

Graphs

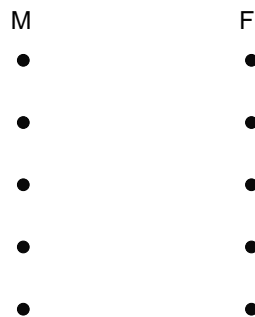
As in Week 4 Notes, we'll refer to undirected graphs simply as "graphs."

1 Counting by Degrees

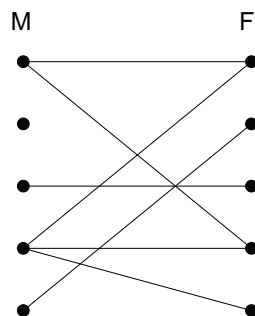
1.1 Sex in America

A 1994 University of Chicago study entitled *The Social Organization of Sexuality* found that on average men have 74% more opposite-gender partners than women, confirming a view that men are more promiscuous. But whoever reported this finding was lying or confused! We'll show you how to use the properties of vertex degrees in a graph to prove this.

Let's recast opposite-gender partnering in graph theoretic terms. Let G be a graph where the set of vertices V consists of everyone in America. Now each vertex either represents either a male or a female, so we can partition V into two subsets: M , which contains all the male vertices, and F , which contains all the female vertices. Let's draw all the M vertices on the left and the F vertices on the right:



Now, without getting into a lot of specifics, *sometimes an edge appears* between an M vertex and a F vertex:



Since we're only considering opposite-gender relationships, every edge connects an M vertex on the left to a F vertex on the right. So the sum of the degrees of the M vertices equals the number of edges and so does the sum of the degrees of the F vertices. So these sums are equal:

$$\sum_{x \in M} \deg(x) = \sum_{y \in W} \deg(y)$$

Now suppose we divide both sides of this equation by the product of the sizes of the two sets, $|M| \cdot |F|$:

$$\left(\frac{\sum_{x \in M} \deg(x)}{|M|} \right) \cdot \frac{1}{|F|} = \left(\frac{\sum_{y \in W} \deg(y)}{|F|} \right) \cdot \frac{1}{|M|}$$

The terms above in parentheses are the *average degree of an M vertex* and the *average degree of a F vertex*. So we know:

$$\frac{\text{Avg. deg in } M}{|F|} = \frac{\text{Avg. deg in } F}{|M|}$$

$$\text{Avg. deg in } M = \frac{|F|}{|M|} \cdot \text{Avg. deg in } F$$

In other words, we've proved that the average number of opposite-gender partners of males in the population compared to those of females is determined solely by the relative number of males and females in the population.

Now the Census Bureau reports that there are slightly more females than males in America; in particular $|F|/|M|$ is about 1.035. So we know that on average, males have 3.5% more opposite-gender partners than females, and this tells us nothing about any sex's predilection toward promiscuity. We can only wonder where the University of Chicago researchers got their 74% difference.

1.2 Handshaking Lemma

The previous argument hinged on the connection between a sum of degrees and the number of edges. There is a simple connection between these in any graph:

Theorem 1.1. *The sum of the degrees of the vertices in a graph equals twice the number of edges.*

Proof. Every edge contributes two to the sum of the degrees, one for each of its endpoints. □

Theorem 1.1 is sometimes called the *Handshake Theorem*: if we total up the number of people each person at a party shakes hands with, the total will be twice the number of handshakes that occurred.

2 Connectedness

2.1 Paths and Simple Cycles

Paths in graphs are defined in pretty much that same way as for digraphs, but they are going to be particularly important, so let's define them precisely.

Definition 2.1. Let G be a graph with vertices, V , and edges, E . A *path* in G is a sequence of vertices

$$v_0, \dots, v_k$$

with $k \geq 0$ such that $v_i - v_{i+1}$ is an edge in E for $0 \leq i < k$. The path is *simple* iff all the v_i 's are different, that is, $v_i = v_j$ only if $i = j$.

The path is said to *start* at v_0 , to *end* at v_k , and *length* of the path is defined to be k . An edge, e , is *traversed* n times by the path if there are n different values of i such that edge $v_i - v_{i+1}$ is e .

For example, the graph in Figure 2.1 has a length 6 simple path A,B,C,D,E,F,G. This is the longest simple path in the graph.

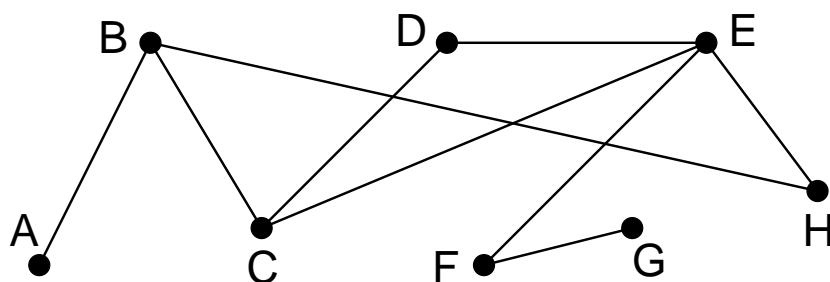


Figure 1: A graph with 3 simple cycles.

Notice that the *length* of a path is the total number of times it traverses edges, which is *one less* than its length as a sequence of vertices. The length 6 path A,B,C,D,E,F,G is actually a sequence of seven vertices.

Cycles are paths that begin and end with the same vertex, same as for digraphs. *Simple cycles* will be cycles that don't cross themselves. For example, the graph in Figure 2.1 has three simple cycles B,H,E,C,B and C,D,E,C and B,C,D,E,H,B

A simple cycle can be described by any of the paths that go around it. For example, the path B,H,E,C,B describes the cycle as beginning and going clockwise around to end at B, while the path E,H,B,C,E, describes the same cycle as beginning and ending at E going counterclockwise.

To capture this precisely, we'll define a simple cycle to be a *subgraph* isomorphic to the simple cycle graph, C_n . A subgraph is obtained by restricting a graph to a subset of its vertices and then possibly deleting some additional edges. Formally:

Definition 2.2. A subgraph, G' , of a graph, G , is a graph whose vertices, V' , are a nonempty subset of the vertices of G and whose edges are a subset of the edges of G .

Notice that since a subgraph is itself a graph, the endpoints of every edge of G' must be vertices in V' .

Definition 2.3. For $n \geq 3$, let C_n be the graph with vertices $1, \dots, n$ and edges

$$1-2, 2-3, \dots, (n-1)-n, n-1.$$

A graph, C , is a *simple cycle of length n* iff it is isomorphic to C_n for some $n \geq 3$. A *simple cycle of a graph, G* , is a subgraph of G that is a simple cycle.

If we describe a simple cycle with a path around it, then the *length* of a cycle is the number of times the path traverses an edge which is *the same* as the number of distinct vertices on the path. For example, the cycle with 5 vertices described by the path B,C,D,E,H,B has length 5.

2.2 Connected Components

Definition 2.4. Two vertices in a graph are said to be *connected* if there is a path that begins at one and ends at the other.

Now if there is a path from vertex u to vertex v , then v is connected to u by the reverse path, so connectedness is a symmetric relation. Also, if there is a path from u to v , and also a path from v to w , then these two paths can be combined to form a path from u to w . So the connectedness relation is transitive. It is also reflexive, since every vertex is by definition connected to itself by a path of length zero. This proves;

Lemma. *Connectness is an equivalence relation on the vertices of any graph.*

The diagram in Figure 2.2 looks like a picture of three graphs, but is intended to be a picture of *one* graph. This graph consists of three connected pieces, with no paths between vertices in different pieces.



Figure 2: A graph with 3 connected components.

Definition 2.5. A graph is said to be *connected* if every pair of vertices are connected.

These connected pieces of a graph are called its *connected components*. A rigorous definition is easy: a connected component is the set of all the vertices connected to some single vertex. That is, the connected components of a graph are the *equivalence classes* of its connectedness relation.

So a graph is connected iff it has exactly one connected component. The empty graph on n vertices has n connected components.

2.3 How Well Connected?

If we think of a graph as modelling cables in a telephone network, or oil pipelines, or electrical power lines, then we not only want connectivity, but we want connectivity that survives component failure. This leads us to the following definition:

Definition 2.6. Two vertices in a graph are k -connected if they remain connected in any subgraph obtained by deleting $k - 1$ edges. A graph is k -connected if every pair of its vertices are k -connected.

So 1-connected is the same as connected for both vertices and graphs. Another way to say that a graph is k -connected is that every subgraph obtained from it by deleting at most $k - 1$ edges is connected.

For example, in the graph in Figure 2.1, vertices B and E are 2-connected, G and E are 1-connected, and no vertices are 3-connected. The graph as a whole is only 1-connected.

More generally, any simple cycle is 2-connected, and the complete graph, K_n , is $(n - 1)$ -connected.

If two vertices are connected by k edge-disjoint paths (that is, no two paths traverse the same edge), then they are obviously k -connected. A fundamental fact, whose ingenious proof we omit, is Menger's theorem which confirms that the converse is also true: if two vertices are k -connected, then there are k edge-disjoint paths connecting them. It takes some ingenuity to prove even for the case $k = 2$.

2.4 Connection by Simple Path

Where there's a path there's a simple path. This is sort of obvious, but proving it carefully provides a nice illustration of the Least Number Principle.

Lemma 2.7. *If vertex u is connected to vertex v in a graph, then there is a simple path from u to v .*

Proof. Since there is a path from u to v , there must, by the Least Number Principle, be a minimum length path from u to v . If the minimum length is zero or one, this minimum length path is itself a simple path from u to v .

Otherwise, there is a minimum length path

$$v_0, v_1, \dots, v_k$$

from $u = v_0$ to $v = v_k$ where $k \geq 2$. We claim this path must be simple.

To prove the claim, suppose to the contrary that the path is not simple, that is, some vertex on the path occurs twice. This means that there are integers i, j such that $0 \leq i < j \leq k$ with $v_i = v_j$. Then deleting the subsequence

$$v_{i+1}, \dots, v_j$$

yields a strictly shorter path from u to v , contradicting the minimality of the given path. \square

Actually, we proved something stronger:

Corollary 2.8. *For any path of length k in a graph, there is a simple path of length at most k with the same endpoints.*

2.5 The Minimum Number of Edges in a Connected Graph

The following theorem says that a graph with few edges must have many connected components. To be interesting, there must be fewer edges than vertices.

Theorem 2.9. *Every graph with k vertices and n edges has at least $k - n$ connected components.*

Proof. We use induction on the number of edges. Let $P(n)$ be the proposition that

for every k , every graph with k vertices and n edges has at least $k - n$ connected components.

Base case: In a graph with 0 edges and k vertices, each vertex is itself a connected component, and so there are exactly $k = k - 0$ connected components.

Inductive step: Now we assume that the induction hypothesis holds for every n -edge graph in order to prove that it holds for every $(n + 1)$ -edge graph, where $n \geq 0$.

Consider a graph, G , with $n + 1$ edges and k vertices. We want to prove that G has at least $k - (n + 1)$ connected components.

To do this, remove an arbitrary edge $u - v$ and call the resulting graph G' . By the induction assumption, G' has at least $k - n$ connected components.

Now add back the edge $u - v$ to obtain the original graph G . If u and v were in the same connected component of G' , then G has the same connected components as G' , so G has at least $k - n > k - (n + 1)$ components. Otherwise, if u and v were in different connected components of G' , then these two components are merged into one in G , but all other components remain unchanged, reducing the number of components by 1. Therefore, G has at least $(k - n) - 1 = k - (n + 1)$ connected components.

The theorem follows by induction. □

Corollary 2.10. *Every connected graph with n vertices has at least $n - 1$ edges.*

A couple of points about the proof of Theorem 2.9 are worth noting. First, notice that we used induction on the number of edges in the graph. This is very common in proofs involving graphs, and so is induction on the number of vertices. When you're presented with a graph problem, these two approaches should be among the first you consider.

The second point is more subtle. Notice that in the inductive step, we took an arbitrary $(n + 1)$ -edge graph, threw out an edge so that we could apply the induction assumption, and then put the edge back. You'll see this shrink-down, grow-back process very often in the inductive steps of proofs related to graphs. This might seem like needless effort; why not start with an n -edge graph and add one more to get an $(n + 1)$ -edge graph? That would work fine in this case, but opens the door to a very nasty logical error in similar arguments. You'll see an example in class. Always use shrink-down, grow-back arguments, and you'll never fall into this trap.

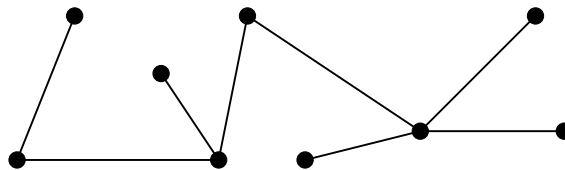
3 Trees

Trees are a fundamental data structure in Computer Science, and there are many kinds, for example rooted, ordered, or binary trees. In this section we focus on the purest kind of tree. Namely, we use the *tree* to mean a connected graph without simple cycles.

A graph with no simple cycles is called *acyclic*; so trees are acyclic connected graphs.

3.1 Tree Properties

Here is an example of a tree:



A vertex of degree one is called a *leaf*. In this example, there are 5 leaves.

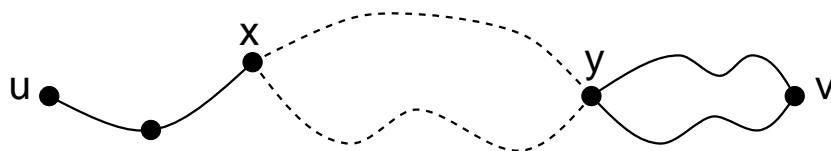
The graph shown above would no longer be a tree if any edge were removed, because it would no longer be connected. The graph would also not remain a tree if any edge were added between two of its vertices, because then it would contain a simple cycle. Furthermore, note that there is a unique path between every pair of vertices. These features of the example tree are actually common to all trees.

Theorem 3.1. *Every tree has the following properties:*

1. *Any connected subgraph is a tree.*
2. *There is a unique simple path between every pair of vertices.*
3. *Adding an edge between two vertices creates a cycle.*
4. *Removing any edge disconnects the graph.*
5. *If it has at least two vertices, then it has at least two leaves.*
6. *The number of vertices is one larger than the number of edges.*

Proof. 1. A simple cycle in a subgraph is also a simple cycle in the whole graph, so any subgraph of an acyclic graph must also be acyclic. If the subgraph is also connected, then by definition, it is a tree.

2. There is at least one path, and hence one simple path, between every pair of vertices, because the graph is connected. Suppose that there are two different simple paths between vertices u and v . Beginning at u , let x be the first vertex where the paths diverge, and let y be the next vertex they share. Then there are two simple paths from x to y with no common edges, which defines a simple cycle. This is a contradiction, since trees are acyclic. Therefore, there is exactly one simple path between every pair of vertices.



3. An additional edge $u-v$ together with the unique path between u and v forms a simple cycle.
4. Suppose that we remove edge $u-v$. Since a tree contained a unique path between u and v , that path must have been $u-v$. Therefore, when that edge is removed, no path remains, and so the graph is not connected.
5. Let v_1, \dots, v_m be the sequence of vertices on a longest simple path in the tree. Then $m \geq 2$, since a tree with two vertices must contain at least one edge. There cannot be an edge v_1-v_i for $2 < i \leq m$; otherwise, vertices v_1, \dots, v_i would form a simple cycle. Furthermore, there cannot be an edge $u-v_1$ where u is not on the path; otherwise, we could make the path longer. Therefore, the only edge incident to v_1 is v_1-v_2 , which means that v_1 is a leaf. By a symmetric argument, v_m is a second leaf.
6. We use induction on the number of vertices. For a tree with a single vertex, the claim holds since it has no edges and $0 + 1 = 1$.

Now suppose that the claim holds for all n -vertex trees and consider an $(n + 1)$ -vertex tree, T . Let v be a leaf of the tree.

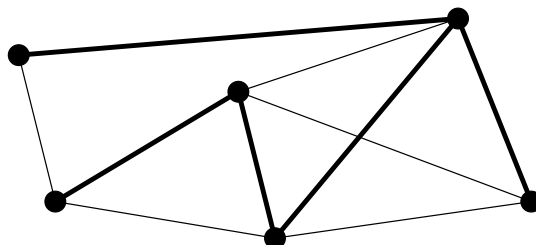
We will let the reader verify that deleting a vertex of degree 1 (and its incident edge) from any connected graph leaves a connected subgraph. So by (1), deleting v and its incident edge gives a smaller tree, and this smaller tree has one more vertex than edge by induction. If we reattach the vertex, v , and its incident edge, then the equation still holds because the number of vertices and number of edges both increase by 1. Thus, the claim holds for T and, by induction, for all trees.

□

Various subsets of these properties provide alternative characterizations of trees, though we won't prove this. For example, a *connected* graph with a number of vertices one larger than its number of edges is necessarily a tree. Also, a graph with unique paths between every pair of vertices is necessarily a tree.

3.2 Spanning Trees

Trees are everywhere. In fact, every connected graph contains a subgraph that is a tree with the same vertices as the graph. This is called a *spanning tree* for the graph. For example, here is a connected graph with a spanning tree highlighted.



Theorem 3.2. *Every connected graph contains a spanning tree.*

Proof. Let T be a connected subgraph of G with the smallest number of edges. We show that T is acyclic by contradiction. So suppose that T has a cycle with the following edges:

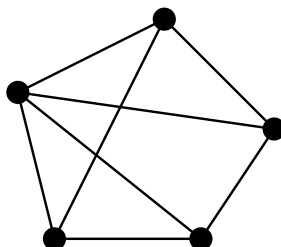
$$v_0—v_1, v_1—v_2, \dots, v_n—v_0$$

Suppose that we remove the last edge, $v_n—v_0$. If a pair of vertices x and y was joined by a path not containing $v_n—v_0$, then they remain joined by that path. On the other hand, if x and y were joined by a path containing $v_n—v_0$, then they remain joined by a path containing the remainder of the cycle. This is a contradiction, since T was defined to be a connected subgraph of G with the smallest number of edges. Therefore, T is acyclic. \square

4 Coloring Graphs

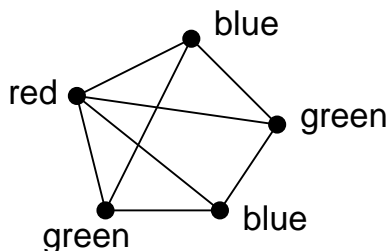
Each term, the MIT Schedules Office must assign a time slot for each final exam. This is not easy, because some students are taking several classes with finals, and a student can take only one test during a particular time slot. The Schedules Office wants to avoid all conflicts, but to make the exam period as short as possible.

We can recast this scheduling problem as a question about coloring the vertices of a graph. Create a vertex for each course with a final exam. Put an edge between two vertices if some student is taking both courses. For example, the scheduling graph might look like this:



Next, identify each time slot with a color. For example, Monday morning is red, Monday afternoon is blue, Tuesday morning is green, etc.

Assigning an exam to a time slot is now equivalent to coloring the corresponding vertex. The main constraint is that adjacent vertices must get different colors; otherwise, some student has two exams at the same time. Furthermore, in order to keep the exam period short, we should try to color all the vertices using as few different colors as possible. For our example graph, three colors suffice:



This coloring corresponds to giving one final on Monday morning (red), two Monday afternoon (blue), and two Tuesday morning (green).

4.1 k-Coloring

Many other resource allocation problems boil down to coloring some graph. In general, a graph G is k -colorable if each vertex can be assigned one of k colors so that adjacent vertices get different colors. The smallest sufficient number of colors is called the *chromatic number* of G . The chromatic number of a graph is generally difficult to compute, but the following theorem provides an upper bound:

Theorem 4.1. *A graph with maximum degree at most k is $(k + 1)$ -colorable.*

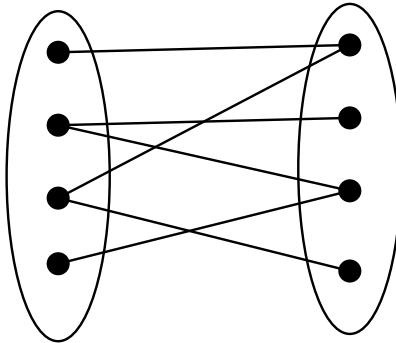
Proof. We use induction on the number of vertices in the graph, which we denote by n . Let $P(n)$ be the proposition that an n -vertex graph with maximum degree at most k is $(k + 1)$ -colorable. A 1-vertex graph has maximum degree 0 and is 1-colorable, so $P(1)$ is true.

Now assume that $P(n)$ is true, and let G be an $(n + 1)$ -vertex graph with maximum degree at most k . Remove a vertex v and its incident edges, leaving an n -vertex graph G' . The maximum degree of G' is at most k , and so G' is $(k + 1)$ -colorable by our assumption $P(n)$. Now put back vertex v and its incident edges. We can assign v a color different from all adjacent vertices, since v has degree at most k and $k + 1$ colors are available. Therefore, G is $(k + 1)$ -colorable. The theorem follows by induction. \square

4.2 Bipartite Graphs

The 2-colorable graphs are important enough to merit a special name; they are called *bipartite graphs*. Suppose that G is bipartite. This means we can color every vertex in G either black or white so that adjacent vertices get different colors. Then we can put all the black vertices in

a clump on the left and all the white vertices in a clump on the right. Since every edge joins differently-colored vertices, every edge must run between the two clumps. Therefore, every bipartite graph looks something like this:

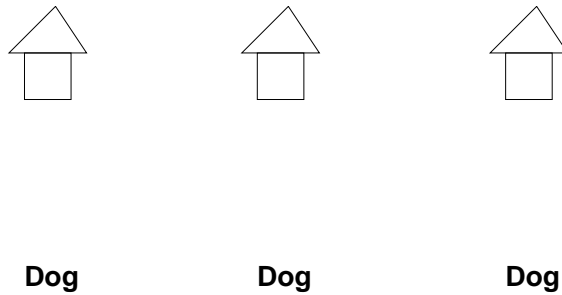


Bipartite graphs are both useful and common. For example, every path, every tree, and every even-length cycle is bipartite. In turns out, in fact, that every graph not containing an odd length cycle is bipartite and vice verse.

Theorem 4.2. *A graph is bipartite if and only if it contains no odd length cycle.*

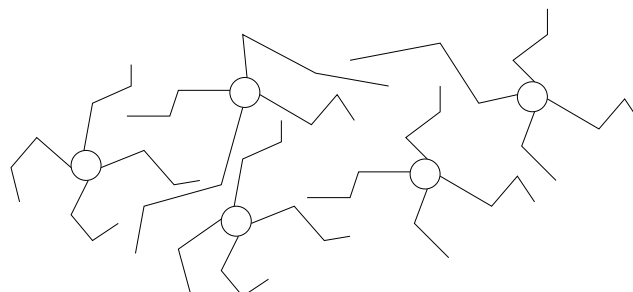
5 Planar Graphs

Here are three dogs and three houses.



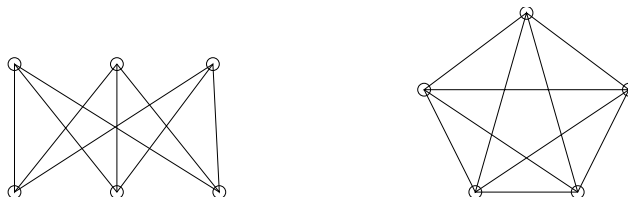
Can you find a path from each dog to each house such that no two paths intersect?

A *quadapus* is a little-known animal similar to an octopus, but with four arms. Here are five quadapi resting on the seafloor:



Can each quadrapus simultaneously shake hands with every other in such a way that no arms cross?

Informally, a *planar graph* is a graph that can be drawn in the plane so that no edges cross. Thus, these two puzzles are asking whether the graphs below are planar; that is, whether they can be redrawn so that no edges cross.



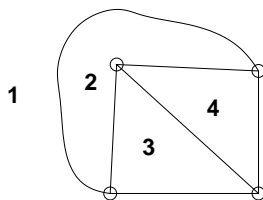
In each case, the answer is, “No— but almost!” In fact, each drawing *would* be possible if any single edge were removed.

Planar graphs have applications in circuit layout and are helpful in displaying graphical data, for example, program flow charts, organizational charts, and scheduling conflicts. We will use them to prove a wonderful mathematical fact that was first proved by the ancient Greeks.

5.1 Euler’s Formula

A graph is planar if it has a *planar embedding* (or *drawing*). This is a way of assigning each vertex to a distinct point in the plane and each edge to a smooth, non-self-intersecting curve between the points assigned to the vertices incident to the edge. None of the curves may “cross,” no point other than a single common endpoint can appear on two of the curves.

This unattractive definition hints at a theoretical problem associated with planar graphs: while the idea seems intuitively simple, rigorous arguments about planar graphs must start off with reasoning about regions of the plane bounded by smooth curves. We’ll get around this simply by *assuming* a few intuitive properties of planar graphs.¹ To start, we’ll assume that a drawing of a planar graph divides the plane into *faces*, connected regions bounded by edges of the graph. For example, the drawing below has four faces:



Face 1, which extends off to infinity in all directions, is called the *outside face*. Notice that each edge that is on a cycle in a planar graph serves as a boundary between two faces.

It turns out that the number of vertices and edges in a connected planar graph determine the number of faces in every drawing:

¹A way to avoid all these assumptions is described in an Appendix.

Theorem 5.1 (Euler's Formula). For every drawing of a connected planar graph

$$v - e + f = 2$$

where v is the number of vertices, e is the number of edges, and f is the number of faces.

For example, in the drawing above, $|V| = 4$, $|E| = 6$, and $f = 4$. Sure enough, $4 - 6 + 4 = 2$, as Euler's Formula claims.

Proof. We use induction on the number of edges in the graph. Let $P(e)$ be the proposition that $v - e + f = 2$ for every drawing of a graph G with e edges.

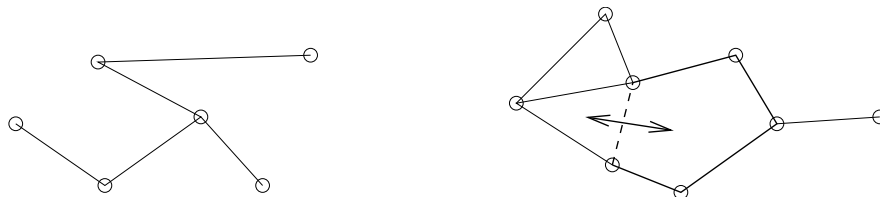
Base case: A connected graph with $e = 0$ edges has $v = 1$ vertices, and every drawing of the graph has $f = 1$ faces (the outside face). Thus, $v - e + f = 1 - 0 + 1 = 2$, and so $P(0)$ is true.

Inductive step: Now we assume that $P(e)$ is true in order to prove $P(e + 1)$ where $e \geq 0$. Consider a connected graph G with $e + 1$ edges. There are two cases:

1. If G is acyclic, then the graph is a tree. Thus, there are $e + 2$ vertices and every drawing has only the outside face. Since $(e + 2) - (e + 1) + 1 = 2 - 1 + 1 = 2$, $P(e + 1)$ is true.
2. Otherwise, G has at least one cycle. Select a spanning tree and an edge $u-v$ in the cycle, but not in the tree. (The spanning tree can not contain all edges in the cycle, since trees are acyclic.) Removing $u-v$ merges the two faces on either side of the edge and leaves a graph G' with only e edges and some number of vertices v and faces f . Graph G' is connected, because there is a path between every pair of vertices within the spanning tree. So $v - e + f = 2$ by the induction assumption $P(e)$. Thus, the original graph G had v vertices, $e + 1$ edges, and $f + 1$ faces. Since $v - (e + 1) + (f + 1) = v - e + f = 2$, $P(e + 1)$ is again true.

The theorem follows by the principle of induction. □

In this argument, we implicitly assumed two geometric facts: a drawing of a tree has only one face, and removing an edge on a cycle merges two faces into one.



5.2 Number of Edges versus Vertices

Lemma 5.2. Suppose a connected, planar graph has $v > 2$ vertices, e edges. Then

$$e \leq 3v - 6.$$

Proof. Suppose the graph is drawn in the plane with f faces. Every edge is traversed exactly twice by the cycles that are the face boundaries, as will be proved in a class problem. So the sum of the length of the face boundaries is exactly $2e$. Also, when $v > 2$, each face boundary traverses at least 3 edges, so this sum is at least $3f$. So we have

$$2e \geq 3f. \quad (1)$$

But $f = e - v + 2$ by Euler's formula, and substituting into (1) gives

$$\begin{aligned} 2e &\geq 3(e - v + 2) \\ 0 &\geq e - 3v + 6 \\ 3v - 6 &\geq e \end{aligned}$$

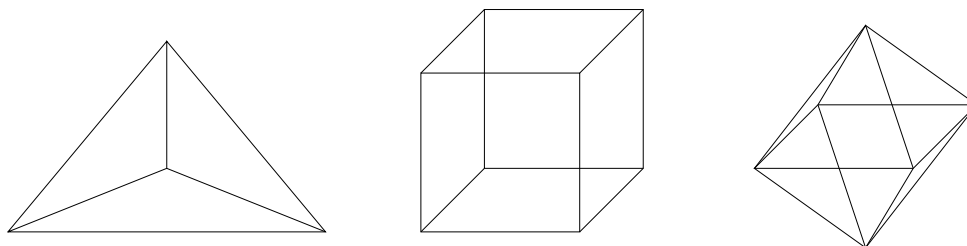
□

Euler's formula lets us prove that the quadapi can't all shake hands without crossing. Representing quadapi by vertices and the necessary handshakes by edges, we get the complete graph, K_5 . Shaking hand without crossing amounts to drawing K_5 in the plane. But K_5 is connected with 5 vertices and 10 edges, and $10 > 3 \cdot 5 - 6$, so violates the condition of Lemma 5.2 necessary for planarity.

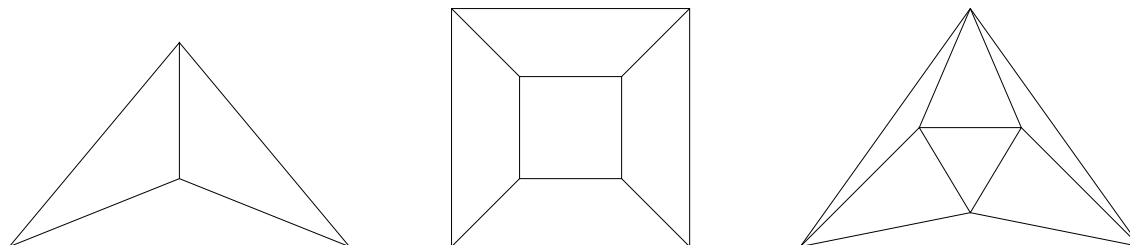
5.3 Classifying Polyhedra

The Pythagoreans had two great mathematical secrets, the irrationality of 2 and a geometric construct that we're about to rediscover!

A *polyhedron* is a convex, three-dimensional region bounded by a finite number of polygonal faces. If the faces are identical regular polygons and an equal number of polygons meet at each corner, then the polyhedron is *regular*. Three examples of regular polyhedra are shown below: the tetraheron, the cube, and the octahedron.



How many more polyhedra are there? Imagine putting your eye very close to one face of a translucent polyhedron. The edges of that face would ring the periphery of your vision and all other edges would be visible within. For example, the three polyhedra above would look something like this:



Thus, we can regard the corners and edges of these polyhedra as the vertices and edges of a planar graph. (This is another logical leap based on geometric intuition.) This means Euler's formula for planar graphs can help guide our search for regular polyhedra.

Let m be the number of faces that meet at each corner of a polyhedron, and let n be the number of sides on each face. In the corresponding planar graph, there are m edges incident to each of the v vertices. Since each edge is incident to two vertices, we know:

$$mv = 2e$$

Also, each face is bounded by n edges. Since each edge is on the boundary of two faces, we have:

$$nf = 2e$$

Solving for v and f in these equations and then substituting into Euler's formula gives:

$$\begin{aligned} \frac{2e}{m} - e + \frac{2e}{n} &= 2 \\ \frac{1}{m} + \frac{1}{n} &= \frac{1}{e} + \frac{1}{2} \end{aligned}$$

The last equation places strong restrictions on the structure of a polyhedron. Every nondegenerate polygon has at least 3 sides, so $n \geq 3$. And at least 3 polygons must meet to form a corner, so $m \geq 3$. On the other hand, if either n or m were 6 or more, then the left side of the equation could be at most $\frac{1}{3} + \frac{1}{6} = \frac{1}{2}$, which is less than the right side. Checking the finitely-many cases that remain turns up five solutions. For each valid combination of n and m , we can compute the associated number of vertices v , edges e , and faces f . And polyhedra with these properties do actually exist:

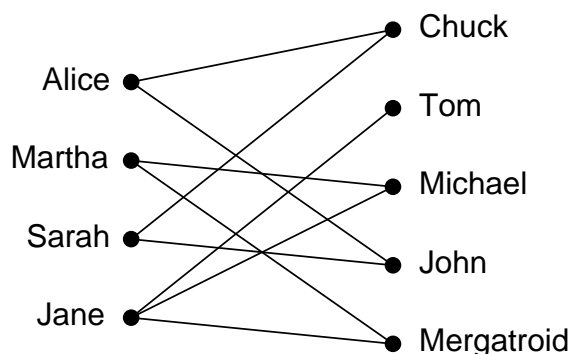
n	m	v	e	f	polyhedron
3	3	4	6	4	tetrahedron
4	3	8	12	6	cube
3	4	6	12	8	octahedron
3	5	12	30	20	icosahedron
5	3	20	30	12	dodecahedron

The last polyhedron in this list, the dodecahedron, was the other great mathematical secret of the Pythagorean sect!

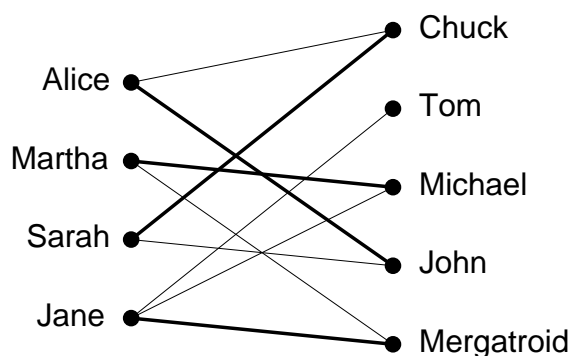
6 Hall's Marriage Theorem

A class contains some girls and some boys. Each girl likes some boys and does not like others. Under what conditions can each girl be paired up with a boy that she likes?

We can model the situation with a bipartite graph. Create a vertex on the left for each girl and a vertex on the right for each boy. If a girl likes a boy, put an edge between them. For example, we might obtain the following graph:



In graph terms, our goal is to find a *matching* for the girls; that is, a subset of edges such that exactly one edge is incident to each girl and at most one edge is incident to each boy. (If there are more boys than girls, some boys won't be paired up, which means no edge is incident to them.) For example, here is one possible matching for the girls:



Hall's Marriage Theorem states necessary and sufficient conditions for the existence of a matching in a bipartite graph. It turns out to be a remarkably useful mathematical tool.

We'll state and prove Hall's Theorem using girl-likes-boy terminology. Define *the set of boys liked by a given set of girls* to consist of all boys liked by at least one of those girls. For example, the set of boys liked by Martha and Jane consists of Tom, Michael, and Mergatroid.

For us to have any chance at all of matching up the girls, the following *marriage condition* must hold:

Every subset of girls likes at least as large a set of boys.

For example, we can not find a matching if some 4 girls like only 3 boys. Hall's Theorem says that this necessary condition is actually sufficient; if the marriage condition holds, then a matching exists.

Theorem 6.1. *A matching for a set of girls G with a set of boys B can be found if and only if the marriage condition holds.*

Proof. First, let's suppose that a matching exists and show that the marriage condition holds. Consider an arbitrary subset of girls. Each girl likes at least the boy she is matched with. Therefore, every subset of girls likes at least as large a set of boys. Thus, the marriage condition holds.

Next, let's suppose that the marriage condition holds and show that a matching exists. We use strong induction on $|G|$, the number of girls. If $|G| = 1$, then the marriage condition implies that the lone girl likes at least one boy, and so a matching exists. Now suppose that $|G| \geq 2$. There are two possibilities:

1. Every proper subset of girls likes a *strictly larger* set of boys. In this case, we have some latitude: we pair an arbitrary girl with a boy she likes and send them both away. The marriage condition still holds for the remaining boys and girls, so we can match the rest of the girls by induction.
2. Some proper subset of girls $X \subset G$ likes an *equal-size* set of boys $Y \subset B$. We match the girls in X with the boys in Y by induction and send them all away. We will show that the marriage condition holds for the remaining boys and girls, and so we can match the rest of the girls by induction as well.

To that end, consider an arbitrary subset of the remaining girls $X' \subseteq G - X$, and let Y' be the set of remaining boys that they like. We must show that $|X'| \leq |Y'|$. Originally, the combined set of girls $X \cup X'$ liked the set of boys $Y \cup Y'$. So, by the marriage condition, we know:

$$|X \cup X'| \leq |Y \cup Y'|$$

We sent away $|X|$ girls from the set on the left (leaving X') and sent away an equal number of boys from the set on the right (leaving Y'). Therefore, it must be that $|X'| \leq |Y'|$ as claimed.

In both cases, there is a matching for the girls. The theorem follows by induction. \square

The proof of this theorem gives an algorithm for finding a matching in a bipartite graph, albeit not a very efficient one. However, efficient algorithms for finding a matching in a bipartite graph do exist. Thus, if a problem can be reduced to finding a matching, the problem is essentially solved from a computational perspective.

6.1 A Formal Statement

Let's restate Hall's Theorem in abstract terms so that you'll not always be condemned to saying, "Now this group of little girls likes at least as many little boys..." Suppose S is a set of vertices in a graph. Define $N(S)$ to be the set of all neighbors of S ; that is, all vertices that are adjacent to a vertex in S , but not actually in S .

Theorem 6.2 (Hall's Theorem). *Let G be a graph whose vertices can be partitioned into two sets, L and R , such that every edge has one endpoint in L and one endpoint in R . There is a matching for the L vertices if and only if $|N(S)| \geq |S|$ for every set $S \subseteq L$.*

A Appendix: Drawing Planar Graphs

[Optional]

A *planar drawing script*, PDS for short, defines how to draw a connected graph in the plane. It specifies the order in which to draw the edges, and, as each edge is drawn, which face of the current drawing that the edge goes into. It also specifies the one or two new faces that result from drawing the edge. This is illustrated in animated slides at

<http://theory.csail.mit.edu/classes/6.042/fall05/planar-animation.pdf>.

It's worth viewing these slides before trying to study the following formal definitions.

Formally a PDS will consist of a sequence of distinct edges, and a set of cycles called *boundaries*. Intuitively, the edges are listed in the order in which they should be drawn in the plane, and the boundaries of the PDS describe the successive vertices on the boundaries of each of the faces of the planar drawing.

PDS's are officially defined recursively as follows:

1. A single edge $v-w$ is a PDS. It has one boundary, namely the length two cycle described by the path v, w, v .
2. (Attaching an edge with a new endpoint to a face.) Suppose S is a PDS, and a boundary of S is described by a path of the form vxv , where v is a vertex and x is a positive length sequence of vertices. If w is a "fresh" vertex to which none of the edges of S are incident, then we can obtain another PDS, T , by adding edge $v-w$ to the list of edges of S . The boundaries of T will be the same as those of S , except that boundary described by vxv is replaced by a boundary described by the path $vxwv$. (Notice that two traversals of the new edge are actually added to the cycle.)
3. (Attaching an edge between two vertices on a face.) Suppose S is a PDS and the edge $v-w$ is not in the edge-list of S . Also suppose some boundary of S is represented by a path of the form $vxwyv$, where x and y are positive length sequences of vertices. Then we can obtain another PDS, T , by adding edge $v-w$ to the list of edges of S . The boundaries of T will be the same as those of S , except that boundary described by $vxwyv$ is replaced by two boundaries described by the two paths $vxwv$ and wyv .²

The *graph of a PDS* is defined to be the graph whose edges are those listed in the PDF and whose vertices are exactly those to which these edges are incident.

There are a bunch of properties of any PDS that can be proved easily by induction on the length of its edge-list.

Lemma.

No edge appears more than once in the edge-list.

Every boundary is a cycle that traverses only edges in the edge-list.

Every edge in the edge-list is traversed exactly twice by one boundary, or exactly once by each of two boundaries, and is not traversed by any other boundaries.

If the graph of the PDS has 3 or more vertices, then every boundary traverses at least three edges.

The graph of the PDS is connected.

If e is the length of the edge-list, v is the number of distinct vertices to which these edges are incident, and f is the number of boundaries, then

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

²There is one minor exception to this rule. If $vxwv$ and wyv turn out to be the same cycle, then both "copies" of this cycle must be considered boundaries of T . Technically, this means that the set of boundaries must be a "multiset" which can contain elements more than once. But the only situation in which this occurs is when there is when the graph is a simple cycle, C_n whose embedding will have two faces —an "inside" face and an "outside" face —that share the same boundary. This exception is so special that it is better handled in a footnote.

These properties support the intuitive claim that all connected planar drawings are captured by PDS's. We will accept this without proof:

Theorem. *A graph is connected and planar iff it is the graph of a PDS.*

PDS's are a data structure based only on the connection properties of a graph, and they are nicely suited to proving planar properties by induction without the need for any geometric facts or assumptions. However, there are still some important properties of planarity that are awkward to prove using PDS's. An example is the fact noted earlier that any subgraph of a planar graph is planar.