Graph Theory

The Definition of a Graph

A *graph* is an ordered pair $G = (V, E)$, where V is a finite, non-empty set of objects called **vertices**, and E is a (possibly empty) set of unordered pairs of distinct vertices *i.e.*, 2-subsets of V called **edges**.
The set V (or $V(G)$ to emphasize that it belongs to the graph G) is called the **vertex set** of G, and E (or $E(G)$ to emphasize as above) is called the **edge set** of G.

If $e = \{u, v\} \in E(G)$, we say that vertices u and v are *adjacent* in G, and that e *joins* u and v. We'll aslo say that u and v are the *ends* of e. The edge e is said to be *incident* with u (and v), and vice versa. We write uv (or vu) to denote the edge $\{u, v\}$, on the understanding that no order is implied.

Notes: (1) $E(G)$ is a set. This means that two vertices either are adjacent or are not adjacent. There is no possibility of more than one edge joining a pair of vertices.

(2) The elements of E are 2-subsets of V. Thus a vertex can not be adjacent to itself.

There are more general definitions in which there may be more than one edge between two vertices, and/or vertices may be adjacent to themselves. Sometimes these go by the names *multigraphs* or *pseudographs*. Sometimes authors intend this more general situation when they say "graphs", and the objects we have defined above are then usually called *simple graphs*.

Representing Graphs

By definition, a graph is a pair of sets. Graphs are usually represented pictorially with a point (or dot) in the plane corresponding to each vertex and a line segment (or curve of some sort) joining the corresponding points for each pair of adjacent vertices. The picture tells you what the graph is, that is, it tells you what the vertices are, and what the edges are.

Another way to represent this information, and a reasonable way to do so in a computer program, is with an *adjacency matrix*. Suppose $V(G) = \{v_1, v_2, \ldots, v_n\}$. Then the **adjacency matrix** of G is the $n \times n$ matrix A whose (i, j) entry, A_{ij}, is 1 if $v_iv_j \in E$ and 0 if $v_iv_j \notin E$. That is A_{ij} is the truth value of the statement "v_i and v_j are adjacent". Since v_iv_j is the same as v_jv_i, the matrix A is symmetric. Since no vertex can be adjacent to itself, the diagonal entries of A are zeros.

Graph Isomorphism

Two graphs are *equal* if they have the same vertex set and the same edge set. But there are other ways in which two graphs could be regarded the same. For example, one could regard two graphs as being "the same" if it is possible to rename the vertices of one and obtain the other. Such graphs are identical in every respect except for the names of the vertices. In this case, we call the graphs *isomorphic*.

Formally, graphs G and H are **isomorphic** if there is a 1-1 correspondence

$$f : V(G) \to V(H) \text{ such that } xy \in E(G) \Leftrightarrow f(x)f(y) \in E(H).$$

The function f is called an **isomorphism**.

The relation $\mathcal{R}$ on the set of all graphs defined by $G\mathcal{R}H$ if and only if G and H are isomorphic (*i.e.*, the vertices of G can be renamed so as to obtain H) is an equivalence relation, and the equivalence classes are collections of graphs which are "the same" in this sense. We frequently draw pictures of graphs without labelling the vertices. These *unlabelled graphs* are understood to represent any of the possible graphs obtained by giving names to the vertices.

It is **important** to recognise that isomorphic graphs are identical in every respect other than the names given to the vertices. To prove two graphs are *not* isomorphic, find something that is true about one of the graphs that is not true about the other. (Something that does not involve the vertex names.) One way to prove two graphs are isomorphic is to relabel the vertices of one and obtain the other.

The *complete graph* on vertex set $\{v_1, v_2, \ldots, v_n\}$ is the graph (with this vertex set) in which every pair of distinct vertices are adjacent. Any two such graphs on the same number of vertices are easily seen to be isomorphic. For this reason we talk about **the complete graph on n vertices** and denote it by K_n.

Vertex degrees

The **degree** of a vertex $v \in V$, denoted by $\deg(v)$ (or $\deg_G(v)$), is the number of vertices adjacent to v.

Since vertex degrees count adjacencies, $\deg(v) \geq 0$ for any vertex v. If G has n vertices, then any vertex v can be adjacent to at most $n-1$ vertices (all vertices except itself), so $\deg(v) \leq n-1$.

Fact 1. *Let $G = (V, E)$ be a graph. Then $\sum_{v \in V} \deg(v) = 2|E|$.*

To prove Fact 1, notice that every edge xy contributes two to the sum: one to $\deg(x)$ and one to $\deg(y)$.

It is **important** to recognise that Fact 1 tells you that the sum of the vertex degrees is even. Thus, a corollary to Fact 1 is that in any graph *the number of vertices of odd degree is even.*

A graph is **r-regular** (or *regular of degree r*) if every vertex has degree r. A graph is regular if it is r-regular for some r.

A consequence of Fact 1 (or, its corollary) is that there does not exist an r regular graph on n vertices if r and n are both odd. The contrapositive is that if there exists an r regular graph on n vertices, then r or n is even (or both). It turns out that such graphs exist in all such cases.

Fact 2. *If a graph G has at least two vertices, then it has two vertices of the same degree.*

Fact 2 is often called the *Handshaking Lemma*. This name is derived from using a graph to model which pairs among a group of n people at a gathering shake hands (*i.e.*, $xy \in E$ if and only if person x and person y shook hands). It is always the case that two people in the group shook the same number of hands (that's what the Fact says). The proof uses the Pigeonhole Principle.

A **degree sequence** of a graph G with n vertices is a sequence of length n whose elements are the degrees of the vertices of G (in some order).

Notice that under this definition a graph could have many degree sequences. It is also possible for different (in the sense of being non-isomorphic) graphs to have the same degree sequence.

A sequence $d_1, d_2, \ldots, d_n$ of non-negative integers is called **graphical** if it is the degree sequence of some graph.

It follows from Fact 1 above that if a sequence of non-negative integers is graphical, then the sum of the elements in the sequence is even. It is also true that the largest element in the sequence must be at most $n-1$, and that $n-1$ and 0 can not both occur in the sequence. These conditions, however, do not characterise the graphical sequences. One characterisation is given by the following theorem.

Havel-Hakimi Theorem. *A sequence $d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$ of non-negative integers is graphical if and only if the sequence $d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, \ldots, d_n$ is graphical.*

The theorem can be applied recursively to determine if the shorter sequence is graphical. The base case is that the sequence 0 is graphical. If at any point the sum of the elements in the sequence is odd, or one of the elements is negative, then the sequence is not graphical. Note that the shorter sequence may need to be sorted before the Theorem can be re-applied. Here it is important to note that the subscripts on the d_i's are only there to index the sequence, it is **not** being assumed that the vertices are $\{1, 2, \ldots, n\}$ and that d_1 is the degree of vertex 1, etc.

The theorem says that if the given sequence is graphical, then it can be realised by a graph in which a vertex of degree d_1 is adjacent to vertices of degree $d_2, d_2, \ldots, d_{d_1+1}$. (It is possible that it can also be realised by many other graphs.) This gives a construction. Once the $n-1$ vertex graph with degree sequence $d_2-1, d_3-1, \ldots, d_{d_1+1}-1, d_{d_1+2}, d_{d_1+3}, \ldots, d_n$ has been constructed, add a new vertex and join it to vertices of degree $d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1$ (in the $n-1$ vertex graph).

Subgraphs

A graph H is a **subgraph** of a graph $G = (V, E)$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

That is, a subgraph of G is a graph, all of whose vertices are vertices of G, and all of whose edges are edges of G. It is important that a subgraph if a graph in its own right. Thus, every edge of a subgraph joins two vertices belonging to the subgraph.

You can think of a subgraph of G as being obtained from G by first selecting some (non-empty collection) of the vertices of G to belong to the subgraph, and then selecting some of the edges of G joining vertices in this collection to be edges of the subgraph. This is consistent with the usual use of the term "sub": here we have a graph which is "contained" inside of G.

A **spanning** subgraph of a graph G is a subgraph H of G such that $V(H) = V(G)$. (Think: the vertex set of H spans the vertex set of G in the sense that all vertices of G are in $V(H)$.)

Note: Any graph with vertex set $\{1, 2, \ldots, n\}$ is a subgraph of K_n. Thus, the number of *labelled* graphs on n vertices equals the number of spanning subgraphs of K_n, which is $2^{\binom{n}{2}}$ (for each edge there are two choices: in the subgraph, or not in the subgraph). The term *labelled* means the names of the vertices is important. Many of these graphs are isomorphic. For instance, there are $2^3 = 8$ labelled graphs on 3 vertices, but only 4 non-isomorphic (or, *unlabelled*) graphs on 3 vertices.

The **complement** of a graph G, denoted $\overline{G}$ is the graph with the same vertex set as G, and where distinct vertices x and y are adjacent in $\overline{G}$ if and only if they are not adjacent in G. You can think of $\overline{G}$ as being obtained from the complete graph on $|V(G)|$ vertices by deleting the edges that belong to G. It is not hard to prove that if G and H are isomorphic, then so are $\overline{G}$ and $\overline{H}$. And, since $\overline{\overline{G}} = G$, the converse is also true. A graph is *self-complementary* if it is isomorphic to its complement. Self complementary graphs exist if and only if the number of vertices is congruent to 0 or 1 modulo 4.

Walks, Trails, Paths, and Cycles

A **walk** (or, $v_0 - v_n$ walk) in a graph is an alternating sequence of vertices and edges, $v_0, e_1, v_1, e_2, v_2, e_3, v_3, \ldots, e_n, v_n$ such that $e_i = v_{i-1}v_i$ for $1 \leq i \leq n$. The integer n is the **length** of the walk. It is the number of edges in the walk, one less than the number of vertices.

You can think of a walk in a graph G as the result of the process of starting at a vertex of G, physically walking along the edges of G (no turning around half way!), and recording the names of the vertices encountered (including the start and end), and the edges used. Thus the length equals the number of edges you'd travel over. According to our definition of a graph there is at most one edge between any two vertices. Thus, recording the edges in a walk amounts to recording redundant information. (However, in a more general situation where there can be more than one edge between two vertices it would be important to record which edge was used.) In our situation, an **equivalent definition of a walk** is a sequence of vertices $v_0, v_1, \ldots, v_n$ such that $v_{i-1}v_i \in E$, $1 \leq i \leq n$. It is then implied which edges belong to the walk. **We'll use this definition**.

Note. If $x \in V(G)$, then x is a perfectly good $x - x$ walk.

In a walk there is no requirement that vertices and/or edges not be repeated. There are special terms to designate the situations in which they are not.

A **trail** is a walk in which no edge is repeated. A **path** is a walk in which no vertex is repeated.

Thus, every path is a trail (to repeat an edge you must repeat one of its ends) and every trail is a walk. There are walks which are not trails, and trails which are not paths. But, the following statment is true.

Fact 3. *If a graph G has an $x-y$ walk, then it has an $x-y$ path.*

One proof of Fact 3 is algorithmic: if you have a walk that's not a path, then you can chop out a part between a repeated vertex (there must be one!) and get a shorter walk. If you keep doing this, you will eventually remove all repeated vertices (the process terminates because the length of the walk is finite) and be left with a path. A more slick argument does all the chopping out beforehand. Consider a shortest $x-y$ walk and argue that it can not have any repeated vertices and is therefore a path.

A **closed walk** is a walk that starts and ends at the same vertex. Similarly, a **closed trail** is a trail that starts and ends at the same vertex. Since each of these is a (special case of a) walk, it is meaningful to talk about the length.

Note: By its definition a path can not be closed – it is impossible for the vertices to be distinct and the first and last vertex to be the same!

A **cycle** is a closed walk of length at least three in which the vertices are distinct except the first and last.

The restriction on the length is necessary to prevent the sequence consisting of a single vertex from being a cycle of length zero. We could have said "trail" instead of "walk" in the above definition, and nothing would have changed. A closed trail of length at least three (which may have repeated vertices) is often called a *circuit*. There are circuits which are not cycles, but the following is true.

Fact 4. *If a graph G has a closed trail of positive length containing a vertex v, then it has a cycle containing v.*

The proof can be carried out in two ways similarly to Fact 3. The quickest argument is that a shortest closed trail of positive length containing v must be a cycle.

Note: it is not true that of a graph contains a closed walk, then it contains a cycle.

A graph is **connected** if there exists a $u-v$ walk for each pair of vertices u and v; otherwise it is *disconnected*. A (connected) *component* of a graph G is a maximal connected subgraph of G.

Notes. (1) *maximal* is with respect to inclusion. An object is maximal with respect to some property if it is not contined in a larger object with the same property. Here, we mean that no component is "contained in" (*i.e.*, a subgraph of) some other subgraph having more vertices and/or edges. By contrast, "maximum" is with respect to size. Something that is maximum is the largest, according to some criteria. A maximum object is maximal, but not the other way around.

(2) A graph is connected if and only if it has only one component, and disconnected if and only if it has at least two components.

(3) The relation $\sim$ on $V(G)$, defined by $u \sim v$ if there exists a $u-v$ walk, is an equivalence relation. The equivalence classes are the components of G.

(4) Because of Fact 3, the term "walk" in the definition of connected could be (and often is) replaced by "path". The same is true for note (3) immediately above.

Notation: If G is a graph and xy is an edge of G, we use $G - xy$ to denote the graph obtained from G by deleting the edge xy. Formally, this graph has vertex set $V(G)$, and edge set $E(G) - \{xy\}$.

An edge $xy \in E(G)$ is a **cut edge** (or bridge) if $G - xy$ has more components than G.

Thus, a cut edge of a connected graph is an edge whose removal disconnects the graph. Not every graph has such an edge.

Fact 5. *An edge xy of a graph G is a cut edge if and only if it is not contained in a cycle.*

Fact 5 characterises the edges that are cut edges. Thus, to demonstrate that e is not a cut edge it is enough to give a cycle containing e. To demonstrate that e is a cut edge, show that e is not in a cycle. Or, use the definition and show that $G - e$ has more components that G

To prove Fact 5, argue the contrapositive. An edge whose removal disconnects the graph can not be in a cycle, and (conversely) an edge whose removal does not disconnect the graph must be contained in a cycle.

Another way of stating Fact 5 is *G has no cut edge if and only if every edge of G is in a cycle.* Another equivalent statement is *An edge xy of a graph G is a cut edge if and only if there exist distinct vertices u and v, such that every $u - v$ path contains xy.*

The vertex analogue of a cut edge is a *cut vertex*: a vertex whose deletion increases the number of components (of the graph). Not every graph has a cut vertex (but, every graph has several vertices which are guaranteed not to be cut vertices). A graph with a cut vertex may or may not have a cut edge, but every graph with at least three vertices and a cut edge has a cut vertex.

Trees

A **tree** is a connected graph that has no cycles (*i.e.*, a connected acyclic graph).

A **forest** is an acyclic graph. That is, a forest is a graph in which each component is a tree.

A **spanning tree** of a graph is a spanning subgraph which is a tree.

Fact 6. *If G is a connected graph, then G has a spanning tree.*

To prove Fact 6, repeatedly apply Fact 5. If G is connected and has a cycle then, for any edge xy in the cycle, $G - xy$ is connected. The process eventually terminates in an acyclic graph because G has a finite number of edges.

Fact 6 implies that every connected graph with n vertices has at least $n - 1$ edges. The converse is false.

A *leaf* (or, pendant vertex) of a tree is a vertex of degree one.

Fact 7. *If T is a tree with at least two vertices, then T has at least two leaves.*

To prove Fact 7, start by considering a longest path and argue that the first and last vertex in the path must have degree 1.

Notation. If G is a graph and x is a vertex of G, then $G - x$ is the graph obtained from G by deleting the vertex x and all edges incident with x.

Fact 8. *If T is a tree and x is a leaf of T, then $T - x$ is a tree.*

To prove Fact 8, you must argue that $T - x$ satisfies the definition of a tree. It is acyclic because any cycle in $T - x$ is a cycle in T. It is connected because a vertex x of degree 1 can not occur on a path between two other vertices.

Fact 8 is the key to many proofs of facts about trees. In a proof by induction, one can often remove a leaf, argue that the smaller tree has the desired property (use the induction hypothesis), and argue from there that the whole tree has the desired property. A typical example of such an argument is used to prove:

Fact 9. *A tree with n vertices has exactly $n - 1$ edges.*

There are a number of statements which are equivalent to a graph being a tree. Many of these are summarised in the theorem below. The notation $T + xy$ in statement (c) means the graph obtained from T by adding an edge joining the vertices x and y (of T).

Theorem 10. *The following statements are all equivalent:*
(a) T is a tree.
(b) T is connected and every edge of T is a cut edge.
(c) T is acyclic, and if $xy \notin E$ then $T + xy$ contains a unique cycle.
(d) T is acyclic and $|E| = |V| - 1$.
(e) T is connected and $|E| = |V| - 1$.
(f) Any two vertices of T are joined by a unique path.

Statements *(b)* through *(e)* say that, in the definition of a tree, the conditions "is connected" and "is acyclic" can sometimes be replaced by other conditions. Here are some ideas that can be used in the proofs:
(a) $\Leftrightarrow$ *(b)*: Note that if some edge of T is not a cut edge, the T contains a cycle.

(a) ⇔ (c): Note that every cycle in $T + xy$ must contain xy, otherwise T has a cycle. If $T + xy$ has two different cycles, then you can show that T has a cycle.
(a) ⇔ (d): If T is acyclic and not connected, then each component is a tree. Count the edges using Fact 9 and show that you get less than $|V| - 1$.
(a) ⇔ (e): Since T is connected, it has a spanning tree by Fact 6. By Fact 9, the spanning tree contains all of the edges of T, hence it is T.
(a) ⇔ (f): Argue that if there are two distinct paths from x to y (say), then T contains a cycle.

Theorem 11. *The sequence of integers $d_1 \geq d_2 \geq \ldots \geq d_n = 1$ is the degree sequence of a tree if and only if $\sum_{i=1}^{n} d_i = 2(n-1)$.*

One direction of the proof of Theorem 11 is straightforward. The degrees of the vertices in a tree must sum to twice the number of edges. The other implication can be established by induction. One way to do the induction step is to delete d_n and reduce the last element of the sequence which is greater than one (if any) by one.

Eulerian graphs

An **Euler trail** in a graph G is a trail that includes all the edges of G. A closed Euler trail is called an **Euler tour**. A graph is **Eulerian** if it has an Euler tour.

Graphs with an Euler trail correspond to figures that can be drawn without lifting one's pencil from the paper, or tracing over any line segnemt more than once. Similarly, they can be viewed as corresponding to the figures that can be drawn with an "etch-a-sketch" (without drawing any line segment more than once, but of you've played with one of these you know this statement is essentially redundant).

Theorem 12. *A graph G has an Euler trail if and only if it is connected and has 0 or 2 vertices of odd degree.*

To prove that if G has an Euler trail then it has 0 or 2 vertices of odd degree, note that each time an Euler trail passes through a vertex, it accounts for exactly two edges incident with that vertex. Thus, the only vertices which can have odd degree are the first and last vertices of the trail, and the number of vertices of odd degree is even. To prove the converse, consider a longest trail and argue that no edges can be omitted. Induction helps. See below.

Theorem 12 gives an easy criteria for proving that a graph has/doesn't have an Euler trail. If it does, describe the trail, and if it doesn't then show that it (the graph) is disconneced or has more than 2 vertices of odd degree. The proof also gives an algorithm for constructing the trail when it exists. Start at a vertex of odd degree (or any vertex if they all have even degree), and construct a maximal trail. If it contains all of the edges in the graph, then you're done. Otherwise, delete all of the edges in your trail. In each component of the resulting graph, all vertices will have even degree. The graph is necessarily disconnected because the trail contains all edges incident with its last vertex (otherwise it would not be maximal!). There will be a last vertex w on the trail which belongs to a component that has edges. This component has an Euler tour starting and ending at w, and this "little tour" can be "added in" to your trail to give a longer trail. The little tour can be be found using the same procedure (or, assumed to exist by induction). Repeating this process for each extended trail eventually results in an Euler trail of the graph.

It follows from the proof of Theorem 12 that if there are two vertices of odd degree, then the Euler trail starts at one of them and ends at the other. And, if there are no vertices of odd degree, then the Euler trail starts and ends at the same vertex. (So, it is a tour.) That is, *a graph is Eulerian if and only if the degree of every vertex is even.*

Hamilton Paths and Cycles

A path that contains every vertex of a graph G is called a **Hamilton path**. A cycle that contains every vertex of G is called a **Hamilton cycle**. A graph is called **hamiltonian** if it has a Hamilton cycle.

Notice that a graph with a Hamilton cycle has a Hamilton path. There are graphs with a Hamilton path which have no Hamilton cycle. There is no direct relationship between graphs with a Hamilton cycle and those with an Euler tour, even though the former is a closed trail that contains every vertex exactly once, and the latter is a closed trail that contains every edge exactly once. Similarly for graphs with a Hamilton path compared to those which have an Euler trail.

By contrast to Eulerian graphs, there is no known theorem that completely characterises the hamiltonian graphs. There are theorems which one can sometimes apply to determine that a graph is Hamiltonian. Most of these assert that graphs with "enough" edges have a Hamilton cycle (and most of these can be adapted to give conditions under which a graph has a Hamilton path). An example is Ore's theorem: *If* $\deg_G(x) + \deg_G(y) \geq |V|$ *for every two non-adjacent vertices* x *and* y, *then* G *is hamiltonian.* There are also theorems that can sometimes be applied to assert that a graph is not hamiltonian. For example, notice that if any set of $k < n$ vertices (and the edges incident with them) is deleted from a cycle of length n, then the resulting graph has at most k components. Therefore, *if there is a proper subset* $S \subset V(G)$ *such that deleting the vertices in* S *(and the edges incident with them) leaves a graph with more than* $|S|$ *components, then* G *is not hamiltonian.* Neither of these results works in all possible circumstances. For example, the graph which is a cycle of length $n \geq 5$ is Hamiltonian, but one can not deduce this from Ore's Theorem. Also, the Petersen Graph (look it up) is not Hamiltonian, but one can not deduce this by applying the result above.

The n**-cube**, Q_n, is the graph whose vertex set is the set of bit strings of length n, and in which two vertices are adjacent if and only if they (remember, *they* are bit strings) differ in exactly one bit.

Fact 13. *For all $n \geq 2$, the graph Q_n is Hamiltonian.*

One proof of Fact 13 is by induction. The key observation is that Q_{n+1} can be viewed as being formed from two copies of Q_n by adding edges joining corresponding vertices (*i.e.*, copies of the same vertex). This makes it possible to use a Hamilton cycle in Q_n to obtain a Hamilton cycle in Q_{n+1}: follow a Hamilton path in Q_n obtained by omitting the last edge of a Hamilton cycle, jump to the other copy of Q_n and follow the same path in reverse, then return to the vertex where you started (these two vertices are adjacent).

A Hamilton cycle in Q_n is a listing of the 2^n bit strings of length n in such a way that consecutive strings in the list differ in exactly one bit, including the first and last strings. Such a list is called a *Gray code.*

Colouring

A k**-colouring** of a graph G is an assignment of k colours $\{1, 2, \ldots, k\}$ to the vertices of G so that adjacent vertices are assigned different colours. A graph G is called k**-colourable** if there exists a k-colouring of G. The **chromatic number** of a graph G, denoted $\chi(G)$, is the smallest k such that G is k-colourable.

Notes: (1) There is no requirement in the definition that all colours be used. Thus, a graph which is k-colourable is t-colourable for every $t \geq k$.

(2) Since $V \neq \emptyset$, colouring a graph requires that at least one colour be used. Every graph with n vertices is n-colourable: assign a different colour to every vertex. Hence, there is a smallest k such that G is k-colourable. That is, $\chi(G)$ is well-defined.

(3) A graph has $\chi = 1$ if and only if it has no edges.

A graph G is called **bipartite** if $V(G)$ can be partitioned into subsets V_1 and V_2 so that every edge joins a vertex in V_1 and a vertex in V_2. (Think: bi-partite means two parts, with all edges going between the parts.)

The **complete bipartite graph** $K_{m,n}$ is the bipartite graph where the sets in the vertex partition have size m and n, and any two vertices belonging to different sets (in the partition) are adjacent. We assume m and n are both at least 1.

Since the vertices of a cycle in a bipartite graph must alternate between the two sets in the vertex partition, the length of the cycle must be even. The converse is also true, and this characterises bipartite graphs.

Theorem 14. *A graph is bipartite if and only if it contains no cycles of odd length.*

Theorem 14 gives a method for demonstrating that a graph is/is not bipartite. To show a graph is bipartite, give the partition of the vertex set. To show that a graph is not bipartite, describe an odd cycle.

If a graph is bipartite, it is easy to find the vertex partition. Repeat the following for each component. Pick a vertex x and put it in V_1. All vertices adjacent to x must belong to V_2

since all edges have one end in V_1 and the other in V_2. All vertices adjacent to these must be in V_1, and so on.

If the above procedure results in V_1 or V_2 containing a pair of adjacent vertices, then there is an odd cycle. You can find it by tracing the procedure backwards.

By Theorem 14, and the definition of a tree, every tree is a bipartite graph. Since there are no cycles at all, there are no odd cycles.

Notice that bipartite graphs are 2-colourable: assign colour 1 to the vertices in V_1 and colour 2 to the vertices in V_2. The converse is also true since, by definition of a k-colouring, no edge joins two vertices of the same colour.

The procedure described above works for finding a 2-colouring, if one exists. Colour a vertex 1 if it would be assigned to V_1, and colour it 2 if it would be assigned to V_2. If the procedure assigns two adjacent vertices the same colour then there is an odd cycle (which can be found by tracing the assignments back).

Thus, a graph has $\chi = 2$ if and only if it is bipartite and has at least one edge. (Graphs with at least two vertices and no edges meet the criteria for being bipartite.) Theorem 14 therefore gives a method for proving that a graph is/is not 2-colourable, or has/doesn't have chromatic number 2.

By contrast to the above, there is no characterisation of graphs with $\chi = k$ for any k greater than or equal to 3. The best we can do is use Theorem 14 to give a certificate that a graph has chromatic number 3: *A proof that* $\chi(G) = 3$ *consists of a 3-colouring of* G*, and an odd cycle in* G. The latter demonstrates that $\chi > 2$.

The following fact gives an upper bound for the chromatic number of a graph in terms of the maximum degree of a vertex in the graph. Its proof can be implemented to give a recursive algorithm to provide a colouring with the advertised number of colours.

Fact 15. *Let* G *be a graph with maximum degree* Δ*. Then* $\chi(G) \leq \Delta + 1$.

The proof is by induction on the number of vertices. In the induction step, delete a vertex x of maximum degree. The resulting graph has maximum degree at most Δ, so by the induction hypothesis it can be coloured with at most $\Delta + 1$ colours. Considering these vertices as being coloured in G, it remains only to colour x. But, at most Δ colours can be used on vertices adjacent to x, so one of the $\Delta + 1$ colours can be assigned to x.

Notice that a graph in which all vertices have large degree need not have large chromatic number. For example, in the complete bipartite graph $K_{n,n}$ every vertex has degree n and the graph has chromatic number 2.

On the other hand, Fact 15 says that a graph in which all vertices have "small" degree also has "small" chromatic number. For example, 4-regular graph has chromatic number at most 5, no matter how many vertices it has.

The bound in Fact 15 is attained only for complete graphs and graphs which are cycles of odd length. Graphs in these families have chromatic number equal to $\Delta + 1$, where Δ

is the maximum degree. That these are the only graphs with chromatic number equal to the maximum degree plus one (and all other graphs have chromatic number at most Δ) is known as *Brooks' Theorem*.

Planar graphs

A graph is called *planar* if it can be drawn in the plane so that edges intersect only at their ends (if at all). A *planar embedding* of a planar graph is such a drawing (embedding) of the graph in the plane. A planar embedding of a planar graph partitions the plane into a number of connected regions called *faces*. (Here, "connected" means that any two points in the region can be joined by a curve that does not touch an edge or vertex.) Notice that one of the faces is always infinite.

Euler's formula. *Let G be a connected planar graph. Then, in any planar embedding of G, $|V| - |E| + f = 2$, where f is the number of faces in the embedding.*

One proof of Euler's formula is by induction on the number of edges. Since the graph is connected, it has at least $|V| - 1$ edges (by Fact 6). For the base case, when $|E| = |V| - 1$, notice that G is a tree and a planar embedding of a tree has only one face because a tree has no cycles. In the induction step, since the graph under consideration has $|E| \geq |V|$, it has a cycle. Removing an edge from this cycle (and the graph) results in the number of edges and the number of faces both being decreased by one. (The edge separated the two faces.) The graph is still connected by Fact 5. The result then follows from applying the induction hypothesis.

Euler's formula implies that two planar embeddings of a planar graph G have the same number of faces. A planar graph can have many (fundamentally) different embeddings, however. (Here "fundamentally" means more substantially different than curving an edge or moving a vertex by a little bit.)

The **degree of a face** x, denoted $\deg(x)$, in a planar embedding of a graph G equals the number of edges incident with (*i.e.* touching) x.

Fact 16. *In any planar embedding of a planar graph G, $\sum_{x \text{ a face}} \deg(x) \leq 2|E|$.*

The proof follows from the observation that every edge is incident with at most 2 faces. (Edges not in a cycle are incident with only 1 face.)

Notice that if G is a connected planar graph with at least 4 vertices (and therefore at least 3 edges), the degree of every face is at least three. Thus, $2|E| \geq \sum_{x \text{ a face}} \deg(x) \geq \sum_{x \text{ a face}} 3 = 3f$, and so $f \leq 2/3|E|$. Substituting this inequality into Euler's formula yields:

Fact 17. *For any planar graph G with $n \geq 3$ vertices has at most $3n - 6$ edges.*

The case $n = 3$ must be treated separately.

Fact 17 is important in graph algorithms because it says that for planar graphs the number of edges is a linear function of the number of vertices. Thus, any algorithm whose running time is a linear function of the number of edges in fact has a running time which is linear in the number of vertices on planar graphs.

A consequence of Fact 17 is that *every planar graph has a vertex of degree at most 5.* The proof is by contradiction (of Fact 17).

Fact 17 can be used to prove that K_5 is not planar. If K_5 were planar, then we would have $10 = |E| \leq 3 \cdot 5 - 6 = 9$, a contradiction.

One can use a similar argument to show that the complete bipartite graph $K_{3,3}$ is not planar. The key observation is that all cycles in a bipartite graph are even, so (as long as there are at least 5 vertices), each face has degree at least 4. This leads to $f \leq (1/2)|E|$, and $|E| \leq 2|V| - 4$. The last inequality can be used to obtain the desired result.

A edge xy in a graph G is **subdivided** if it is deleted and replaced by the path x, w, y, where w is a new vertex. A **subdivision** of G is any graph that results from a sequence of subdivisions of edges starting with the graph G. A subdivision of G can be viewed as being constructed from G by replacing some of the edges of G by (graphs which are) paths (of possibly differing lengths).

Notice that if G is not planar, then every subdivision of G is not planar. Therefore, no subdivision of $K_{3,3}$ or K_5 can be planar. This turns out to characterise the planar graphs.

Kuratowski's Theorem. *A graph G is planar if and only if it has no subgraph which is a subdivision of K_5 or $K_{3,3}$.*

Thus, in some sense, K_5 and $K_{3,3}$ are the only non-planar graphs.

Kuratowski's Theorem gives a method for proving that a graph is/is not planar (and leads to fast algorithms for doing so). To show that a graph is planar, give a planar embeding. To show that a graph is not planar, describe a subgraph which is a subdivision of $K_{3,3}$ or K_5.

The **Four Colour Theorem** says that *every planar graph is 4-colourable.* (The converse is false, as there are bipartite graphs which are not planar.) The known proofs of this theorem are long, difficult, and involve examination of hundreds of cases by computer. It is not difficult, however, to prove that 5 colours suffice, and easier still to show that 6 will do. The latter is an application of the technique discussed in the colouring section together with the consequence of Fact 17 mentioned above.

Fact 18. *Every planar graph is 6-colourable.*

The proof is by induction on $|V|$. In the induction step, delete a vertex x with degree at most 5. Colour the remaining vertices by induction, and extend this colouring to G as in the proof of Fact 15.