*, $G \cup H$, is the graph whose vertex set is the disjoint union of $V(G)$ and $V(H)$ and whose edges are exactly the edges in G and H.

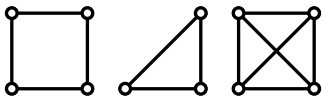


G $\quad$ H $\quad$ $G \cup H$

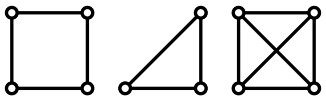
A graph is called *connected* if it is not the union of two graphs. Each of the graphs G and H in the example above is connected. The graph



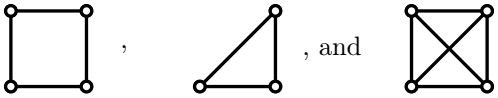
is not connected, nor is



Because every graph G is a union of one or more connected graphs, we may define a *component* of G to be any of the connected graphs of which it is a union. For example, the connected components of the graph



are



Theorem 2.1 *Let G be a simple graph with n vertices, m edges, and k components. Then*

$$n - k \leq m \leq \binom{n-k+1}{2}.$$

Proof. First we prove the lower bound on m, proceeding by induction on the number of edges. If G has no edges, then it is a null graph on n vertices and therefore has n components as well, so $n - k = 0 = m$. Now let G be a simple graph with $m \geq 1$ edges, and assume the bound is true for simple graphs with fewer than m edges. By removing edges if necessary, we obtain a new simple graph G' with $m' \leq m$ edges and with the property that removing any edge from G' creates an extra connected component. Thus we have this situation:

- G' has the same number of vertices as G.
- The number of edges in G' is less than or equal to the number of edges in G.
- G' has the same number of connected components as G, but removing any edge of G' increases the number of connected components by one.

Then if we remove any edge e of G', we create a simple graph with n vertices, $m' - 1$ edges, and $k + 1$ components, so by the inductive hypothesis,

$$m' - 1 \geq n - (k + 1) = n - k - 1,$$

i.e., $m' \geq n - k$. Since $m \geq m'$, we are done.

For the other inequality, we first add edges until every connected component is complete. We then choose a connected component C with the greatest number of vertices, s say. Let C' be another connected component, complete on $t \leq s$ vertices. If $t > 1$, we replace C and C' with K_{s+1} and K_{t-1}. The number of vertices remains the same, but the number of edges changes by

$$\binom{s+1}{2} - \binom{s}{2} + \binom{t-1}{2} - \binom{t}{2} = s - (t-1) = s - t + 1 > 0.$$

(Here, we have used case $l = 2$ of the general fact that $\binom{x+1}{l} - \binom{x}{l} = \binom{x}{l-1}$ for any $l \in \mathbb{Z}_{\geq 1}$; the $l = 2$ case gives $\binom{x+1}{2} - \binom{x}{2} = \binom{x}{1} = x$.) Continuing in this way, we arrive at a new graph that is a complete graph on $n - k + 1$ vertices together with $k-1$ isolated points, and it therefore has $\binom{n-k+1}{2}$ edges. However, by construction, it has at least as many edges as the original graph G. ∎

Corollary 2.2 *A simple graph on n vertices that has more than $\binom{n-1}{2}$ edges is necessarily connected.*

Proof. If G has $k \geq 2$ components, then

$$m \leq \binom{n-k+1}{2} \leq \binom{n-2+1}{2} = \binom{n-1}{2}.$$

■

2.2 Walks

Let G be a graph.A *walk* in G is a finite sequence $(v, e_1, e_2, \ldots, e_k)$ consisting of a vertex v and edges $e_1, \ldots, e_k$ (k may be 0), for which there exist $v_0, v_1, \ldots, v_k \in V(G)$ such that

- $v_0 = v$ and
- for every $i \in \{1, \ldots, k\}$, the set of ends of e_i is $\{v_{i-1}, v_i\}$.

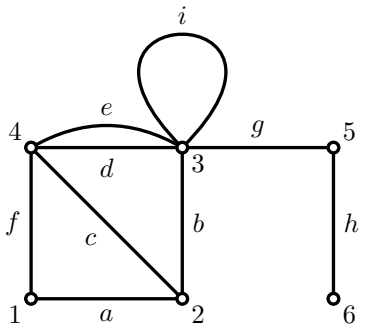
The *length* of a walk is the number k. We allow walks of length 0. Such a walk consists simply of a vertex.

The vertex v at the beginning of a walk is the *initial* vertex of the walk. The *final* vertex is the one called v_k above. Vertices $v_1, \ldots, v_{k-1}$ are called *internal*. Although not necessary, it can sometimes help the reader to include the internal vertices in the notation:

$$(v_0, e_1, v_1, e_2, \ldots, e_{k-1}, v_{k-1}, e_k, v_k).$$

If v and w are vertices, then a (v, w)-walk is a walk with initial vertex v and final vertex w.

Example 2.3 Consider the following graph on vertices $1, \ldots, 6$:



A $(1, 5)$-walk is

$$(1, a, c, d, e, d, i, g).$$

A *trail* is a walk in which all the edges are distinct, a *path* is a walk in which all vertices are distinct, *closed walk* is a walk whose initial and final vertices are equal, and a *cycle* is a closed walk whose internal vertices are distinct, that is, no vertices are repeated except for the initial vertex, which occurs once at the beginning and once at the end. In the preceding example,

- $(2, b, e, f)$ and $(4, d, i, b)$ are trails,
- $(5, g, e, c, a)$ is a path,
- $(1, a, c, c, a)$ and $(4, c, b, i, d)$ are closed walks, and
- $(2, c, d, b)$ is a cycle.

We say that a graph is *path connected* if for all vertices v and w, there is a path from v to w. We will also say that two given vertices are path connected in the graph if there is a path from one to the other.

Proposition 2.4 *A graph is connected if and only if it is path connected.*

Proof. Suppose that there is a path between any two vertices of G. Fix any vertex v_0, and let G_0 be its connected component. Now take any other vertex w. We show that w is in G_0, establishing that G has only one connected component. Choose a path P from v_0 to w, and let the edges in P be $e_1, \ldots, e_k$. We prove by induction on $i \geq 1$ that both ends of e_i are entirely in G_0. For this, we use the fact that an edge cannot connect two distinct connected components. Thus, since one end of e_1 is in G_0, so is the other. And if both ends of e_i are in G_0 for some $i = 1, \ldots, k-1$, one end of e_{i+1} is in G_0 and therefore so is the other. The induction is complete.

Conversely, suppose that there are vertices in G that have no path between them, i.e., that are not path connected. Since path connectedness is an equivalence relation on vertices, we may partition $V(G)$ into components under path connection. By assumption, there is more than one component. Let G_C be the induced subgraph of G relative to the vertices in the path-connected component C. We show that G is the union of the G_C. This amounts to showing that G_C and $G_{C'}$ have no edge between them when $C \neq C'$. But this is clear, since if there were, then a vertex in C would be adjacent to a vertex in C'. ■

2.3 Walks and the adjacency matrix

Proposition 2.5 *Let G be a graph on vertices $v_1, \ldots, v_n$, and let A be its adjacency matrix with respect to this numbering of the vertices. If $k \in \mathbb{Z}_{\geq 0}$, then the number of (v_i, v_j)-walks of length k is the (i, j)-entry of A^k.*

Proof. We prove this by induction on k. For $k = 0$, either $i = j$, in which case there is only one walk of length 0 from v_i to itself, or $i \neq j$, in which case there

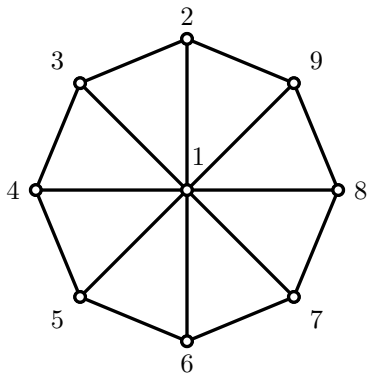
are no (v_i, v_j)-walks of length 0. Thus, the number is equal to the (i, j)-entry of the identity matrix, i.e., of A^0.

Now suppose the statement is true for some $k \geq 0$. Each (v_i, v_j)-walk of length $k+1$, if one exists, is obtained by choosing a (v_i, v)-walk of length k for some vertex v adjacent to v_j and then choosing an edge between v and v_j. Thus, writing $w_k(v)$ for the number of (v_i, v)-walks of length k, we see that

$$w_{k+1}(v_j) = \sum_{l=1}^{n} w_k(v_l) a_{l,j}. \tag{2.1}$$

(Remember that $a_{l,j} = 0$ for any v_l not adjacent to v_j.) However, by the inductive hypothesis, $w_k(v_l)$ is the (i, l)-entry of A^k, so the right-hand side of (2.1) is the (i, j)-entry of A^{k+1}. ■

Consider the graph



This graph, the wheel W_9, has adjacency matrix

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

The powers A^2, A^3, and A^4 are

$$A^2 = \begin{pmatrix} 8 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 1 & 2 & 1 & 1 & 1 & 2 & 1 \\ 2 & 1 & 3 & 1 & 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 1 & 3 & 1 & 2 & 1 & 1 & 1 \\ 2 & 1 & 2 & 1 & 3 & 1 & 2 & 1 & 1 \\ 2 & 1 & 1 & 2 & 1 & 3 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 & 2 & 1 & 3 & 1 & 2 \\ 2 & 2 & 1 & 1 & 1 & 2 & 1 & 3 & 1 \\ 2 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 3 \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 16 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 12 & 4 & 7 & 4 & 5 & 4 & 5 & 4 & 7 \\ 12 & 7 & 4 & 7 & 4 & 5 & 4 & 5 & 4 \\ 12 & 4 & 7 & 4 & 7 & 4 & 5 & 4 & 5 \\ 12 & 5 & 4 & 7 & 4 & 7 & 4 & 5 & 4 \\ 12 & 4 & 5 & 4 & 7 & 4 & 7 & 4 & 5 \\ 12 & 5 & 4 & 5 & 4 & 7 & 4 & 7 & 4 \\ 12 & 4 & 5 & 4 & 5 & 4 & 7 & 4 & 7 \\ 12 & 7 & 4 & 5 & 4 & 5 & 4 & 7 & 4 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 96 & 40 & 40 & 40 & 40 & 40 & 40 & 40 & 40 \\ 40 & 26 & 20 & 24 & 20 & 22 & 20 & 24 & 20 \\ 40 & 20 & 26 & 20 & 24 & 20 & 22 & 20 & 24 \\ 40 & 24 & 20 & 26 & 20 & 24 & 20 & 22 & 20 \\ 40 & 20 & 24 & 20 & 26 & 20 & 24 & 20 & 22 \\ 40 & 22 & 20 & 24 & 20 & 26 & 20 & 24 & 20 \\ 40 & 20 & 22 & 20 & 24 & 20 & 26 & 20 & 24 \\ 40 & 24 & 20 & 22 & 20 & 24 & 20 & 26 & 20 \\ 40 & 20 & 24 & 20 & 22 & 20 & 24 & 20 & 26 \end{pmatrix}$$

This last matrix, for example, tells us how many walks of length 4 there are between any two vertices. For example, there are 22 walks of length 4 from vertex 4 to vertex 8. Can you find them all?

Proposition 2.6 *Let G be a finite graph with no loops and A its adjacency matrix with respect to some numbering of the vertices.*

(i) The number of triangles in G is $\frac{1}{6}\operatorname{Tr}(A^3)$.

(ii) If, in addition, G has no multiple edges (so G is now simple), then the number of edges in G is $\frac{1}{2}\operatorname{Tr}(A^2)$.

Proof. (i) Every closed walk of length 3 is contained in a triangle (because G has no loops), and that triangle is unique of course, and also each triangle contains 6 closed walks of length 3. Therefore, there are 6 times as many closed walks of length 3 as there are triangles. But the number of closed walks of length 3

is the sum over all vertices v of the number of closed walks of length 3 starting and ending at v, which is the sum of the diagonal entries of A^3, i.e., $\mathrm{Tr}(A^3)$.

(ii) The proof is essentially the same, except this time, because G is simple, each closed walk of length 2 is contained in an edge, which is also necessarily unique, and each edge contains 2 closed walks of length 2. ■

2.4 Bipartite graphs and cycles

Proposition 2.7 *A graph is bipartite if and only if it contains no cycles of odd length.*

Proof. First suppose G is bipartite, with a partition being (A, B), say, and let $C = (v_0, e_1, v_1, e_2, \cdots, e_k, v_k)$ be a k-cycle (so $v_k = v_0$). Assume, without loss of generality, that $v_0 \in A$. Then v_i is in A if i is even and B if i is odd. Therefore, since $v_k = v_0 \in A$, k is even.

Conversely, suppose G contains no odd cycles. Since a graph is bipartite if and only if each of its connected components is, we may assume that G is connected. Choose any vertex v. Note that, for any vertex v', it is not possible to find simultaneously a (v, v')-path P of even length and a (v, v')-path Q of odd length, since if we could, then P followed by the reverse of Q would contain a cycle of odd length. Therefore, we may partition the vertices of G into the two disjoint sets

$$A = \{v' \mid \text{there is a } (v, v')\text{-path of even length}\},$$
$$B = \{v' \mid \text{there is a } (v, v')\text{-path of odd length}\}.$$

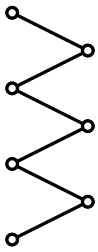
No two vertices in A are adjacent: If $v_i \in A$ and P_i is a (v, v_i)-path of even length $(i = 1, 2)$, then an edge e between v_1 and v_2 would yield a cycle of odd length: P_1 followed by e followed by the reverse of P_2 would contain an odd-length cycle.

Similarly, no two vertices in B are adjacent: If $v_i \in B$ and P_i is a (v, v_i)-path of odd length $(i = 1, 2)$, then an edge e between v_1 and v_2 would yield a cycle of odd length: P_1 followed by e followed by the reverse of P_2 would contain an odd-length cycle. ■

Example 2.8 Let B be the adjacency matrix of the path graph P_n on n vertices, where $n \geq 2$ (with respect to some numbering of the vertices). Is the following statement true or false?

For all $k \in \mathbb{Z}_{\geq 0}$, the matrix B^k has at least one zero entry.

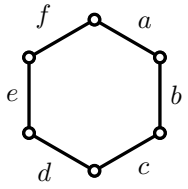
Solution: The statement is true. Note that a path graph is bipartite:



Now let $k \geq 0$. If k is even, choose vertices v and w in opposite parts of the vertex partition. Then every (v, w)-walk has odd length, so there are no (v, w)-walks of length k. If, instead, k is odd, choose v and w in the same part of the partition. In that case, every (v, w)-walk has even length, so again there are none of length k.

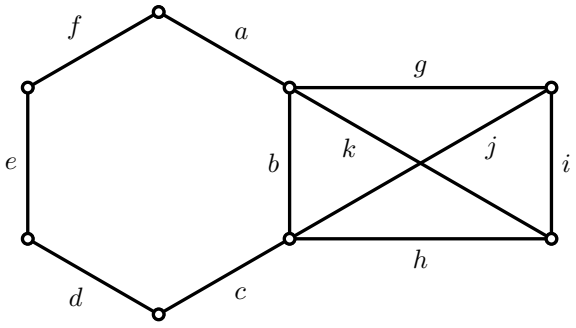
2.5 Edge-connectivity

Let G be a connected graph. A *disconnecting set* is a set of edges whose removal would disconnect G. For example, in the graph below, $\{a, b\}$ is a disconnecting set, as is $\{a, d\}$. (There are several others besides.)



A *cutset* is a minimal disconnecting set, meaning a disconnecting set that contains no other disconnecting set. In the previous example, $\{a, b, c\}$ is a disconnecting set, but it is not minimal, because it contains the disconnecting set $\{a, b\}$. On the other hand, $\{a, b\}$ is a cutset, because neither $\{a\}$ nor $\{b\}$ is a disconnecting set.

Example 2.9 Consider the following graph:

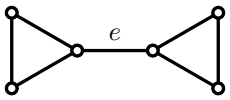


Here,

- $\{a, f\}$ is a cutset of size 2,
- $\{g, i, j\}$ is a cutset of size 3, and
- $\{g, h, j, k\}$ is a cutset of size 4.

There are no cutsets of size 1.

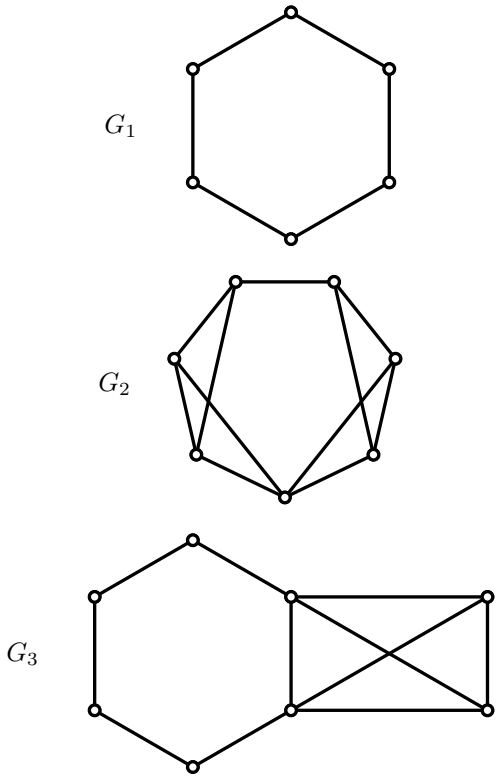
If a cutset consists of just one edge, then we call that edge a *bridge*, as in the edge e in this graph:



In fact, it is common to use the word *bridge* in a more general setting: If G is any graph, not necessarily connected, then a bridge is an edge whose removal would increase the number of connected components.

The *edge-connectivity* of a connected graph G is defined to be the size of a smallest cutset. It is denoted $\lambda(G)$. If $k \in \mathbb{Z}_{\geq 1}$, we say that G is *k-edge-connected* if $k \leq \lambda(G)$.

Example 2.10 Consider the graphs G_1, G_2, and G_3 below:



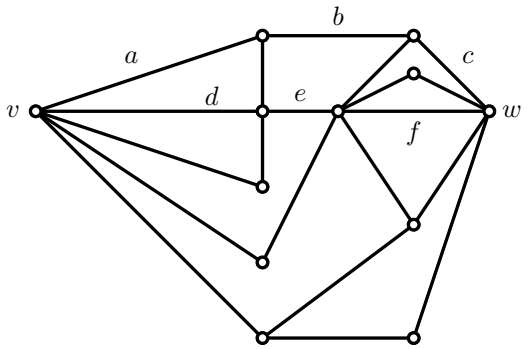
We see that $\lambda(G_1) = 2$, so G_1 is 1-edge-connected and 2-edge-connected. Next, $\lambda(G_2) = 3$ (what is a cutset of size 3 in G_2, and how do you know there are no smaller ones?), so G is k-edge-connected for all $k \in \{1, 2, 3\}$. Finally, $\lambda(G_3) = 2$ (see above), so G_3 is 1-edge-connected and 2-edge-connected.

If P and Q are paths in a given graph, we call them *edge-disjoint* is they have no edges in common.

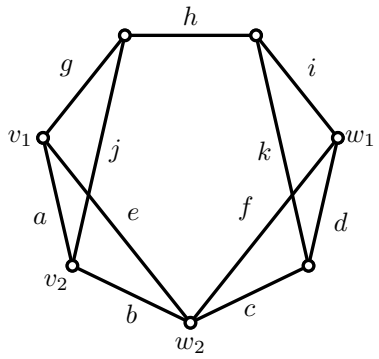
Theorem 2.11 (Menger) *Let G be a connected graph, and let $k \in \mathbb{Z}_{\geq 1}$. Then G is k-edge-connected if and only if for every pair of distinct vertices v and w in G, there is a set of k edge-disjoint paths from v to w.*

For a proof, see page 135 of Wilson.

Example 2.12 The following graph has edge-connectivity 2, and two edge-disjoint paths from v to w are (v, a, b, c) and (v, d, e, f):



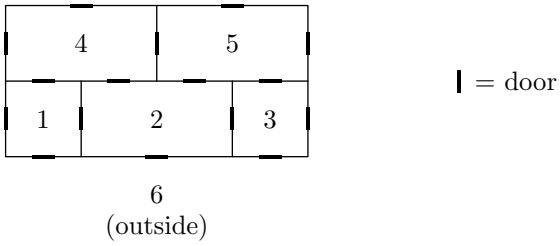
Example 2.13 Consider now this graph, which has edge-connectivity 3:



Three edge-disjoint paths from v_1 to w_1 are (v_1, a, b, c, d), (v_1, e, f), and (v_1, g, h, i), and three edge-disjoint paths from v_2 to w_2 are (v_2, b), (v_2, a, e), and (v_2, j, h, k, c).

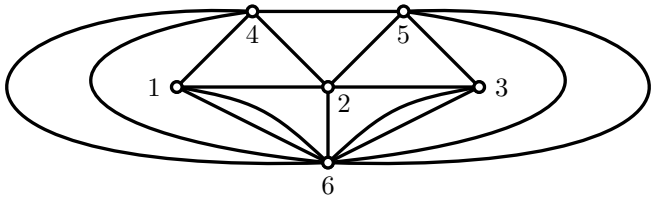
2.6 Euler circuits and Euler trails

Consider a building with five rooms as follows:



The rooms are numbered 1 to 5, and we give the outside the label 6. The various doors in the building are indicated. Is it possible to start somewhere, possibly outside, and walk through every door once and only once?

Let us model the situation with a graph in which the vertices represent rooms and the edges doors:



The problem has thus been translated into this question: Does the graph have a trail that includes every edge? (Recall that a trail is a walk that visits no edge more than once.)

We have some related terminology: If G, is a graph,

- an *Euler trail* in G is a trail that visits every edge of G, and
- an *Euler circuit* in G is a closed Euler trail (i.e., ends where it begins).

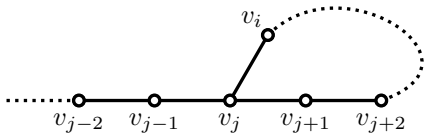
A connected graph is called *Eulerian* if it has an Euler circuit, and *semi-Eulerian* if it has an Euler trail but no Euler circuit.

Lemma 2.14 *A graph in which every vertex has degree at least* 2 *contains a cycle.*

Proof. If the graph, call it G, has a loop or multiple edges, we are done immediately, so we may assume that it is simple. Choose any vertex v_0. Then choose a vertex v_1 adjacent to v_0, and for each $i \geq 1$, choose a vertex v_{i+1} adjacent to v_i but not equal to v_{i-1}, possible because $d(v_i) \geq 2$. Because G has only finitely many vertices, there are $i, j \geq 0$ with $j < i$ such that $v_{i+1} = v_j$. If i is least such that v_{i+1} is equal to some previous vertex, then

$$v_j, v_{j+1}, \ldots, v_i, v_{i+1} = v_j$$

describes a cycle in G.



■

Theorem 2.15 *Let G be a connected graph.*

(i) G is Eulerian if and only if every vertex has even degree.

(ii) G is semi-Eulerian if and only if all but exactly two of its vertices have even degree.

Proof. We prove only (i), as (ii) is similar. Note that since removing or adding loops does not change whether a graph is Eulerian (or semi-Eulerian) and does not change the parity of any vertex in the graph, we may assume that G has no loops.

Suppose first that G is Eulerian, i.e., has an Euler circuit C. Each time C passes through a vertex, it enters via an edge not already encountered and then leaves via a different edge, again, not already encountered. Thus, each pass through a given vertex v counts two towards the degree of v. Since C passes through every edge, the count must eventually *equal* the degree, so the degree is even.

For the converse, we proceed by induction on the number of edges, the case of no edges being vacuous. Suppose that G has a positive number of edges and that every vertex has even degree. Because G is connected, every vertex has positive degree and therefore has degree at least 2, so by Lemma 2.14, G contains a cycle C. If every edge is in C, we are done. Otherwise, let H be the graph obtained by removing the edges of C, which leaves the parity of every vertex degree unchanged (so all vertices still have even degree). Consequently, each connected component of H contains an Euler circuit by induction and shares a vertex with C because G is connected. We can therefore form an Euler circuit in G by going around C, passing around each connected component of H as we reach it, so long as we have not already been around it. ■

We may now solve the puzzle of the five rooms easily. We wish to decide whether the graph in our model has an Euler trail. By the theorem, it has an Euler trail if and only if at most two vertices have odd degree. But the degree sequence is $(9, 5, 5, 5, 4, 4)$, so in fact four vertices have odd degree. Thus, the problem has a negative answer.

Theorem 2.16 (Fleury's Algorithm) *If G is an Eulerian graph, then the following algorithm produces an Euler circuit in G:*

(i) Start at any vertex v_0 and make it the initial vertex of a trail.

(ii) Whatever vertex v you are currently at, choose any edge incident with v that remains, choosing a bridge only if you are forced to (i.e., there are no non-bridges available). Then remove that edge and add it to the trail. If that was the last edge remaining on v, remove v as well.

(iii) Repeat step (ii) until no edges remain in the graph. The trail produced will be an Euler circuit for G.

Proof. Before delving into the proof proper, we point out that once we start removing edges in the algorithm and we are at some vertex v, both $d(v)$ and $d(v_0)$ are odd—unless $v = v_0$, in which case $d(v) = d(v_0)$ is even.

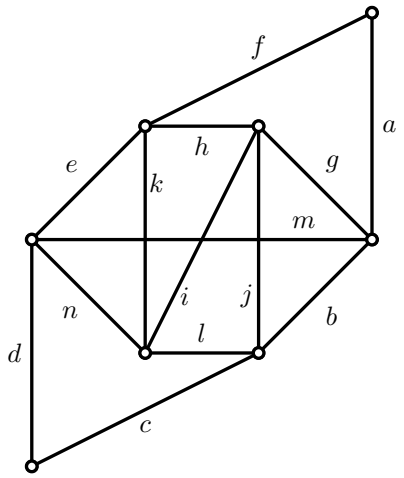
Now, we argue that Fleury's algorithm ensures the graph remains connected at every stage. The graph is connected by assumption at the start of the algorithm, so suppose that we have arrived at some vertex v and that the graph is still connected. We show that it remains connected after we have followed the rule in step (ii).

If v is equal to the initial vertex v_0 (either because we have returned momentarily to v_0 or because the algorithm is complete), then v must now have even degree, as remarked upon above. Hence, if there is any edge e still incident with v by this point, then e cannot be a bridge. Indeed, if it were, then because v has even degree in this case, removing e would yield two components each of which has a single vertex of odd degree, contradicting the handshaking lemma. Thus, any remaining edge incident with v must be a non-bridge, so removing it results in another connected graph. (And if there are no more edges incident with $v = v_0$, we have already found an Euler circuit.)

The other case is that $v \neq v_0$. Then v has odd degree by our remark at the beginning of the proof. Suppose that every edge incident with v is a bridge. If there were more than one bridge incident with v, then we could remove the edges incident with v, along with v itself, to yield a graph with connected components $C_1, \ldots, C_r$, none of which contains v but at least one of which, C_i say, does not contain v_0 either. Since the only odd-degree vertices in the graph before removing these edges are v and v_0, there can be only one vertex of odd degree in C_i (the vertex that had been adjacent to v). This again contradicts the handshaking lemma. Therefore, v has only one bridge incident with it, so removing this bridge along with v, as instructed in step (ii), results in another connected graph.

Hence, having shown that the graph remains connected throughout Fleury's algorithm, we deduce that a trail exists that includes every edge. Once we have used up all the edges, the vertex v that we finally arrive at has even degree (because it has degree 0), so we must have $v = v_0$ by the remark at the beginning of the proof. Thus, the trail is closed, as desired. ■

Example 2.17 Use Fleury's algorithm to find an Euler circuit in the following Eulerian graph:



Proposition 2.18 *Let G be a connected graph, not Eulerian, and let k be the number of vertices of odd degree. Then there is a set of $\frac{k}{2}$ edge-disjoint trails in G whose edge sets partition $E(G)$, and $\frac{k}{2}$ is the smallest possible. Further, if S is the set of vertices of odd degree in G (necessarily a set of even cardinality), the following algorithm produces such a set of trails:*

(i) Choose a vertex $v \in S$, and starting there carry out Fleury's algorithm (avoiding bridges whenever possible) until you meet another vertex in S.

(ii) Repeat step (i) until all vertices in S have been visited, at which point the remaining graph will have no vertices of odd degree.

(iii) If necessary (i.e., if edges remain), find an Euler circuit starting and ending at the last vertex of S arrived at and append it to the last trail.

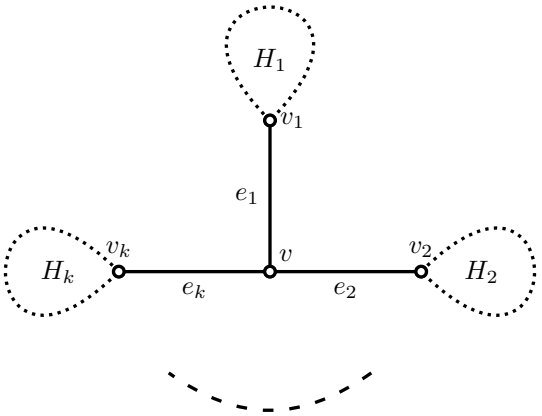
Proof. Pair the odd vertices into $k/2$ pairs (k is even by the handshaking lemma), and for each pair (v, w) add an edge between v and w. Call the new graph G'. Then G' has no vertices of odd degree, so it is Eulerian. Let C be an Euler circuit in G'. If E is the set of edges that we added to obtain G', then E has size $k/2$ and no two edges in E are incident with each other, so $C - E$ breaks up into $k/2$ trails. These trails are edge-disjoint, because C is an Euler circuit and therefore repeats no edges.

To see that $k/2$ is the least possible, observe that each trail in S can contribute only 2 to the number of vertices of odd degree in G, namely, the initial and final vertices of the trail. Therefore, if S has size l, then $k \leq 2l$, i.e., $l \geq k/2$.

Let us now prove that the algorithm works. Choose some $v_0 \in S$. We want to construct a trail from v_0 to some other vertex in S, removing edges as we go. The idea is to remain always in a component that contains a vertex in $S \setminus \{v_0\}$, and, if we take a bridge, to also leave behind either nothing or a component that contains a vertex in S (so that we may return to that component later and repeat the process there with a new v_0).

Remark: The vertices in $S \setminus \{v_0\}$ are precisely the ones that have odd degree when we are not at them and even degree when we are.

Now, suppose we are at some vertex v. If there is a non-bridge incident with it, take that edge. Then we remain in a component that contains a vertex in $S \setminus \{v_0\}$. If, instead, all edges $e_1, \ldots, e_k$ incident with v are bridges, choose any one, e_1 say, and let v_1 be the vertex at the other end of e_1. Let similarly v_i be the vertex at the other end of e_i for each i. We also declare H_i to be the graph at the other end of e_i, as illustrated here:



If $k = 1$, i.e., if e_1 is the only edge incident with v, then there is no component left behind when we take it. Otherwise, the component left behind will contain a vertex of S. Indeed, if v_2 has odd degree, then it is in $S \setminus \{v_0\}$ by the remark above, and if v_2 has even degree, then in H_2 alone (i.e., without e_2 incident with v_2) it would have odd degree, so H_2 must have another vertex of odd degree by the handshaking lemma, which would therefore be a vertex in $S \setminus \{v_0\}$.

We claim that the component H_1 that we arrive at after taking e_1 contains a vertex in $S \setminus \{v_0\}$. We consider two cases.

Case (i): v_1 has odd degree (before we move to it)

In this case, $v_1 \in S \setminus \{v_0\}$ by the remark above. Follow the edge e_1 and remove it.

Case (ii): v_1 has even degree (before we move to it)

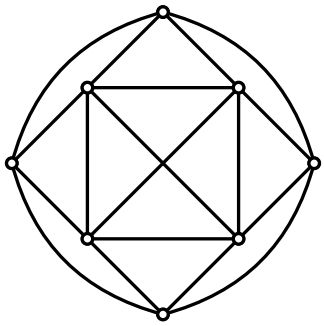
In this case, once we pass along e_1 and remove it, v_1 has odd degree, so the resulting component has another vertex of odd degree by the handshaking lemma. Being non-current, that other vertex of odd degree belongs to $S \setminus \{v_0\}$.

Finally, if any component we find ourselves in contains no vertices of odd degree (a situation that occurs in the subcase of Case (i) above where v_0 is in H_1 and v_0, v_1 are the only vertices of S in H_1), then we may find an Euler circuit

in it and append it to the trail just finished to obtain a new trail from v_0 to v_1 that exhausts all the edges in the component.

Of course, once we have exhausted a component, we return to some other component, taking one of its vertices in S, guaranteed to exist by the foregoing argument, to be the new v_0. ■

Example 2.19 In the graph below, use the algorithm of Proposition 2.18 to find two trails whose edge sets partition the edge set of the graph.

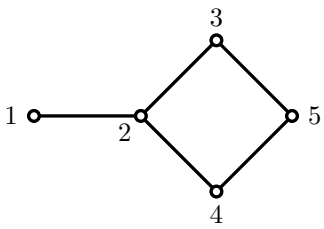


2.7 Vertex connectivity

Let G be a connected graph. We make the following definitions:

- A *separating set* is a proper subset of $V(G)$ whose removal disconnects G. (One may not remove all vertices from a graph.)
- A *cut-vertex* is a vertex whose removal disconnects G.
- A *minimal* separating set is one that contains no other separating sets.
- If G has at least one pair of distinct non-adjacent vertices, then the *connectivity* of G, also for clarity sometimes called the *vertex connectivity*, is the size of a smallest separating set and is denoted $\kappa(G)$. (If all pairs of distinct vertices are adjacent, then G has no separating sets.)

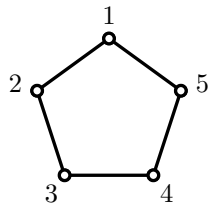
Example 2.20 In the graph below, each of $\{2\}$ and $\{3, 4\}$ is a minimal separating set.



While $\{2, 3, 4\}$ is a separating set, it is not minimal, because it contains the separating sets $\{2\}$ and $\{3, 4\}$.

Remark. Any set of edges that contains a disconnecting set is a disconnecting set, but it is not true that a set of vertices containing a separating set is itself necessarily a separating set. In the preceding example, $\{1, 2\}$ is a set of vertices containing the separating set $\{2\}$, but $\{1, 2\}$ is not a separating set.

In fact, a union of two separating sets, even minimal ones, need not be a separating set. For example, in the graph below, $\{2, 4\}$ and $\{3, 5\}$ are minimal separating sets, but their union, $\{2, 3, 4, 5\}$, is not a separating set:



Proposition 2.21 *If G is a connected graph, then $\lambda(G) \leq \delta(G)$ (the edge-connectivity is less than or equal to the smallest vertex degree). If, in addition, there is a pair of distinct non-adjacent vertices in G (so that $\kappa(G)$ is defined), then $\kappa(G) \leq \lambda(G)$, and therefore*

$$\kappa(G) \leq \lambda(G) \leq \delta(G).$$

Proof. Let v be a vertex of degree $\delta(G)$, and let S be the set of non-loops incident with v. Then S is a disconnecting set, so $\lambda(G) \leq \#S \leq d(v) \leq \delta(G)$.

For the other inequality, we may restrict attention henceforth to simple graphs. Indeed, removing loops and replacing multiple edges with single edges has no effect on the vertex connectivity and can only decrease the edge-connectivity if anything.

We prove by induction on $n \geq 3$ the assertion that $\kappa(G) \leq \lambda(G)$ for every non-complete simple graph on n vertices. If $n = 3$, then G is a path graph on three vertices. The central vertex is therefore a cut-vertex, so $\kappa(G) = 1 = \lambda(G)$.

Now let $n \geq 3$, assume the assertion for this n, and let G be a non-complete simple graph on $n + 1$ vertices. Choose a cutset C of G of cardinality $l = \lambda(G)$, and let H_1 and H_2 be the two components of $G - C$. Suppose first that some vertex v of G is incident with no edge in C, and assume without loss of generality that v is in H_1. If V is the set of vertices in H_1 incident with edges in C, then V is a separating set for G of cardinality at most $\#C = l$.

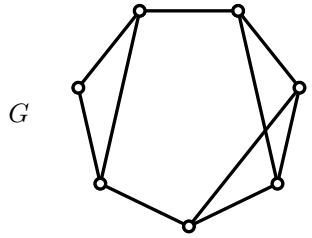
We may now assume, therefore, that every vertex of G is incident with at least one edge in C. It is easy to verify the inequality in the situation where the vertex sets of H_1 and H_2 both have cardinality at most two, so we may assume that one of them, say H_1, has at least three vertices. In that case, if $G - v$ was

complete for every $v \in V(H_1)$, then G itself would be complete, contrary to a hypothesis of the proposition, so there is a vertex v in H_1 such that $G-v$ is not complete. Because v is incident with some edge e in C, that edge is missing in $G-v$, so $C \setminus \{e\}$ is a disconnecting set for $G-v$. Thus, $\lambda(G-v) \leq \lambda(G)-1$. Of course, $\#V(G-v) = n$, so we may apply the inductive hypothesis to see that $\kappa(G-v) \leq \lambda(G-v)$. Hence, if T' is a separating set for $G-v$ of cardinality $\kappa(G-v)$, then $T = T' \cup \{v\}$ is a separating set for G of cardinality

$$\kappa(G-v)+1 \leq \lambda(G-v)+1 \leq \lambda(G),$$

so $\kappa(G) \leq \lambda(G)$. The induction is complete. ■

Example 2.22 Consider this graph:

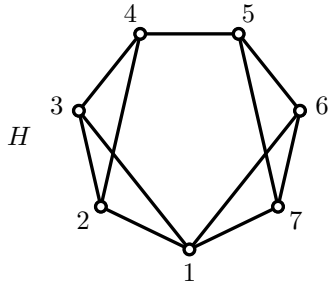


There is no cut vertex, so $\kappa(G) \geq 2$, but there is a vertex of degree 2, so $\delta(G) \leq 2$. Therefore,

$$2 \leq \kappa(G) \leq \lambda(G) \leq \delta(G) \leq 2,$$

so $\kappa(G) = \lambda(G) = \delta(G) = 2$.

Consider this time the following graph, whose vertices we label:

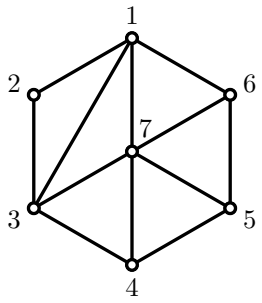


Again, there is no cut-vertex, so $\kappa(H) \geq 2$, but in fact $\{1,5\}$ is a separating set, so $\kappa(H) = 2$. Therefore $\lambda(H) \geq \kappa(H) = 2$. But we can do better: There is no disconnecting set of cardinality 3, because $H-e$ contains a cycle for every edge e, so $\lambda(H) \geq 3$. Because $\delta(H) = 3$, it follows that $\lambda(H) = 3$ as well.

2.8 Hamilton cycles

A *Hamilton cycle* in a connected graph G is a cycle that includes every vertex of G. A graph that contains a Hamilton cycle is said to be *Hamiltonian*. A graph that has no Hamilton cycle but does have a path containing every vertex is said to be *semi-Hamiltonian*.

Example 2.23 This graph is Hamiltonian, a Hamilton cycle being $(1, 2, 3, 4, 5, 6, 7, 1)$:



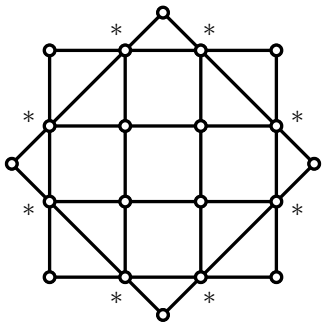
(It is permissible to describe any walk in a simple graph by giving a sequence of vertices.)

Proposition 2.24 *Let G be a graph and S a proper subset of $V(G)$. If G is Hamiltonian, then $G - S$ has at most $\#S$ components.*

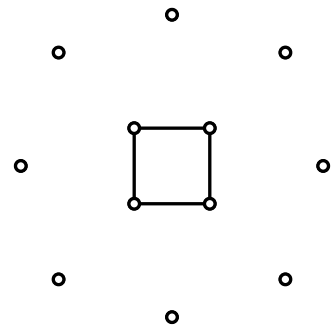
Proof. Suppose that C is a Hamilton cycle in G. Because G contains every edge in C, $G - S$ has at most as many components as $C - S$, and this in turn is at most k. ■

We may sometimes use Proposition 2.24 to show that a graph is not Hamiltonian.

Example 2.25 Consider this graph:



If S is the set of vertices marked $*$, then the number of components in $G - S$ is $9 > 8 = \#S$:



Therefore, by Proposition 2.24, the graph is not Hamiltonian.

Theorem 2.26 (Ore) *If G is a simple graph with $n \geq 3$ vertices, and if $d(v) + d(w) \geq n$ for all pairs v, w of distinct non-adjacent vertices, then G is Hamiltonian.*

Proof. Let G satisfy the assumptions in the theorem, and suppose that G is not Hamiltonian. Create a new simple graph G' from G by adding edges until the addition of one more edge would create a Hamiltonian graph. Thus, G' is not Hamiltonian. Note that G' still satisfies the assumptions of the theorem.

Since the addition of another edge in G' would create a Hamilton cycle, there must be a path $P = (v_1, v_2, \ldots, v_n)$ in G' that contains every vertex in G'. The vertices v_1 and v_n cannot be adjacent in G', for otherwise we could complete P to a Hamilton cycle. Therefore, $d(v_1) + d(v_n) \geq n$ by our assumption on G'.

We argue that, for some $i \in \{2, \ldots, n-1\}$, v_i is adjacent to v_1 and v_{i-1} is adjacent to v_n. Indeed, let S be the set of $i \in \{2, \ldots, n-1\}$ such that v_i is adjacent to v_1, so that $\{v_i \mid i \in S\}$ is the set of neighbours of v_1 and therefore has cardinality $d(v_1)$. If $\{v_{i-1} \mid i \in S\}$ contained no neighbour of v_n, then we would have at least $d(v_1)$ vertices in G' not adjacent to v_n (and none of them equal to v_n either). But, by definition, we have $d(v_n)$ vertices adjacent to v_n, so counting v_n as well, we find that G' would have at least $d(v_1) + d(v_n) + 1$ vertices. This is impossible, since $d(v_1) + d(v_n) \geq n$.

So, having found $i \in \{2, \ldots, n-1\}$ such that v_i is adjacent to v_1 and v_{i-1} is adjacent to v_n, we may construct a Hamilton cycle in G', namely, $(v_1, v_2, \ldots, v_{i-1}, v_n, v_{n-1}, \ldots, v_i, v_1)$. Such a cycle contradicts our assumption that G' is not Hamiltonian. ■

Note: The assumptions in the theorem are sufficient but not necessary. For example, if $n \geq 5$, C_n fails to meet the assumptions yet is nonetheless Hamiltonian.

Corollary 2.27 (Dirac's Theorem) *If G is a simple graph on $n \geq 3$ vertices and $d(v) \geq \frac{n}{2}$ for every vertex v, then G is Hamiltonian.*

Proof. This follows immediately from the theorem. ■

Corollary 2.28 *Let G be a finite simple graph on n vertices, and assume it has at least $\binom{n-1}{2} + 2$ edges. Then G is Hamiltonian.*

Proof. To apply Ore's Theorem, we show that for *any* two distinct vertices v, w (whether adjacent or not), $d(v) + d(w) \geq n$. Let

$$E = \{e \in E(G) \mid e \text{ is incident with } v \text{ or } w\},$$
$$E' = E(G) \setminus E.$$

Then

$$d(v) + d(w) = \begin{cases} \#E + 1 & \text{if } v \text{ and } w \text{ are adjacent,} \\ \#E & \text{otherwise.} \end{cases}$$

Either way, $d(v) + d(w) \geq \#E$. We will be done, therefore, if we can show that $\#E \geq n$.

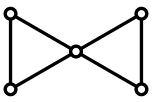
Now, the number of vertices incident with an edge in E' is at most $n-2$, so because G is simple, $\#E' \leq \binom{n-2}{2}$, and so

$$\begin{aligned} \#E &= \#E(G) - \#E' \\ &\geq \binom{n-1}{2} + 2 - \binom{n-2}{2} \\ &= \frac{1}{2}(n-1)(n-2) + 2 - \frac{1}{2}(n-2)(n-3) \\ &= n. \end{aligned}$$

■

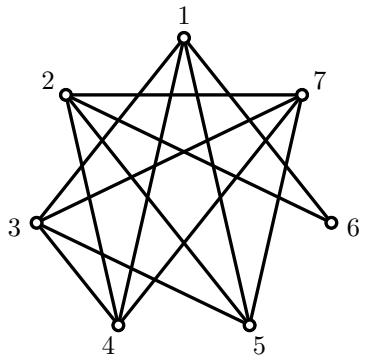
Corollary 2.28 is best possible in the sense that, for each n, there is a finite simple graph G on n vertices such that G has $\binom{n-1}{2} + 1$ edges but is not Hamiltonian. Indeed, we can just take K_{n-1} and add a new vertex, which we join with one edge to some vertex in K_{n-1}.

Remark. An Eulerian graph need not be Hamiltonian:



Also, a Hamiltonian graph need not be Eulerian:

Example 2.29 Decide whether this graph is Hamiltonian:



Solution: It is, a Hamilton cycle being $(1, 6, 2, 4, 7, 5, 3, 1)$. Note that it is not possible to justify that the graph is Hamiltonian using Ore's Theorem or the corollaries given, because the hypotheses are not met. For example, vertices 6 and 7 are distinct non-adjacent vertices, but the sum of their vertex degrees is $2 + 4 = 6 < 7 = \#V(G)$.

2.9 Paths of least weight: Dijkstra's algorithm

A *weighted graph* is a graph G together with a function $E(G) \to \mathbb{R}$ (thus assigning a number to each edge). Typically, weights are positive, and we will make that asssumption here. The weight of a path in a weighted graph is defined to be the sum of the weights of its edges.

Here is a typical problem related to weighted graphs:

> Given a weighted graph and a choice of two vertices v and w, find a path of least weight from v to w.

For example, if a weighted graph represents a road network and its weights represent distances between junctions, then the problem above corresponds to finding a shortest route between two points. In fact, below, we will think in terms of distances rather than weights for this very reason.

Let G be a simple, connected, weighted graph, and let v be a vertex in G. We describe Dijkstra's algorithm to construct a path of least length from v to any other vertex and then prove that the algorithm succeeds.

The algorithm

We distinguish between *complete* vertices and *incomplete* ones. To begin with, only v is complete. Each complete vertex will have a permanent distance, which for v is 0, and a permanent path, which for v is the trivial path starting and ending at v. All vertices are incomplete until they become complete. We also use the notion of the *temporary distance* of an incomplete vertex. All incomplete vertices start with an infinite temporary distance. An *active* vertex is an incomplete vertex that has a finite temporary distance. Each active vertex has a temporary path (to be defined in the algorithm).

We now proceed as follows: At each stage, look for all incomplete vertices adjacent to the most recent complete vertex v'. If, for any incomplete neighbour w' of v', the permanent distance of v' plus the weight of the edge $v'w'$ is less than the current temporary distance of w', then we replace the temporary distance of w' with that number, and we set the temporary path of w' to be the permanent path of v' followed by the edge $v'w'$. Among all active vertices (not just those adjacent to v'), we look for one with smallest temporary distance. We then make that vertex complete, define its permanent distance to be its last temporary distance, and define its permanent path to be its last temporary path. We then let this vertex be the current one and repeat. The algorithm ends once all vertices are complete.

Remark. In practice, one does not need to keep track of the temporary and permanent paths, only the temporary and permanent distances. Once all permanent distances have been found, shortest paths can be found by backtracking, as follows. To find a path of shortest length from the starting vertex v to any given vertex w, start at w and find a neighbour w_1 of w such that the permanent distance to u (from v) plus the weight of the edge w_1w is equal to the permanent distance to w. Then do the same for w_1 to obtain a suitable neighbour w_2, and so on until you reach v. If w_k is the last one before v, then $vw_kw_{k-1}\cdots w_2w_1w$ is a path of shortest length from v to w.

Theorem 2.30 *In the algorithm described above, the permanent path of a complete vertex w is a path of least length from v to w, and the length of the permanent path is the permanent distance of w. Every vertex in the graph eventually becomes complete.*

Proof. We proceed by induction on the number of complete vertices. The statements about the permanent path and permanent distance are clearly true when v is the only complete vertex.

Now suppose we have established $n \geq 1$ complete vertices. Update all of the temporary distances and temporary paths of the neighbours of the last complete vertex, as described in the algorithm. Let w be an active vertex of least temporary distance, and let v_w be the last complete vertex in the temporary path P_w of w. Let P be any path from v to w, and let w' be the first incomplete vertex that P hits. Then by the point where P reaches w', its length is already at least the temporary distance of w' (this claim, which is key, will be justified at the end of the proof). This distance is at least equal to the temporary distance of w, by the very choice of w, and that is the permanent distance of v_w plus the weight of the edge v_ww. This last sum is simply the length of P_w, so P is at least as long P_w, and therefore P_w is a path of least length from v to w. Symbolically,

$$l(P) \geq t(w') \geq t(w) = p(v_w) + \mathrm{wt}(v_ww) = l(P_w),$$

where l (length), t (temporary distance), and wt (weight) have the obvious meanings. By definition, its length is the temporary distance of w. Since w is

now made complete, we thus have that P_w is the permanent path of w and has length equal to the permanent distance of w.

That all vertices eventually become complete follows immediately from the algorithm, in light of the connectedness of G.

It remains to justify the claim made above, namely, that by the point where P reaches w', its length is already at least the temporary distance of w'. Let v' be the complete vertex immediately before w' in P. (Remember how w' was defined in P.) Also, let $v_{w'}$ be the last complete vertex in the temporary path to w' (which may not necessarily follow P).

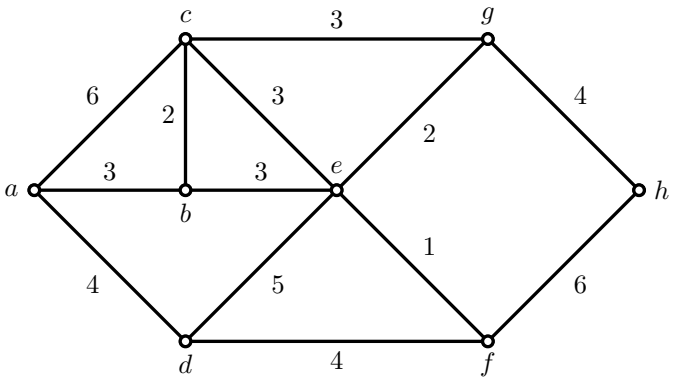
Case (i): $v' = v_{w'}$

In this case, if P follows exactly the permanent path to v', then its length up to w' will be precisely the temporary distance of w' by definition. If P takes any other route to v', then its length up to that point must be at least the length of the permanent path of v', since the permanent path is shortest to v'.

Case (ii): $v' \neq v_{w'}$

Observe that the current temporary path to w' can only ever be an improvement (if anything) on any previous temporary paths via other complete vertices. Therefore, if $v' \neq v_{w'}$, then the length of the path along P from v up to w' via v' is at least the length of the temporary path to w', which is equal to the temporary length of w'. This completes the claim. ■

Example 2.31 Use Dijkstra's algorithm to (a) calculate the length of a shortest path from vertex a to vertex h in the graph below, and (b) find such a path.



Solution: (1)

b	$3*$
c	6
d	4

(2)

c	$3+2=5$
d	$4*$
e	$3+3=6$

(3)

c	$5*$
e	6
f	$4+4=8$

(4)

e	$6*$
f	8
g	$5+3=8$

(5)

f	$6+1=7*$
g	8

(6)

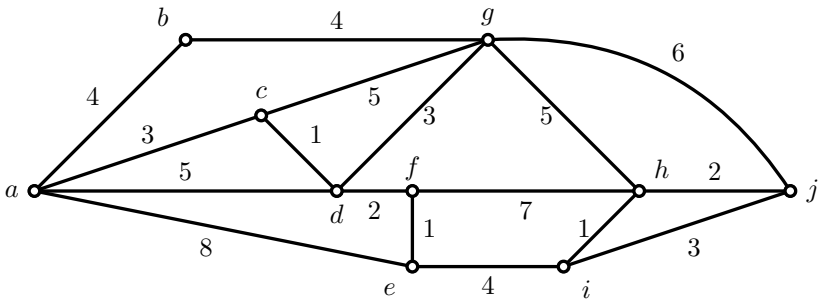
g	$8*$
h	$7+6=13$

(7)

h	$8+4=12*$

Thus, the length of a shortest path from a to h is 12, and such a path, found by backtracking, is (a, b, e, g, h). Another is (a, b, c, g, h).

Example 2.32 Use Dijkstra's algorithm to (a) calculate the length of a shortest path from vertex a to vertex j in the graph below, and (b) find such a path.



Solution: (1)

b	4
c	$3*$
d	5
e	8

(2)

b	$4*$
d	$3+1=4$
e	8
g	$3+5=8$

(3)

d	$4*$
e	8
g	8

(4)

e	8
f	$4+2=6*$
g	$4+3=7$

(5)

e	$6+1=7*$
g	7
h	$6+7=13$

(6)

g	$7*$
h	13
i	$7+4=11$

(7)

h	$7+5=12$
i	$11*$
j	$7+6=13$

(8)

h	$12*$
j	13

(9)

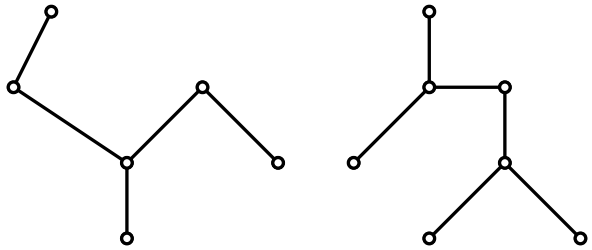
j	$13*$

Thus, the length of a shortest path from a to j is 13, and such a path, found by backtracking, is (a, c, d, g, j). This is the unique shortest path in this case.

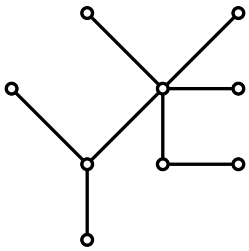
3 Trees

3.1 Forests, trees, and first properties

A *forest* is a graph that has no cycles, and a *tree* is a connected forest. Here is a forest:



And here is a tree (which is also a forest):



Observe that every edge in a forest is a bridge. Indeed, if T is a tree and e were an edge such that $T - e$ is still connected, then a path in $T - e$ joining the ends of e could be combined with e to create a cycle in T, contrary to the definition of a tree, and this argument extends to forests.

Theorem 3.1 *Let G be a graph with n vertices. The following are equivalent:*

(i) *G is a tree.*

(ii) *G contains no cycles and has $n - 1$ edges.*

(iii) *G is connected and has $n - 1$ edges.*

(iv) *G is connected, and each edge is a bridge.*

(v) *Any two vertices in G are connected by exactly one path.*

(vi) *G contains no cycles, but the addition of any edge creates a cycle. (Note: In this case, the cycle created is the only cycle in the resulting graph.)*

Proof. (i) $\Rightarrow$ (ii): We show by induction on $m \geq 0$ that a tree with m edges has $m + 1$ vertices. The case $m = 0$ is immediate because a tree is connected. Now let $m \geq 1$, and assume the assertion for all smaller values of m. If T is a tree

with m edges, then choosing an edge e, we may consider $T - e$, a union of two trees T_1 and T_2. By the inductive hypothesis, $n(T_i) = m(T_i) + 1$ for each i, so

$$n(T) = n(T_1) + n(T_2) = m(T_1) + m(T_2) + 2 = m(T) + 1,$$

because T has one more edge than T_1 and T_2 combined.

(ii) $\Rightarrow$ (iii): Let G be a cycle-free graph such that $m(G) = n(G) - 1$. Then G is a union of trees, say $G = T_1 \cup \cdots \cup T_k$, and we already know that the number of vertices in a tree is one more than its number of edges, so

$$\begin{aligned} m(G) + 1 &= n(G) \\ &= \sum_{i=1}^{k} n(T_i) \\ &= \sum_{i=1}^{k} (m(T_i) + 1) \\ &= \sum_{i=1}^{k} m(T_i) + k \\ &= m(G) + k, \end{aligned}$$

so $k = 1$.

(iii) $\Rightarrow$ (iv): If G is a connected graph such that $m(G) = n(G) - 1$, form a graph from G as follows: If G has a cycle, remove an edge from the cycle to obtain a connected graph with one fewer edge, and repeat until no cycles remain. Call this graph G'. By construction, G' is a connected cycle-free graph, i.e., a tree, so $m(G') = n(G') - 1$ by the above. Because G and G' have the same number of vertices,

$$m(G') = n(G) - 1 = m(G),$$

so they also have the same number of edges. Because of how G' was constructed, this means that G and G' are equal, so G is a tree, and so every edge in G is a bridge.

(iv) $\Rightarrow$ (v): Two paths would create a cycle somewhere and therefore an edge that is not a bridge.

(v) $\Rightarrow$ (vi): A cycle would imply the existence of two vertices connected by two different paths. If a new edge is added, then since every pair of vertices is already connected, the new edge would create a cycle. Only one cycle can be created by the addition of an edge: In any graph, if there are two distinct cycles containing a common edge, then there is a cycle in the graph that does not contain that edge.

(vi) $\Rightarrow$ (i): If G had more than one component, then adding an edge between one component and another would not create a cycle. ■

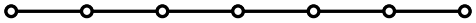
An end-vertex in a graph is a vertex of degree 1. A tree T has at least two end-vertices: If V is the set of end-vertices in T and $V' = V(T) \setminus V$, then by the

handshaking lemma,

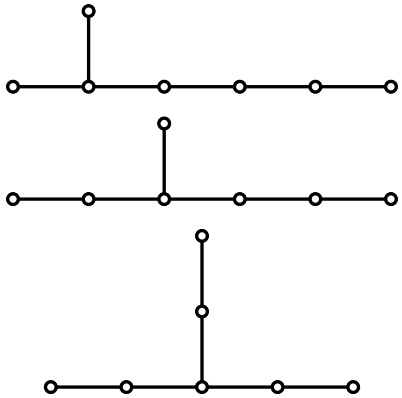
$$\begin{aligned} 2n-2 &= \#V + \sum_{v \in V'} d(v) \quad \text{where } n \text{ is the number of vertices in } T \\ &\geq \#V + 2(n - \#V) \\ &= 2n - \#V, \end{aligned}$$

i.e., $\#V \geq 2$.

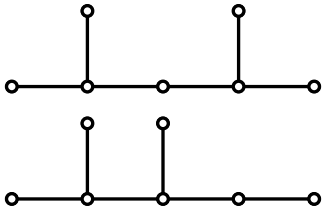
By way of exercise, let us find all trees on 7 vertices. One way to do this is to categorize them according to vertex degrees. Let Δ denote the greatest vertex degree, as usual. First, if $\Delta = 2$, we have only



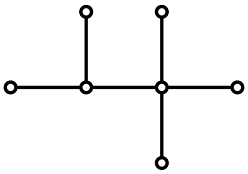
Next, we consider the situation where $\Delta = 3$ and there is only one vertex of degree 3:



If $\Delta = 3$ and two vertices (but no more) have degree 3, then we have these possibilities:



We move up now to $\Delta = 4$. Only one vertex may have degree 4, and the only possibility where the other has degree 3 is



If, instead, there is a vertex of degree 4 and the rest have degree 2 or less, we have these possibilities:

The remaining situations are $\Delta = 5$ and $\Delta = 6$:

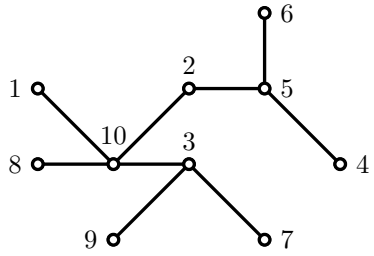
Thus, there are 11 trees on 7 vertices.

Proposition 3.2 *If G is a forest with n vertices, m edges, and k components, then $m = n - k$.*

Proof. By joining them with $k-1$ edges, string the k components of G together to form a tree. This tree has n vertices and $m+k-1$ edges, so by Theorem 3.1, $m + k - 1 = n - 1$, i.e., $n - k = m$. ■

3.2 Labelled trees

A *labelled tree* is a tree whose vertices are given distinct labels. Typically, if the tree has n vertices, then the labels will be $1, \ldots, n$. For example, here is a labelled tree on 10 vertices in which the labels are $1, \ldots, 10$:



Prüfer sequences

Every labelled tree on $n \geq 2$ vertices with labels $1, \ldots, n$ can be encoded into a sequence of $n - 2$ positive integers. Specifically, the sequence is of the form

$(a_1, a_2, \ldots, a_{n-2})$ where $a_i \in \{1, \ldots, n\}$ for each i. (Repetitions are allowed.) This sequence is called the *Prüfer sequence* of the labelled tree. No two labelled trees have the same Prüfer sequence, and every sequence of $n-2$ integers in $\{1, \ldots, n\}$ is the Prüfer sequence of some labelled tree.

A consequence of this correspondence between labelled trees and sequences is that there are n^{n-2} labelled trees on n vertices, because this is how many Prüfer sequences there are.

Remark. Two trees that are different as labelled trees may nonetheless be isomorphic simply as trees:

Method to obtain the Prüfer sequence of a labelled tree

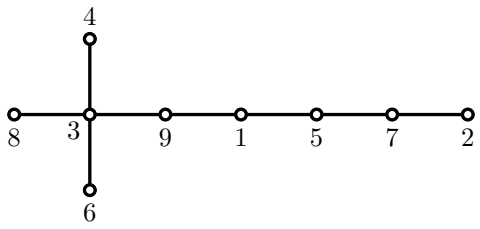
Input: A tree T on $n \geq 2$ vertices, labelled $1, \ldots, n$
Initial objects: An empty sequence S
Output: The Prüfer sequence of T
Algorithm:

(1) Find the end-vertex with smallest label, i, and let j be its neighbour.

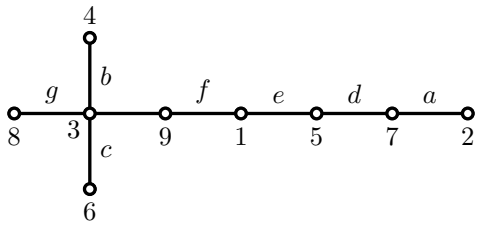
(2) Add the neighbour j to the right end of S and remove i from the tree, striking out the edge incident with i as you do so.

(3) Repeat steps (1) and (2) until only two vertices remain. Do not add these last two to the sequence. You are already done.

Example 3.3 Find the Prüfer sequence of the following labelled tree:



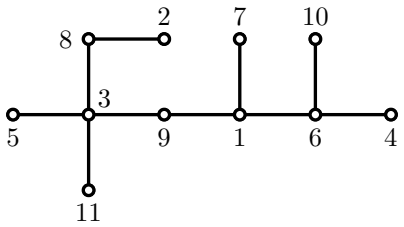
Solution: Let us indicate with letters $a, b, c, \ldots$ the order in which we remove edges:



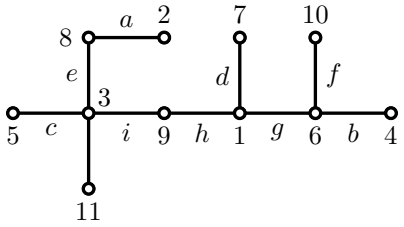
The process results in the sequence $(7, 3, 3, 5, 1, 9, 3)$.

You do not need to label the edges as you remove them. It would be enough simply to strike them out as you go. We labelled them above (and will label them in the next example) purely for illustrative purposes.

Example 3.4 Repeat the previous exercise with the following labelled tree:



Solution:



We arrive this way at the Prüfer sequence $(8, 6, 3, 1, 3, 6, 1, 9, 3)$.

Method to reconstruct a labelled tree from a Prüfer sequence

Input: An integer $n \geq 2$ and sequence S of $n-2$ integers in $\{1, \ldots, n\}$.
Initial objects: The list of numbers $1, \ldots, n$, called 'the List'
Output: The labelled tree whose Prüfer sequence is S
Algorithm:

(1) Find the least i in the List that is not in the sequence.

(2) Join i to the first (i.e., left-most) entry j in the sequence, and cross off j from the front of the sequence and i from the List.

(3) Repeat steps (1) and (2) until the sequence is empty. This will leave 2 entries in the List. Join them.

Exercise 3.5 Find the labelled tree with Prüfer sequence $(7, 3, 7, 5, 2, 1)$.

Exercise 3.6 Find the labelled tree with Prüfer sequence $(8, 6, 3, 1, 3, 6, 1, 9, 3)$. (Actually, we already know the corresponding tree because of Example 3.4, but reproduce that tree from the given sequence using the algorithm just provided.)

Remark. The number of times a label i appears in the Prüfer sequence of a labelled tree is $d(i) - 1$, where $d(i)$ is the degree of the vertex with label i. This is because in the encoding algorithm (tree to sequence), we write 'i' in the sequence each time we remove one of its edges, up to the point where i becomes an end-vertex itself, thus $d(i) - 1$ times.

For example, the number 7 appears 4 times in the sequence $(2, 1, 7, 7, 6, 4, 7, 9, 7)$, so the vertex with that label has degree $4 + 1 = 5$ in the corresponding labelled tree.

Example 3.7 Let $n \geq 4$. Find the number of labelled trees on n vertices in which there are exactly 3 end-vertices.

Solution: End-vertices are the vertices whose labels appear $1 - 1 = 0$ times in the Prüfer sequence, so we want to count Prüfer sequences of length $n - 2$ in which exactly 3 labels are absent. There are

- $\binom{n}{3}$ ways to choose the three labels that do not appear.

Because the Prüfer sequence has $n-2$ entries and only $n-3$ labels are available to be placed in it, and because each of these $n - 3$ appears at least once (these numbers correspond to vertices of degree at least 2), it follows that one label appears twice in the sequence and all the others once each. There are

- $n - 3$ ways to choose the label that appears twice, and
- $\binom{n-2}{2}$ ways to choose the two slots in which to place the repeated label.

This leaves $(n - 2) - 2 = n - 4$ slots in the sequence in which to place the remaining $n - 4$ labels, and there are therefore

- $(n - 4)!$ ways to assign the remaining $n - 4$ labels to slots.

Therefore, the number of labelled trees of the desired kind is

$$\binom{n}{3}(n-3)\binom{n-2}{2}(n-4)! = \frac{1}{12}n!(n-2)(n-3).$$

3.3 Spanning trees

Let G be a connected graph. A *spanning tree* in G is a subgraph that is a tree containing every vertex of G. Here are two options for finding a spanning tree:

(i) Find a cycle (if one exists), remove an edge from it, and repeat until there are no more cycles.

(ii) Start with only the vertices of G, and add the edges of G back in one by one until you cannot add any more without creating a cycle.

Example 3.8 Here is a connected graph with a spanning tree indicated on the right, the edges of the spanning tree being the solid (not dotted) ones:

Remark. If G is not connected, we may still find a spanning *forest*, consisting of a spanning tree in each component.

If G is a connected graph and T a spanning tree, the *complement* of T in G is the subgraph of G obtained by removing from G the edges of T. That is, the complement is $G - E(T)$.

Proposition 3.9 *Let G be a connected graph and T a spanning tree.*

(i) Each cutset of G has an edge in common with T.

(ii) Each cycle in G has an edge in common with the complement of T.

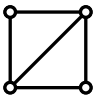
Proof. (i) If E is a set of edges of G contained in the complement of T, then $G - E$ still includes all edges in T and is therefore connected.
(ii) This is immediate, because T cannot contain a cycle. ■

Fundamental sets of cycles

Let G be a connected graph with n vertices and m edges, and suppose that T is a spanning tree for G. Observe that T has $n - 1$ edges. Each edge e in the complement of T defines a cycle in G: adding e creates a unique cycle in the subgraph $T + e$. The set of cycles obtained this way as we take all $m - (n - 1) = m - n + 1$ edges in the complement of T is called the *fundamental set of cycles* of G with respect to T.

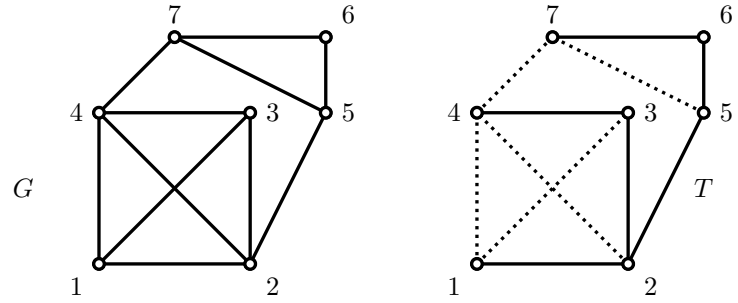
Remark. Strictly speaking, what we are counting when looking at fundamental sets of cycles are not cycles as such but what one might call *cycle sets*. By a

cycle set we will mean the set of edges in some cycle of the graph. The number of cycle sets is fewer than the number of cycles (unless all cycles are loops), because if C is a cycle of length n and E is its set of edges, then there are, in total, $2n$ cycles having E as their sets of cycles. For example, the graph

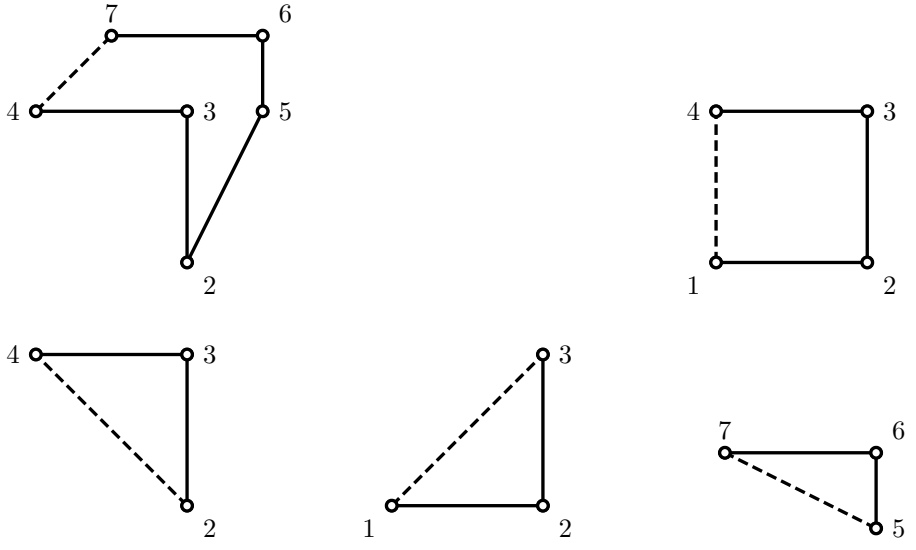


has 3 cycle sets, two of size 3 and one of size 4, but many more actual cycles.

Example 3.10 Consider the graph G below, together with the spanning tree T indicated on the right by the solid edges:



Because G has 7 vertices and 11 edges, it has $11 - 7 + 1 = 5$ cycles in a fundamental set of cycles. For the given tree T, they are as follows, the edge indicated by a dashed line being the one that creates the cycle:



Application to electrical circuits

The notion of a fundamental set of cycles has an application to electrical circuits. Let us first recall some key points concerning circuits, restricting ourselves to the situation where there is a single energy source with a positive terminal and a negative one:

- Electrons, being negatively charged, flow from the negative terminal to the positive.
- The positive terminal has the higher electrical potential, so electrons flow from points of lower potential to points of higher potential.
- It is nonetheless the accepted convention that the current in a circuit flows from the positive terminal to the negative, that is, in the opposite direction to the actual flow of electrons. (The discovery of the electron in 1897 by J. J. Thomson came decades after the notion of electrical current was first introduced.)
- Ohm's law is the law asserting that the drop in electical potential from a point A in a circuit to a point B is proportional to the current flowing from A to B. If V is the fall in potential and I the current, the value $R = V/I$ is called the *resistance*.

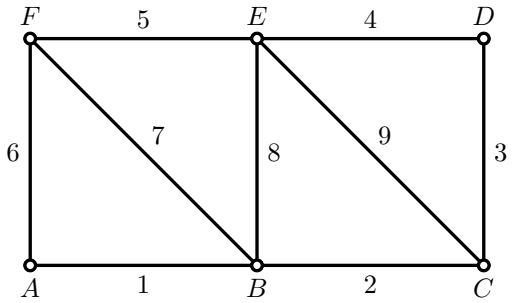
Kirchhoff's circuit laws for a closed circuit are as follows:

(i) **(Current law)** At each junction of the circuit, the sum of the currents flowing in is equal to the sum flowing out.

(ii) **(Voltage law)** The sum of the potential differences around any closed loop in the circuit is zero.

The voltage law has a lot of redundancy in it, because the question of whether the zero-sum condition holds for all cycles boils down, in fact, to whether it holds on a fundamental set of cycles with respect to a spanning tree T. We illustrate with an example.

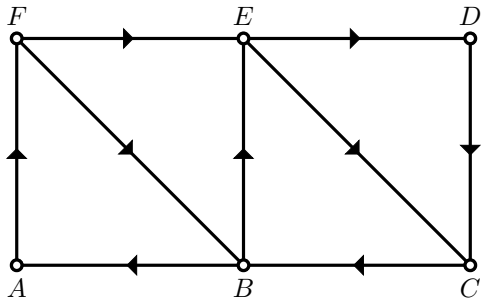
Example 3.11 Consider the graph below representing a circuit, in which the edges represent wires with components and the vertices represent junctions in the circuit. The letters and numbers are merely labels for the vertices and edges and do not represent physical information about the circuit.



Suppose that an electromotive force keeps A at a higher potential than B and that the potential difference between the points is 10 (in some units of potential difference), and suppose that the resistances (in some units of resistance) in each wire (edge) are as follows:

Wire	1	2	3	4	5	6	7	8	9
Resistance	–	2	1	2	2	1	3	1	3

(We do not consider the resistance in the energy source at edge 1.) Our goal is to determine the currents in the various parts of the circuit. For this, we will need to introduce arrows to represent the direction of current flow. We may begin by assigning directions arbitrarily:



This is called a *digraph* (directed graph), and the directed edges are called *arcs*. Let a_i be the current along edge i in the direction indicated by the arc, taking the units of current to be compatible with the units of potential difference and resistance chosen earlier. If the current flows in the opposite direction to the arc, then a_i will have a negative value. (Actually, we have chosen one of the arcs deliberately to be in the 'incorrect' direction. We will see this when we complete our calculations below.)

We apply Kirchhoff's laws. There are six vertices, so the current law gives us six equations. However, we will need only five of them: If the law holds at all except possibly one vertex, then it holds at all of them because of a general fact about digraphs which says that the sum of all *in-degrees* in the digraph is equal to the sum of all *out-degrees*, the in-degree at a vertex being the sum of the values of the arcs pointing in to the vertex and the out-degree being the sum of the values pointing out from it. We choose vertices $A, \ldots, E$ and obtain these equations:

$$\begin{aligned} a_1 - a_6 &= 0 \\ a_1 - a_2 - a_7 + a_8 &= 0 \\ a_2 - a_3 - a_9 &= 0 \\ a_3 - a_4 &= 0 \\ a_4 - a_5 - a_8 + a_9 &= 0 \end{aligned}$$

If we now choose the spanning tree T to be, for example, the one with edges $1, 2, 4, 5, 6$, then the voltage law applied to the four corresponding cycles, together with the equation

$$\text{fall in potential} = \text{current} \times \text{resistance},$$

gives these equations:

$$\begin{aligned} a_6 + 3a_7 &= 10 \\ 2a_5 + a_6 - a_8 &= 10 \\ 2a_2 + 2a_5 + a_6 + 3a_9 &= 10 \\ 2a_2 + a_3 + 2a_4 + 2a_5 + a_6 &= 10 \end{aligned}$$

These 9 linear equations in $a_1, \ldots, a_9$ can be solved by Gaussian elimination to yield the following unique solution for the currents:

i	1	2	3	4	5	6	7	8	9
a_i	$\frac{520}{127}$	$\frac{60}{127}$	$\frac{30}{127}$	$\frac{30}{127}$	$\frac{270}{127}$	$\frac{520}{127}$	$\frac{250}{127}$	$-\frac{210}{127}$	$\frac{30}{127}$

The negative value for a_8 shows that the current in that wire is $\frac{210}{127}$ in the opposite direction from the one we chose for that arc.

Fundamental sets of cutsets

Running alongside the foregoing theory is a parallel theory for cutsets. Let T again be a spanning tree in a connected graph G.

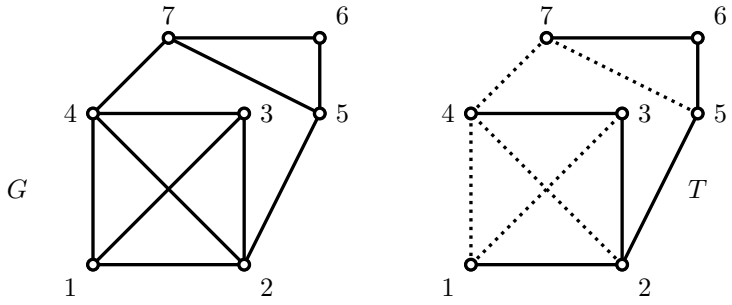
Proposition 3.12 *Let e be an edge in T, let X and Y be the two components of $T - e$, and let C be the set of edges of G having one end in X and the other in Y. Then C is a cutset in G.*

Proof. Suppose that E is a set of edges in G such that $G - E$ is connected, and let v and w be vertices in X and Y respectively. By assumption, there is a path in $G - E$ from v to w. Since the path begins in X and ends in Y, some edge e in $G - E$ has one end in X and the other in Y, so $e \in C$. Thus, C is not a subset of E. Therefore, $G - C$ is disconnected.

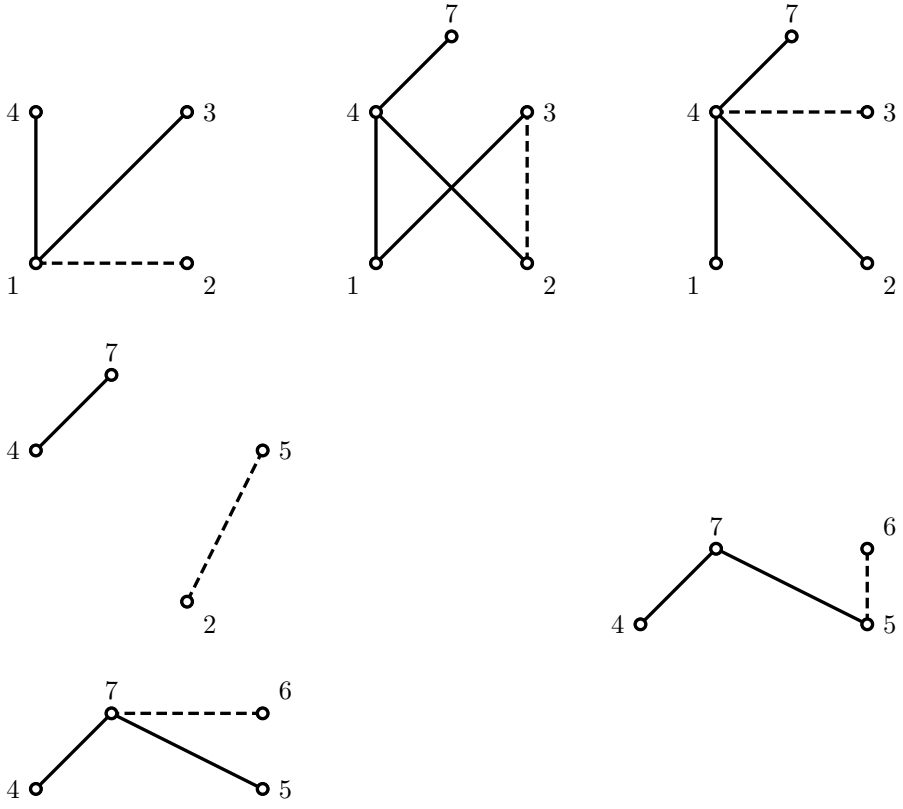
Now, if we take any edge of C and add it back in to $G - C$, then some vertex in X is joined to some vertex in Y, so because X and Y are connected in T, and therefore also in G, we have reconnected the graph. ■

The set of cutsets obtained via the process described in Proposition 3.12 is called the *fundamental set of cutsets* of G with respect to T. This set contains one cutset for each edge in T, so there are $n - 1$ cutsets in a fundamental set, where $n = \#V(G)$.

Example 3.13 Consider again the graph G and spanning tree T appearing in Example 3.10:



The cutsets in the corresponding fundamental set are as follows, the dashed edge in each case representing, in the notation of Proposition 3.12, the edge e of T used to create the two components X and Y:



The minimum-connector problem

If G is a connected weighted graph, the minimum-connector problem for G is to find a spanning tree of least weight in G, called a *minimum spanning tree*. We describe an algorithm to do this and prove that it works.

Input: A connected weighted graph G on n vertices
Initial objects: An empty set E of edges of G
Output: A set of edges in G forming a spanning tree in G (which will be minimal; see Proposition 3.14)
Algorithm:

(1) Among all edges e of $E(G) - E$ such that the subgraph of G with edge set $E \cup \{e\}$ has no cycles, choose one of least weight, and replace E with $E \cup \{e\}$.

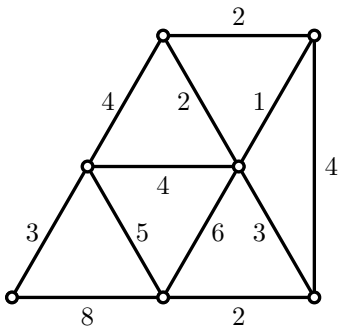
(2) Repeat step (1) until E contains $n - 1$ edges. Output the spanning tree with edge set E.

Proposition 3.14 *Any spanning tree output by the algorithm above is a spanning tree of least weight in G.*

Proof. Let the sequence of edges chosen in the algorithm be, in order, $e_1, \ldots, e_{n-1}$, and let T be the spanning tree with these edges. Suppose now that T' is any spanning tree in G. We show that the weight $\mathrm{wt}(T)$ of T is less than or equal to the weight $\mathrm{wt}(T')$ of T'. If $T' = T$, then we are done immediately, so assume otherwise. Then there is some $k \in \{1, \ldots, n-1\}$ such that e_k is not in T'. Because T' is a tree, $T' + e_k$ contains a cycle C, and some edge e of C must lie outside the tree T. But e is in $T' + e_k$ and is not equal to e_k (e_k is in T while e is not), so e is in T'. Therefore, if $T'' = (T' + e_k) - e$, then T'' is a spanning tree. Indeed, it is connected because it is obtained by removing an edge from the cycle C, and it has both n vertices and $n - 1$ edges. Further, because e is not in T, it is not equal to any of $e_1, \ldots, e_{k-1}$.

Now, since $e_1, \ldots, e_{k-1}, e$ are all in T', if $E = \{e_1, \ldots, e_{k-1}\}$ then $E \cup \{e\}$ contains no cycles. Therefore, because in the algorithm e_k is chosen with least weight, $\mathrm{wt}(e_k) \leq \mathrm{wt}(e)$, so $\mathrm{wt}(T'') \leq \mathrm{wt}(T')$. But T'' has one more edge in common with T than T' does, so continuing in this way we obtain $\mathrm{wt}(T) \leq \mathrm{wt}(T')$. ■

Exercise 3.15 Find a minimum spanning tree in the weighted graph below.



The minimum-connector problem has an application to the travelling salesman problem, which is to find a Hamilton cycle of least weight in a weighted complete graph. This problem is often phrased in terms of somebody who wishes to make a tour of n given cities and return to the first city in such a way as to minimize the total distance travelled.

To see how the minimum-connector problem gives a lower bound on the distance, suppose that C is a Hamilton cycle in a weighted complete graph G. Let v be a vertex in G, and let e_1, e_2 be the edges in C incident with v. Then $C - v$ is a spanning tree in $G - v$, and

$$\begin{aligned} \mathrm{wt}(C) &= \mathrm{wt}(C - v) + \mathrm{wt}(e_1) + \mathrm{wt}(e_2) \\ &\geq \mathrm{wt}(T) + \mathrm{wt}(e_1') + \mathrm{wt}(e_2'), \end{aligned}$$

where T is a spanning tree of minimum weight in $G - v$ and e_1', e_2' are the two edges of least weight in G incident with v.

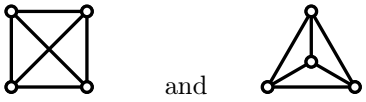
Example 3.16 By considering the minimum connector problem in a suitable graph, find a lower bound on the length of a solution to the travelling salesman problem for the cities A, B, C, D, E, F with distances given as follows in tens of kilometres:

	A	B	C	D	E	F
A	–	4	4	3	2	4
B	–	–	1	9	1	7
C	–	–	–	8	4	4
D	–	–	–	–	3	6
E	–	–	–	–	–	4
F	–	–	–	–	–	–

Solution: We remove a city, B say, and find a minimum spanning tree in the remaining complete graph on 5 vertices. In the case where B is removed, a minimum spanning tree consists of the edges AE, DE, AC, AF and has weight 13. Since the two edges of least weight incident with B both have weight 1, a lower bound for the travelling salesman problem for the six cities is $13 + 1 + 1 = 15$, corresponding to a distance of 150 km. Other lower bounds are possible if the city removed is not B, but no city yields a better lower bound than 150 km.

4 Planarity

When we try to draw a graph in the plane (on paper, so to speak), we may end up drawing edges that cross. Sometimes, doing this is a matter of convenience, but other times it is forced. For example, while the drawings



and

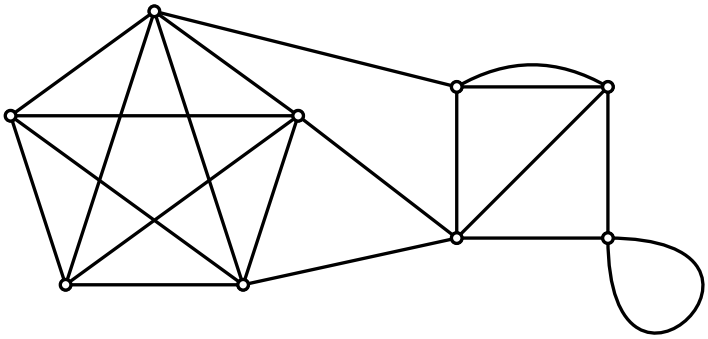
show that K_4 can be drawn in the plane both with and without crossings, it is not possible to draw K_5 in the plane without crossings. One crossing is the best one can do. (Try it.)

A graph that can be drawn in the plane without crossings is said to be *planar*. Otherwise, it is *non-planar*. A *plane drawing* of a planar graph is a crossing-free drawing of that graph in the plane. The graph $K_{3,3}$ is not planar. Try to draw it in the plane and see what is the least number of crossings you can achieve.

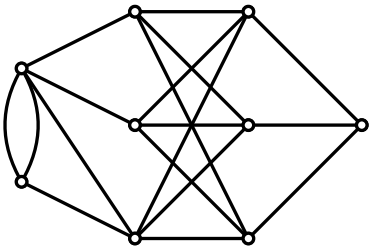
Remark. A subgraph of a planar graph is again planar, which is to say that if a graph G has a non-planar subgraph, then G itself is not planar either.

Example 4.1 The following graphs G and H have subgraphs isomorphic to K_5 and $K_{3,3}$ respectively, so neither G nor H is planar.

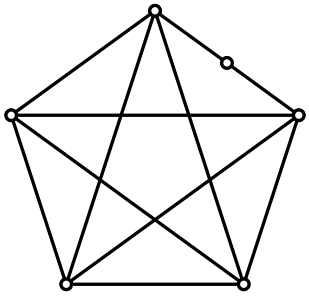
G



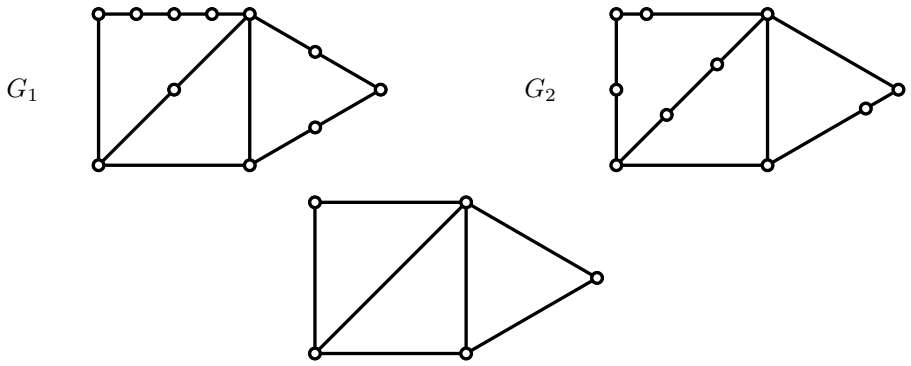
H



Now, a graph that is not planar need not necessarily have a subgraph isomorphic to K_5 or $K_{3,3}$. An example is this non-planar graph, which has no subgraph isomorphic to either of those two:



However, we do have something close to that. Let us say that two graphs G_1 and G_2 are *homeomorphic* if there is a graph G such that each of G_1 and G_2 can be obtained by inserting vertices of degree two into edges of G:

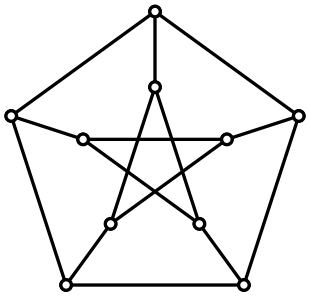


Theorem 4.2 (Kuratowski) *A graph is planar if and only if it has no subgraph homeomorphic to K_5 or $K_{3,3}$.*

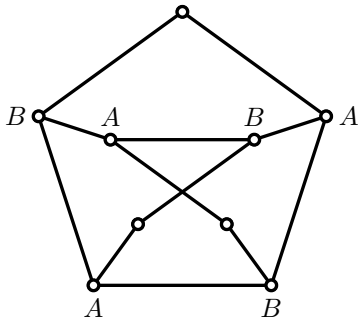
A sketch of the proof is given in Section 4.1 of Wilson. Quite often, however, one uses only the easier direction of Kuratowski's Theorem, that a graph with a subgraph homeomorphic to K_5 or $K_{3,3}$ is non-planar. The next example illustrates the use of that direction.

Example 4.3 By finding a subgraph homeomorphic to K_5 or $K_{3,3}$, show that the Petersen graph is non-planar.

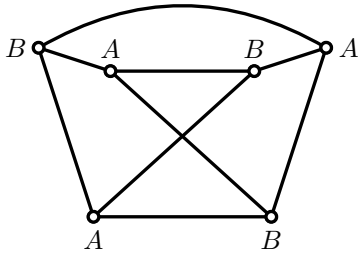
Solution: Here is the Petersen graph:



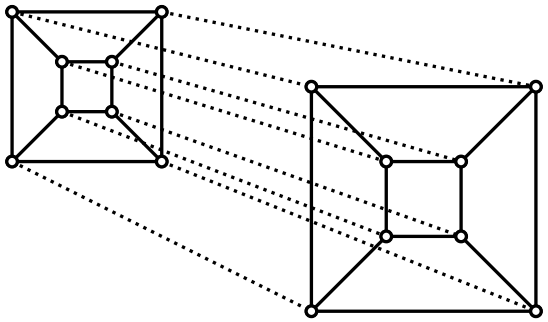
And here is a subgraph homeomorphic to $K_{3,3}$:



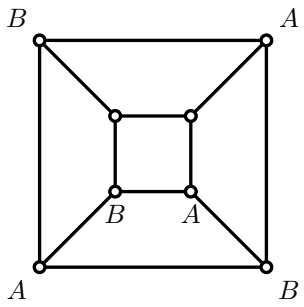
Observe how one obtains $K_{3,3}$ itself by replacing combinations edge-v-edge, where v is a vertex of degree 2, by a single edge:



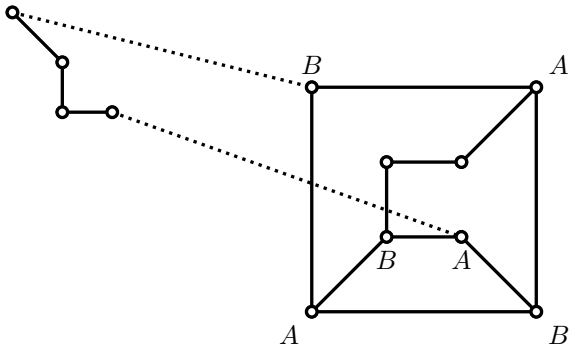
Example 4.4 Here is the 4-cube, Q_4:



We have dotted some of the edges to make the graph easier to visualize. We claim that it has a subgraph homeomorphic to $K_{3,3}$. To do this, we label six of the vertices in the 'front face' with A's and B's, corresponding to the two parts of $K_{3,3}$:



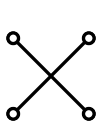
And here is a subgraph homeomorphic to $K_{3,3}$:



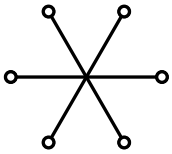
The path extending out to the back, joining an A to a B, corresponds to one edge in $K_{3,3}$, as does the path of length 3 from an A to a B in the front face.

4.1 The crossing number of a graph

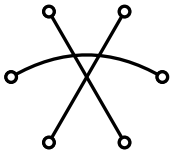
There are two types of edge crossing: crossings that involve only two edges, which we will call *proper*, and ones that involve more than three edges, which we will call *improper*.



A proper crossing



An improper crossing

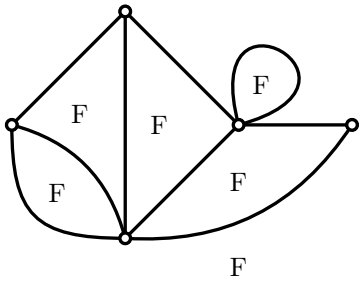


Three proper crossings

Any graph can be drawn in the plane without improper crossings. Such a drawing is called *proper*. The *crossing number* of a graph, denoted $\mathrm{cr}(G)$, is the least possible number of crossings in a proper drawing of G in the plane. For example, $\mathrm{cr}(K_5)$ and $\mathrm{cr}(K_{3,3})$ are both 1. Note that $\mathrm{cr}(G) = 0$ if and only if G is planar.

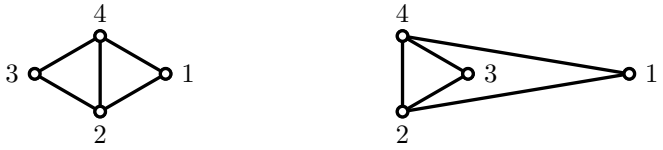
4.2 Faces

If G is a plane graph, i.e., a plane drawing of a planar graph, then the edges divide the plane into regions, called *faces*.



Each F here is a face. The F 'outside' the graph represents the *infinite face*, although it should not be considered special. Indeed, the correct way to view graph drawings on the plane is as drawings on a sphere, and then all faces have equal status and there is no face that could be called the infinite face.

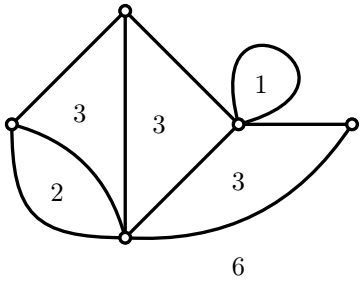
Example 4.5 Here are two plane drawings of the same planar graph:



In the left-hand drawing, the edges $\{1,2\}$, $\{2,3\}$, $\{3,4\}$, and $\{1,4\}$ 'bound' the infinite face, whereas the same four edges in the right-hand drawing bound a face 'inside' the graph. But if we consider these drawings as representing drawings on the sphere instead of the plane, as is more natural, then in both cases, the faces bounded by those edges are essentially the same. In fact, the two drawings on a sphere would be the same; try it on your own sphere, either physical or mental.

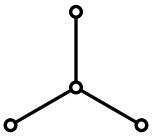
Each edge in a plane graph has two sides. A formal definition of the notion of a side of an edge is given below, but the intuitive meaning is what you think it is. We will say that a face and an edge side are *incident* if the edge side 'meets' the face, in a sense that can also be made precise. The *degree* of a face F in a plane graph is the number of edge sides incident with it, and is denoted $d(F)$. We will also say that a face and an edge are incident if the face is incident with at least one side of the edge.

Example 4.6 The various face degrees in the plane graph below are indicated:



It is worth pointing out the face analogue of loops. The unique face in the graph

is incident with two edge sides of the same edge and therefore has face degree 2. This is analogous to the way that a vertex with a loop on it is situated at both ends of that one loop. Similarly, the unique face in the tree



has degree 6, being incident with all 6 edge sides in the tree.

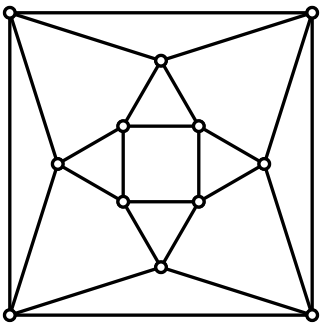
Lemma 4.7 (Face-kissing lemma) *If G is a plane graph whose number of edges is m, then*

$$\sum_F d(F) = 2m,$$

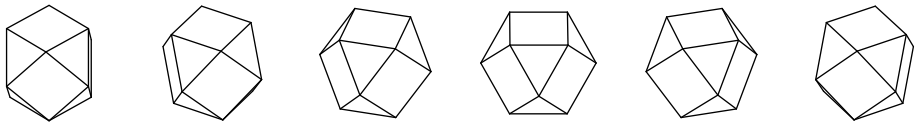
the sum here running over all faces in G.

Proof. Each edge side is incident with a unique face and counts one to the degree of that face, so the number of edge sides is the sum of all face degrees. But the number of edge sides is also twice the number of edges, each edge having two sides. ■

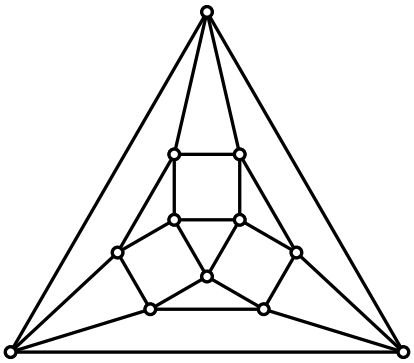
Example 4.8 The following plane graph represents the vertices and edges of a 4-regular convex polyhedron in which every face has degree 3 or 4:



To drive home the point that there is nothing special about the infinite face (appearing above as one of the faces of degree 4), let us illustrate what the polyhedron itself looks like in three dimensions, rotating it so that we see it from several angles:



If we were to imagine being able to bring our eye up close to one of the faces of degree 4 in the polyhedron and look inside, what we would see is something like the graph drawn above. But if we were to peer into one of the degree-3 faces, we would instead see something like this:



The sides of an edge made formal

Let us give a formal definition of the notion of a side of an edge. Not much is lost if we assume that an edge in a plane graph is described by a smooth parametric curve that does not intersect itself. (More general types of curve can be handled in a similar way, with appropriate consideration of complicating factors.) Call the edge in question e. At each point of e, except at its ends, there is a unit tangent vector, and the two unit vectors in the plane orthogonal to this tangent vectors are called the *unit normal vectors* at that point. (This can be done for

graphs drawn on spheres and other surfaces as well via consideration of the tangent plane at a given point.) It is possible to choose a family of unit normal vectors indexed by points on e other than the end points in such a way that the vectors in the family vary continuously along e. Informally, we mean that as we move along e, we do not allow the unit normal vector to suddenly reverse direction. One may show that there are two such continuous families, and we call them *edges* of e.

We can also give a formal meaning to the notion of incidence of faces and edge sides. Let e be an edge and F a face. If σ is a choice of side of e, and if p is a point on e not equal to an end, let $N_{\sigma,p}$ be the unit normal vector at p corresponding to σ. We will say that the face F and the edge side σ are *incident* if there exist a point p on e (not an end) and a positive real number λ such that

$$\{p + tN_{\sigma,p} \mid 0 < t \leq \lambda\} \subseteq F.$$

Informally, one might say that as we head out from p on the side σ, the first thing we encounter is the face F. This can also be done on a sphere (and on other surfaces of a sufficiently nice kind) by projecting the tangent plane at p onto the sphere and requiring that the projection of the set above be contained in F.

4.3 Euler's formula

Theorem 4.9 (Euler's formula) *Let G be a plane drawing of a connected planar graph, and let n, m, and f be the number of vertices, edges, and faces, respectively, in G. Then $n - m + f = 2$.*

Proof. We proceed by induction on $m \geq 0$. If there are no edges, then being connected, G has only one vertex, and then there is only one face, so $n-m+f = 2$.

Now let $m \geq 1$, and assume the formula holds for all connected planar graphs with $m - 1$ edges. Let G be a plane drawing of a connected planar graph with m edges. If G is a tree, then $m = n-1$ and $f = 1$, so $n-m+f = 2$. Otherwise, G has a cycle. Let e be an edge in that cycle. The graph $G' = G - e$ is still connected, and if n', m', and f' are the relevant quantities for G', then $n' = n$, $m' = m - 1$, and $f' = f - 1$. Therefore,

$$n - m + f = n' - (m' + 1) + (f' + 1) = n' - m' + f' = 2$$

by the inductive hypothesis applied to G'. This completes the induction. ■

Corollary 4.10 *If G is a plane drawing of a planar graph with n vertices, m edges, f faces, and k components, then $n - m + f = k + 1$.*

Proof. Suppose that $k > 1$, the case $k = 1$ having been done already. Join the k components of G into a connected plane graph G' via the addition of $k - 1$

edges. Basic topological reasoning shows that it is indeed possible to do this in such away that the result is a plane graph. The numbers n', m', and f' for G' are $n' = n$, $m' = m + k - 1$, and $f' = f$, so

$$n - m + f = n' - (m' - k + 1) + f' = n' - m' + f' + k - 1 = 2 + k - 1 = k + 1,$$

the second-to-last equality being Euler's formula applied to the connected plane graph G'. ■

An important consequence of Corollary 4.10 is that all plane drawings of the same planar graph have the same number of faces, $f = k + 1 - n + m$. We can this call this number, that is, $k + 1 - n + m$, the number of faces in the planar graph.

Corollary 4.11 *If G is a simple planar graph on $n \geq 3$ vertices, then the number of edges, m, satisfies $m \leq 3n - 6$. If, in addition, G has no triangles, then $m \leq 2n - 4$.*

Proof. It is easy to reduce to the case where G is connected by joining components together with edges, along the lines of the proof of Corollary 4.10. Assume, then, that G is a connected plane graph with n vertices and m edges. Because G is simple, each face has degree at least 3, so by the face-kissing lemma,

$$2m = \sum_F d(F) \geq 3f,$$

where f is the number of faces. Hence, Euler's formula gives

$$6 = 3n - 3m + 3f \leq 3n - m.$$

If, in addition, G has no triangles, then every face degree is at least 4, and the inequality in that case follows in the same way. ■

Example 4.12 We show via Corollary 4.11 that K_5 and $K_{3,3}$ are not planar. Note that both are simple and have at least 3 vertices. Now, K_5 has $n = 5$ vertices, and a simple planar graph on 5 vertices has at most $3\dot{5} - 6 = 9$ edges by the corollary. Since K_5 has 10 edges, it is not planar. For $K_{3,3}$ the argument is the same except that we appeal to the fact that it has no triangles. A triangle-free simple planar graph on 6 vertices has at most $2 \cdot 6 - 4 = 8$ edges, so $K_{3,3}$ having 9 edges, is not planar.

Corollary 4.13 *Let G be a simple graph with $n \geq 3$ vertices and m edges. Then $\mathrm{cr}(G) \geq m - 3n + 6$. If, in addition, G has no triangles, then $\mathrm{cr}(G) \geq m - 2n + 4$.*

Proof. We use Corollary 4.11. Let H be a proper drawing of G in the plane with $r = \mathrm{cr}(G)$ crossings. Remove r appropriate edges to obtain a plane graph H'. If n' and m' denote the relevant quantities in H',

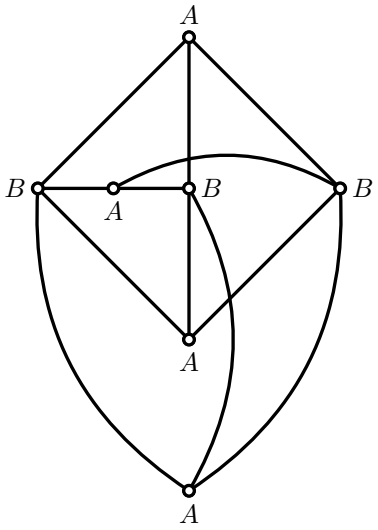
$$r = m - m' \geq m - (3n' - 6) = m - 3n' + 6 = m - 3n + 6.$$

The proof in the case of no triangles is the same. ■

Example 4.14 Let us find $\mathrm{cr}(K_{3,4})$. Here, $n = 7$ and $m = 12$, and $K_{3,4}$ is bipartite and therefore has no triangles, so

$$\mathrm{cr}(K_{3,4}) \geq 12 - 2 \cdot 7 + 4 = 2.$$

However, we can find a plane drawing with only 2 crossings:



Thus, $\mathrm{cr}(K_{3,4}) = 2$.

Corollary 4.15 *Every simple planar graph has at least one vertex of degree less than or equal to* 5*.*

Proof. Let δ be the least vertex degree. Then using the handshaking lemma and Corollary 4.11, we obtain

$$\delta n \leq 2m \leq 2(3n - 6) = 6n - 12 < 6n,$$

so $\delta < 6$. ■

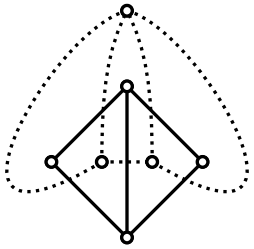
4.4 Duality

Let G be a plane graph, i.e., a plane drawing of a planar graph. We may construct from G another plane graph called the *geometric dual* of G, denoted G^*. The construction goes as follows:

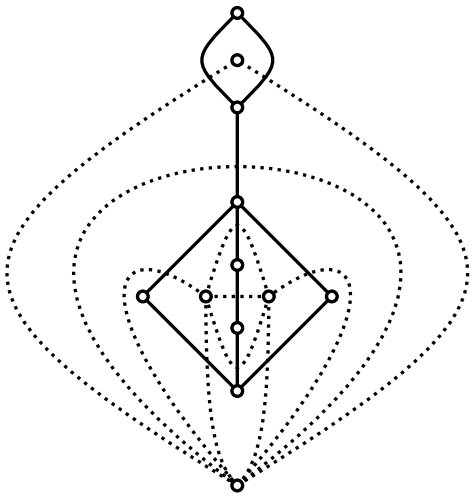
(i) Inside each face F of G, choose a point F^*. (Remember to include the infinite face.) These points F^* will be the vertices of the new graph.

(ii) For each edge e of G, draw a curve e^* that crosses e (and only e) and joins the points F^* representing the faces either side of e. If both sides of e are incident with the same face, then the curve will be a loop on the associated vertex F^*, crossing e and no other edges. The curves e^* are the edges of G^*. Make sure that none of the edges of G^* cross one another.

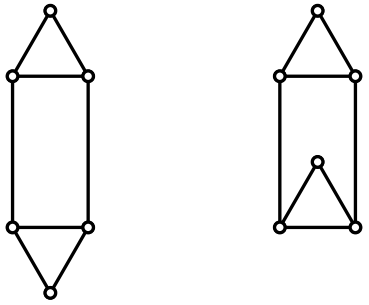
Example 4.16 Here are a graph (solid edges) and its geometric dual (dotted edges):



Example 4.17 We illustrate by example how a bridge in a plane graph G gives rise to a loop in G^*, and illustrate also the appearance of multiple edges in G^* when two faces in G have more than one edge in common.

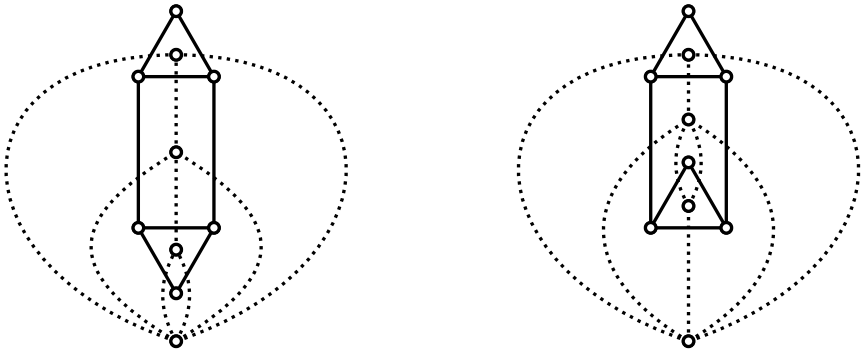


Remark. Consider these two plane graphs G and H:



G H

They are not isomorphic as crossing-free graph drawings on the sphere. An easy way to see this is by considering the face degrees. For example, H has faces of degree 5, but G does not. Now notice what happens when we take their geometric duals:



G and its dual H and its dual

We see that G^* and H^* are not isomorphic even as abstract graphs, their degree sequences being $(6, 4, 3, 3)$ and $(5, 5, 3, 3)$. This example illustrates the importance of taking care over which plane drawing of a planar graph one is considering.

Proposition 4.18 *If G is a connected plane graph with n vertices, m edges, and f faces, then*

$$n^* = f, \quad m^* = m, \quad f^* = n,$$

where n^*, m^*, f^* are the number of vertices edges, and faces, respectively, in G^*.

Proof. The equalities $n^* = f$ and $m^* = m$ hold by construction. For the third equality, we use Euler's formula. The geometric dual of any plane graph is connected, so we may apply the formula to both G and G^*. Thus,

$$f^* = 2 - n^* + m^* \quad \text{(formula applied to } G^*\text{)}$$

$$= 2 - f + m$$
$$= n \quad \text{(formula applied to } G\text{)}.$$

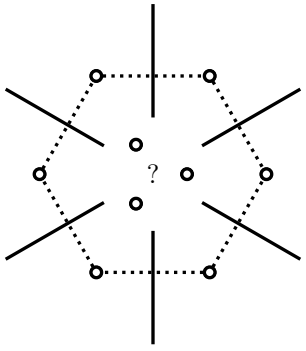
■

The geometric dual of the geometric dual of a plane graph G is denoted G^{**}, a shorthand for $(G^*)^*$.

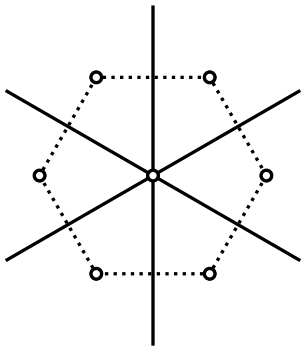
Theorem 4.19 *If G is a connected plane graph, then G and G^{**} are isomorphic as crossing-free graph drawings on the sphere.*

Proof. The case where G has no edges is trivial, so we may assume otherwise, and then G^* has at least one edge as well. We attempt to reconstruct G from G^*. Let Φ be a face of G^*, and choose any edge ε incident with Φ. By the construction of G^* from G, ε crosses a unique edge e of G, and further, ε and e cross only once and e crosses no other edges of G^*. Therefore, at least one end of e is in Φ. Thus, every face of G^* contains at least one vertex of G. But Proposition 4.18 shows that G^* has as many faces as G has vertices, so it follows that each face Φ of G^* contains exactly one vertex $v(\Phi)$ of G.

Now, for each edge ε of G^* incident with Φ, if Φ is incident with only one of its sides, then the edge e of G crossing ε has one end in Φ (at $v(\Phi)$) and the other in a neighbouring face, and if Φ is incident with both sides of ε, then e is a loop crossing ε and joining $v(\Phi)$ to itself. But then the vertices $v(\Phi)$ and edges e as just described have exactly the relationship to G^* as do the vertices and edges of G^{**}: one vertex of G (and likewise G^{**}) in each face of G^*, and edges in G (and likewise G^{**}) crossing edges in G^* in the way indicated above. Thus, G and G^{**} are isomorphic. Here is a diagram illustrating the situation in the more straightforward case where Φ is incident with only one side of each of its incident edges:



The dotted edges are those in G^* bounding the face Φ, and the solid lines we represent the edges e of G crossing edges ε of G^* incident with Φ. The number of vertices of G inside Φ is initially unknown, but one may use a counting argument (see above) to determine that there is one and only one vertex of G in each face of G^*:



■

If G is a connected plane graph, then by construction, the faces of G are in canonical bijection with the vertices of G^*. If F is a face of G, let F^* denote the corresponding vertex in G^*. On the other hand, every face of G^* contains a unique vertex of G, as we saw in the proof of Theorem 4.19, and, further, every vertex of G is in a unique face of G^*. Therefore, it is also the case that the vertices of G are in canonical bijection with the faces of G^*. (One could also reason in terms of G^* and G^{**}, but it comes to the same thing.) If v, then, is a vertex of G, let v^* be the corresponding face of G^*.

Proposition 4.20 *Let G be a connected plane graph. For all vertices v and faces F in G,*

$$d_G(v) = d_{G^*}(v^*)$$
$$\text{and} \quad d_G(F) = d_{G^*}(F^*).$$

Proof. The second equality is true by construction. Now, if v is a vertex in G, then the degree of the face v^* in G^* is equal to the degree of the vertex v^{**} in G^{**}, again by construction. But if we take G^{**} to be G as in the proof of Theorem 4.19, then the vertex v^{**} is just v and has the same degree as v. ■

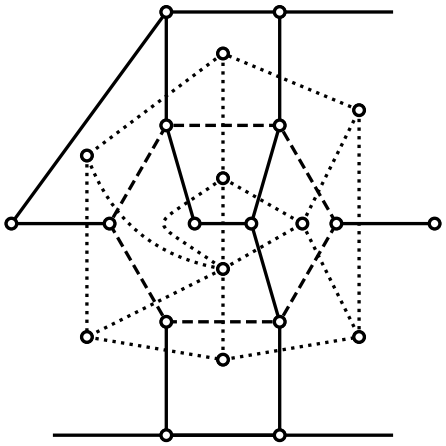
For the next result, we extend the notion of *cutset* to graphs that are not necessarily connected. Let us say that a set of edges is a *disconnecting set* if its removel increases the number of components, and then let us define a *cutset* to be a minimal disconnecting set. These notions, thus defined, coincide with the notions defined earlier.

Theorem 4.21 *Let G be a plane graph.*

(i) A set of edges of G is a cycle set if and only if the corresponding set of edges in G^* is a cutset.

(ii) A set of edges of G is a cutset if and only if the corresponding set of edges in G^* is a cycle set.

Proof. Consider first the case where G is connected. Let E be any set of edges in G and E^* the corresponding set of edges in G^*. Suppose first that E encloses a region of the plane. In the point of view of drawings on a sphere, this would mean that E divides the sphere in two. Any edge in G^* that joins a vertex of G^* on one side of E to a vertex on the other must cross some edge in E, i.e., must be in E^*, so removing the edges of E^* from G^* would disconnect it. Here is an illustration, in which the dashed edges form a cycle set in G, the bold edges are selected other edges in G, and the dotted edges are selected edges in G^*:



Conversely, suppose that E does not enclose a region of the plane (i.e., does not divide the sphere). Then between any two faces F_1 and F_2 of G, there is a path of faces from one to the other that does not cross any edges of E. This corresponds to a path from the vertices F_1^* to F_2* in G^* that does not use any edges in E^*. Thus, $G^* - E^*$ is connected, so E^* is not a cutset in G^*.

We have thus shown that sphere-dividing sets of edges in G correspond to disconnecting sets in G^*, so minimal sphere-dividing sets correspond to minimal disconnecting sets. But a minimal sphere-dividing set is a cycle, and a minimal disconnecting set is a cutset. The first assertion of the theorem is proven.

For the second assertion, we may simply replace G be G^* in the first assertion and use the fact, established in Theorem 4.19, that G^{**} is isomorphic to G.

Finally, we turn to the general case of a plane graph G that is not necessarily connected. In fact, we can prove that cycles in G correspond to cutsets in G^* in the same way as in the connected case above, and we shall not repeat the argument here. To show that cutsets in G correspond to cycles in G^*, suppose first that C is a cutset in G. Then C is contained entirely in one component H of G. The edges of C^* are contained entirely in H^*, so we may apply our result about connected plane graphs to see that C^* is a cycle in H^* and therefore a cycle in G^*. Conversely, any cycle C^* in G^* must be contained entirely in H^* for some component H of G, and again we may apply our result for connected plane graphs to deduce that the corresponding set C in H is a cutset of H and therefore a cutset of G. ■

4.5 Vertex regularity, face regularity, and Platonic graphs

An early use of graph-theoretic reasoning was in the study of convex polyhedra, such as cubes, tetrahedra, and dodecahedra. The vertices and edges of these shapes, in the geometric senses of these words, could be represented by vertices and edges in a graph that represented the essential combinatorial properties of the polyhedra, and this graph is necessarily planar.

Graph theory can be put to effective use to count the possibilities for the numbers of vertices, edges, and faces in a convex polyhedron that exhibits some amount of regularity in its vertex and face degrees. The next proposition illustrates a typical situation.

Proposition 4.22 *Let G be a connected cubic plane graph in which every face has degree 5 or 6. Then the number of faces of degree 5 is 12.*

Proof. Let f_k be the number of faces of degree k. We have the following equalities:

$$\begin{aligned} 2 &= n - m + f \quad \text{(Euler's formula)} \\ 3n &= 2m \quad \text{(handshaking lemma in a cubic graph)} \\ 5f_5 + 6f_6 &= 2m \quad \text{(face-kissing lemma)} \\ f_5 + f_6 &= f \quad \text{(every face has degree 5 or 6)} \end{aligned}$$

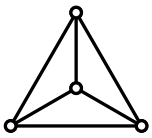
Hence,

$$\begin{aligned} 12 &= 6n - 6m + 6f \\ &= 6f - 2m \quad (6n - 4m) \\ &= 6(f_5 + f_6) - (5f_5 + 6f_6) \\ &= f_5. \end{aligned}$$

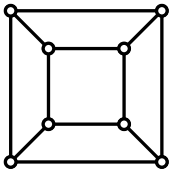
■

Platonic graphs

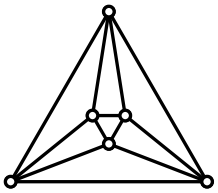
Let us say that a plane graph is *l-face regular* if every face has degree l. (As always, remember the infinite face if the graph is drawn in the plane rather than on the sphere.) We shall call it *(k, l)-regular* if it is k-regular in the usual sense and l-face regular. With these definitions made, we define a *Platonic graph* to be a connected plane graph that is (k, l)-regular for some $k, l \geq 3$. We will show that there are only five Platonic graphs, namely, these:



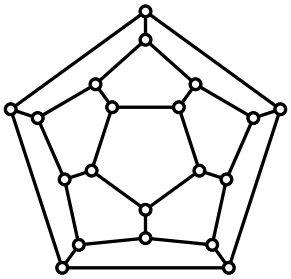
Tetrahedron, $(3, 3)$-regular



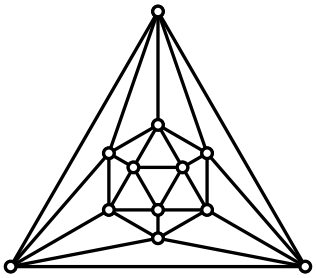
Cube, $(3,4)$-regular



Octahedron, $(4,3)$-regular



Dodecahedron, $(3,5)$-regular



Icosahedron, $(5,3)$-regular

Theorem 4.23 *The five plane graphs above are the only Platonic graphs.*

Proof. Let G be a connected (k,l)-regular plane graph with $k,l \geq 3$, and let n, m, and f denote, as usual, the numbers of vertices, edges, and faces respectively. Then

$$\begin{aligned} 2 &= n - m + f \quad \text{(Euler's formula)},\\ 2m &= nk \quad \text{(handshaking lemma)},\\ 2m &= fl \quad \text{(face-kissing lemma)}, \end{aligned}$$

so we may eliminate n and f to obtain $2 = \frac{2m}{k} - m + \frac{2m}{l}$, which, once rearranged, says

$$\frac{1}{k} - \frac{1}{m} + \frac{1}{l} = \frac{1}{2}.$$

Hence,

$$k = \frac{1}{\frac{1}{2} + \frac{1}{m} - \frac{1}{l}} < \frac{1}{\frac{1}{2} - \frac{1}{l}} = 2\,\frac{l}{l-2} \leq 2 \cdot 3 \quad \text{because } l \geq 3$$
$$= 6,$$

so $k \in \{3, 4, 5\}$. A similar argument shows that

$$l < 2\,\frac{k}{k-2}.$$

Therefore,

- if $k = 3$, then $l < 6$ (so $l \in \{3, 4, 5\}$),
- if $k = 4$, then $l < 4$ (so $l = 3$), and
- if $k = 5$, then $l < 4$ (so $l = 3$),

so the only possibilities for the pair (k, l) are

$$(3,3), \quad (3,4), \quad (3,5), \quad (4,3), \quad (5,3).$$

Any attempt to draw a $(3, 3)$-regular, connected plane graph results in the tetrahedron, and similarly the other four possibilities for (k, l) yield only the other four Platonic graphs. ■

Corollary 4.24 *The tetrahedron is self-dual (is isomorphic to its own dual), the cube and the octahedron are dual to each other, and the dodecahedron and the icosahedron are dual to each other.*

Proof. If G is (k, l)-regular, then G^* is (l, k)-regular. Therefore, since the vertex and face regularities of a Platonic graph determine the graph, we are done by the theorem. For example, the cube, which is $(3, 4)$-regular, has a dual that is a $(4, 3)$-regular Platonic graph, which is necessarily the octahedron. ■

Biparite plane graphs and Eulerian plane graphs

Lemma 4.25 *Let G be any connected graph, not necessarily planar. Then G is Eulerian if and only if every cutset has even cardinality.*

Proof. If every cutset has even cardinality, then in particular every vertex v has even degree. Indeed, the set of non-loops incident with v is a cutset and therefore has even cardinality, and each loop incident with v counts for 2 in the vertex degree.

Conversely, suppose that G is Eulerian, so that every vertex has even degree. Let C be a cutset, let X be the vertex set of one of the two components of $G - C$,

and let X_0 be the set of vertices $v \in X$ such that v is an end of some edge in C. Also, given $v \in X$, write $d_X(v)$ for the degree of v in the induced subgraph on X. For emphasis, we write $d_G(v)$ for the degree of v in G itself. Then

$$\begin{aligned}\sum_{v\in X} d_X(v) &= \sum_{v\in X\smallsetminus X_0} d_X(v) + \sum_{v\in X_0} d_X(v)\\ &= \sum_{v\in X\smallsetminus X_0} d_G(v) + \sum_{v\in X_0} d_G(v) - \#C\\ &= \sum_{v\in X} d_G(v) - \#C.\end{aligned}$$

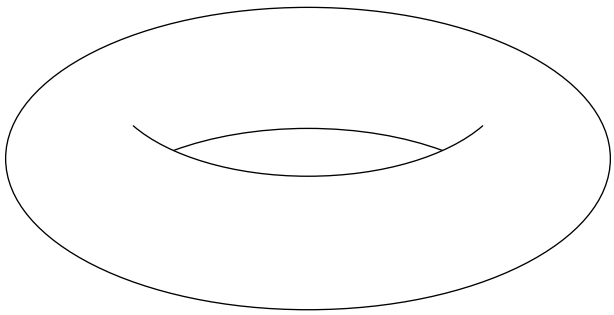
By assumption, $d_G(v)$ is even for all v, and $\sum_{v\in X} d_X(v)$ is even by the handshaking lemma applied to the induced subgraph on X. Thus, $\#C$ is even, as desired. ■

Proposition 4.26 *A plane graph is bipartite if and only if its dual is Eulerian.*

Proof. Let G be a plane graph. We know from Proposition 2.7 that G is bipartite if and only if every cycle has even length, so by Theorem 4.21 it is bipartite if and only if every cutset of G^* has even cardinality, if and only if G^* is Eulerian by Lemma 4.25. (Note that the dual of any plane graph is connected.) ■

4.6 Graphs on other surfaces

A compact Riemann surface in $\mathbb{R}^3$ can be given a *genus*, a non-negative integer, which in very intuitive terms is the number of 'holes' in the surface. A sphere has no holes, whereas a torus (also called a 1-torus to distinguish it from a torus of higher genus g, which would be called a g-torus) has one:

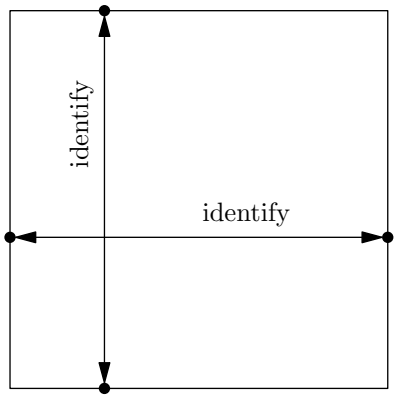


Compact Riemann surfaces of any given genus exist. Throughout, we will use the term *compact Riemann surface* to refer to a compact Riemann surface in $\mathbb{R}^3$, as opposed to some higher dimension.

For any graph G, there is some $g \geq 0$ such that G can be drawn on a compact Riemann surface in $\mathbb{R}^3$ of genus g without any crossings. One has only to add

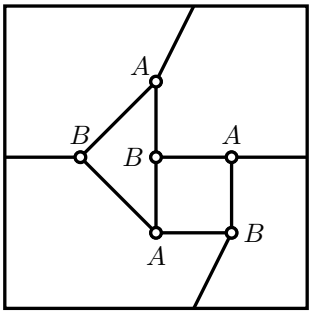
a ‘handle’ at each crossing, forming a bridge that lifts one edge over the other so that they no longer cross on the new surface. In practice, it is usually not necessary to join a separate handle for each crossing, as we shall see.

A common way to represent the torus is as a rectangle in which opposite sides are identified:

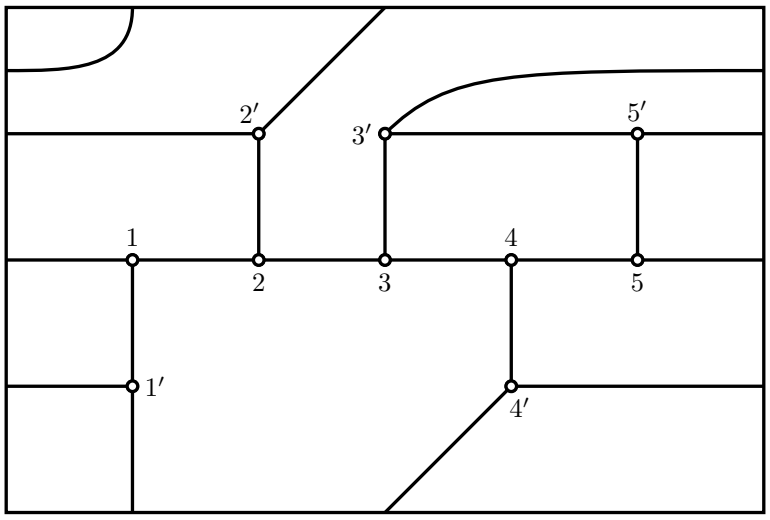


Thus, if we ‘walk’ off the right-hand edge, we reappear at the corresponding point on the left-hand edge, and similarly ‘walking’ off the top brings is back round to the bottom at the corresponding point.

For example, here is $K_{3,3}$ drawn without crossings on the torus:



Also, here is the Petersen graph drawn without crossings on the torus:



The *genus* of a graph G, denoted $g(G)$, is the smallest $g \geq 0$ such that a graph G can be drawn without crossings on a compact Riemann surface of genus g. The genus of a graph G is positive if and only if G is non-planar.

Example 4.27 We have just seen that $K_{3,3}$ can be drawn on a torus without crossings, so its genus is at most 1, but it is non-planar so its genus is positive and therefore equal to 1.

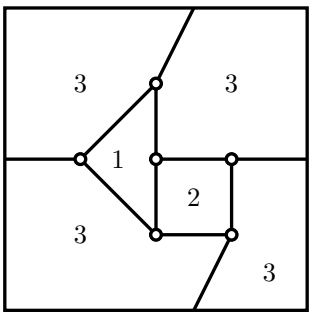
Because crossings in a drawing of a graph can be eliminated by the addition of a handle for each one, we have the inequality

$$g(G) \leq \mathrm{cr}(G)$$

for all graphs G. The inequality is often strict.

Faces

If G is a graph drawn on some compact Riemann surface, then a *face* in this drawing consists of points that can all be connected to each other with curves without crossing edges of the graph. For example, when drawn on a torus, $K_{3,3}$ has three faces, labelled here with $1, 2, 3$:



Theorem 4.28 *If G is a connected graph of genus g drawn without crossings on a compact Riemann surface of genus g, then*

$$n - m + f = 2 - 2g,$$

where n, m, and f are the numbers of vertices, edges, and faces of G respectively.

Proof. We only sketch the key idea. It may be done by induction on the genus, the case $g = 0$ being Euler's formula. For the inductive step, one chooses a handle and removes it, in so doing creating a new graph G' drawn without crossings on a surface of genus $g-1$, in such a way that $n' = n+i$, $m' = m+i$, and $f' = f + 2$, where i is the number of edges in each of the two new faces created. By the inductive hypothesis,

$$n' - m' + f' = 2 - 2g',$$

but $n' - m' = n - m$, so the equalities $f' = f + 2$ and $g' = g - 1$ result in $n - m + f = 2 - 2g$, completing the induction. ■

Corollary 4.29 *If G is a graph of genus g drawn without crossings on a compact Riemann surface of genus g, then*

$$n - m + f = 2 - 2g,$$

where n, m, f, and k are the numbers of vertices, edges, faces, and components of G respectively.

Proof. Join the k components together with $k-1$ edges and apply the theorem to the resulting connected graph. ■

In light of the corollary, the number of faces in a crossing-free drawing of a genus-g graph on a compact Riemann surface of genus g is independent of the choice of drawing, being equal to $k + 1 - 2g - n + m$. Thus, any graph (finite, of course) has a well-defined number of faces.

Here are some useful facts about the genus of a graph:

(1) If G is a connected, simple graph, then $g(G) \geq \frac{1}{6}(m - 3n + 6)$. (Use the theorem above together with the inequality $3f \leq 2m$.)

(2) **(Theorem of Ringel and Youngs)** For all $n \geq 3$,

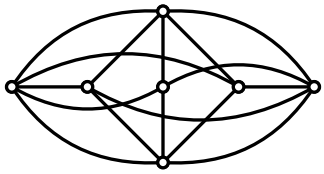
$$g(K_n) = \left\lceil \frac{1}{12}(n-3)(n-4) \right\rceil,$$

where for a real number x, $\lceil x \rceil$ is the least integer greater than or equal to x.

The genera of the complete graphs K_1 to K_9 inclusive are therefore

$$0, 0, 0, 0, 1, 1, 1, 2, 3.$$

Example 4.30 Find the genus of the following graph G, and show that it is less than the crossing number.



Solution: Because G is a subgraph of K_7, we have $g(G) \leq g(K_7) \leq 1$. Also, because G is simple with $m = 16$ edges and $n = 7$ vertices,

$$g(G) \geq \frac{1}{6}(m - 3n + 6) = \frac{1}{6},$$

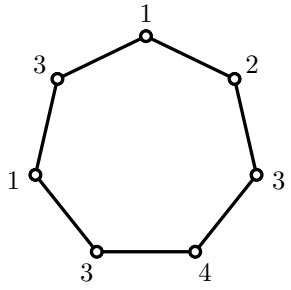
so $g(G) = 1$.

As for the crossing number, observe that G contains a subgraph isomorphic to $K_{3,4}$, which has crossing number 2, so $\mathrm{cr}(G) \geq 2 > 1 = g(G)$.

5 Colouring

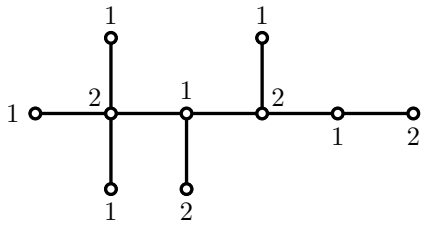
Let G be a graph and S a finite set, which will be referred to as the *palette*. Typically, S is a set of the form $\{1, 2, 3, \ldots, k\}$ for some $k \in \mathbb{Z}_{\geq 1}$. A *vertex colouring* (or just *colouring*, usually) of G by S is a function $f : V(G) \to S$ such that $f(v) \neq f(w)$ whenever v and w are adjacent vertices. A *k-colouring* of G is a colouring of G by a set of cardinality k.

Example 5.1 Here is a 4-colouring of C_7:



It is also a 5-colouring by any finite set containing $1, 2, 3, 4$, and a 6-colouring, and so on. There is no requirement for all colours in the palette to be used.

Example 5.2 Here is a 2-colouring of a tree:



Remark. We will only ever consider simple graphs. A graph with a loop would have no colourings, and in a loop-free graph with multiple edges, repeated edges make no difference.

We say that a simple graph G is *k-colourable* if it has a k-colouring. The *chromatic number* of G, denoted $\chi(G)$, is the least $k \geq 1$ such that G is k-colourable. If a graph is k-colourable, then it is also l-colourable for all $l \geq k$. Every graph on n vertices is n-colourable: just assign each vertex its own colour.

Example 5.3 A graph is 1-colourable if and only if it has no edges, i.e., is null. A graph is 2-colourable if and only if it is bipartite.

Proposition 5.4 *If G is a simple graph with greatest vertex degree Δ, then G is $(\Delta + 1)$-colourable, i.e., $\chi(G) \leq \Delta + 1$.*

Proof. We proceed by induction on n, the number of vertices. The case $n = 1$ is immediate, because then both $\chi(G)$ and $\Delta + 1$ are equal to 1. Now let $n \geq 2$, assume the assertion for graphs on $n - 1$ vertices, and suppose that G has n vertices and greatest vertex degree Δ. Choose any $v \in V(G)$. Then $G - v$ has $n - 1$ vertices, and its greatest vertex degree is at most Δ, so $G - v$ is colourable by a set S of cardinality $\Delta + 1$. It remains to assign a colour to v. Since it has at most Δ neighbours in G, whatever colours its neighbours have, there is still at least one in S left for v. ■

This proposition can be improved, as follows.

Theorem 5.5 (Brooks) *If G is a simple graph that has greatest vertex degree $\Delta \geq 3$ and is not complete, then $\chi(G) \leq \Delta$.*

We refer the reader to Section 5.1 of Wilson for a proof.

5.1 Colouring planar graphs

Via an argument similar to that used above for Proposition 5.4, one may show that every simple planar graph is 6-colourable, taking advantage of Corollary 4.15, which says that every simple planar graph has a vertex of degree not exceeding 5. However, we may do a little better without much more work.

Theorem 5.6 (5-colour theorem) *Every simple planar graph is 5-colourable.*

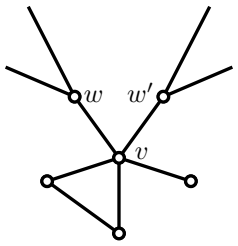
Proof. We prove this by induction on n, the number of vertices. If $n \leq 5$, then the assertion is immediate. Now let $n \geq 6$, assume the assertion for graphs with fewer than n vertices, and suppose that G has n vertices. We consider two cases.

Case (i): G has a vertex of degree 4 or less

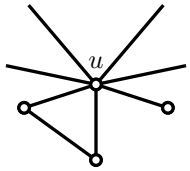
If $d(v) \leq 4$, then we apply the inductive hypothesis to $G - v$ to obtain a colouring of that graph by a set S of cardinality 5. Because v has at most 4 neighbours, there is a colour in S left over for v.

Case (ii): Every vertex has degree at least 5

In this case, G has a vertex v of degree 5 by Corollary 4.15. Because K_5 is not planar, two of the neighbours of v, say w and w', are not adjacent.



Now we contract the edges $\{v, w\}$ and $\{v, w'\}$ so that the vertices v, w, w' collapse into a single vertex, which we call u:

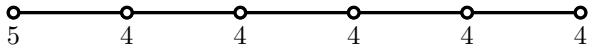


By the inductive hypothesis, the resulting graph G' is colourable by a set S of cardinality 5. Let the colour of u be c, and then in the original graph G, give all vertices in $V(G) \setminus \{v, w, w'\}$ the colours they had in G'. Then assign both w and w' the colour c, possible because they are not adjacent. Thus, at most 4 colours have been used for the five neighbours of v, so there is a fifth colour in S left over for v. This completes the induction. ■

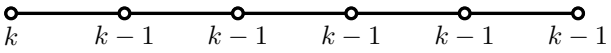
5.2 The chromatic polynomial of a simple graph

Let G be a simple graph. The *chromatic function* of G is the function $P_G : \mathbb{Z}_{\geq 1} \to \mathbb{Z}_{\geq 0}$ defined by letting $P_G(k)$ be the number of colourings of G by a set of cardinality k.

Example 5.7 Consider G is a path graph on 6 vertices $v_1, \ldots, v_6$ in order from one end to the other. If we start at v_1, say, then we have 5 choices for the colour of that vertex, then 4 choices for v_2 because we have to avoid the colour used for v_1, and 4 choices for v_3 because of having to avoid the colour used for v_2, and so on:



Thus, $P_G(5) = 5 \cdot 4 \cdot 4 \cdot 4 \cdot 4 \cdot 4 = 5210$. There is nothing special about 5-colourings here. If S is a palette of k colours, where k is any positive integer, then we have k choices of the first vertex, then $k-1$ for each of the remaining ones in turn:

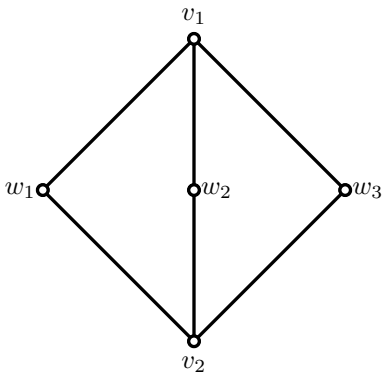


Thus, $P_G(k) = k(k-1)^5$.

Example 5.8 The same argument as above shows that a tree T on n vertices satisfies $P_T(k) = k(k-1)^{n-1}$. Start at any vertex v, giving it any of the k available colours, and branch out (quite literally) from there, taking paths from the first vertex to all others. Each vertex w has only to be given a different colour from the one before it on the unique path from v to w. This argument works because T has no cycles.

If G is a forest with n vertices and r components, then in each component, we begin with a vertex given any of the k colours and branch out from there, so that $P_G(k) = k^r(k-1)^{n-r}$.

Example 5.9 Consider $K_{2,3}$:



Let us count by hand the number of colourings of G by a fixed palette of size k. We consider two cases:

(i) v_1 and v_2 have the same colour. Here, there are k choices for the shared colour of v_1 and v_2, leaving $k-1$ choices for the colour of w_2, $k-1$ for the colour of w_3, and $k-1$ for the colour of w_4. In this case, then, there are $k(k-1)^3$ colourings.

(ii) v_1 and v_2 have different colours. This time, we may choose the colour of v_1 in k ways, then the colour of v_2 in $k-1$ ways, and finally choose the colours of w_1, w_2, w_3 independently of one another in $k-2$ ways each. The total for this case, then, is $k(k-1)(k-2)^3$.

Hence,

$$\begin{aligned} P_G(k) &= k(k-1)^3 + k(k-1)(k-2)^3 \\ &= k(k-1)((k^2-2k+1) + (k^3-6k^2+12k-8)) \\ &= k(k-1)(k^3-5k^2+10k-7). \end{aligned}$$

Exercise 5.10 Use the idea in the previous example to find the chromatic polynomial of $K_{m,n}$.

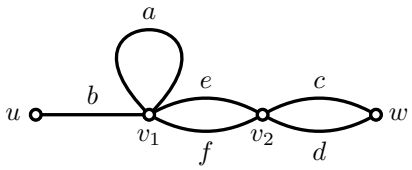
5.3 Edge deletion and contraction

If G is any graph, simple or not, and e is an edge with distinct ends v_1 and v_2 (so it is not a loop), we denote by $G \setminus e$ the graph obtained by bringing its ends together to a single vertex v. The effect on edges in the graph is as follows:

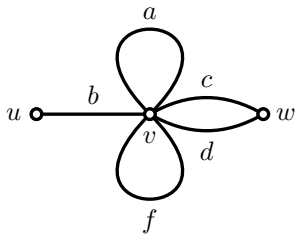
- The edge e disappears.
- Any loop incident with either v_1 or v_2 becomes a loop on v.
- Any edge joining a vertex $w \notin \{v_1, v_2\}$ to v_1 or v_2 becomes an edge from w to v.
- Any edge other than e joining v_1 to v_2 becomes a loop on v.

We say that $G \setminus e$ is the graph obtained by *contracting* e.

For example, contracting edge e in the graph



results in

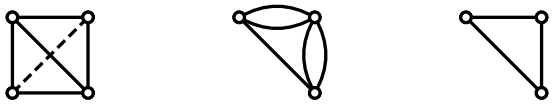


Notice that if G is simple, then $G \setminus e$ need not be simple:

G $\qquad\qquad$ $G \setminus e$

It will be convenient to denote by $G \setminus\setminus e$ the underlying simple graph in $G \setminus e$; that is if G is any graph and e any non-loop, then $G \setminus\setminus e$ is obtained from $G \setminus e$ be removing loops and replacing multiple edges between two given vertices by a single edge. Note that if G is simple, then $G \setminus e$ will have no loops, so replacing multiple edges by single edges is all that would be required.

Example 5.11 Here is a graph G along with $G \setminus e$ and $G \setminus\setminus e$, where e is the dashed edge:



G $\qquad$ $G \setminus e$ $\qquad$ $G \setminus\setminus e$

Recall that if e is an edge in a graph G, then $G-e$ denotes the graph obtained by removing e.

Proposition 5.12 (Deletion-contraction) *If G is simple and e is an edge of G, then*

$$P_G(k) = P_{G-e}(k) - P_{G\setminus\setminus e}(k).$$

Proof. If the ends of e are v and w, then the colourings of $G - e$ fall into two kinds:

(i) Colourings in which v and w have the same colour. These are in bijection with colourings of $G \setminus\setminus e$.

(ii) Colourings in which v and w have different colours. These are in bijection with colourings of G.

■

Proposition 5.13 *Let G be a simple graph with n vertices and m edges. Then the following hold:*

(i) $P_G(k)$ is a polynomial in k of degree n.

(ii) $P_G(k)$ is monic, that is, the term of highest degree n is k^n and not ak^n for some $a \neq 1$.

(iii) The coefficient of k^{n-1} is equal to $-m$.

(iv) If G has r components, then for every $i \in \{1, \ldots, n-r\}$, the coefficient of k^{n-i} lies in $(-1)^i\mathbb{Z}_{>0}$, and the coefficient of k^j is zero for all $j < r$. (Here, $\mathbb{Z}_{>0}$ denotes the set of positive integers, and then $(-1)^i\mathbb{Z}_{>0}$ is the set of positive integers if i is even, and the negative integers if i is odd.)

Proof. Let us prove (i), (ii), and (iii) simultaneously by induction on the number of edges. The case of no edges is immediate because in that case G is null on n vertices and therefore has chromatic polynomial k^n. If G has $e \geq 1$ edges, then both $G - e$ and $G \setminus\setminus e$ have fewer edges, and they have n and $n-1$ vertices respectively, so by the inductive hypothesis $P_{G-e}(k)$ is a monic polynomial of degree n beginning

$$k^n - (m-1)k^{n-1} + \text{lower terms} \quad (G - e \text{ has } m-1 \text{ edges}),$$

and $P_{G\setminus\setminus e}$ is a monic polynomial of degree $n-1$. Therefore, by Proposition 5.12, $P_G(k)$ is a monic polynomial of degree n beginning

$$\Big(k^n - (m-1)k^{n-1} + \text{lower terms}\Big) - (k^{n-1} + \text{lower terms})$$
$$= k^n - mk^{n-1} + \text{lower terms}.$$

We leave the proof of (iv) as an exercise, again involving induction on the number of edges. ∎

Example 5.14 If G is a simple graph and

$$P_G(k) = k^8 - 9k^7 + 34k^6 - 69k^5 + 79k^4 - 48k^3 + 12k^2,$$

then G has 8 vertices (the degree), 9 edges (the negative of the coefficient of k^7), and 2 components (the least power of k in which the coefficient is non-zero).

Remark.

(i) If $G = G_1 \cup G_2$ where G_1 and G_2 are simple, then $P_G(k) = P_{G_1}(k)P_{G_2}(k)$, because the two components may be coloured independently of each other.

(ii) If T is a tree on n vertices, then $P_T(k) = k(k-1)^{n-1}$, as we may verify easily by induction on the number of vertices.

(iii) If simple graph G is a forest if and only if $P_G(k) = k^r(k-1)^{n-r}$ for some positive integers n and r. In this case, n is the number of vertices and r the number of components. (Exercise)

(iv) For every $n \geq 1$, $P_{K_n}(k) = k(k-1)(k-2)\cdots(k-(n-1))$. Indeed, each time a colour is assigned to a vertex, each next vertex to be coloured has to be given a colour that avoids all previously used ones.

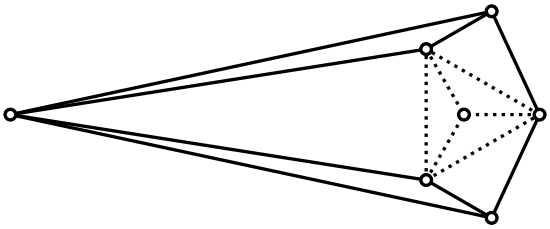
Exercise 5.15 Find the chromatic polynomial of the following graph in two different ways: by counting directly, and by using Proposition 5.12.

5.4 Tools for calculating chromatic polynomials

When finding chromatic polynomials, using deletion-contraction (Proposition 5.12) every time can be time consuming. When a graph has one or more of several features, some shortcuts are available.

Cliques

A *clique* in a graph is a subgraph isomorphic to a complete graph. If the clique has r vertices in it, we call it an *r-clique*. This graph, for example, has a 4-clique, whose edges are the dotted ones:



If a simple graph G contains an r-clique, then $\chi(G) \geq r$, because the r vertices of the clique, being mutually adjacent, need to be coloured with r different colours. We will use the idea of cliques to help us find chromatic polynomials of certain types of graph. First, we introduce some notation and give a lemma.

If G is a simple graph and H a subgraph, let $\mathcal{P}_H(k)$ denote the set of k-colourings of H. Then for $\mathfrak{c} \in \mathcal{P}_H(k)$, let $\mathcal{P}_G^{\mathfrak{c}}(k)$ denote the set of k-colourings of G that extend $\mathfrak{c}$, and let $P_G^{\mathfrak{c}}(k) = \#\mathcal{P}_G^{\mathfrak{c}}(k)$.

Lemma 5.16 *If G is a simple graph and H a subgraph isomorphic to K_r, then for any $k \geq 1$ and any $\mathfrak{c} \in \mathcal{P}_H(k)$,*

$$P_G^{\mathfrak{c}}(k) = \frac{P_G(k)}{k(k-1)\cdots(k-r+1)}.$$

Proof. We show first that $P_G^{\mathfrak{c}}(k)$ is independent of $\mathfrak{c}$. So, let $\mathcal{P}_H(k)$ be the set of k-colourings of H, and suppose that $\mathfrak{c}, \mathfrak{c}' \in \mathcal{P}_H(k)$. Then we choose a permutation σ of $\{1, \ldots, k\}$ that sends $\mathfrak{c}(v)$ to $\mathfrak{c}'(v)$ for every vertex v in H. Such a permutation exists because both of the colourings $\mathfrak{c}$ and $\mathfrak{c}'$ are injective maps $V(H) \to \{1, \ldots, k\}$. The map

$$\mathcal{P}_G^{\mathfrak{c}}(k) \quad \to \quad \mathcal{P}_G^{\mathfrak{c}'}(k)$$

$$\overline{\mathfrak{c}} \quad \mapsto \quad \sigma \circ \overline{\mathfrak{c}}$$

is then a bijection, as desired. Hence,

$$\begin{aligned} P_G(k) &= \sum_{\mathfrak{c} \in \mathcal{P}_H(k)} \#\mathcal{P}_G^{\mathfrak{c}}(k) \\ &= \#\mathcal{P}_H(k) \#\mathcal{P}_G^{\mathfrak{c}_0}(k) \quad \text{for any chosen } \mathfrak{c}_0 \in \mathcal{P}_H(k) \\ &= P_H(k) P_G^{\mathfrak{c}_0}(k) \\ &= k(k-1)\cdots(k-r+1) P_G^{\mathfrak{c}_0}(k). \end{aligned}$$

■

Proposition 5.17 (Clique-intersection trick) *Let G be a simple graph, let V_1 and V_2 be subsets of $V(G)$ such that $V_1 \cup V_2 = V(G)$ and $V_1 \cap V_2 \neq \emptyset$, and suppose that the induced subgraph on $V_1 \cap V_2$ is a complete graph on r vertices for some $r \geq 1$. Finally, assume that there is no edge between $V_1 \smallsetminus V_2$ and $V_2 \smallsetminus V_1$. Then*

$$P_G(k) = \frac{P_{H_1}(k) P_{H_2}(k)}{k(k-1)\cdots(k-r+1)},$$

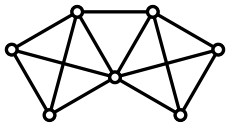
where H_i is the induced subgraph on V_i.

Proof. Let H be the induced subgraph on $V_1 \cap V_2$, and fix some $k \geq 1$. Each k-colouring of G arises from choosing a k-colouring $\mathfrak{c}$ of H and then choosing, for $i = 1, 2$, a k-colouring of H_i that extends $\mathfrak{c}$. Thus, letting $y = k(k-1)\cdots(k-r+1)$, we see that

$$\begin{aligned} P_G(k) &= \sum_{\mathfrak{c} \in \mathcal{P}_H(k)} P_{H_1}^{\mathfrak{c}}(k) P_{H_2}^{\mathfrak{c}}(k) \\ &= \sum_{\mathfrak{c} \in \mathcal{P}_H(k)} \frac{1}{y} P_{H_1}(k) \cdot \frac{1}{y} P_{H_2}(k) \quad \text{by Lemma 5.16} \\ &= \frac{1}{y} P_{H_1}(k) P_{H_2}(k). \end{aligned}$$

■

Example 5.18 Consider this graph G:



Let us take H_1 and H_2 to be these subgraphs:

H_1 H_2

Via deletion-contraction and the fact that K_4 has chromatic polynomial $f(k) = k(k-1)(k-2)(k-3)$, we find that each of H_1 and H_2 has chromatic polynomial $(k-2)f(k)$. Therefore, by the clique-intersection trick in the case $r = 3$,

$$\begin{aligned} P_G(k) &= \frac{P_{H_1}(k)P_{H_2}(k)}{k(k-1)(k-2)} \\ &= \frac{(k-2)^2 f(k)^2}{k(k-1)(k-2)} \\ &= \frac{(k-2)^2k^2(k-1)^2(k-2)^2(k-3)^2}{k(k-1)(k-2)} \\ &= k(k-1)(k-2)^3(k-3)^2. \end{aligned}$$

Triangle trick

Proposition 5.19 (Triangle trick) *If G is a simple graph that has a vertex v of degree 2 forming one vertex of a triangle in G, then*

$$P_G(k) = (k-2)P_H(k)$$

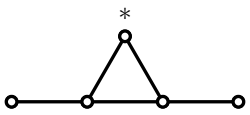
where $H = G - v$.

Proof. If C is the triangle that v is a vertex of, apply the clique-intersection trick with $r = 2$, $H_1 = H$, and $H_2 = C$:

$$P_G(k) = \frac{P_H(k)P_C(k)}{k(k-1)} = (k-2)P_H(k),$$

the chromatic polynomial of C being $k(k-1)(k-2)$. ■

Example 5.20 The chromatic polynomial of



is equal to $k-2$ times the chromatic polynomial of the path graph left by removing the indicated vertex and is therefore equal to

$$(k-2) \cdot k(k-1)^3 = k(k-1)^3(k-2).$$

Cycles

The triangle trick can be extended to include the removal of paths, under appropriate conditions. First, we give the chromatic polynomial of the cycle graph C_n. For each $n \geq 3$,

$$P_{C_n}(k) = (k-1)((k-1)^{n-1} + (-1)^n).$$

We leave it as a short exercise to prove this by induction on n using deletion-contraction.

Proposition 5.21 *Let H be a simple graph, and suppose that v and w are adjacent vertices. Let $t \in \mathbb{Z}_{\geq 2}$, and join a path graph of length t to H be joining one end to v and the other to w. If G is the resulting graph, then*

$$P_G(k) = \frac{1}{k}((k-1)^t - (-1)^t)P_H(k).$$

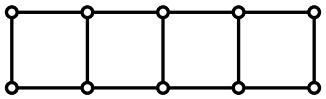
Proof. Let C be the $(t+1)$-cycle obtained by adding the edge between v and w to the path in the proposition. Then we may apply the clique-intersection trick with $r = 2$, $H_1 = H$, and $H_2 = C$:

$$\begin{aligned} P_G(k) &= \frac{P_H(k)P_C(k)}{k(k-1)} \\ &= \frac{P_H(k) \cdot (k-1)((k-1)^t + (-1)^{t+1})}{k(k-1)} \\ &= \frac{1}{k}((k-1)^t - (-1)^t)P_H(k). \end{aligned}$$

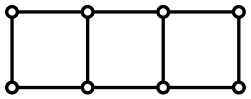
■

We will call the special case $t = 3$ the *square trick*.

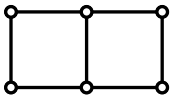
Example 5.22 Let us find the chromatic polynomial of this graph G:



We can do it by repeated use of the square trick. Indeed, let H_1, H_2, and H_3 be the following graphs:



H_1



H_2

H_3

Then by the square trick

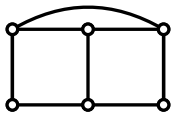
$$\begin{aligned} P_G(k) = (k^2 - 3k + 3)P_{H_1}(k) &= (k^2 - 3k + 3)^2 P_{H_2}(k) \\ &= (k^2 - 3k + 3)^3 P_{H_3}(k) \\ &= k(k-1)(k^2 - 3k + 3)^4 \end{aligned}$$

because H_3 is a copy of C_4.

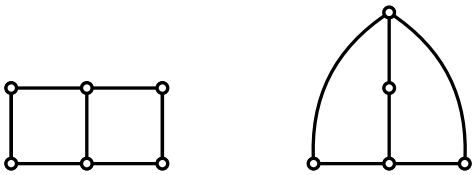
5.5 Further examples involving chromatic polynomials

Ahead of our next example, let us point out a useful shortcut that can be used for end-vertices in the calculation of chromatic polynomials. If G is any simple graph and v an end-vertex of it, then $P_G = (k-1)P_G - v$. We will call this the *end-vertex trick*. It can be proven either directly or via deletion-contraction.

Example 5.23 Find the chromatic polynomial of this graph G:

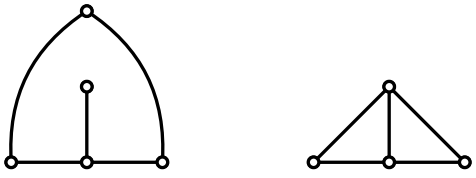


Solution: If e is the curved edge, then $G - e$ and $G \setminus\setminus e$ are as follows:



$G - e$ $\qquad$ $G \setminus\setminus e$

By the square trick, $P_{G-e}(k) = k(k-1)(k^2 - 3k + 3)^2$. To tackle the graph $H' = G \setminus\setminus e$, we apply deletion-contraction again, this time using the upper vertical edge, which we will call e':



$H' - e'$ $\qquad$ $H' \setminus\setminus e'$

The chromatic polynomials of these last two graphs may be found using the end-vertex trick and the triangle trick:

$$P_{H'-e'}(k) = (k-1)P_{C_4}(k) = k(k-1)^2(k^2 - 3k + 3),$$

$$P_{H' \setminus\setminus e'}(k) = (k-2)P_{K_3}(k) = k(k-1)(k-2)^2.$$

Hence,

$$\begin{aligned} P_G(k) &= P_{G-e}(k) - P_{G \setminus\setminus e}(k) \\ &= P_{G-e}(k) - P_{H'-e'}(k) + P_{H' \setminus\setminus e'}(k) \\ &= k(k-1)(k^2-3k+3)^2 - k(k-1)^2(k^2-3k+3) + k(k-1)(k-2)^2 \\ &= k^6 - 8k^5 + 27k^4 - 48k^3 + 44k^2 - 16k \\ &= k(k-1)(k-2)^2(k^2-3k+4). \end{aligned}$$

Some forensics via chromatic polynomials

Proposition 5.24 *If a graph has n vertices, m edges, and r components, then it has at least $m-n+r$ cycles.*

Proof. If G' is a connected graph formed by the addition of $r-1$ edges to G, then G' has the same number of cycles as G, and its number of vertices and edges is $n' = n$ and $m' = m+r-1$. The number of cycles in a fundamental set for G' with respect to a spanning tree is $m'-n'+1 = m-n+r$. ■

Example 5.25 Let $f(k) = k^6 - 6k^5 + 15k^4 - 17k^3 + 7k^2$. Show that there is a unique simple graph such that $P_G(k) = f(k)$.

Solution: If G is a simple graph such that $P_G(k) = f(k)$, then it has $n = 6$ vertices, $m = 6$ edges, and $r = 2$ components, so its number of cycles is at least

$$n - m + r = 6 - 6 + 2 = 2$$

by Proposition 5.24. Now,

$$\begin{aligned} P_G(1) &= 1 - 6 + 15 - 17 + 7 = 0, \\ P_G(2) &= 2^6 - 6 \cdot 2^5 + 15 \cdot 2^4 - 17 \cdot 2^3 + 7 \cdot 2^2 \\ &= 2^2(\underbrace{\underbrace{2^4 - 6 \cdot 2^3 + 15 \cdot 2^2 - 17 \cdot 2}_{\text{even}} + 7}_{\text{odd}}), \end{aligned}$$

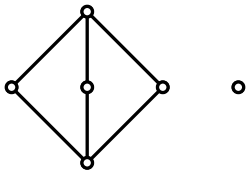
so $P_G(2) \neq 0$ and therefore G is bipartite. Every cycle, then, has even length.

Let the two components of G be G_1 and G_2. At least one of them, say G_1, contains a cycle C. The length of C is greater than 2 because G is simple, and it cannot exceed the number of vertices in G_1, which is no more than 5, so having even length C must be a 4-cycle.

The number of vertices in G_1 is necessarily 5, because otherwise, G_2 would have 2 vertices and one edge, so G_1 would have 4 vertices and 5 edges and would therefore have a 3-cycle. Hence, G_2 consists of a single vertex, and G_1 is a connected simple graph having all of the following properties:

- it has 5 vertices,
- it has 6 edges,
- it contains a 4-cycle,
- it contains no 3-cycle.

The only such graph is $K_{2,3}$. Thus, G is the union of $K_{2,3}$ and the null graph on one vertex:



Finally, we must verify that the chromatic polynomial of $K_{2,3} \cup N_1$, where N_1 denotes the null graph on one vertex, is equal to $f(k)$. In fact, we found $P_{K_{2,3}}(k)$ in Example 5.9, and we obtain

$$\begin{aligned} P_{K_{2,3}\cup N_1}(k) &= kP_{K_{2,3}}(k) \\ &= k \cdot k(k-1)(k^3 - 5k^2 + 10k - 7) \\ &= k^6 - 6k^5 + 15k^4 - 17k^3 + 7k^2 \\ &= f(k). \end{aligned}$$

Families of chromatic polynomials via induction

If one has a family $(G_n)_{n\in I}$ of simple graphs, where $I = \mathbb{Z}_{n_0}$ for some $n_0 \in \mathbb{Z}$, then it may be possible to use induction to determine $P_{G_n}(k)$ by applying deletion-contraction to G_{n+1} and attempting to relate $P_{G_{n+1}}(k)$ to chromatic polynomials of earlier graphs in the family.

Example 5.26 If $n \in \mathbb{Z}_{\geq 4}$, show that the chromatic polynomial of the wheel graph W_n on n vertices is equal to $k(k-2)((k-2)^{n-2} - (-1)^n)$.

Solution: We proceed by induction on n. The case $n = 4$ follows quickly after the observations that $W_4 \cong K_4$ and $P_{K_4}(k) = k(k-1)(k-2)(k-3)$.

Now let $n \geq 4$, and assume the equality for this n. If e is an edge not incident with the central vertex (the vertex of degree n), then

$$P_{W_{n+1}-e}(k) = (k-2)^{n-2}P_{K_3}(k) = k(k-1)(k-2)^{n-1}$$

by repeated use of the triangle trick. Further, $W_{n+1} \backslash\backslash e \cong W_n$, so by the inductive hypothesis,

$$P_{W_{n+1}\backslash\backslash e} = k(k-2)((k-2)^{n-2} - (-1)^n).$$

Hence, by deletion-contraction,

$$P_{W_{n+1}} = k(k-1)(k-2)^{n-1} - k(k-2)((k-2)^{n-2} - (-1)^n)$$

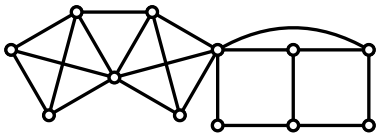
$$= k(k-2)\Big((k-1)(k-2)^{n-2} - (k-2)^{n-2} + (-1)^n\Big)$$
$$= k(k-2)((k-2)^{n-1} - (-1)^{n+1}),$$

completing the induction.

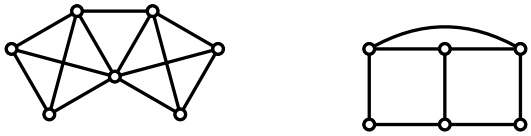
The cut-vertex trick

We will call the case of the clique-intersection trick where $r = 1$ the *cut-vertex trick*. Under the relevant hypotheses (see Proposition 5.17), the cut-vertex says that $P_G(k) = \frac{1}{k} P_{H_1} P_{H_2}$.

Example 5.27 Consider this graph G:



By the cut-vertex trick, $P_G(k) = \frac{1}{2} P_{H_1}(k) P_{H_2}(k)$ where



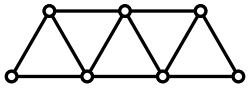
H_1 $\qquad\qquad$ H_2

We have already found $P_{H_1}(k)$ and $P_{H_2}(k)$, in Examples 5.18 and 5.23 respectively, and then the cut-vertex trick gives

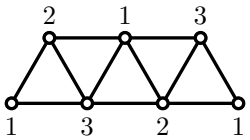
$$\begin{aligned} P_G(k) &= \frac{1}{k} P_{H_1}(k) P_{H_2}(k) \\ &= \frac{1}{k} \cdot k(k-1)(k-2)^3(k-3)^2 \cdot k(k-1)(k-2)^2(k^2-3k+4) \\ &= k(k-1)^2(k-2)^5(k-3)^2(k^2-3k+4). \end{aligned}$$

A calculation of a chromatic number

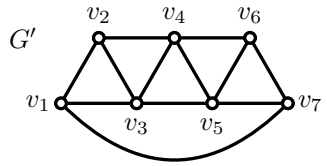
Consider first this graph G:



We could find its chromatic polynomial first (how might you find it?), but if all we want is the chromatic number, a very direct argument suffices instead. We can see that there are 3-cliques in G, so $\chi(G) \geq 3$, but we may readily exhibit a 3-colouring to show that $\chi(3) = 3$, in fact:



Consider, now, this modification of G, called G', in which the labels are the names of the vertices:



Suppose, for a contradiction, that G' is 3-colourable, and let f be a 3-colouring. If $c_i = f(v_i)$ for $i \in \{1, 2, 3\}$, then c_1, c_2, c_3 have to be distinct because those three vertices form a 3-clique. But then, as we progress from left to right along the triangles, the assumption that f is a 3-colouring forces the following, in this order:

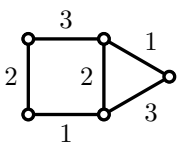
$$\begin{aligned} f(v_4) &= c_1, \\ f(v_5) &= c_2, \\ f(v_6) &= c_3, \\ f(v_7) &= c_1. \end{aligned}$$

Thus, v_1 and v_7 have the same colour, a contradiction because they are adjacent. We conclude, then, that no 3-colouring exists. We can, however, find a 4-colouring simply by choosing v_7 to have a fourth colour, different from c_1, c_2, c_3. Therefore, $\chi(G') = 4$.

5.6 Edge colouring

Let G be a loop-free graph. Multiple edges are allowed. An *edge colouring* of G is a function $f : E(G) \to S$, where S is a finite non-empty set, such that $f(e_1) \neq f(e_2)$ whenever the edges e_1 and e_2 are adjacent. Then G is *k-edge colourable* if there is a colouring of the edges by a set of cardinality k. The least $k \geq 1$ such that G has a k-edge colouring is called the *chromatic index* of G, denoted $\chi'(G)$.

Example 5.28 We indicate a 3-edge colouring of the following graph G:



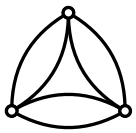
This shows that $\chi'(G) \leq 3$, but in fact it cannot be less, because there is a triangle of edges, which are therefore mutually adjacent. Thus, $\chi'(G) = 3$.

Theorem 5.29 (Vizing) *If G is a simple graph, then $\Delta \leq \chi'(G) \leq \Delta + 1$, where Δ is the greatest vertex degree in G.*

We refer the reader to Douglas B. West's *Intoduction to Graph Theory*, second edition, for a proof. Note, however, that the lower bound is clear, the Δ edges incident with a vertex of greatest degree being mutually adjacent.

Both values of $\chi'(G) - \Delta_G$, i.e., 0 and 1, are possible. Indeed, it is 0 if $G = C_4$, and it is 1 if $G = C_3$.

Remark. The assertion would be false if the word *simple* were removed, as this example shows:

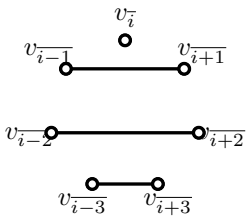


$\Delta = 4$, $\chi' = 6$

Theorem 5.30 *If $n \geq 2$, then*

$$\chi'(K_n) = \begin{cases} n\,(= \Delta_{K_n} + 1) & \text{if } n \text{ is odd,} \\ n-1\,(= \Delta_{K_n}) & \text{if } n \text{ is even.} \end{cases}$$

Proof. Let us first tackle the case where n is odd. It will be convenient to index the vertices of K_n by $\mathbb{Z}/n\mathbb{Z}$, so that, for example, the vertices are $v_{\overline{0}}, \ldots, v_{\overline{n-1}}$ where $\overline{a}$ denotes the residue class of an integer a modulo n. For a given $a \in \{0, \ldots, n-1\}$, the $\frac{n-1}{2}$ edges $\{v_{\overline{a-i}}, v_{\overline{a+i}}\}$ for $i = 1, 2, \ldots, \frac{n-1}{2}$ are pairwise non-adjacent. Indeed, suppose the edges $\{v_{\overline{a-i}}, v_{\overline{a+i}}\}$ and $\{v_{\overline{a-j}}, v_{\overline{a+j}}\}$ share a vertex, where $i, j \in \{1, \ldots, \frac{n-1}{2}\}$. If $\overline{a-i} = \overline{a-j}$ or $\overline{a+i} = \overline{a+j}$, then immediately we have $i = j$, and neither of the other cases is possible, because if, say, $\overline{a+i} = \overline{a-j}$, then n divides $i + j \in \{2, \ldots, n-1\}$, a contradiction. Here is a visual representation in the case $n = 7$, showing the $\frac{7-1}{2} = 3$ edges associated to vertex $v_{\overline{i}}$:



We can therefore colour all $\frac{n-1}{2}$ of these edges with the same colour, say colour c_a. Doing this for all n vertices allows us to colour $n \cdot \frac{n-1}{2}$ edges with n colours. But this is $\binom{n}{2}$ edges and therefore all of them in K_n.

Now we turn to the case where n is even. In that case, we first colour K_{n-1} in the way just described, $n-1$ being odd. Thus, the $\frac{n-2}{2}$ edges associated, as above, with vertex a $v_{\overline{a}}$ in K_{n-1} have colour c_a. Since none of these edges is incident with $v_{\overline{a}}$, any edge that *is* incident with $v_{\overline{a}}$ will have colour $c_{\overline{b}}$ for some $b \in \{0, \ldots, n-1\} \setminus \{a\}$. Therefore, if w is the nth vertex to be joined to K_{n-1} to form K_n, we can join w to $v_{\overline{a}}$ with an edge coloured with $c_{\overline{a}}$. Thus, we obtain an edge colouring of K_n by the set $\{c_{\overline{a}} \mid a \in \{0, \ldots, n-1\}\}$. ■

Theorem 5.31 (Kőnig) *If G is a bipartite graph, then $\chi'(G) = \Delta_G$.*

Proof. We proceed by induction on the number of edges. The case of no edges is vacuous, so now take a bipartite graph G with at least one edge, and assume the truth of the assertion for all bipartite graphs with fewer edges. Choose any edge e in G, call its ends v and w, and apply the inductive hypothesis to the bipartite graph $G' = G - e$. Thus, there is an edge colouring of G' by $\Delta_{G'}$ colours, so because $\Delta_{G'} \leq \Delta_G$, there is an edge colouring of G' by a palette S of cardinality $\Delta_{G'} \leq \Delta_G$.

For brevity, it will be helpful to say that a colour in S is *present* at a given vertex of G' if some edge of G' incident with the vertex has that colour. Otherwise, we will say that the colour is *absent* at the vertex.

Note that the degrees of v and w in G' are both less than Δ_G, because e has been removed, so there is at least one colour $\alpha \in S$ absent at v, and at least one colour $\beta \in S$ absent at w. If $\alpha = \beta$, then we may assign e that colour and thus obtain a colouring of G by S. We may therefore assume that $\alpha \neq \beta$, and then α is absent at v and present at w, and β is absent at w and present at α.

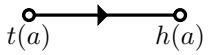
Now form a path in G' using only edges coloured α and β, as follows: Start at v, and note that v is incident with one edge of colour β and no edges of colour α. Our path then begins with the edge coloured β. If the vertex arrived at has an edge coloured α, then take that edge; otherwise stop. We continue in this manner, alternately taking edges of colour β or α until we arrive at a vertex with neither colour (or we exhaust all edges). Let the resulting path be $(v, e_1, e_2, \ldots, e_{k-1}, e_k)$, and notice that β is present at every vertex in the path. Because e_1 has colour β, and α is absent at v, and because e_k is incident with no edge coloured α or β aside from e_{k-1}, we may interchange α and β in every edge $e_1, \ldots, e_k$ and still have an edge colouring of G'. Thus, β is now absent at v. But it is absent from w still, as well, because w is not in the path so none of its incident edges have changed colour. Consequently, we may now assign edge e the colour β and so obtain an edge colouring of G by S. This completes the induction. ■

6 Digraphs

6.1 Basic definitions

A *digraph* (directed graph) consists of a finite, non-empty set of *vertices*, V, a finite set of *arcs*, A, and a function $\psi : A \to V \times V$.

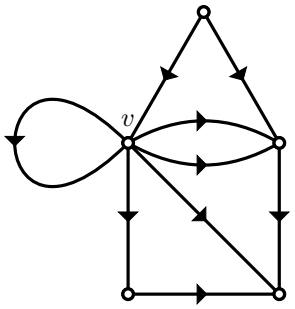
If $\psi(a) = (v, w)$, then v is called the *tail* of a, denoted $t(a)$, and w the *head*, denoted $h(a)$. An arc is always written with an arrow from tail to head:



If $v \in V(G)$, then

$$\begin{aligned} \text{indeg}(v) &= \#\{a \in A(G) \mid h(a) = v\} \quad (\text{the } \textit{in-degree}), \\ \text{outdeg}(v) &= \#\{a \in A(G) \mid t(a) = v\} \quad (\text{the } \textit{out-degree}). \end{aligned}$$

Example 6.1 For the vertex v below, $\text{indeg}(v) = 2$ and $\text{outdeg}(v) = 5$:



Lemma 6.2 (Handshaking dilemma) *If G is a digraph, then*

$$\sum_{v \in V(G)} \text{indeg}(v) = \sum_{v \in V(G)} \text{outdeg}(v).$$

Proof. Each arc has one head and one tail and therefore contributes one head to the sum of the in-degrees and one tail to the sum of the out-degrees. ■

6.2 Flows in weighted digraphs

A *weighted digraph* is a digraph G together with a *weight function*, meaning a function $c : A(G) \to \mathbb{R}_{\geq 0}$. If (G, c) is a weighted digraph and $a \in A(G)$, it is common to call $c(a)$ the *capacity* of a.

We extend the notions of *in-degree* and *out-degree* defined above to weighted graphs. If (G, c) is a weighted graph, then

$$\text{indeg}_c(v) = \sum_{h(a)=v} c(a),$$

$$\mathrm{outdeg}_c(v) = \sum_{t(a)=v} c(a).$$

In the situation where $c(a) = 1$ for all arcs a, we recover the previous notions of in-degree and out-degree.

Lemma 6.3 *If (G, c) is a weighted digraph, then*

$$\sum_{v \in V(G)} \mathrm{indeg}_c(v) = \sum_{v \in V(G)} \mathrm{outdeg}_c(v),$$

the in-degrees and out-degrees here being those given by c.

Proof. An arc a contributes $c(a)$ to the sum of the in-degrees and the same amount to the sum of the out-degrees. ■

Of course, we recover Lemma 6.2 if we take $c(a) = 1$ for all arcs a.

A *source* in a weighted digraph (G, c) is a vertex v with $\mathrm{indeg}_c(v) = 0$, and a target is one with $\mathrm{outdeg}_c(v) = 0$. We will henceforth restrict ourselves to the not-uncommon situation where the digraph has a unique source and a unique target, and we will call such a digraph a *network*. (If there are several sources and targets, one may reduce to the situation of a single one of each, thus. Create two new vertices, v and w, the first to be the unique source and the second the unique target. For each given source v', create an arc from v to v' and assign it a capacity greater than or equal to the sum of the capacities of the arcs leaving v'. Similarly, for each given target w', create an arc from w' to w with capacity at least equal to the sum of the capacities of the arcs entering w'.

Definition of a flow

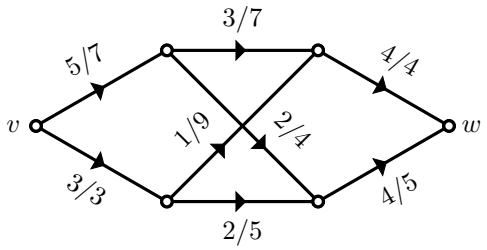
Let (G, c) be a network, and let its source and target be v and w respectively. A *flow* on (G, c) is a weight function ϕ on G such that

(i) $\phi(a) \leq c(a)$ for all $a \in A(G)$, and

(ii) $\mathrm{indeg}_\phi(u) = \mathrm{outdeg}_\phi(u)$ for all $u \in V(G) \setminus \{v, w\}$.

We will typically refer to out-degrees and in-degrees in a flow as *out-flows* and *in-flow*.

It is customary to indicate the flow and capacity on the arcs of a digraph in the format x/y where x is the value of the flow at the arc and y the value of the capacity, thus:



Here, the source is v and the target w.

Proposition 6.4 *If ϕ is a flow on a network (G, c), and if v and w are the source and target respectively, then*

$$\text{outdeg}_\phi(v) = \text{indeg}_\phi(w).$$

Proof. Remembering that $\text{outdeg}_\phi(w) = \text{indeg}_\phi(v) = 0$, we have

$$\begin{aligned}\text{outdeg}_\phi(v) &= \sum_{u \in V(G)} \text{outdeg}_\phi(u) - \sum_{u \in V(G) \setminus \{v,w\}} \text{outdeg}_\phi(u) \\ &= \sum_{u \in V(G)} \text{indeg}_\phi(u) - \sum_{u \in V(G) \setminus \{v,w\}} \text{indeg}_\phi(u) = \text{indeg}_\phi(w).\end{aligned}$$

Note that the second equality holds because of Lemma 6.3 and the fact that the in-flow at a vertex u is equal to the out-flow as long as $u \notin \{v, w\}$. ■

In light of Proposition 6.4, we may define the *value* of a flow to be either the sum of the out-flows at the source or the sum of the inflows at the target, both sums being equal.

A set of arcs in a network is called a *cut* if every directed path from source to sink includes at least one of those arcs. In the previous example, the two arcs emanating from v form a cut.

The *capacity* of a cut is the sum of the capacities of the arcs in the cut. The cut comprising the two arcs emanating from v, in the example above, has capacity $7 + 3 = 10$.

If P is a path from source to target in some network (G, c), and if ϕ is a flow on (G, c), let $\phi(P) = \min\{\phi(a) \mid a \text{ is an arc of } P\}$, the flow through P.

Lemma 6.5 *If (G, c) is a network, and if ϕ is a flow on (G, c) with positive value, then there is a path P from source to target such that $\phi(P)$ is at least the minimum flow among all non-zero arc flows in G.*

Proof. If a cycle of positive flow x exists, subtract x from each of the arcs in the cycle. Doing so does not change the value of the flow on the digraph, because neither v nor w is in a cycle. Repeat this step until no cycles of positive flow remain.

Now start at v and choose an arc of positive flow, which exists by assumption. The vertex v_1 at the other end of the arc has positive inflow and therefore has positive outflow as well, so we may choose another arc of positive flow, this time leaving v_1. Continuing in this way produces a sequence of arcs of positive flow. No vertex can be repeated, because we have replaced the original flow with one having no positive-flow cycles, so the sequence of arcs must eventually arrive at w. The resulting path from v to w has flow equal to the minimum flow of its arcs, and all such arcs have positive flow because the path does. ■

For flows ϕ on a network, define a positive real number m_ϕ recursively as follows. If ϕ is the zero flow, let $m_\phi = 0$. Otherwise, let $\mathcal{P}$ denote the set of paths from source to target with $\phi(P) > 0$, and for each $P \in \mathcal{P}$, let ϕ_P be the flow obtained by subtracting $\phi(P)$ from every arc in P. Then define m_ϕ to be the minimum of the set

$$\{\phi(a) \mid a \in A(G) \text{ and } \phi(a) > 0\} \cup \{m_{\phi_P} \mid P \in \mathcal{P} \text{ and } \phi_P \text{ is not the zero flow}\}.$$

Induction on the number of arcs of positive flow shows that m_ϕ is positive for every non-zero flow ϕ. We note as well that, simply by definition, $m_{\phi_P} \geq m_\phi$ for every $P \in \mathcal{P}$ such that ϕ_P is not the zero flow.

Proposition 6.6 *Let (G, c) be a network. If C is a cut and ϕ a flow, then the value of ϕ is at most the capacity of C.*

Proof. If ϕ is a flow on some (G, c), let $|\phi|$ denote the value of ϕ, and if A is a set of arcs, let $\Sigma_\phi(A) = \sum_{a \in A} \phi(a)$.

Fix a positive real number M. If $n \in \mathbb{Z}_{\geq 0}$, let $P(n)$ be the assertion that if ϕ is a flow on (G, c) with $m_\phi \geq M$ and $|\phi| \leq nM$, then $|\phi| \leq \Sigma_\phi(C)$ for every cut C of G. We prove $P(n)$ by induction.

The case $n = 0$ is vacuous, so now let $n \geq 1$ and assume $P(n-1)$. Let ϕ be a flow on (G, c) such that $m_\phi \geq M$ and $|\phi| \leq nM$. Suppose also that C is a cut in G. If $|\phi| = 0$, there is nothing to show. Otherwise, by Lemma 6.5, there is a path P from source to target such that

$$\phi(P) \geq \min\{\phi(a) \mid a \in A(G) \text{ and } \phi(a) > 0\} \geq m_\phi \geq M.$$

For brevity, let $x = \phi(P)$.

We consider two cases.

Case (i): ϕ_P is the zero flow

Because of the way ϕ_P is constructed, the only possibility in this case is that $\phi(a) = x$ for all arcs a in P, and $\phi(a) = 0$ for all other arcs in G. But then $|\phi| = x$ and $\Sigma_\phi(C) \geq x$ for all cuts C, so $|\phi| \leq \sum_\phi C$, as desired.

Case (ii): ϕ_P is not the zero flow

Let C' be the set of arcs in C that are also in P. Observe that $\#C' \geq 1$, because C, being a cut, contains at least one arc in P. Now,

$$|\phi_P| = |\phi| - x \leq |\phi| - M \leq nM - M = (n-1)M,$$

and of course $m_{\phi_P} \geq m_\phi \geq M$, so we may apply the inductive hypothesis to see that

$$|\phi_P| \leq \Sigma_{\phi_P}(C) = \Sigma_\phi(C) - x\#C',$$

so

$$|\phi| = |\phi_P| + x \leq \Sigma_\phi(C) - x\#C' + x$$

$$= \Sigma_\phi(C) - x(\#C' - 1)$$
$$\leq \Sigma_\phi(C),$$

completing the induction.

Now, let ϕ be any non-zero flow on (G, c). We may apply the just-proven fact in the situation where M is any positive real number less than or equal to m_ϕ (such as m_ϕ) itself and n is any positive integer such that $|\phi| \leq nM$. The conclusion is that $|\phi| \leq \Sigma_\phi(C)$ for every cut C. We are thus done because of the simple observation that $\Sigma_\phi(C)$ is, by definition of a flow, less than the capacity of C. ■

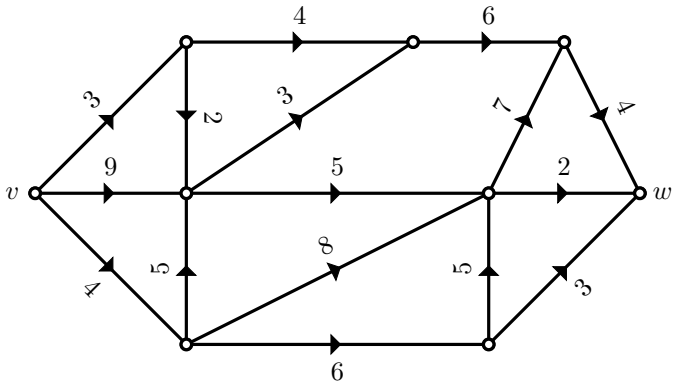
Given any weighted digraph, there are always a flow and a cut that attain the bound in Proposition 6.6.

Theorem 6.7 (Max flow, min cut) *Let (G, c) be a network. Then there are a flow ϕ on (G, c) and a cut A such that the value of ϕ is equal to the capacity of A. Thus, the maximum possible flow is equal to the minimum possible cut.*

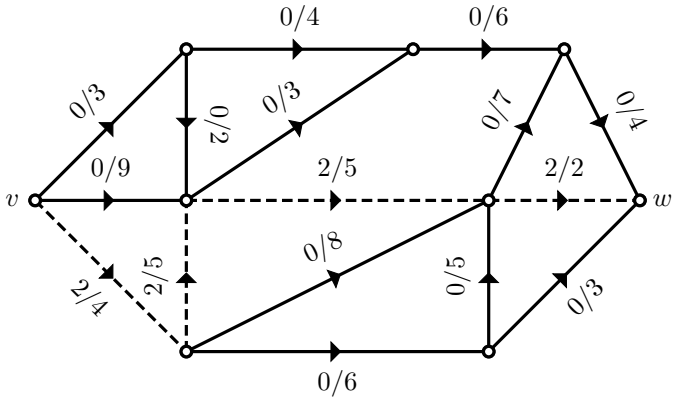
We refer the reader to Wilson for a proof.

Ford and Fulkerson described a method for finding a maximum flow in a network. The idea is to begin with the zero flow on the network and seek directed paths from source to target that increase the flow. Such a path is called an *augmenting path*. Let us illustrate by example.

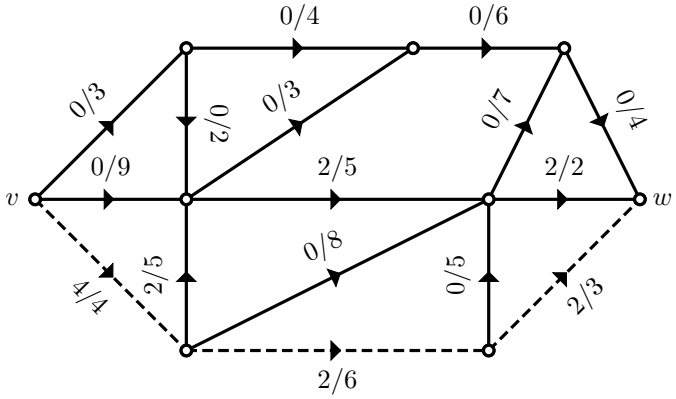
Example 6.8 Consider the following network, with source v and target w, in which the capacities are indicated:



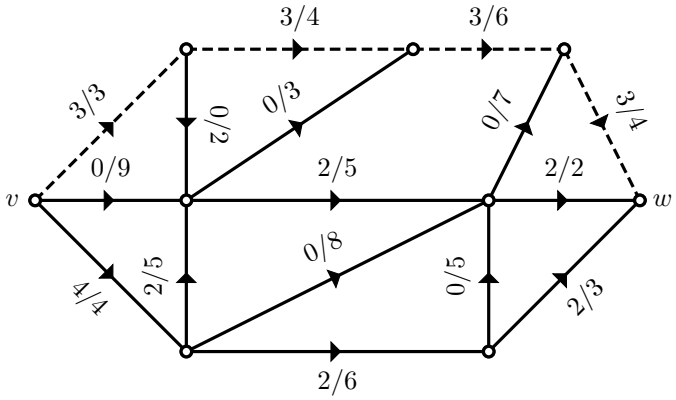
We look for a directed path in which the flow can be increased without exceeding the capacity of any arc in the path. One example is this, in which the dashed path has a flow of 2:



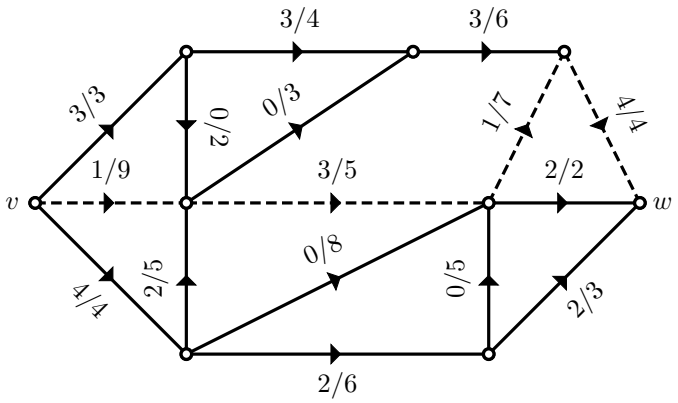
We cannot increase the flow along that path any further, because its final arc is at capacity. However, there is still capacity to increase the flow along the bottom path:



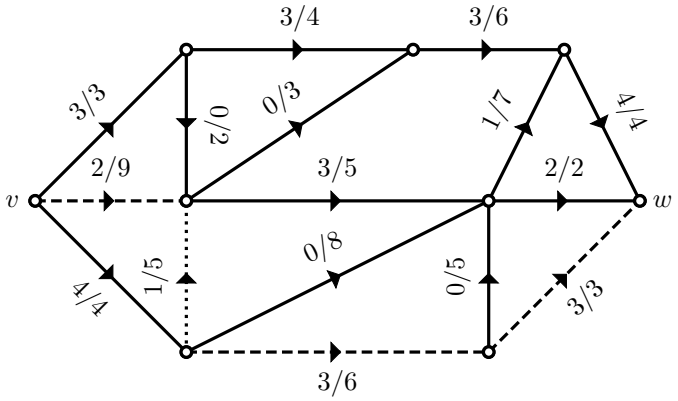
The first arc in this path is now at capacity, so we look for another arc from v to increase the flow through:



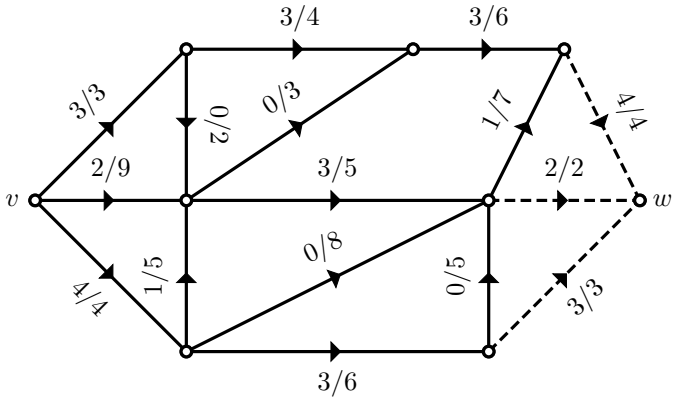
Again, we have saturated another arc from v, so we go to the final one. We observe that there is a directed path whose flow can be increased by 1:



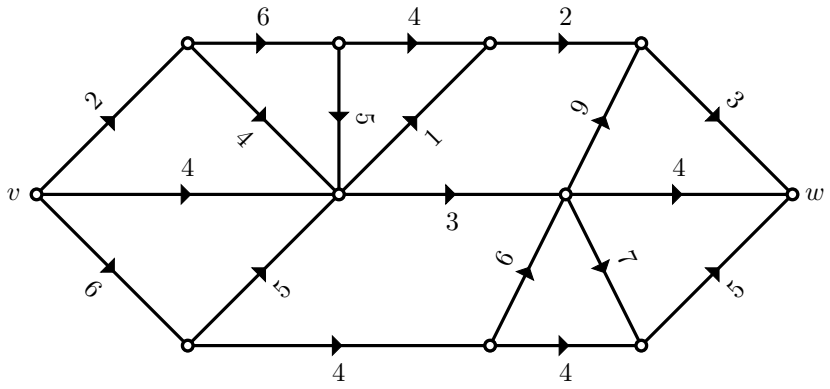
At this point, it appears as though we are stuck. If we simply follow the arrows in their directions, then no directed path can have its flow increased. However, observe that any arc that has a positive flow can have that flow reduced. In particular, we can increase 1/9 to 2/9 out of v, then decrease 2/5 to 1/5 in the dotted arc below, and from there increase the flow along the bottom:



The value of the resulting flow (remember, its value is the outflow at v and the inflow at w) is now 9. In fact, this is a maximum flow, because no flow can exceed the capacity of any cut, and a cut of capacity 9 exists:



Exercise 6.9 Find a maximum flow and a minimum cut in the following network:



Solution: