3/17

11 Simple Graphs

arrow always in both directions

Simple graphs model relationships that are *symmetric*, meaning that the relationship is mutual. Example of such mutual relationships are being married, speaking the same language, not speaking the same language, occurring during overlapping time intervals, or being connected by a conducting wire. They come up in all sorts of applications, including scheduling, constraint satisfaction, computer graphics, and communications, but we'll start with an application designed to get your attention: we are going to make a professional inquiry into sexual behavior. Namely, we'll look at some data about who, on average, has more opposite-gender partners, men or women.

Sexual demographics have been the subject of many studies. In one of the largest, researchers from the University of Chicago interviewed a random sample of 2500 people over several years to try to get an answer to this question. Their study, published in 1994, and entitled *The Social Organization of Sexuality* found that on average men have 74% more opposite-gender partners than women.

Other studies have found that the disparity is even larger. In particular, ABC News claimed that the average man has 20 partners over his lifetime, and the average woman has 6, for a percentage disparity of 233%. The ABC News study, aired on Primetime Live in 2004, purported to be one of the most scientific ever done, with only a 2.5% margin of error. It was called "American Sex Survey: A peek between the sheets," —which raises some question about the seriousness of their reporting.

20-6 / 6 ?

Yet again, in August, 2007, the N.Y. Times reported on a study by the National Center for Health Statistics of the U.S. government showing that men had seven partners while women had four. Anyway, whose numbers do you think are more accurate, the University of Chicago, ABC News, or the National Center? —don't answer; this is a setup question like "When did you stop beating your wife?" Using a little graph theory, we'll explain why none of these findings can be anywhere near the truth.

11.1 Vertex Adjacency and Degrees

Simple graphs are defined as digraphs in which edges are *undirected* —they just connect two vertices without pointing in either direction between the vertices. So instead of a directed edge $\langle v \to w \rangle$ which starts at vertex v and ends at vertex w, a

simple graph only has an undirected edge, v—w, that connects v and w.

Definition 11.1.1. A *simple graph*, G, consists of a nonempty set, $V(G)$, called the *vertices* of G, and a set $E(G)$ called the *edges* of G. An element of $V(G)$ is called a *vertex*. A vertex is also called a *node*; the words "vertex" and "node" are used interchangeably. An element of $E(G)$ an *undirected edge* or simply an "edge." A undirected edge has two vertices $u \neq v$ called its *endpoints*. Such an edge can be represented by the two element set $\{u, v\}$. The notation u—v denotes this edge.

Both u—v and v—u define the same undirected edge, namely the one whose endpoints are u and v.

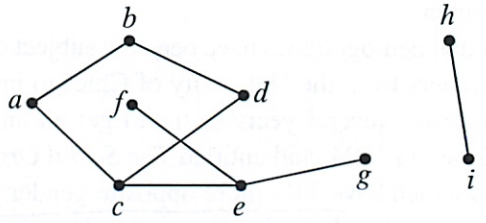


Figure 11.1 An example of a graph with 9 nodes and 8 edges.

For example, let H be the graph pictured in Figure 11.1. The vertices of H correspond to the nine dots in Figure 11.1, that is,

$$V(H) = \{a, b, c, d, e, f, g, h, i\}.$$

The edges correspond to the eight lines, that is,

$$E(H) = \{a\text{—}b, a\text{—}c, b\text{—}d, c\text{—}d, c\text{—}e, e\text{—}f, e\text{—}g, h\text{—}i\}.$$

Mathematically, that's all there is to the graph H.

Definition 11.1.2. Two vertices in a simple graph are said to be *adjacent* iff they are the endpoints of an edge, and an edge is said to be *incident* to each of its endpoints. The number of edges incident to a vertex v is called the *degree* of the vertex and is denoted by $\deg(v)$. Equivalently, the degree of a vertex is equals the number of vertices adjacent to it.

For example, for the graph H of Figure 11.1, vertex a is adjacent to vertex b, and b is adjacent to d. The edge a—c is incident to vertices a and c. Vertex h has degree 1, d has degree 2, and $\deg(e) = 3$. It is possible for a vertex to have degree 0, in which case it is not adjacent to any other vertices. A simple graph, G, does not need to have any edges at all, namely $|E(G)|$ could be zero, which would

implies that the degree of every vertex is also zero. But a simple graph must have at least one vertex, that is, $|V(G)|$ is required to be at least one.

An edge whose endpoints are the same is called a *self-loop*. Self-loops aren't allowed in simple graphs.[1] In a more general class of graphs called *multigraphs* there can be more than one edge with the same two endpoints, but this doesn't happen in simple graphs since every edge is uniquely determined by its two endpoints.

Sometimes graphs with no vertices, with self-loops, or with more than one edge between the same two vertices are convenient to have, but we don't need them, and sticking with simple graphs is simpler. :-)

For the rest of this chapter we'll use "graphs" as an abbreviation for "simple graphs."

A synonym for "vertices" is "*nodes*," and we'll use these words interchangeably. Simple graphs are sometimes called *networks*, edges are sometimes called *arcs*. We mention this as a "heads up" in case you look at other graph theory literature; we won't use these words.

11.2 Sexual Demographics in America

Let's model the question of heterosexual partners in graph theoretic terms. To do this, we'll let G be the graph whose vertices, V, are all the people in America. Then we split V into two separate subsets: M, which contains all the males, and F, which contains all the females.[2] We'll put an edge between a male and a female iff they have been sexual partners. This graph is pictured in Figure 11.2 with males on the left and females on the right.

Actually, this is a pretty hard graph to figure out, let alone draw. The graph is *enormous*: the US population is about 300 million, so $|V| \approx 300M$. Of these, approximately 50.8% are female and 49.2% are male, so $|M| \approx 147.6M$, and $|F| \approx 152.4M$. And we don't even have trustworthy estimates of how many edges there are, let alone exactly which couples are adjacent. But it turns out that we don't need to know any of this —we just need to figure out the relationship between the average number of partners per male and partners per female. To do this, we note that every edge is incident to exactly one M vertex (remember, we're only considering male-female relationships); so the sum of the degrees of the M vertices equals the number of edges. For the same reason, the sum of the degrees

real technicalities of set theory → very precise language

[1] You might try to represent a self-loop going between a vertex v and itself as $\{v, v\}$, but this equals $\{v\}$, and it wouldn't be an edge which is defined to be a set of *two* vertices.

[2] For simplicity, we'll ignore the possibility of someone being both, or neither, a man and a woman.

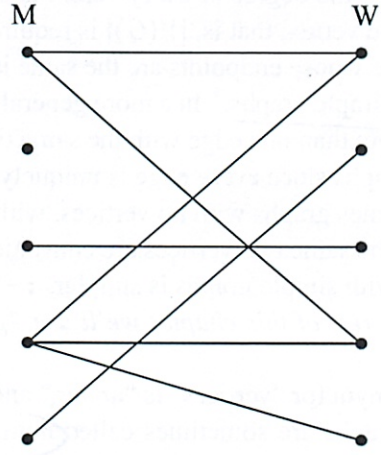


Figure 11.2 The sex partners graph

of the F vertices equals the number of edges. So these sums are equal:

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

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

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

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

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

In other words, we've proved that the average number of female partners of males in the population compared to the average number of males per female 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 promiscuity or selectivity. Rather, it just has to do with the relative number of males and females. Collectively, males and females have the same number of opposite gender partners, since it takes one of each set for every partnership,

but there are fewer males, so they have a higher ratio. This means that the University of Chicago, ABC, and the Federal government studies are way off. After a huge effort, they gave a totally wrong answer.

There's no definite explanation for why such surveys are consistently wrong. One hypothesis is that males exaggerate their number of partners —or maybe females downplay theirs —but these explanations are speculative. Interestingly, the principal author of the National Center for Health Statistics study reported that she knew the results had to be wrong, but that was the data collected, and her job was to report it.

The same underlying issue has led to serious misinterpretations of other survey data. For example, a couple of years ago, the Boston Globe ran a story on a survey of the study habits of students on Boston area campuses. Their survey showed that on average, minority students tended to study with non-minority students more than the other way around. They went on at great length to explain why this "remarkable phenomenon" might be true. But it's not remarkable at all —using our graph theory formulation, we can see that all it says is that there are fewer minority students than non-minority students, which is, of course what "minority" means.

11.2.1 Handshaking Lemma

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

Lemma 11.2.1. *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. ■

Lemma 11.2.1 is sometimes called the *Handshake Lemma*: 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.

11.3 Some Common Graphs

Some graphs come up so frequently that they have names. A *complete graph* K_n has n vertices and an edge between every two vertices, for a total of $n(n-1)/2$ edges. For example, K_5 is shown in Figure 11.3.

The *empty graph* has no edges at all. For example, the empty graph with 5 nodes is shown in Figure 11.4.

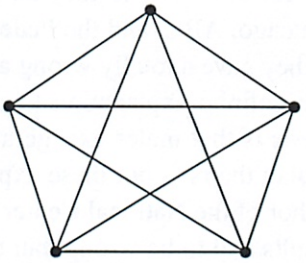


Figure 11.3 K_5: the complete graph on 5 nodes.

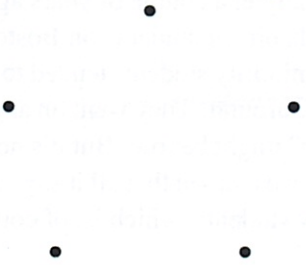


Figure 11.4 An empty graph with 5 nodes.

An n-node graph containing $n-1$ edges in sequence is known as a *line graph* L_n. More formally, L_n has

$$V(L_n) = \{v_1, v_2, \ldots, v_n\}$$

and

$$E(L_n) = \{\, v_1\text{—}v_2, v_2\text{—}v_3, \ldots, v_{n-1}\text{—}v_n \,\}$$

For example, L_5 is pictured in Figure 11.5.

There is also a one-way infinite line graph L_∞ which can be defined by letting the nonnegative integers $\mathbb{N}$ be the vertices with edges $k, k+1$— for all $k \in \mathbb{N}$.

If we add the edge v_n—v_1 to the line graph L_n, we get a graph called a *length-n cycle* C_n. Figure 11.6 shows a picture of length-5 cycle.

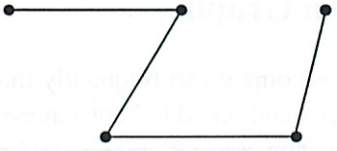


Figure 11.5 L_5: a 5-node line graph.

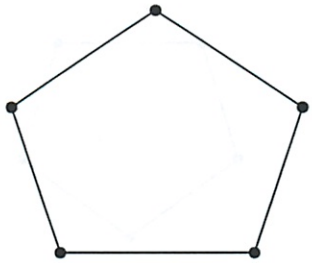


Figure 11.6 C_5: a 5-node cycle graph.

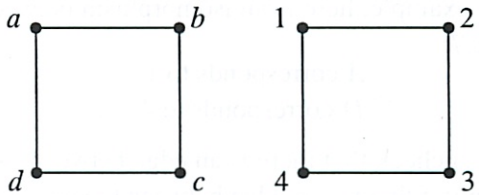


Figure 11.7 Isomorphic L_4 graphs.

11.4 Isomorphism

Two graphs that look the same might actually be different in a formal sense. For example, the two graphs in Figure 11.7 are both line graphs with 4 vertices, but one graph has vertex set $\{a, b, c, d\}$ while the other has vertex set $\{1, 2, 3, 4\}$.

Strictly speaking, these graphs are different mathematical objects, but this difference doesn't reflect the fact that the two graphs can be described by the same picture—except for the labels on the vertices. This idea of having the same picture "up to relabeling" can be captured neatly by adapting Definition 9.7.1 of isomorphism of digraphs to handle simple graphs. An isomorphism between two graphs is an edge-preserving bijection between their sets of vertices:

Definition 11.4.1. An isomorphism between graphs G and H is a bijection $f : V(G) \to V(H)$ such that

$$u—v \in E(G) \quad \text{iff} \quad f(u)—f(v) \in E(H)$$

for all $u, v \in V(G)$. Two graphs are isomorphic when there is an isomorphism between them.

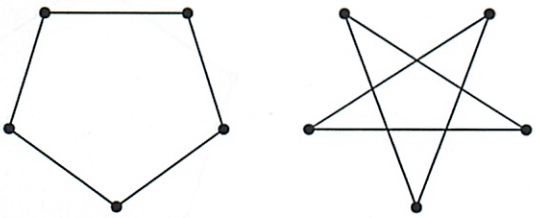


Figure 11.8 Isomorphic C_5 graphs.

For example, here is an isomorphism between vertices in the two graphs in Figure 11.7:

A corresponds to 1	B corresponds to 2
D corresponds to 4	C corresponds to 3.

You can check that there is an edge between two vertices in the graph on the left if and only if there is an edge between the two corresponding vertices in the graph on the right.

Two isomorphic graphs may be drawn very differently. For example, Figure 11.8 shows two different ways of drawing C_5:

Notice that f is an isomorphism between G and H, then f^{-1} is an isomorphism between H and G. Isomorphism is also transitive because by Lemma 5.2.1, the composition of bijections is a bijection. So isomorphism is in fact an equivalence relation.

Isomorphism preserves the connection properties of a graph, abstracting out what the vertices are called, what they are made out of, or where they appear in a drawing of the graph. More precisely, a property of a graph is said to be *preserved under isomorphism* if whenever G has that property, every graph isomorphic to G also has that property. For example, since an isomorphism is a bijection between sets of vertices, isomorphic graphs must have the same number of vertices. What's more, if f is a graph isomorphism that maps a vertex, v, of one graph to the vertex, $f(v)$, of an isomorphic graph, then by definition of isomorphism, every vertex adjacent to v in the first graph will be mapped by f to a vertex adjacent to $f(v)$ in the isomorphic graph. That is, v and $f(v)$ will have the same degree. So if one graph has a vertex of degree 4 and another does not, then they can't be isomorphic. In fact, they can't be isomorphic if the number of degree 4 vertices in each of the graphs is not the same.

Looking for preserved properties can make it easy to determine that two graphs are not isomorphic, or to guide the search for an isomorphism when there is one. It's generally easy in practice to decide whether two graphs are isomorphic. However, no one has yet found a procedure for determining whether two graphs are

This class still seems like it is in intro - we rarely use previous skills

Why not? — oh searching each like

isomorphic that is *guaranteed* to run in polynomial time on all pairs of graphs.[3]

Having such a procedure would be useful. For example, it would make it easy to search for a particular molecule in a database given the molecular bonds. On the other hand, knowing there is no such efficient procedure would also be valuable: secure protocols for encryption and remote authentication can be built on the hypothesis that graph isomorphism is computationally exhausting.

The definitions of bijection and isomorphism apply infinite graphs as well as finite graphs, as do most of the results in the rest of this chapter. But graph theory focuses mostly on finite graphs, and we will too. So super

in the rest of this chapter we'll assume graphs are finite.

We've actually been taking isomorphism for granted ever since we wrote "K_n has n vertices..." at the beginning of section 11.3.

key → *Graph theory is all about properties preserved by isomorphism.*

11.5 Bipartite Graphs & Matchings

2 parts (M and F)

There were two kinds of vertices in the "Sex in America" graph—males and females, and edges only went between the two kinds. Graphs like this come up so frequently that they have earned a special name—they are called *bipartite graphs*.

Definition 11.5.1. A *bipartite graph* is a graph whose vertices can be partitioned[4] into two sets, $L(G)$ and $R(G)$, such that every edge is incident to a vertex in $L(G)$ and to a vertex in $R(G)$. one in L() and one in G()

So every bipartite graph looks something like the graph in Figure 11.2.

11.5.1 The Bipartite Matching Problem

The bipartite matching problem is related to the sex-in-America problem that we just studied; only now the goal is to get everyone happily married. As you might imagine, this is not possible for a variety of reasons, not the least of which is the fact that there are more women in America than men. So, it is simply not possible to marry every woman to a man so that every man is married at most once.

But what about getting a mate for every man so that every woman is married at most once? Is it possible to do this so that each man is paired with a woman that

[3] A procedure runs in *polynomial time* when it needs an amount of time of at most $p(n)$, where n is the total number of vertices and $p()$ is a fixed polynomial.

[4] Partitioning a set means cutting it up into *nonempty* pieces. In this case, it means that $L(G)$ and $R(G)$ are nonempty, $L(G) \cup R(G) = V(G)$, and $L(G) \cap R(G) = \emptyset$.

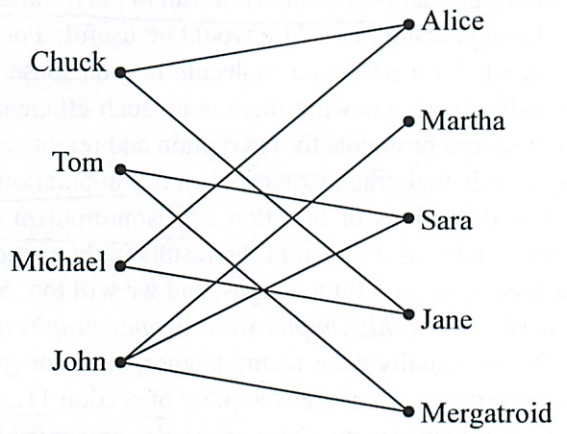


Figure 11.9 A graph where an edge between a man and woman denotes that the man likes the woman.

he likes? The answer, of course, depends on the bipartite graph that represents who likes who, but the good news is that it is possible to find natural properties of the who-likes-who graph that completely determine the answer to this question.

In general, suppose that we have a set of men and an equal-sized or larger set of women, and there is a graph with an edge between a man and a woman if the man likes the woman. In this scenario, the "likes" relationship need not be symmetric, since for the time being, we will only worry about finding a mate for each man that he likes.[5] (Later, we will consider the "likes" relationship from the female perspective as well.) For example, we might obtain the graph in Figure 11.9.

In this problem, a *matching* will mean a way of assigning every man to a woman so that different men are assigned to different women, and a man is always assigned to a woman that he likes. For example, one possible matching for the men is shown in Figure 11.10.

The Matching Condition

A famous result known as Hall's Matching Theorem gives 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 man-likes-woman terminology. Define *the set of women liked by a given set of men* to consist of all women liked by

[5] By the way, we do not mean to imply that marriage should or should not be of a heterosexual nature. Nor do we mean to imply that men should get their choice instead of women. It's just that with bipartite graphs, the edges only connected male nodes to female nodes and there are fewer men in America. So please don't take offense.

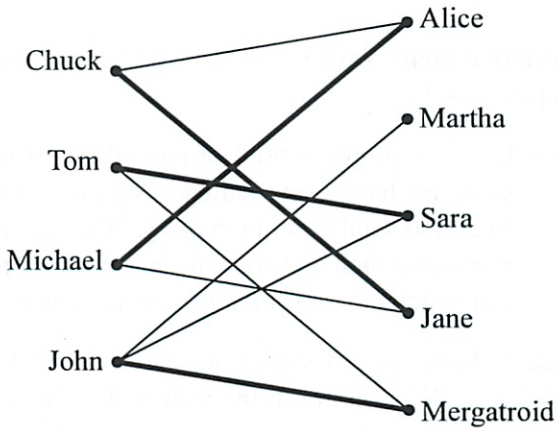


Figure 11.10 One possible matching for the men is shown with bold edges. For example, John is matched with Jane.

at least one of those men. For example, the set of women liked by Tom and John in Figure 11.9 consists of Martha, Sarah, and Mergatroid. For us to have any chance at all of matching up the men, the following *matching condition* must hold:

The Matching Condition: every subset of men likes at least as large a set of women.

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

Theorem 11.5.2. *A matching for a set M of men with a set W of women can be found if and only if the matching condition holds.*

Proof. First, let's suppose that a matching exists and show that the matching condition holds. For any subset of men, each man likes at least the woman he is matched with and a woman is matched with at most one man. Therefore, every subset of men likes at least as large a set of women. Thus, the matching condition holds.

Next, let's suppose that the matching condition holds and show that a matching exists. We use strong induction on $|M|$, the number of men, on the predicate:

$P(m)$ ::= for any set M of m men, if the matching condition holds for M, then there is a matching for M.

Base case ($|M| = 1$): If $|M| = 1$, then the matching condition implies that the lone man likes at least one woman, and so a matching exists.

Inductive Step: We need to show that $P(m)$ IMPLIES $P(m+1)$. Suppose that $|M| = m + 1 \geq 2$.

Case 1: Every proper subset[6] of men likes a *strictly larger* set of women. In this case, we have some latitude: we pair an arbitrary man with a woman he likes and send them both away. The matching condition still holds for the remaining men and women since we have removed only one woman, so we can match the rest of the men by induction.

Case 2: Some proper subset of men $X \subset M$ likes an *equal-size* set of women $Y \subset W$. We match the men in X with the women in Y by induction and send them all away. We can also match the rest of the men by induction if we show that the matching condition holds for the remaining men and women. To check the matching condition for the remaining people, consider an arbitrary subset of the remaining men $X' \subseteq (M - X)$, and let Y' be the set of remaining women that they like. We must show that $|X'| \leq |Y'|$. Originally, the combined set of men $X \cup X'$ liked the set of women $Y \cup Y'$. So, by the matching condition, we know:

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

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

So in both cases, there is a matching for the men, which completes the proof of the Inductive step. The theorem follows by induction. ■

The proof of Theorem 11.5.2 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.

A Formal Statement

Let's restate Theorem 11.5.2 in abstract terms so that you'll not always be condemned to saying, "Now this group of men likes at least as many women..."

Definition 11.5.3. A *matching* in a graph G is a set M of edges of G such that no vertex is incident to more than one edge in M. A matching is said to *cover* a set, S,

[6] A subset A of B is *proper* if $A \neq B$, see Section ??.

of vertices iff each vertex in S is incident to an edge of the matching. A matching is said to be *perfect* if it covers $V(G)$. In any graph, the set $N(S)$ of *neighbors* of some set S of vertices is the set of all vertices adjacent to some vertex in S. That is,

$$N(S) ::= \{ r \mid s\text{—}r \text{ is an edge of } G \text{ for some } s \in S \}.$$

S is called a *bottleneck* if

$$|S| > |N(S)|.$$

Theorem 11.5.4 (Hall's Theorem)**.** *Let G be a bipartite graph. There is matching in G that covers $L(G)$ iff no subset of $L(G)$ is a bottleneck.*

An Easy Matching Condition

The bipartite matching condition requires that *every* subset of men has a certain property. In general, verifying that every subset has some property, even if it's easy to check any particular subset for the property, quickly becomes overwhelming because the number of subsets of even relatively small sets is enormous—over a billion subsets for a set of size 30. However, there is a simple property of vertex degrees in a bipartite graph that guarantees the existence of a matching. Namely, call a bipartite graph *degree-constrained* if vertex degrees on the left are at least as large as those on the right. More precisely,

Definition 11.5.5. A bipartite graph G is *degree-constrained* when $\deg(l) \geq \deg(r)$ for every $l \in L(G)$ and $r \in R(G)$.

For example, the graph in Figure 11.9 is degree constrained since every node on the left is adjacent to at least two nodes on the right while every node on the right is incident to at most two nodes on the left.

Theorem 11.5.6. *If G is a degree-constrained bipartite graph, then there is a matching that covers $L(G)$.*

Proof. The proof is by contradiction. Suppose that G is degree constrained but that there is no matching that covers $L(G)$. By Theorem 11.5.4, this means that there must be a bottleneck $S \subseteq L(G)$.

Let d be a value such that $\deg(l) \geq d \geq \deg(r)$ for every $l \in L$ and $r \in R$. Since every edge incident to some node in S is by definition incident to a node in $N(S)$, and every node in $N(S)$ is incident to at most d edges, we know that

$$d|N(S)| \geq \#\text{edges incident to a vertex in } S.$$

Also, since every node in S is incident to at most d edges

$$\#\text{edges incident to a vertex in } S \geq d|S|.$$

It follows that $d|N(S)| \geq d|S|$ and so

$$|N(S)| \geq |S|.$$

This means that S is not a bottleneck, which is a contradiction. Hence G has a matching that covers $L(G)$. ■

Regular graphs are a large class of degree constrained graphs that often arise in practice. Hence, we can use Theorem 11.5.6 to prove that every regular bipartite graph has a perfect matching. This turns out to be a surprisingly useful result in computer science.

Definition 11.5.7. A graph is said to be *regular* if every node has the same degree.

Theorem 11.5.8. *Every regular bipartite graph has a perfect matching.*

Proof. Let G be a regular bipartite graph. Since regular graphs are degree-constrained, we know by Theorem 11.5.6 that there must be a matching in G that covers $L(G)$. Such a matching is only possible when $|L(G)| \geq |R(G)|$. But G is also degree-constrained if the roles of $L(G)$ and $R(G)$ are switched, which implies that $|R(G)| \geq |L(G)|$ also. That is, $L(G)$ and $R(G)$ are the same size, and any matching covering $L(G)$ will also cover $R(G)$. So every node in G is incident to an edge in the matching, and thus G has a perfect matching. ■

need to get good at proofs!

11.6 The Stable Marriage Problem

We next consider a version of the bipartite matching problem where there are an equal number of men and women, and where each person has preferences about who they would like to marry. In fact, we assume that each man has a complete list of all the women ranked according to his preferences, with no ties. Likewise, each woman has a ranked list of all of the men.

The preferences don't have to be symmetric. That is, Jennifer might like Brad best, but Brad doesn't necessarily like Jennifer best. The goal is to marry everyone: every man must marry exactly one woman and vice-versa—no polygamy. Moreover, we would like to find a matching between men and women that is *stable* in the sense that there is no pair of people that prefer each other to their spouses.

For example, suppose *every* man likes Angelina best, and every woman likes Brad best, but Brad and Angelina are married to other people, say Jennifer and Billy Bob. Now *Brad and Angelina prefer each other to their spouses*, which puts their

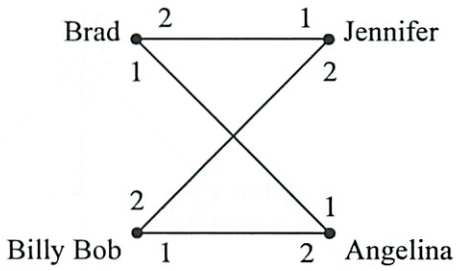


Figure 11.11 Preferences for four people. Both men like Angelina best and both women like Brad best.

marriages at risk: pretty soon, they're likely to start spending late nights together working on problem sets!

This unfortunate situation is illustrated in Figure 11.11, where the digits "1" and "2" near a man shows which of the two women he ranks first second, respectively, and similarly for the women.

More generally, in any matching, a man and woman who are not married to each other and who like each other better than their spouses, is called a *rogue couple*. In the situation shown in Figure 11.11, Brad and Angelina would be a rogue couple.

Having a rogue couple is not a good thing, since it threatens the stability of the marriages. On the other hand, if there are no rogue couples, then for any man and woman who are not married to each other, at least one likes their spouse better than the other, and so they won't be tempted to start an affair.

Definition 11.6.1. A *stable matching* is a matching with no rogue couples.

The question is, given everybody's preferences, how do you find a stable set of marriages? In the example consisting solely of the four people in Figure 11.11, we could let Brad and Angelina both have their first choices by marrying each other. Now neither Brad nor Angelina prefers anybody else to their spouse, so neither will be in a rogue couple. This leaves Jen not-so-happily married to Billy Bob, but neither Jen nor Billy Bob can entice somebody else to marry them, and so there is a stable matching.

Surprisingly, there always is a stable matching among a group of men and women. The surprise springs in part from considering the apparently similar "buddy" matching problem. That is, if people can be paired off as buddies, regardless of gender, then a stable matching *may not* be possible. For example, Figure 11.12 shows a situation with a love triangle and a fourth person who is everyone's last choice. In this figure Mergatroid's preferences aren't shown because they don't even matter. Let's see why there is no stable matching.

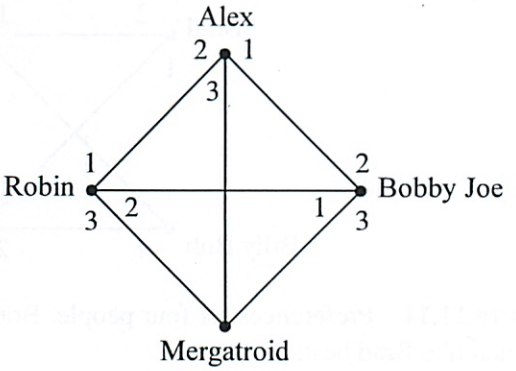


Figure 11.12 Some preferences with no stable buddy matching.

Lemma 11.6.2. *There is no stable buddy matching among the four people in Figure 11.12.*

Proof. We'll prove this by contradiction.

Assume, for the purposes of contradiction, that there is a stable matching. Then there are two members of the love triangle that are matched. Since preferences in the triangle are symmetric, we may assume in particular, that Robin and Alex are matched. Then the other pair must be Bobby-Joe matched with Mergatroid.

But then there is a rogue couple: Alex likes Bobby-Joe best, and Bobby-Joe prefers Alex to his buddy Mergatroid. That is, Alex and Bobby-Joe are a rogue couple, contradicting the assumed stability of the matching. ■

So getting a stable *buddy* matching may not only be hard, it may be impossible. But when mens are only allowed to marry women, and vice versa, then it turns out that a stable matching can always be found.[7]

11.6.1 The Mating Ritual

The procedure for finding a stable matching involves a *Mating Ritual* that takes place over several days. The following events happen each day:

Morning: Each woman stands on her balcony. Each man stands under the balcony of his favorite among the women on his list, and he serenades her. If a man has no women left on his list, he stays home and does his math homework.

Afternoon: Each woman who has one or more suitors serenading her, says to her favorite among them, "We might get engaged. Come back tomorrow." To the other suitors, she says, "No. I will never marry you! Take a hike!"

[7] Once again, we disclaim any political statement here—its just the way that the math works out.

how in all world to prove?

1. break it up

Evening: Any man who is told by a woman to take a hike, crosses that woman off his list.

Termination condition: When a day arrives in which every woman has at most one suitor, the ritual ends with each woman marrying her suitor, if she has one.

There are a number of facts about this Mating Ritual that we would like to prove:

- The Ritual eventually reaches the termination condition.
- Everybody ends up married.
- The resulting marriages are stable.

11.6.2 There is a Marriage Day

It's easy to see why the Mating Ritual has a terminal day when people finally get married. Every day on which the ritual hasn't terminated, at least one man crosses a woman off his list. (If the ritual hasn't terminated, there must be some woman serenaded by at least two men, and at least one of them will have to cross her off his list). If we start with n men and n women, then each of the n men's lists initially has n women on it, for a total of n^2 list entries. Since no women ever gets added to a list, the total number of entries on the lists decreases every day that the Ritual continues, and so the Ritual can continue for at most n^2 days.

same as other P-set qu

11.6.3 They All Live Happily Every After...

We still have to prove that the Mating Ritual leaves everyone in a stable marriage. To do this, we note one very useful fact about the Ritual: if a woman has a favorite suitor on some morning of the Ritual, then that favorite suitor will still be serenading her the next morning—because his list won't have changed. So she is sure to have today's favorite man among her suitors tomorrow. That means she will be able to choose a favorite suitor tomorrow who is at least as desirable to her as today's favorite. So day by day, her favorite suitor can stay the same or get better, never worse. This sounds like an invariant, and it is.

Definition 11.6.3. Let P be the predicate: For every woman, w, and every man, m, if w is crossed off m's list, then w has a suitor whom she prefers over m.

Lemma 11.6.4. *P is an invariant for The Mating Ritual.*

always gets better

Proof. By induction on the number of days.

Base case: In the beginning—that is, at the end of day 0—every woman is on every list. So no one has been crossed off, and P is vacuously true.

Inductive Step: Assume P is true at the end of day d and let w be a woman that has been crossed off a man m's list by the end of day $d+1$.

Case 1: w was crossed off m's list on day $d+1$. Then, w must have a suitor she prefers on day $d+1$.

Case 2: w was crossed off m's list prior to day $d+1$. Since P is true at the end of day d, this means that w has a suitor she prefers to m on day d. She therefore has the same suitor or someone she prefers better at the end of day $d+1$.

In both cases, P is true at the end of day $d+1$ and so P must be an invariant. ■

With Lemma 11.6.4 in hand, we can now prove:

Theorem 11.6.5. *Everyone is married by the Mating Ritual.*

Proof. By contradiction. Assume that it is the last day of the Mating Ritual and someone does not get married. Since there are an equal number of men and women, and since bigamy is not allowed, this means that at least one man (call him Bob) and at least one woman do not get married.

Since Bob is not married, he can't be serenading anybody and so his list must be empty. This means that Bob has crossed every woman off his list and so, by invariant P, every woman has a suitor whom she prefers to Bob. Since it is the last day and every woman still has a suitor, this means that every woman gets married. This is a contradiction since we already argued that at least one woman is *not* married. Hence our assumption must be false and so everyone must be married. ■

Theorem 11.6.6. *The Mating Ritual produces a stable matching.*

Proof. Let Brad and Jen be any man and woman, respectively, that are *not* married to each other on the last day of the Mating Ritual. We will prove that Brad and Jen are not a rogue couple, and thus that all marriages on the last day are stable. There are two cases to consider.

Case 1: Jen is not on Brad's list by the end. Then by invariant P, we know that Jen has a suitor (and hence a husband) that she prefers to Brad. So she's not going to run off with Brad—Brad and Jen cannot be a rogue couple.

Case 2: Jen is on Brad's list. But since Brad is not married to Jen, he must be choosing to serenade his wife instead of Jen, so he must prefer his wife. So he's not going to run off with Jen—once again, Brad and Jenn are not a rogue couple. ■

11.6.4 ...Especially the Men

Who is favored by the Mating Ritual, the men or the women? The women *seem* to have all the power: they stand on their balconies choosing the finest among their suitors and spurning the rest. What's more, we know their suitors can only change for the better as the Ritual progresses. Similarly, a man keeps serenading the woman he most prefers among those on his list until he must cross her off, at which point he serenades the next most preferred woman on his list. So from the man's perspective, the woman he is serenading can only change for the worse. Sounds like a good deal for the women.

But it's not! The fact is that from the beginning, the men are serenading their first choice woman, and the desirability of the woman being serenaded decreases only enough to ensure overall stability. The Mating Ritual actually does as well as possible for all the men and does the worst possible job for the women.

Women don't have top picks come to them!

To explain all this we need some definitions. Let's begin by observing that while The Mating Ritual produces one stable matching, there may be other stable matchings among the same set of men and women. For example, reversing the roles of men and women will often yield a different stable matching among them.

But some spouses might be out of the question in all possible stable matchings. For example, given the preferences shown in Figure 11.11, Brad is just not in the realm of possibility for Jennifer, since if you ever pair them, Brad and Angelina will form a rogue couple.

Graph theory is cool!

Definition 11.6.7. Given a set of preference lists for all men and women, one person is in another person's *realm of possible spouses* if there is a stable matching in which the two people are married. A person's *optimal spouse* is their most preferred person within their realm of possibility. A person's *pessimal spouse* is their least preferred person in their realm of possibility.

Everybody has an optimal and a pessimal spouse, since we know there is at least one stable matching, namely, the one produced by the Mating Ritual. Now here is the shocking truth about the Mating Ritual:

Theorem 11.6.8. *The Mating Ritual marries every man to his optimal spouse.*

Proof. By contradiction. Assume for the purpose of contradiction that some man does not get his optimal spouse. Then there must have been a day when he crossed off his optimal spouse—otherwise he would still be serenading (and would ultimately marry) her or some even more desirable woman.

By the Well Ordering Principle, there must be a *first* day when a man (call him "Keith") crosses off his optimal spouse (call her Nicole). According to the rules of

the Ritual, Keith crosses off Nicole because Nicole has a preferred suitor (call him Tom), so

Nicole prefers Tom to Keith. (∗)

Since this is the first day an optimal woman gets crossed off, we know that Tom had not previously crossed off his optimal spouse, and so

Tom ranks Nicole at least as high as his optimal spouse. (∗∗)

By the definition of an optimal spouse, there must be some stable set of marriages in which Keith gets his optimal spouse, Nicole. But then the preferences given in (∗) and (∗∗) imply that Nicole and Tom are a rogue couple within this supposedly stable set of marriages (think about it). This is a contradiction. ■

Theorem 11.6.9. *The Mating Ritual marries every woman to her pessimal spouse.*

Proof. By contradiction. Assume that the theorem is not true. Hence there must be a stable set of marriages $\mathcal{M}$ where some woman (call her Nicole) is married to a man (call him Tom) that she likes less than her spouse in The Mating Ritual (call him Keith). This means that

Nicole prefers Keith to Tom. (+)

By Theorem 11.6.8 and the fact that Nicole and Keith are married in the Mating Ritual, we know that

Keith prefers Nicole to his spouse in $\mathcal{M}$. (++)

This means that Keith and Nicole form a rogue couple in $\mathcal{M}$, which contradicts the stability of $\mathcal{M}$. ■

11.6.5 Applications

The Mating Ritual was first announced in a paper by D. Gale and L.S. Shapley in 1962, but ten years before the Gale-Shapley paper was published, and unknown by them, a similar algorithm was being used to assign residents to hospitals by the National Resident Matching Program (NRMP)[8]. The NRMP has, since the turn of the twentieth century, assigned each year's pool of medical school graduates to hospital residencies (formerly called "internships") with hospitals and graduates playing the roles of men and women. (In this case, there may be multiple women married to one man, a scenario we consider in the problem section at the end of the

[8] Of course, there is no serenading going on in the hospitals—the preferences are submitted to a program and the whole process is carried out by a computer.

chapter.). Before the Ritual-like algorithm was adopted, there were chronic disruptions and awkward countermeasures taken to preserve assignments of graduates to residencies. The Ritual resolved these problems so successfully, that it was used essentially without change at least through 1989.[9]

The Internet infrastructure company, Akamai, also uses a variation of the Mating Ritual to assign web traffic to its servers. In the early days, Akamai used other combinatorial optimization algorithms that got to be too slow as the number of servers (over 65,000 in 2010) and requests (over 800 billion per day) increased. Akamai switched to a Ritual-like approach since it is fast and can be run in a distributed manner. In this case, web requests correspond to women and web servers correspond to men. The web requests have preferences based on latency and packet loss, and the web servers have preferences based on cost of bandwidth and colocation.

Not surprisingly, the Mating Ritual is also used by at least one large online dating agency. Even here, there is no serenading going on—everything is handled by computer.

need to understand - perhaps this Summer

Cool!

11.7 Coloring

3/31

In Section 11.2, we used edges to indicate an affinity between a pair of nodes. But there are lots of situations where edges will correspond to *conflicts* between nodes. Exam scheduling is a typical example.

11.7.1 An Exam Scheduling Problem

after did in class

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 (even at MIT) a student can take only one test during a particular time slot. The Schedules Office wants to avoid all conflicts. Of course, you can make such a schedule by having every exam in a different slot, but then you would need hundreds of slots for the hundreds of courses, and the exam period would run all year! So, the Schedules Office would also like to keep exam period short.

The Schedules Office's problem is easy to describe as a graph. There will be a vertex for each course with a final exam, and two vertices will be adjacent exactly when some student is taking both courses. For example, suppose we need to schedule exams for 6.041, 6.042, 6.002, 6.003 and 6.170. The scheduling graph might

[9] Much more about the Stable Marriage Problem can be found in the very readable mathematical monograph by Dan Gusfield and Robert W. Irving, The Stable Marriage Problem: Structure and Algorithms, MIT Press, Cambridge, Massachusetts, 1989, 240 pp.

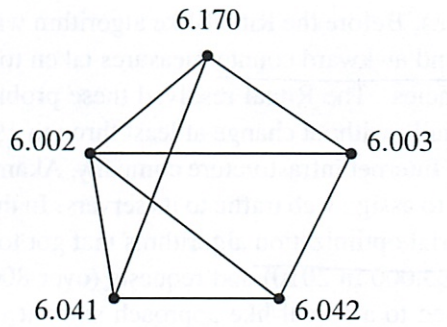


Figure 11.13 A scheduling graph for five exams. Exams connected by an edge cannot be given at the same time.

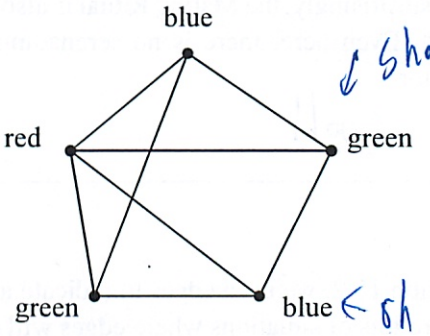


Figure 11.14 A 3-coloring of the exam graph from Figure 11.13.

appear as in Figure 11.13.

6.002 and 6.042 cannot have an exam at the same time since there are students in both courses, so there is an edge between their nodes. On the other hand, 6.042 and 6.170 can have an exam at the same time if they're taught at the same time (which they sometimes are), since no student can be enrolled in both (that is, no student *should* be enrolled in both when they have a timing conflict).

We 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 then 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*. As shown in Figure 11.14, three colors suffice for our example.

The coloring in Figure 11.14 corresponds to giving one final on Monday morning (red), two Monday afternoon (blue), and two Tuesday morning (green). Can we use

fewer than three colors? No! We can't use only two colors since there is a triangle in the graph, and three vertices in a triangle must all have different colors.

This is an example of a *graph coloring* problem: given a graph G, assign colors to each node such that adjacent nodes have different colors. A color assignment with this property is called a *valid coloring* of the graph—a "*coloring*," for short. A graph G is *k-colorable* if it has a coloring that uses at most k colors.

Definition 11.7.1. The minimum value of k for which a graph, G, has a valid coloring is called its *chromatic number*, $\chi(G)$.

So G is k-colorable iff $\chi(G) \leq k$.

In general, trying to figure out if you can color a graph with a fixed number of colors can take a long time. It's a classic example of a problem for which no fast algorithms are known. In fact, it is easy to check if a coloring works, but it seems really hard to find it. (If you figure out how, then you can get a $1 million Clay prize.)

11.7.2 Some Coloring Bounds

There are some simple properties of graphs that give useful bounds on colorability. The simplest property is being a cycle: an even-length closed cycle is 2-colorable, and since by definition it must have some edges, it is not 1-colorable. So

$$\chi(C_{\text{even}}) = 2.$$

On the other hand, an odd-length cycle requires 3 colors, that is,

$$\chi(C_{\text{odd}}) = 3. \tag{11.1}$$

You should take a moment to think about why this equality holds. Another simple example is a complete graph K_n:

$$\chi(K_n) = n$$

since no two vertices can have the same color.

Being bipartite is another property closely related to colorability. If a graph is bipartite, then you can color it with 2 colors usiing one color for the nodes on the "left" and a second color for the nodes on the "right." Conversely, graphs with chromatic number 2 are all bipartite with all the vertices of one color on the "left" and those with the other color on the right. The graphs with chromatic number 1 are the graphs with no edges—the *empty graphs*. Empty graphs are bipartite as long they have at least two vertices: a graph with only one vertex is not bipartite because its vertex set cannot be partitioned into two *nonempty* subsets. In short:

Lemma 11.7.2. *A graph with more than one vertex is 2-colorable iff it is bipartite.*

The chromatic number of a graph can also be shown to be small if the vertex degrees of the graph are small. In particular, if we have an upper bound on the degrees of all the vertices in a graph, then we can easily find a coloring with only one more color than the degree bound.

Theorem 11.7.3. *A graph with maximum degree at most k is $(k+1)$-colorable.*

Since k is the only nonnegative integer valued variable mentioned in the theorem, you might be tempted to try to prove this theorem using induction on k. Unfortunately, this approach leads to disaster—we don't know of any reasonable way to do this and expect it would ruin your week if you tried it on a problem set. When you encounter such a disaster using induction on graphs, it is usually best to change what you are inducting on. In graphs, typical good choices for the induction parameter are n, the number of nodes, or e, the number of edges.

Proof of Theorem 11.7.3. 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.

Base case ($n = 1$): A 1-vertex graph has maximum degree 0 and is 1-colorable, so $P(1)$ is true.

Inductive step: 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 all edges incident to it), leaving an n-vertex subgraph, H. The maximum degree of H is at most k, and so H is $(k+1)$-colorable by our assumption $P(n)$. Now add back vertex v. We can assign v a color (from the set of $k+1$ colors) that is different from all its adjacent vertices, since there are at most k vertices adjacent to v and so at least one of the $k+1$ colors is still available. Therefore, G is $(k+1)$-colorable. This completes the inductive step, and the theorem follows by induction. ■

Sometimes $k+1$ colors is the best you can do. For example, $\chi(K_n) = n$ and every node in K_n has degree $k = n-1$ and so this is an example where Theorem 11.7.3 gives the best possible bound. By a similar argument, we can show that Theorem 11.7.3 gives the best possible bound for *any* graph with degree bounded by k that has K_{k+1} as a subgraph.

But sometimes $k+1$ colors is far from the best that you can do. For example, the n-node *star graph* shown in Figure 11.15 has maximum degree $n-1$ but can be colored using just 2 colors.

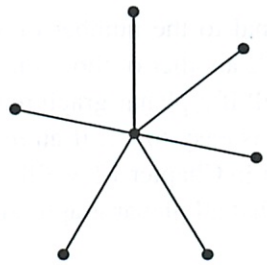


Figure 11.15 A 7-node star graph.

11.7.3 Why coloring?

One reason coloring problems frequently arise in practice is because scheduling conflicts are so common. For example, at Akamai, a new version of software is deployed over each of 65,000 servers every few days. The updates cannot be done at the same time since the servers need to be taken down in order to deploy the software. Also, the servers cannot be handled one at a time, since it would take forever to update them all (each one takes about an hour). Moreover, certain pairs of servers cannot be taken down at the same time since they have common critical functions. This problem was eventually solved by making a 65,000-node conflict graph and coloring it with 8 colors—so only 8 waves of install are needed!

Another example comes from the need to assign frequencies to radio stations. If two stations have an overlap in their broadcast area, they can't be given the same frequency. Frequencies are precious and expensive, so you want to minimize the number handed out. This amounts to finding the minimum coloring for a graph whose vertices are the stations and whose edges connect stations with overlapping areas.

Coloring also comes up in allocating registers for program variables. While a variable is in use, its value needs to be saved in a register. Registers can be reused for different variables but two variables need different registers if they are referenced during overlapping intervals of program execution. So register allocation is the coloring problem for a graph whose vertices are the variables: vertices are adjacent if their intervals overlap, and the colors are registers. Once again, the goal is to minimize the number of colors needed to color the graph.

Finally, there's the famous map coloring problem stated in Proposition 1.1.6. The question is how many colors are needed to color a map so that adjacent territories get different colors? This is the same as the number of colors needed to color a graph that can be drawn in the plane without edges crossing. A proof that four colors are enough for *planar* graphs was acclaimed when it was discovered about thirty years ago. Implicit in that proof was a 4-coloring procedure that takes

time proportional to the number of vertices in the graph (countries in the map). Surprisingly, it's another of those million dollar prize questions to find an efficient procedure to tell if a planar graph really *needs* four colors or if three will actually do the job. (It is easy to tell if an *arbitrary* graph is 2-colorable, as explained in Section 11.10.) In Chapter 12, we'll develop enough planar graph theory to present an easy proof that all planar graphs are 5-colorable.

11.8 Getting from u to v in a Graph

Walks and paths in simple graphs are esentially the same as in digraphs. We just modify the digraph definitions using undirected edges instead of directed ones. For example, the formal definition of a walk in a simple graph is a virtually that same as Definition 9.2.1 of walks in digraphs:

Definition 11.8.1. A *walk in a simple graph*, G, is an alternating sequence of vertices and edges that begins with a vertex, ends with a vertex, and such that for every edge u—v in the walk, one of the endpoints u, v is the element just before the edge, and the other endpoint is the next element after the edge.

So a walk, $\mathbf{v}$, is a sequence of the form

Same as before $\mathbf{v} ::= v_0\ v_0\text{—}v_1\ v_1\ v_1\text{—}v_2\ v_2\ \ldots\ v_{k-1}\text{—}v_k\ v_k$

where v_i—$v_{i+1} \in E(G)$ for $i \in [0, k)$. The walk is said to *start* at v_0, to *end* at v_k, and the *length*, $|\mathbf{v}|$, of the walk is defined to be k. The walk is a *path* iff all the v_i's are different, that is, if $i \neq j$, then $v_i \neq v_j$.

A *closed walk* is a walk that begins and ends at the same vertex. A *cycle* is a closed walk of length three or more whose vertices are distinct except for the beginning and end vertices.

Note that a single vertex counts as a length zero path and closed walk. But in contrast to digraphs, a single vertex is not considered to be a cycle.

For example, the graph in Figure 11.16 has a length 6 path A, B, C, D, E, F, G. This is the longest path in the graph.

As in digraphs, the length of a walk is the total number of edges in it, which is *one less* than its length as a sequence of vertices. For example, the length 6 path A,B,C,D,E,F,G is actually a sequence of seven vertices.

The vertex sequence B, C, D, E, C, B describes a closed walk in the graph in Figure 11.16. This sequence suggests indicates that the walk begins and ends at vertex B, but a closed walk isn't intended to have a beginning and end, and can be

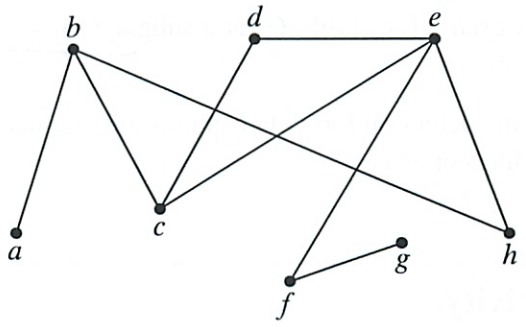


Figure 11.16 *A graph with 3 cycles.*

described by *any* of the paths that go around it. For example, D, E, C, B, C, D describes this same closed walk as though it started and ended at D, and D, C, B, C, E, D describes the same closed walk as though it started and ended at D but went in the opposite direction.

All the paths that describe the same closed walk have the same length which is defined to be the *length of the walk*. (Note that this implies that going around the same walk twice is considered to be different than going around it once.)

The graph in Figure 11.16 has three cycles B, H, E, C, B and C, D, E, C and B, C, D, E, H, B.

11.8.1 Subgraphs

A precise way to explain which closed walks describe the same cycle is to define cycle as a subgraph instead of as a closed walk. Namely, we could define a cycle in G to be a *subgraph* of G that looks like a length-n cycle for $n \geq 3$.

Definition 11.8.2. A graph G is said to be a *subgraph* of a graph H if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$.

abstract/reduce to subgraph

For example, the one-edge graph G where

$$V(G) = \{g, h, i\} \quad \text{and} \quad E(G) = \{h\text{—}i\}$$

is a subgraph of the graph H in Figure 11.1. On the other hand, any graph containing an edge g—h will not be a subgraph of H because this edge is not in $E(H)$. Another example is an empty graph on n nodes, which will be a subgraph of an L_n with same set of nodes; similarly, L_n is a subgraph of C_n, and C_n is a subgraph of K_n.

Definition 11.8.3. For $n \geq 3$, let C_n be the graph with vertices $1, \ldots, n$ and edges

$$1\text{—}2, \quad 2\text{—}3, \quad \ldots, \quad (n-1)\text{—}n, \quad n\text{—}1.$$

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

This definition formally captures the idea that cycles don't have direction or beginnings or ends.

11.9 Connectivity

Definition 11.9.1. Two vertices in a graph are *connected* when there is a path that begins at one and ends at the other. By convention, every vertex is connected to itself by a path of length zero.

Definition 11.9.2. A graph is *connected* when every pair of vertices are connected.

11.9.1 Connected Components

Being connected is usually a good property for a graph to have. For example, it could mean that it is possible to get from any node to any other node, or that it is possible to communicate between any pair of nodes, depending on the application.

But not all graphs are connected. For example, the graph where nodes represent cities and edges represent highways might be connected for North American cities, but would surely not be connected if you also included cities in Australia. The same is true for communication networks like the Internet—in order to be protected from viruses that spread on the Internet, some government networks are completely isolated from the Internet.

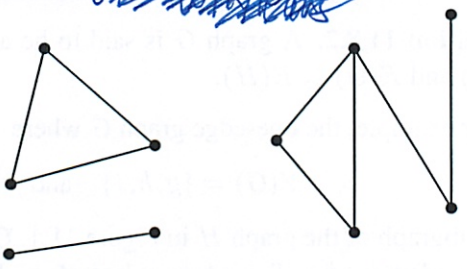
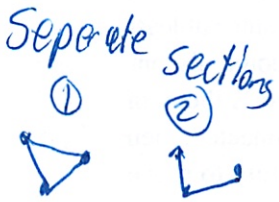


Figure 11.17 One graph with 3 connected components.

Another example, is shown in Figure 11.17, which looks like a picture of three graphs, but is intended to be a picture of *one* graph. This graph consists of three pieces (subgraphs). Each piece by itself is connected, but there are no paths be-

tween vertices in different pieces. These connected pieces of a graph are called its *connected components*.



Definition 11.9.3. A *connected component* of a graph is a subgraph consisting of some vertex and every node and edge that is connected to that vertex.

So a graph is connected iff it has exactly one connected component. At the other extreme, the empty graph on n vertices has n connected components.

11.9.2 k-Connected Graphs

If we think of a graph as modeling 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. So more generally we want to define how strongly two vertices are connected. One measure of connection strength is how many links must fail before connectedness fails. In particular, two vertices are *k-edge connected* when it takes at least k "edge-failures" to disconnect them.. More precisely:

Definition 11.9.4. Two vertices in a graph are *k-edge connected* when they remain connected in every subgraph obtained by deleting up to $k-1$ edges. A graph is k-edge connected when it has more than one vertex, and every subgraph obtained by deleting at most $k-1$ edges is connected.

So two vertices are connected according to Definition 11.9.1 iff they are 1-edge connected according to Definition 11.9.4; likewise for any graph with more than one vertex.

There are other kinds of connectedness but edge-connectedness will be enough for us, so from now on we'll drop the "edge" modifier and just say "connected."[10]

For example, in the graph in Figure 11.16, vertices c and e are 3 connected, b and e are 2 connected, g and e are 1 connected, and no vertices are 4 connected. The graph as a whole is only 1 connected. A complete graph, K_n, is $(n-1)$ connected. Every cycle is 2-connected.

The idea of a *cut edge* is a useful way to explain 2-connectivity.

Definition 11.9.5. If two vertices are connected in a graph G, but not connected when an edge e is removed, then e is called a *cut edge* of G.

So a graph with more than one vertex is 2-connected iff it is connected, and has no cut edges. The following Lemma is another immediate consequence of the definition:

[10] There is an obvious definition of k-vertex connectedness based on deleting vertices rather than edges. Graph theory texts usually use "k-connected" as shorthand for "k-vertex connected."

Lemma 11.9.6. *An edge is a cut-edge iff it is not on a cycle.*

More generally, if two vertices are connected by k edge-disjoint paths—that is, no two paths traverse the same edge—then they must be k connected, since at least one edge will have to be removed from each of the paths before they could disconnect. 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 this just for the case $k = 2$.

11.9.3 The Minimum Number of Edges in a Connected Graph

The following theorem says that a graph with few edges must have many connected components.

Theorem 11.9.7. *Every graph with v vertices and e edges has at least $v - e$ connected components.*

Of course for Theorem 11.9.7 to be of any use, there must be fewer edges than vertices.

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

> for every v, every graph with v vertices and e edges has at least $v - e$ connected components.

Base case:($e = 0$). In a graph with 0 edges and v vertices, each vertex is itself a connected component, and so there are exactly $v = v - 0$ connected components. So $P(e)$ holds.

Inductive step: Now we assume that the induction hypothesis holds for every e-edge graph in order to prove that it holds for every $(e+1)$-edge graph, where $e \geq 0$. Consider a graph, G, with $e + 1$ edges and v vertices. We want to prove that G has at least $v - (e + 1)$ connected components. To do this, remove an arbitrary edge a—b and call the resulting graph G'. By the induction assumption, G' has at least $v - e$ connected components. Now add back the edge a—b to obtain the original graph G. If a and b were in the same connected component of G', then G has the same connected components as G', so G has at least $v-e > v-(e+1)$ components. Otherwise, if a and b were in different connected components of G', then these two components are merged into one component in G, but all other components remain unchanged, reducing the number of components by 1. Therefore, G has at least $(v-e)-1 = v-(e+1)$ connected components. So in either case, $P(e+1)$ holds. This completes the Inductive step. The theorem now follows by induction. ■

Figure 11.18 A counterexample graph to the False Claim.

Corollary 11.9.8. *Every connected graph with v vertices has at least $v - 1$ edges.*

A couple of points about the proof of Theorem 11.9.7 are worth noticing. First, we used induction on the number of edges in the graph. This is very common in proofs involving graphs, as 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 nasty logical error called *buildup error*.

went over in class

Build-Up Error

False Claim. *If every vertex in a graph has degree at least 1, then the graph is connected.*

My thought error in class — I actually assumed this at first

There are many counterexamples; for example, see Figure 11.18.

Bogus proof. We use induction. Let $P(n)$ be the proposition that if every vertex in an n-vertex graph has degree at least 1, then the graph is connected.

Base case ($n = 1$): There is only one graph with a single vertex and it has degree 0. Therefore, $P(1)$ is vacuously true, since the if-part is false.

Inductive step: We must show that $P(n)$ implies $P(n+1)$ for all $n \geq 1$. Consider an n-vertex graph in which every vertex has degree at least 1. By the assumption $P(n)$, this graph is connected; that is, there is a path between every pair of vertices. Now we add one more vertex x to obtain an $(n+1)$-vertex graph as shown in Figure 11.19.

All that remains is to check that there is a path from x to every other vertex z. Since x has degree at least one, there is an edge from x to some other vertex; call

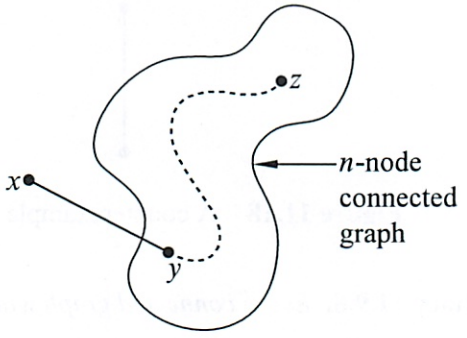


Figure 11.19 Adding a vertex x with degree at least 1 to a connected n-node graph.

it y. Thus, we can obtain a path from x to z by adjoining the edge x—y to the path from y to z. This proves $P(n+1)$.

By the principle of induction, $P(n)$ is true for all $n \geq 1$, which proves the theorem ■

Uh-oh... this proof looks fine! Where is the bug? It turns out that the faulty assumption underlying this argument is that *every $(n+1)$-vertex graph with minimum degree 1 can be obtained from an n-vertex graph with minimum degree 1 by adding 1 more vertex.* Instead of starting by considering an arbitrary $(n+1)$-node graph, this proof only considered $(n+1)$-node graphs that you can make by starting with an n-node graph with minimum degree 1.

The counterexample in Figure 11.18 shows that this assumption is false; there is no way to build the 4-vertex graph in Figure 11.18 from a 3-vertex graph with minimum degree 1. Thus the first error in the proof is the statement "This proves $P(n+1)$."

This kind of flaw is known as "build-up error." Usually, build-up error arises from a faulty assumption that every size $n+1$ graph with some property can be "built up" from a size n graph with the same property. (This assumption is correct for some properties, but incorrect for others—such as the one in the argument above.)

One way to avoid an accidental build-up error is to use a "shrink down, grow back" process in the inductive step, namely, start with a size $n+1$ graph, remove a vertex (or edge), apply the inductive hypothesis $P(n)$ to the smaller graph, and then add back the vertex (or edge) and argue that $P(n+1)$ holds. Let's see what would have happened if we'd tried to prove the claim above by this method:

Revised inductive step: We must show that $P(n)$ implies $P(n+1)$ for all $n \geq 1$.

Consider an $(n+1)$-vertex graph G in which every vertex has degree at least 1. Remove an arbitrary vertex v, leaving an n-vertex graph G' in which every vertex has degree... uh oh!

The reduced graph G' might contain a vertex of degree 0, making the inductive hypothesis $P(n)$ inapplicable! We are stuck—and properly so, since the claim is false!

Always use shrink-down, grow-back arguments and you'll never fall into this trap.

11.10 Odd Cycles and 2-Colorability

We have already seen that determining the chromatic number of a graph is a challenging problem. There is one special case where this problem is very easy, namely, when the graph is 2-colorable.

Theorem 11.10.1. *The following graph properties are equivalent:*

1. *The graph contains an odd length cycle.*
2. *The graph is not 2-colorable.*
3. *The graph contains an odd length closed walk.*

In other words, if a graph has any one of the three properties above, then it has all of the properties.

We will show the following implications among these properties:

1. IMPLIES 2. IMPLIES 3. IMPLIES 1.

So each of these properties implies the other two, which means they all are equivalent.

1 IMPLIES 2 *Proof.* This follows from equation 11.1. ■

2 IMPLIES 3 If we prove this implication for connected graphs, then it will hold for an arbitrary graph because it will hold for each connected component. So we can assume that G is connected.

Proof. Pick an arbitrary vertex r of G. Since G is connected, for every node $u \in V(G)$, there will be a walk $\mathbf{w}_u$ starting at u and ending at r. Assign

colors to vertices of G as follows:

$$\text{color}(u) = \begin{cases} \text{black}, & \text{if } |\mathbf{w}_u| \text{ is even,} \\ \text{white}, & \text{otherwise.} \end{cases}$$

Now since G is not colorable, this can't be a valid coloring. So there must be an edge between two nodes u and v with the same color. But in that case

$$\mathbf{w}_u \, \widehat{r} \, \text{reverse}(\mathbf{w}_v) \, \widehat{v} \, vu.$$

is a closed walk starting and ending at u, and its length is

$$|\mathbf{w}_u| + |\mathbf{w}_v| + 1.$$

This length is odd, since $\mathbf{w}_u$ and $\mathbf{w}_v$ are both even length or are both odd length. ■

3 IMPLIES 1 *Proof.* Since there is an odd length closed walk, the WOP implies there is a odd length closed walk $\mathbf{w}$ of minimum length. We claim $\mathbf{w}$ must be a cycle. To show this, assume to the contrary that there is vertex x that appears twice on the walk, so $\mathbf{w}$ consists of a closed walk from x to x followed by another such walk. That is,

$$\mathbf{w} = \mathbf{f} \, \widehat{x} \, \mathbf{r}$$

for some positive length walks $\mathbf{f}$ and $\mathbf{r}$ that begin and end at x. Since

$$|\mathbf{w}| = |\mathbf{f}| + |\mathbf{r}|$$

is odd, exactly one of $\mathbf{f}$ and $\mathbf{g}$ must have odd length, and that one will be an odd length closed walk shorter than $\mathbf{w}$, a contradiction.

■

This completes the proof of Theorem 11.10.1.

Theorem 11.10.1 turns out to be useful since bipartite graphs come up fairly often in practice. We'll see examples when we talk about planar graphs in Chapter 12.

11.11 Trees

As we have just seen, finding good cycles in a graph can be trickier than you might first think. But what if a graph has no cycles at all? Sounds pretty dull. But graphs without cycles, called *acyclic graphs*, are probably the most important graphs of all when it comes to computer science.

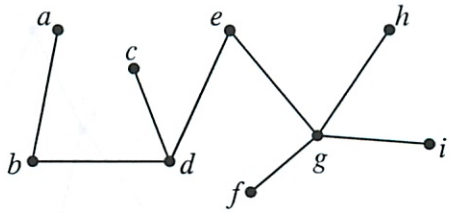


Figure 11.20 A 9-node tree.

Figure 11.21 A 6-node forest consisting of 2 component trees. Note that this 6-node graph is not itself a tree since it is not connected.

11.11.1 Definitions

Definition 11.11.1. A connected acyclic graph is called a *tree*.

For example, Figure 11.20 shows an example of a 9-node tree.

The graph shown in Figure 11.21 is not a tree since it is not connected, but it is a forest. That's because, of course, it consists of a collection of trees.

Definition 11.11.2. If every connected component of a graph G is a tree, then G is a *forest*.

One of the first things you will notice about trees is that they tend to have a lot of nodes with degree one. Such nodes are called *leaves*.

Definition 11.11.3. A *leaf* is a node with degree 1 in a tree (or forest).

For example, the tree in Figure 11.20 has 5 leaves and the forest in Figure 11.21 has 4 leaves.

Trees are a fundamental data structure in computer science. For example, information is often stored in tree-like data structures and the execution of many recursive programs can be modeled as the traversal of a tree. In such cases, it is often useful to draw the tree in a leveled fashion where the node in the top level is identified as the *root*, and where every edge joins a *parent* to a *child*. For example, we have redrawn the tree from Figure 11.20 in this fashion in Figure 11.22. In this example, node d is a child of node e and a parents of nodes b and c.

In the special case of *ordered binary trees*, every node is the parent of at most 2 children and the children are labeled as being a left-child or a right-child.

tree
- graph
- connected
- acyclic

forest
- collection of trees
- each is a Connected Component

only two branches 0 or 1

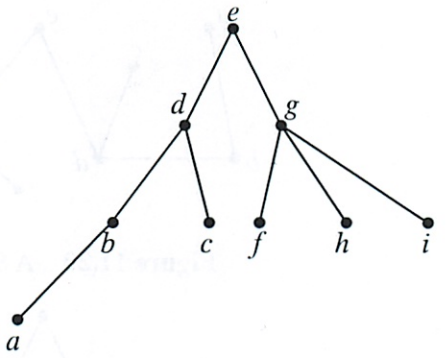


Figure 11.22 The tree from Figure 11.20 redrawn in a leveled fashion, with node E as the root.

11.11.2 Properties

Trees have many unique properties. We have listed some of them in the following theorem.

Theorem 11.11.4. *Every tree has the following properties:*

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

Proof. 1. A cycle in a subgraph is also a 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. Since a tree is connected, there is at least one path between every pair of vertices. Suppose for the purposes of contradiction, that there are two different paths between some pair of 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. (For example, see Figure 11.23.) Then there are two paths from x to y with no common edges, which defines a cycle. This is a contradiction, since trees are acyclic. Therefore, there is exactly one path between every pair of vertices.

Figure 11.23 If there are two paths between u and v, the graph must contain a cycle.

3. An additional edge u—v together with the unique path between u and v forms a cycle.

4. Suppose that we remove edge u—v. Since the 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. Since the tree has at least two vertices, the longest path in the tree will have different endpoints u and v. There cannot be an edge from an endpoint to any other vertex on the path, because that would form a cycle. There also can't be an edge from an endpoint to any vertex not on the path, because that make a longer path. So both endpoints must be leaves.

6. We use induction on the proposition

$$P(n) ::= \text{there are } n-1 \text{ edges in any } n\text{-vertex tree.}$$

Base case ($n = 1$): $P(1)$ is true since a tree with 1 node has 0 edges and $1 - 1 = 0$.

Inductive step: Now suppose that $P(n)$ is true and consider an $(n+1)$-vertex tree, T. Let v be a leaf of the tree. You can verify that deleting a vertex of degree 1 (and its incident edge) from any connected graph leaves a connected subgraph. So by part 1 of Theorem 11.11.4, deleting v and its incident edge gives a smaller tree, and this smaller tree $n-1$ edges by induction. If we re-attach the vertex, v, and its incident edge, we find that T has $= (n+1) - 1$ edges. Hence, $P(n+1)$ is true, and the induction proof is complete. ■

Various subsets of properties in Theorem 11.11.4 provide alternative characterizations of trees. For example,

Lemma 11.11.5. *A graph G is a tree iff G is a forest and $|V(G)| = |E(G)| + 1$.*

We'll leave the proof as an exercise.

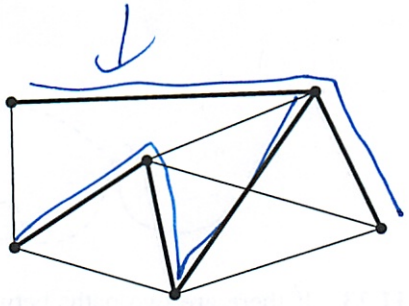


Figure 11.24 A graph where the edges of a spanning tree have been thickened.

11.11.3 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 a called a *spanning tree* for the graph. For example, Figure 11.24 is a connected graph with a spanning tree highlighted.

Theorem 11.11.6. *Every connected graph contains a spanning tree.*

Proof. Suppose G is a connected graph. Define a *spanning subgraph* of G to be a subgraph in which every pair of vertices of G are connected. Because the graph G is connected, G itself is a spanning subgraph. So by WOP, there must be a minimum-edge spanning subgraph T. We claim T is a spanning tree. Since T is a spanning subgraph by definition, all we have to show is that T is acyclic.

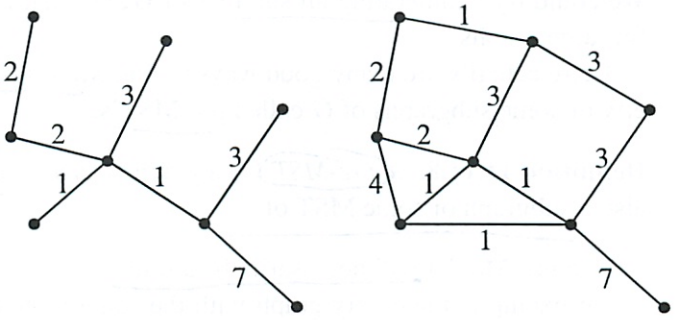
But suppose to the contrary that T contained a cycle C. By Lemma 11.9.6, an edge e of C will not be a cut edge, so removing it would leave a spanning subgraph that was smaller than T, contradicting the minimality to T. ■

11.11.4 Minimum Weight Spanning Trees

Spanning trees are interesting because they connect all the nodes of a graph using the smallest possible number of edges. For example the spanning tree for the 6-node graph shown in Figure 11.24 has 5 edges.

Spanning trees are very useful in practice, but in the real world, not all spanning trees are equally desirable. That's because, in practice, there are often costs associated with the edges of the graph.

For example, suppose the nodes of a graph represent buildings or towns and edges represent connections between buildings or towns. The cost to actually make a connection may vary a lot from one pair of buildings or towns to another. The cost might depend on distance or topography. For example, the cost to connect LA to NY might be much higher than that to connect NY to Boston. Or the cost of a pipe through Manhattan might be more than the cost of a pipe through a cornfield.



(a) (b)

Figure 11.25 A spanning tree (a) with weight 19 for a graph (b).

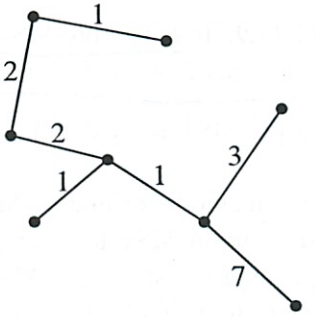


Figure 11.26 An MST with weight 17 for the graph in Figure 11.25(b).

In any case, we typically represent the cost to connect pairs of nodes with a weighted edge, where the weight of the edge is its cost. The weight of a spanning tree is then just the sum of the weights of the edges in the tree. For example, the weight of the spanning tree shown in Figure 11.25 is 19.

The goal, of course, is to find the spanning tree with minimum weight, called the min-weight spanning tree (MST for short).

Definition 11.11.7. A *minimum-weight spanning tree* (MST) of an edge-weighted graph G is a spanning tree of G with the smallest possible sum of edge weights.

Is the spanning tree shown in Figure 11.25(a) an MST of the weighted graph shown in Figure 11.25(b)? Actually, it is not, since the tree shown in Figure 11.26 is also a spanning tree of the graph shown in Figure 11.25(b), and this spanning tree has weight 17.

What about the tree shown in Figure 11.26? Is it an MST? It seems to be, but how do we prove it? In general, how do we find an MST for a connected graph G?

We could try enumerating all subtrees of G, but that approach would be hopeless for large graphs.

There actually are many good ways to find MST's based on an invariance property of some subgraphs of G called pre-MST's.

Definition 11.11.8. A *pre-MST* for a graph G is a spanning subgraph of G that is also a subgraph of some MST of G.

So a pre-MST it will necessarily be a forest.

For example, the empty graph with the same vertices as G guaranteed to be a pre-MST of G, and so is any actual MST of G.

If e is an edge of G and S is a spanning subgraph, we'll write $S + e$ for the spanning subgraph with edges $E(S) \cup \{e\}$.

Definition 11.11.9. If F is a pre-MST and e is a new edge, that is $e \in E(G) - E(F)$, then e *extends* F when $F + e$ is also a pre-MST.

So being a pre-MST is by definition an invariant under addition of extending edges.

The standard methods for finding MST's all start with the empty spanning forest and build up to an MST by adding one extending edge after another. Since the empty spanning forest is a pre-MST, and being a pre-MST is invariant under extensions, every forest built in this way will be a pre-MST. But no spanning tree can be a subgraph of a different spanning tree. So when the pre-MST finally grows enough to become a tree, it will be an MST. By Lemma 11.11.5, this happens after exactly $|V(G)| - 1$ edge extensions.

So the problem of finding MST's reduces to the question of how to tell if an edge is an extending edge. Here's how:

Definition 11.11.10. Let F be a forest, and color the vertices in each connected component of F either all black or all white. At least one component of each color is required. Call this a *solid coloring* of F. A *gray edge* of a solid coloring is an edge of G with different colored endpoints.

Any path in G from a white vertex to a black vertex obviously must traverse a gray vertex, so for any solid coloring, there is guaranteed to be at least one gray edge. In fact, there will have to be at least as many gray edges as there are components with the same color. Here's the punchline:

Lemma 11.11.11. *An edge extends a pre-MST F if it is a minimum weight gray edge in some solid coloring of F.*

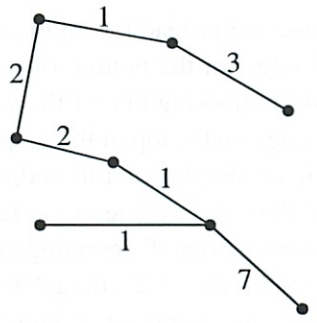


Figure 11.27 A spanning tree found by Algorithm 1.

So to extend a pre-MST, choose any solid coloring, find the gray edges, and among them choose one with minimum weight. Each of these steps is easy to do, so it is easy to keep extending and arrive at an MST. For example, here are three nown algorithms that are explained by Lemma 11.11.11:

Algorithm 1. *Grow a tree one edge at a time by adding a minimum weight edge among the edges that have exactly one endpoint in the tree.*

This is the algorithm that comes from coloring the growing tree white and all the vertices not in the tree black. Then the gray edges are the ones with exactly one endpoint in the tree.

Algorithm 2. *Grow a forest one edge at a time by adding a minimum-weight edge among the edges with endpoints in different connected components.*

The edges between different component are exactly the edges that are gray under some solid coloring, namely any coloring where the components it connects have different colors.

For example, in the weighted graph we have been considering, we might run Algorithm 1 as follows. We would start by choosing one of the weight 1 edges, since this is the smallest weight in the graph. Suppose we chose the weight 1 edge on the bottom of the triangle of weight 1 edges in our graph. This edge is incident to the same vertex as two weight 1 edges, a weight 4 edge, a weight 7 edge, and a weight 3 edge. We would then choose the incident edge of minimum weight. In this case, one of the two weight 1 edges. At this point, we cannot choose the third weight 1 edge: it won't be gray because its endpoints are both in the tree, and so are both colored white. But we can continue by choosing a weight 2 edge. We might end up with the spanning tree shown in Figure 11.27, which has weight 17, the smallest we've seen so far.

Now suppose we instead ran Algorithm 2 on our graph. We might again choose the weight 1 edge on the bottom of the triangle of weight 1 edges in our graph. Now, instead of choosing one of the weight 1 edges it touches, we might choose the weight 1 edge on the top of the graph. This edge still has minimum weight, and will be gray if we simply color its endpoints differently, so Algorithm 2 can choose it. We would then choose one of the remaining weight 1 edges. Note that neither causes us to form a cycle. Continuing the algorithm, we could end up with the same spanning tree in Figure 11.27, though this will depend on how the tie breaking rules used to choose among gray edges with the same minimum weight. For example, if the weight of every edge in G is one, then all spanning trees are MST's with weight $|V(G)|-1$, and both of these algorithms can arrive at each of these spannning trees by suitable tie-breaking.

The coloring that explains Algorithm 1 also justifies a more flexible algorithm which has Algorithm 1 a special case:

Algorithm 3. *Grow a forest one edge at a time by picking any component and adding a minimum-weight edge among the edges leaving that component.*

This algorithm allows components that are not too close to grow in parallel and independently, which is great for "distributed" computation where separate processors share the work with limited communication between processors.

These are examples of greedy approaches to optimization. Sometimes it works and sometimes it doesn't. The good news is that it works to find the MST. So we can be sure that the MST for our example graph has weight 17 since it was produced by Algorithm 2. And we have a fast algorithm for finding a minimum-weight spanning tree for any graph.

Ok, to wrap up this story, all that's left is the proof that extending edges are the same as minimal gray edges. This might sound like a chore, but it just uses the same reasoning we use to be sure there will be a gray edge when you need it.

Proof. (of Lemma 11.11.11)

Let F be a pre-MST that is a subgraph some some MST M of G, and suppose e is a minimum weight gray edge under some solid coloring of F. We want to show that $F+e$ is also a pre-MST.

If e happens to be an edge of M, then $F+e$ remains a subgraph of M, and so is a pre-MST.

The other case is when e is not an edge of M. Then since M is a spannning tree, $M+e$ is a spanning subgraph. Also M has a path $\mathbf{p}$ between the different colored endpoints of e, so $M+e$ has a cycle consisting of e together with $\mathbf{p}$. Now $\mathbf{p}$ has both a black endpoint and a white one, so it must contain some gray edge $g \neq e$.

The trick is to remove g from $M + e$ to obtain a subgraph $M + e - g$. We claim that $M + e - g$ is an MST that contains $F + e$, which shows that e extends F.

We begin proving the claim with the observation that $M + e - g$ contains $F + e$, which follows because gray edges like g are by definition not edges of F. Also, since the weight $w(e)$ is minimum among gray edges, $w(M + e - g)$ is at most $w(M)$, the minimum possible weight of a spanning tree. So to confirm that $M + e - g$ is an MST containing $F + e$, we just have to show that $M + e - g$ is a spanning tree,

But $M + e$ is a spanning subgraph, and g is on a cycle of $M + e$, so by Lemma 11.9.6, removing g won't disconnect anything, which means that $M + e - g$ is still a spanning subgraph. Moreover, $M + e - g$ has the same number of edges as M, so Lemma 11.11.5 confirms that it must be a tree, as claimed.

■

Another interesting fact falls out of the proof of Lemma 11.11.11:

Corollary 11.11.12. *If all edges in a weighted graph have distinct weights, then the graph has a* unique *MST.*

The proof of Corollary 11.11.12 is left to Problem 11.34.

Problems for Section 11.4

Class Problems

Problem 11.1. **(a)** Prove that in every graph, there are an even number of vertices of odd degree.

Hint: The Handshaking Lemma 11.2.1.

(b) Conclude that at a party where some people shake hands, the number of people who shake hands an odd number of times is an even number.

(c) Call a sequence of two or more different people at the party a *handshake sequence* if, except for the last person, each person in the sequence has shaken hands with the next person in the sequence.

Suppose George was at the party and has shaken hands with an odd number of people. Explain why, starting with George, there must be a handshake sequence ending with a different person who has shaken an odd number of hands.

Hint: Just look at the people at the ends of handshake sequences that start with George.

Problem 11.2.
For each of the following pairs of graphs, either define an isomorphism between them, or prove that there is none. (We write ab as shorthand for a—b.)

(a)

$$G_1 \text{ with } V_1 = \{1,2,3,4,5,6\},\ E_1 = \{12,23,34,14,15,35,45\}$$
$$G_2 \text{ with } V_2 = \{1,2,3,4,5,6\},\ E_2 = \{12,23,34,45,51,24,25\}$$

(b)

$$G_3 \text{ with } V_3 = \{1,2,3,4,5,6\},\ E_3 = \{12,23,34,14,45,56,26\}$$
$$G_4 \text{ with } V_4 = \{a,b,c,d,e,f\},\ E_4 = \{ab,bc,cd,de,ae,ef,cf\}$$

(c)

$$G_5 \text{ with } V_5 = \{a,b,c,d,e,f,g,h\},\ E_5 = \{ab,bc,cd,ad,ef,fg,gh,he,dh,bf\}$$
$$G_6 \text{ with } V_6 = \{s,t,u,v,w,x,y,z\},\ E_6 = \{st,tu,uv,sv,wx,xy,yz,wz,sw,vz\}$$

Homework Problems

Problem 11.3.
Determine which among the four graphs pictured in the Figures are isomorphic. If two of these graphs are isomorphic, describe an isomorphism between them. If they are not, give a property that is preserved under isomorphism such that one graph has the property, but the other does not. For at least one of the properties you choose, *prove* that it is indeed preserved under isomorphism (you only need prove one of them).

Problem 11.4. **(a)** For any vertex, v, in a graph, let $N(v)$ be the set of *neighbors* of v, namely, the vertices adjacent to v:

$$N(v) ::= \{u \mid u\text{—}v \text{ is an edge of the graph}\}.$$

Suppose f is an isomorphism from graph G to graph H. Prove that $f(N(v)) = N(f(v))$.

Your proof should follow by simple reasoning using the definitions of isomorphism and neighbors—no pictures or handwaving.

Hint: Prove by a chain of iff's that

$$h \in N(f(v)) \quad \text{iff} \quad h \in f(N(v))$$

for every $h \in V_H$. Use the fact that $h = f(u)$ for some $u \in V_G$.

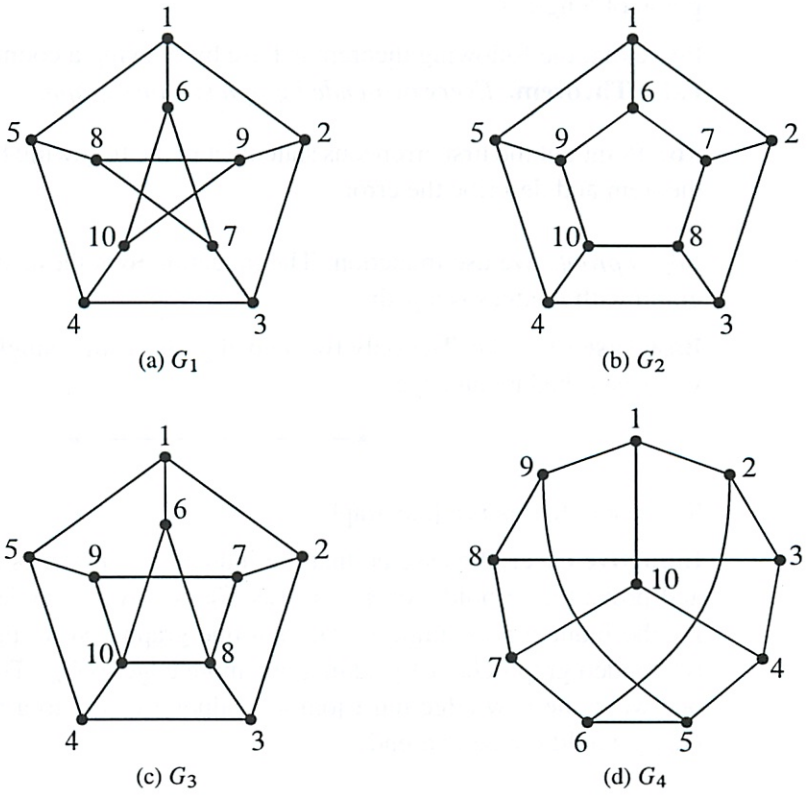


Figure 11.28 Which graphs are isomorphic?

(b) Conclude that if G and H are isomorphic graphs, then for each $k \in \mathbb{N}$, they have the same number of degree k vertices.

Problem 11.5.

Let's say that a graph has "two ends" if it has exactly two vertices of degree 1 and all its other vertices have degree 2. For example, here is one such graph:

(a) A *line graph* is a graph whose vertices can be listed in a sequence with edges between consecutive vertices only. So the two-ended graph above is also a line

graph of length 4.

Prove that the following theorem is false by drawing a counterexample.
False Theorem. *Every two-ended graph is a line graph.*

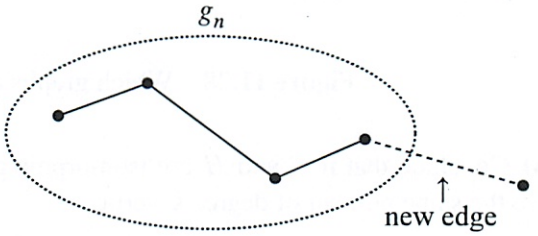
(b) Point out the first erroneous statement in the following bogus proof of the false theorem and describe the error.

Bogus proof. We use induction. The induction hypothesis is that every two-ended graph with n edges is a path.

Base case ($n = 1$): The only two-ended graph with a single edge consists of two vertices joined by an edge:

Sure enough, this is a line graph.

Inductive case: We assume that the induction hypothesis holds for some $n \geq 1$ and prove that it holds for $n + 1$. Let G_n be any two-ended graph with n edges. By the induction assumption, G_n is a line graph. Now suppose that we create a two-ended graph G_{n+1} by adding one more edge to G_n. This can be done in only one way: the new edge must join an endpoint of G_n to a new vertex; otherwise, G_{n+1} would not be two-ended.

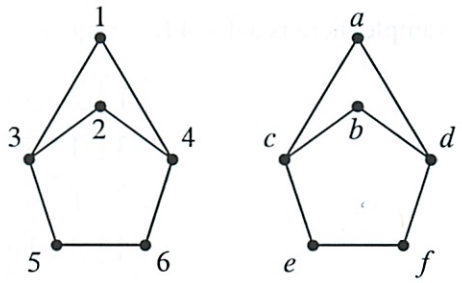


Clearly, G_{n+1} is also a line graph. Therefore, the induction hypothesis holds for all graphs with $n + 1$ edges, which completes the proof by induction.

■

Exam Problems

Problem 11.6.
There are four isomorphisms between these two graphs. List them.



Problems for Section 11.5

Class Problems

Problem 11.7.
A certain Institute of Technology has a lot of student clubs; these are loosely overseen by the Student Association. Each eligible club would like to delegate one of its members to appeal to the Dean for funding, but the Dean will not allow a student to be the delegate of more than one club. Fortunately, the Association VP took Math for Computer Science and recognizes a matching problem when she sees one.

(a) Explain how to model the delegate selection problem as a bipartite matching problem.

(b) The VP's records show that no student is a member of more than 9 clubs. The VP also knows that to be eligible for support from the Dean's office, a club must have at least 13 members. That's enough for her to guarantee there is a proper delegate selection. Explain. (If only the VP had taken an *Algorithms*, she could even have found a delegate selection without much effort.)

Problem 11.8.
A *Latin square* is $n \times n$ array whose entries are the number $1, \ldots, n$. These entries satisfy two constraints: every row contains all n integers in some order, and also every column contains all n integers in some order. Latin squares come up frequently in the design of scientific experiments for reasons illustrated by a little story in a footnote[11]

[11] At Guinness brewery in the eary 1900's, W. S. Gosset (a chemist) and E. S. Beavan (a "maltster") were trying to improve the barley used to make the brew. The brewery used different varieties of barley according to price and availability, and their agricultural consultants suggested a different fertilizer mix and best planting month for each variety.

Somewhat sceptical about paying high prices for customized fertilizer, Gosset and Beavan planned a season long test of the influence of fertilizer and planting month on barley yields. For as many

For example, here is a 4×4 Latin square:

1	2	3	4
3	4	2	1
2	1	4	3
4	3	1	2

(a) Here are three rows of what could be part of a 5×5 Latin square:

2	4	5	3	1
4	1	3	2	5
3	2	1	5	4

Fill in the last two rows to extend this "Latin rectangle" to a complete Latin square.

(b) Show that filling in the next row of an $n \times n$ Latin rectangle is equivalent to finding a matching in some $2n$-vertex bipartite graph.

(c) Prove that a matching must exist in this bipartite graph and, consequently, a Latin rectangle can always be extended to a Latin square.

Exam Problems

Problem 11.9.

Overworked and over-caffeinated, the Teaching Assistant's (TA's) decide to oust

---

months as there were varieties of barley, they would plant one sample of each variety using a different one of the fertilizers. So every month, they would have all the barley varieties planted and all the fertilizers used, which would give them a way to judge the overall quality of that planting month. But they also wanted to judge the fertilizers, so they wanted each fertilizer to be used on each variety during the course of the season. Now they had a little mathematical problem, which we can abstract as follows.

Suppose there are n barley varieties and an equal number of recommended fertilizers. Form an $n \times n$ array with a column for each fertilizer and a row for each planting month. We want to fill in the entries of this array with the integers $1,\ldots,n$ numbering the barley varieties, so that every row contains all n integers in some order (so every month each variety is planted and each fertilizer is used), and also every column contains all n integers (so each fertilizer is used on all the varieties over the course of the growing season).

the lecturer and teach their own recitations. They will run a recitation session at 4 different times in the same room. There are exactly 20 chairs to which a student can be assigned in each recitation. Each student has provided the TA's with a list of the recitation sessions her schedule allows and no student's schedule conflicts with all 4 sessions. The TA's must assign each student to a chair during recitation at a time she can attend, if such an assignment is possible.

(a) Describe how to model this situation as a matching problem. Be sure to specify what the vertices/edges should be and briefly describe how a matching would determine seat assignments for each student in a recitation that does not conflict with his schedule. (This is a *modeling problem*; we aren't looking for a description of an algorithm to solve the problem.)

(b) Suppose there are 65 students. Given the information provided above, is a matching guaranteed? Briefly explain.

Homework Problems

Problem 11.10.

Take a regular deck of 52 cards. Each card has a suit and a value. The suit is one of four possibilities: heart, diamond, club, spade. The value is one of 13 possibilities, $A, 2, 3, \ldots, 10, J, Q, K$. There is exactly one card for each of the 4×13 possible combinations of suit and value.

Ask your friend to lay the cards out into a grid with 4 rows and 13 columns. They can fill the cards in any way they'd like. In this problem you will show that you can always pick out 13 cards, one from each column of the grid, so that you wind up with cards of all 13 possible values.

(a) Explain how to model this trick as a bipartite matching problem between the 13 column vertices and the 13 value vertices. Is the graph necessarily degree constrained?

(b) Show that any n columns must contain at least n different values and prove that a matching must exist.

Problem 11.11.

Scholars through the ages have identified *twenty* fundamental human virtues: honesty, generosity, loyalty, prudence, completing the weekly course reading-response, etc. At the beginning of the term, every student in Math for Computer Science possessed exactly *eight* of these virtues. Furthermore, every student was unique; that is, no two students possessed exactly the same set of virtues. The Math for Com-

puter Science course staff must select *one* additional virtue to impart to each student by the end of the term. Prove that there is a way to select an additional virtue for each student so that every student is unique at the end of the term as well.

Suggestion: Use Hall's theorem. Try various interpretations for the vertices on the left and right sides of your bipartite graph.

Problems for Section 11.6

Practice Problems

Problem 11.12.

Four Students want separate assignments to four VI-A Companies. Here are their preference rankings:

Student	Companies
Albert:	HP, Bellcore, AT&T, Draper
Rich:	AT&T, Bellcore, Draper, HP
Megumi:	HP, Draper, AT&T, Bellcore
Justin:	Draper, AT&T, Bellcore, HP

Company	Students
AT&T:	Justin, Albert, Megumi, Rich
Bellcore:	Megumi, Rich, Albert, Justin
HP:	Justin, Megumi, Albert, Rich
Draper:	Rich, Justin, Megumi, Albert

(a) Use the Mating Ritual to find *two* stable assignments of Students to Companies.

(b) Describe a simple procedure to determine whether any given stable marriage problem has a unique solution, that is, only one possible stable matching.

Problem 11.13.

Suppose that Harry is one of the boys and Alice is one of the girls in the *Mating Ritual*. Which of the properties below are preserved invariants? Why?

a. Alice is the only girl on Harry's list.

b. There is a girl who does not have any boys serenading her.

c. If Alice is not on Harry's list, then Alice has a suitor that she prefers to Harry.

d. Alice is crossed off Harry's list and Harry prefers Alice to anyone he is serenading.

e. If Alice is on Harry's list, then she prefers to Harry to any suitor she has.

Class Problems

Problem 11.14.
Consider a stable marriage problem with 4 boys and 4 girls and the following partial information about their preferences:

B1:	G1	G2	–	–
B2:	G2	G1	–	–
B3:	–	–	G4	G3
B4:	–	–	G3	G4
G1:	B2	B1	–	–
G2:	B1	B2	–	–
G3:	–	–	B3	B4
G4:	–	–	B4	B3

(a) Verify that

$$(B1, G1), (B2, G2), (B3, G3), (B4, G4)$$

will be a stable matching whatever the unspecified preferences may be.

(b) Explain why the stable matching above is neither boy-optimal nor boy-pessimal and so will not be an outcome of the Mating Ritual.

(c) Describe how to define a set of marriage preferences among n boys and n girls which have at least $2^{n/2}$ stable assignments.

Hint: Arrange the boys into a list of $n/2$ pairs, and likewise arrange the girls into a list of $n/2$ pairs of girls. Choose preferences so that the kth pair of boys ranks the kth pair of girls just below the previous pairs of girls, and likewise for the kth pair of girls. Within the kth pairs, make sure each boy's first choice girl in the pair prefers the other boy in the pair.

Homework Problems

Problem 11.15.
The most famous application of stable matching was in assigning graduating medical students to hospital residencies. Each hospital has a preference ranking of students and each student has a preference order of hospitals, but unlike the setup

in the notes where there are an equal number of boys and girls and monogamous marriages, hospitals generally have differing numbers of available residencies, and the total number of residencies may not equal the number of graduating students. Modify the definition of stable matching so it applies in this situation, and explain how to modify the Mating Ritual so it yields stable assignments of students to residencies. No proof is required.

Problem 11.16.
Give an example of a stable matching between 3 boys and 3 girls where no person gets their first choice. Briefly explain why your matching is stable.

Problem 11.17.
In a stable matching between n boys and girls produced by the Mating Ritual, call a person *lucky* if they are matched up with one of their $\lceil n/2 \rceil$ top choices. We will prove:

Theorem. *There must be at least one lucky person.*

To prove this, define the following derived variables for the Mating Ritual:

$q(B) = j$, where j is the rank of the girl that boy B is courting. That is to say, boy B is always courting the jth girl on his list.

$r(G)$ is the number of boys that girl G has rejected.

(a) Let

$$S ::= \sum_{B \in \text{Boys}} q(B) - \sum_{G \in \text{Girls}} r(G). \tag{11.2}$$

Show that S remains the same from one day to the next in the Mating Ritual.

(b) Prove the Theorem above. (You may assume for simplicity that n is even.)

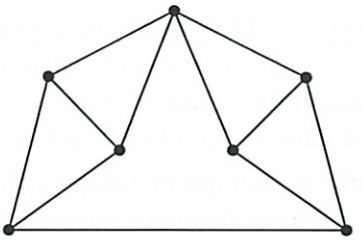
Hint: A girl is sure to be lucky if she has rejected half the boys.

Problems for Section 11.7

Class Problems

Problem 11.18.
Let G be the graph below[12]. Carefully explain why $\chi(G) = 4$.

[12]From *Discrete Mathematics*, Lovász, Pelikan, and Vesztergombi. Springer, 2003. Exercise 13.3.1



Homework Problems

Problem 11.19.
Math for Computer Science is often taught using recitations. Suppose it happened that 8 recitations were needed, with two or three staff members running each recitation. The assignment of staff to recitation sections is as follows:

- R1: Ali, Oshani, Henrique
- R2: Ali, Nick, Kyle
- R3: Oshani, Radhika
- R4: Becky, Nick, Oscar
- R5: Becky, Subha, Kyle
- R6: Subha, Radhika
- R7: Subha, Nick
- R8: Oshani, Radhika, Kyle

Two recitations can not be held in the same 90-minute time slot if some staff member is assigned to both recitations. The problem is to determine the minimum number of time slots required to complete all the recitations.

(a) Recast this problem as a question about coloring the vertices of a particular graph. Draw the graph and explain what the vertices, edges, and colors represent.

(b) Show a coloring of this graph using the fewest possible colors. What schedule of recitations does this imply?

Problem 11.20.
This problem generalizes the result proved Theorem 11.7.3 that any graph with maximum degree at most w is $(w + 1)$-colorable.

A simple graph, G, is said to have *width*, w, iff its vertices can be arranged in a sequence such that each vertex is adjacent to at most w vertices that precede it in the sequence. If the degree of every vertex is at most w, then the graph obviously has width at most w—just list the vertices in any order.

(a) Describe an example of a graph with 100 vertices, width 3, but *average* degree more than 5. *Hint:* Don't get stuck on this; if you don't see it after five minutes, ask for a hint.

(b) Prove that every graph with width at most w is $(w+1)$-colorable.

(c) Prove that the average degree of a graph of width w is at most $2w$.

Exam Problems

Problem 11.21.

False Claim. *Let G be a graph whose vertex degrees are all $\leq k$. If G has a vertex of degree strictly less than k, then G is k-colorable.*

(a) Give a counterexample to the False Claim when $k = 2$.

(b) Underline the exact sentence or part of a sentence that is the first unjustified step in the following bogus proof of the False Claim.

Bogus proof. Proof by induction on the number n of vertices:

Induction hypothesis:

$P(n)$::= "Let G be an n-vertex graph whose vertex degrees are all $\leq k$. If G also has a vertex of degree strictly less than k, then G is k-colorable."

Base case: ($n = 1$) G has one vertex, the degree of which is 0. Since G is 1-colorable, $P(1)$ holds.

Inductive step:

We may assume $P(n)$. To prove $P(n+1)$, let G_{n+1} be a graph with $n+1$ vertices whose vertex degrees are all k or less. Also, suppose G_{n+1} has a vertex, v, of degree strictly less than k. Now we only need to prove that G_{n+1} is k-colorable.

To do this, first remove the vertex v to produce a graph, G_n, with n vertices. Let u be a vertex that is adjacent to v in G_{n+1}. Removing v reduces the degree of u by 1. So in G_n, vertex u has degree strictly less than k. Since no edges were added, the vertex degrees of G_n remain $\leq k$. So G_n satisfies the conditions of the induction hypothesis, $P(n)$, and so we conclude that G_n is k-colorable.

Now a k-coloring of G_n gives a coloring of all the vertices of G_{n+1}, except for v. Since v has degree less than k, there will be fewer than k colors assigned to the nodes adjacent to v. So among the k possible colors, there will be a color not used to color these adjacent nodes, and this color can be assigned to v to form a k-coloring of G_{n+1}. ■

(c) With a slightly strengthened condition, the preceding proof of the False Claim could be revised into a sound proof of the following Claim:

Claim. *Let G be a graph whose vertex degrees are all $\leq k$. If* ⟨statement inserted from below⟩ *has a vertex of degree strictly less than k, then G is k-colorable.*

Circle each of the statements below that could be inserted to make the Claim true.

- G is connected and
- G has no vertex of degree zero and
- G does not contain a complete graph on k vertices and
- every connected component of G
- some connected component of G

Problems for Section 11.9

Class Problems

Problem 11.22.

The n-dimensional hypercube, H_n, is a graph whose vertices are the binary strings of length n. Two vertices are adjacent if and only if they differ in exactly 1 bit. For example, in H_3, vertices `111` and `011` are adjacent because they differ only in the first bit, while vertices `101` and `011` are not adjacent because they differ at both the first and second bits.

(a) Prove that it is impossible to find two spanning trees of H_3 that do not share some edge.

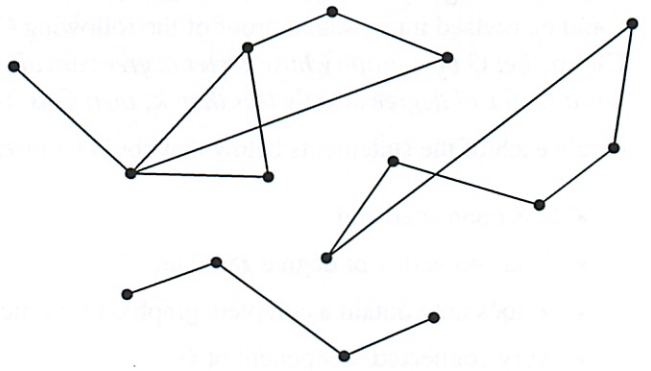
(b) Verify that for any two vertices $x \neq y$ of H_3, there are 3 paths from x to y in H_3, such that, besides x and y, no two of those paths have a vertex in common.

(c) Conclude that the connectivity of H_3 is 3.

(d) Try extending your reasoning to H_4. (In fact, the connectivity of H_n is n for all $n \geq 1$. A proof appears in the problem solution.)

Problem 11.23.
A set, M, of vertices of a graph is a *maximal connected* set if every pair of vertices in the set are connected, and any set of vertices properly containing M will contain two vertices that are not connected.

(a) What are the maximal connected subsets of the following (unconnected) graph?



(b) Explain the connection between maximal connected sets and connected components. Prove it.

Problem 11.24. **(a)** Prove that K_n is $(n-1)$-edge connected for $n > 1$.

Let M_n be a graph defined as follows: begin by taking n graphs with non-overlapping sets of vertices, where each of the n graphs is $(n-1)$-edge connected (they could be disjoint copies of K_n, for example). These will be subgraphs of M_n. Then pick n vertices, one from each subgraph, and add enough edges between pairs of picked vertices that the subgraph of the n picked vertices is also $(n-1)$-edge connected.

(b) Draw a picture of M_4.

(c) Explain why M_n is $(n-1)$-edge connected.

Problem 11.25.

False Claim. *If every vertex in a graph has positive degree, then the graph is connected.*

(a) Prove that this Claim is indeed false by providing a counterexample.

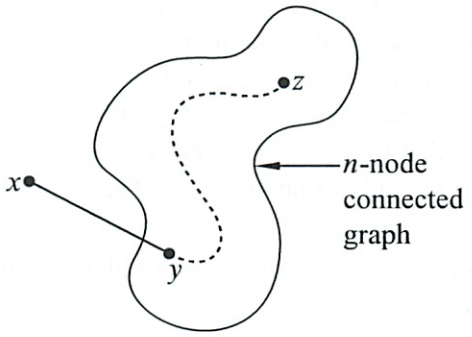
(b) Since the Claim is false, there must be an logical mistake in the following bogus proof. Pinpoint the *first* logical mistake (unjustified step) in the proof.

Bogus proof. We prove the Claim above by induction. Let $P(n)$ be the proposition that if every vertex in an n-vertex graph has positive degree, then the graph is connected.

Base cases: ($n \leq 2$). In a graph with 1 vertex, that vertex cannot have positive degree, so $P(1)$ holds vacuously.

$P(2)$ holds because there is only one graph with two vertices of positive degree, namely, the graph with an edge between the vertices, and this graph is connected.

Inductive step: We must show that $P(n)$ implies $P(n+1)$ for all $n \geq 2$. Consider an n-vertex graph in which every vertex has positive degree. By the assumption $P(n)$, this graph is connected; that is, there is a path between every pair of vertices. Now we add one more vertex x to obtain an $(n+1)$-vertex graph:



All that remains is to check that there is a path from x to every other vertex z. Since x has positive degree, there is an edge from x to some other vertex, y. Thus, we can obtain a path from x to z by going from x to y and then following the path from y to z. This proves $P(n+1)$.

By the principle of induction, $P(n)$ is true for all $n \geq 0$, which proves the Claim.

■

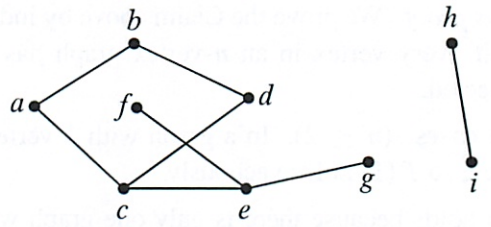
Homework Problems

Problem 11.26.

In this problem we'll consider some special closed walks called *Euler tours*, named after the famous mathematician Leonhard Euler. (Same Euler as for the constant $e \approx 2.718$—he did a lot of stuff.)

Definition 11.11.13. An Euler tour of a graph is a walk that traverses every edge exactly once.

Does the graph in the following figure contain an Euler tour?



Well, if it did, the edge (E, F) would need to be included. If the walk does not start at F then at some point it traverses edge (E, F), and now it is stuck at F since F has no other edges incident to it and an Euler tour can't traverse (E, F) twice. But then the walk could not be a tour. On the other hand, if the walk starts at F, it must then go to E along (E, F), but now it cannot return to F. It again cannot be a tour. This argument generalizes to show that if a graph has a vertex of degree 1, it cannot contain an Euler tour.

So how do you tell in general whether a graph has an Euler tour? At first glance this may seem like a daunting problem (the similar sounding problem of finding a cycle that touches every vertex exactly once is one of those million dollar NP-complete problems known as the *Traveling Salesman Problem*)—but it turns out to be easy.

(a) Show that if a graph has an Euler tour, then the degree of each of its vertices is even.

In the remaining parts, we'll work out the converse: if the degree of every vertex of a connected finite graph is even, then it has an Euler tour. To do this, let's define an Euler *walk* to be a walk that traverses each edge *at most* once.

(b) Suppose that an Euler walk in a connected graph does not include every edge. Explain why there must be an unincluded edge that is incident to a vertex on the walk.

In the remaining parts, let **w** be the *longest* Euler walk in some finite, connected graph.

(c) Show that if **w** is a closed walk, then it must be an Euler tour.

Hint: part (b)

(d) Explain why all the edges incident to the end of **w** must already have been traversed by **w**.

(e) Show that if the end of **w** was not equal to the start of **w**, then the degree of the end would be odd.

Hint: part (d)

(f) Conclude that if every vertex of a finite, connected graph has even degree, then it has an Euler tour.

Homework Problems

Problem 11.27.
An edge is said to *leave* a set of vertices if one end of the edge is in the set and the other end is not.

(a) An n-node graph is said to be *mangled* if there is an edge leaving every set of $\lfloor n/2 \rfloor$ or fewer vertices. Prove the following:
Claim. *Every mangled graph is connected.*

An n-node graph is said to be *tangled* if there is an edge leaving every set of $\lceil n/3 \rceil$ or fewer vertices.

(b) Draw a tangled graph that is not connected.

(c) Find the error in the bogus proof of the following
False Claim. *Every tangled graph is connected.*

Bogus proof. The proof is by strong induction on the number of vertices in the graph. Let $P(n)$ be the proposition that if an n-node graph is tangled, then it is connected. In the base case, $P(1)$ is true because the graph consisting of a single node is trivially connected.

For the inductive case, assume $n \geq 1$ and $P(1), \ldots, P(n)$ hold. We must prove $P(n+1)$, namely, that if an $(n+1)$-node graph is tangled, then it is connected.

So let G be a tangled, $(n+1)$-node graph. Choose $\lceil n/3 \rceil$ of the vertices and let G_1 be the tangled subgraph of G with these vertices and G_2 be the tangled subgraph with the rest of the vertices. Note that since $n \geq 1$, the graph G has a least two vertices, and so both G_1 and G_2 contain at least one vertex. Since G_1 and G_2 are tangled, we may assume by strong induction that both are connected. Also, since G is tangled, there is an edge leaving the vertices of G_1 which necessarily connects to a vertex of G_2. This means there is a path between any two vertices of G: a path within one subgraph if both vertices are in the same subgraph, and a path traversing the connecting edge if the vertices are in separate subgraphs. Therefore, the entire graph, G, is connected. This completes the proof of the inductive case, and the Claim follows by strong induction.

Problem 11.28.
Let G be the graph formed from C_{2n}, the cycle of length $2n$, by connecting every pair of vertices at maximum distance from each other in C_{2n} by an edge in G.

(a) Given two vertices of G find their distance in G.

(b) What is the *diameter* of G, that is, the largest distance between two vertices?

(c) Prove that the graph is not 4-connected.

(d) Prove that the graph is 3-connected.

Problems for Section 11.10

Exam Problems

Problem 11.29.
A researcher analyzing data on heterosexual sexual behavior in a group of m males and f females found that within the group, the male average number of female partners was 10% larger that the female average number of male partners.

(a) Circle all of the assertions below that are implied by the above information on average numbers of partners:

(i) males exaggerate their number of female partners

(ii) $m = (9/10)f$

(iii) $m = (10/11)f$

(iv) $m = (11/10)f$

(v) there cannot be a perfect matching with each male matched to one of his female partners

(vi) there cannot be a perfect matching with each female matched to one of her male partners

(b) The data shows that approximately 20% of the females were virgins, while only 5% of the males were. The researcher wonders how excluding virgins from the population would change the averages. If he knew graph theory, the researcher would realize that the nonvirgin male average number of partners will be $x(f/m)$ times the nonvirgin female average number of partners. What is x?

Problems for Section 11.11

Class Problems

Problem 11.30.

Procedure *Mark* starts with a connected, simple graph with all edges unmarked and then marks some edges. At any point in the procedure a path that traverses only marked edges is called a *fully marked* path, and an edge that has no fully marked path between its endpoints is called *eligible*.

Procedure *Mark* simply keeps marking eligible edges, and terminates when there are none.

Prove that *Mark* terminates, and that when it does, the set of marked edges forms a spanning tree of the original graph.

Problem 11.31.

Procedure **create-spanning-tree**

Given a simple graph G, keep applying the following operations to the graph until no operation applies:

1. If an edge u—v of G is on a cycle, then delete u—v.
2. If vertices u and v of G are not connected, then add the edge u—v.

Assume the vertices of G are the integers $1, 2, \ldots, n$ for some $n \geq 2$. Procedure **create-spanning-tree** can be modeled as a state machine whose states are all possible simple graphs with vertices $1, 2, \ldots, n$. The start state is G, and the final states are the graphs on which no operation is possible.

(a) Let G be the graph with vertices $\{1, 2, 3, 4\}$ and edges

$$\{1\text{—}2, 3\text{—}4\}$$

What are the possible final states reachable from start state G? Draw them.

(b) Prove that any final state of must be a tree on the vertices.

(c) For any state, G', let e be the number of edges in G', c be the number of connected components it has, and s be the number of cycles. For each of the derived

variables below, indicate the *strongest* of the properties that it is guaranteed to satisfy, no matter what the starting graph G is and be prepared to briefly explain your answer.

The choices for properties are: *constant*, *strictly increasing*, *strictly decreasing*, *weakly increasing*, *weakly decreasing*, *none of these*. The derived variables are

(i) e

(ii) c

(iii) s

(iv) $e - s$

(v) $c + e$

(vi) $3c + 2e$

(vii) $c + s$

(viii) (c, e), partially ordered coordinatewise (the *product* partial order 9.9.1).

(d) Prove that procedure **create-spanning-tree** terminates. (If your proof depends on one of the answers to part (c), you must prove that answer is correct.)

Problem 11.32.
Prove that a graph is a tree iff it has a unique path between any two vertices.

Homework Problems

Problem 11.33. (a) Prove that the average degree of a tree is less than 2.

(b) Suppose every vertex in a graph has degree at least k. Explain why the graph has a path of length k.

Hint: Consider a longest path.

Problem 11.34. (a) Show that every minimum spanning tree of a graph can be the result of Algorithm 1.

(b) Conclude Corollary 11.11.12 that if all edges in a weighted graph have distinct weights, then the graph has a *unique* MST.

3/31

12 Planar Graphs

12.1 Drawing Graphs in the Plane

Suppose there are three dog houses and three human houses, as shown in Figure 12.1. Can you find a route from each dog house to each human house such that no route crosses any other route?

For example, Figure 12.4 shows how to route the first two dogs to all threes houses and the third dog to two of the houses, all without any crossings. But in this figure there is no way left to route the third dog to the last house without crossing one of the other routes. Is there a better routing that does the job?

A similar question comes up about a little-known animal called a *quadrapus* that looks like an octopus with four stretchy arms instead of eight. If five quadrapi are resting on the sea floor, as shown in Figure 12.2, can each quadrapus simultaneously shake hands with every other in such a way that no arms cross?

PC circuit boards

Both these puzzles can be understood as asking about drawing graphs in the plane. Replacing dogs and houses by nodes, the dog house puzzle can be rephrased as asking whether there is a planar drawing of the graph with six nodes and edges between each of the first three nodes and each of the second three nodes. This graph is called the *complete bipartite graph* $K_{3,3}$ and is shown in Figure 12.3.(a). The quadrapi puzzle asks whether there is a planar drawing of the complete graph K_5 shown in Figure 12.3.(b).

In each case, the answer is, "No—but almost!" In fact, if you remove an edge from either of these graphs, then the resulting graph *can* be redrawn in the plane so that no edges cross. Figure 12.4 already showed how to do this.

Planar drawings have applications in circuit layout and are helpful in displaying graphical data such as program flow charts, organizational charts, and scheduling conflicts. For these applications, the goal is to draw the graph in the plane with as few edge crossings as possible. (See the box on the following page for one such example.)

12.2 Definitions of Planar Graphs

We took the idea of a planar drawing for granted in the previous section, but if we're going to *prove* things about planar graphs, we better have precise definitions.

nice house d'iagrams

Figure 12.1 Three dog houses and and three human houses. Is there a route from each dog house to each human house so that no pair of routes cross each other?

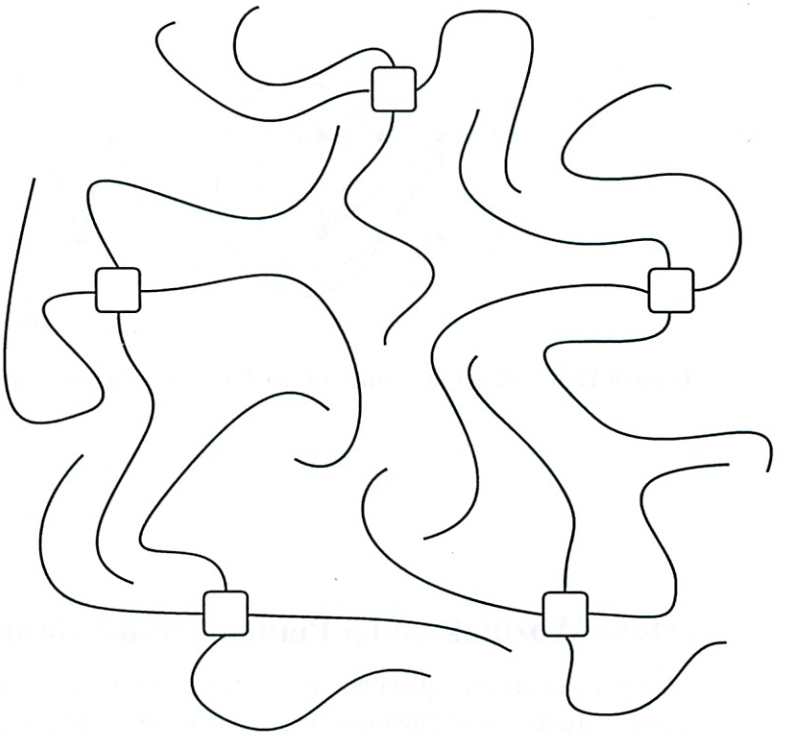


Figure 12.2 Five quadrapi (4-armed creatures).

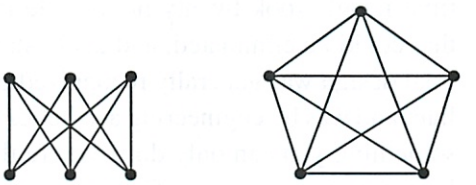


Figure 12.3 $K_{3,3}$ (a) and K_5 (b). Can you redraw these graphs so that no pairs of edges cross?

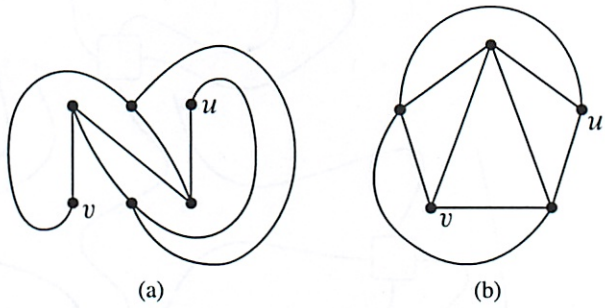


Figure 12.4 Planar drawings of (a) $K_{3,3}$ without u—v, and (b) K_5 without u—v.

Steve Wozniak and a Planar Circuit Design

When wires are arranged on a surface, like a circuit board or microchip, crossings require troublesome three-dimensional structures. When Steve Wozniak designed the disk drive for the early Apple II computer, he struggled mightily to achieve a nearly planar design:

> For two weeks, he worked late each night to make a satisfactory design. When he was finished, he found that if he moved a connector he could cut down on feedthroughs, making the board more reliable. To make that move, however, he had to start over in his design. This time it only took twenty hours. He then saw another feedthrough that could be eliminated, and again started over on his design. "The final design was generally recognized by computer engineers as brilliant and was by engineering aesthetics beautiful. Woz later said, 'It's something you can only do if you're the engineer and the PC board layout person yourself. That was an artistic layout. The board has virtually no feedthroughs.' "[1]

Definition 12.2.1. A *drawing* of a graph assigns to each node a distinct point in the plane and assigns to each edge a smooth curve in the plane whose endpoints correspond to the nodes incident to the edge. The drawing is *planar* if none of the curves cross themselves or other curves, namely, the only points that appear more than once on any of the curves are the node points. A graph is *planar* when it has a planar drawing.

Definition 12.2.1 is precise but depends on further concepts: "smooth planar curves" and "points appearing more than once" on them. We haven't defined these concepts—we just showed the simple picture in Figure 12.4 and hoped you would get the idea.

Pictures can be a great way to get a new idea across, but it is generally not a good idea to use a picture to replace precise mathematics. Relying solely on pictures can sometimes lead to disaster—or to bogus proofs, anyway. There is a long history of bogus proofs about planar graphs based on misleading pictures.[2]

The bad news is that to prove things about planar graphs using the planar drawings of Definition 12.2.1, we'd have to take a chapter-long excursion into continuous mathematics just to develop the needed concepts from plane geometry and topology. The good news is that there is another way to define planar graphs that uses only discrete mathematics. In particular, we can define planar graphs as a recursive data type. In order to understand how it works, we first need to understand the concept of a *face* in a planar drawing.

12.2.1 Faces

In a planar drawing of a graph. the curves corresponding to the edges divide up the plane into connected regions. We call these regions the *continuous faces*.[3] For example, the drawing in Figure 12.5 has four continuous faces. Face IV, which extends off to infinity in all directions, is called the *outside face*.

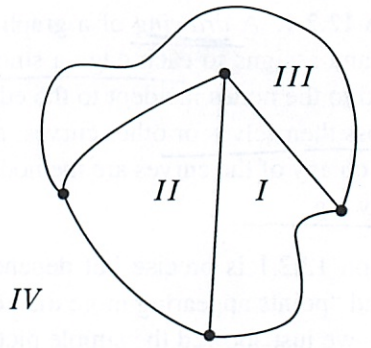
The vertices along the boundary of each continuous face in Figure 12.5 form a cycle. For example, labeling the vertices as in Figure 12.6, the cycles for the face boundaries are

$$abca \qquad abda \qquad bcdb \qquad acda. \tag{12.1}$$

[2]The bogus proof of the 4-Color Theorem for planar graphs is not the only example. Mistakes creep in with statements like,

> As you can see from Figure ABC, it must be that property XYZ holds for all planar graphs.

[3]Most texts drop the adjective *continuous* from the definition of a face as a connected region. We need the adjective to distinguish continuous faces from the *discrete* faces we're about to define.



oh areas inside

Figure 12.5 A planar drawing with four continuous faces.

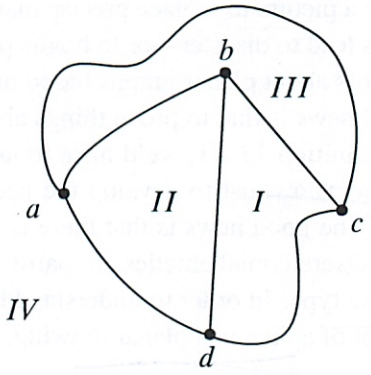


Figure 12.6 The drawing with labeled vertices.

These four cycles correspond nicely to the four continuous faces in Figure 12.6—so nicely, in fact, that we can identify each of the faces in Figure 12.6 by its cycle. For example, the cycle $abca$ identifies face III. The cycles in list 12.1 are called the *discrete faces* of the graph in Figure 12.6. We use the term "discrete" since cycles in a graph are a discrete data type—as opposed to a region in the plane, which is a continuous data type.

Unfortunately, continuous faces in planar drawings are not always bounded by cycles in the graph—things can get a little more complicated. For example, the planar drawing in Figure 12.7 has what we will call a *bridge*, namely, the edge c—e. The sequence of vertices along the boundary of the outer region of the drawing is

$$abcefgecda.$$

so is ∞ face on

This sequence defines a closed walk, but does not define a cycle since the walk traverses the bridge c—e twice.

both sides of c-e;

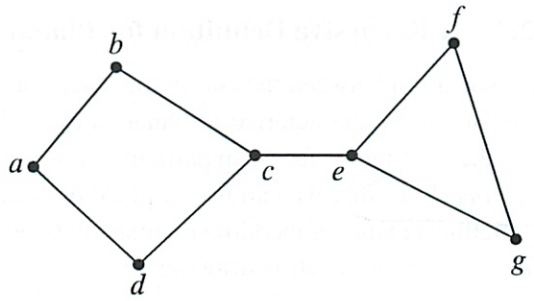


Figure 12.7 A planar drawing with a *bridge*.

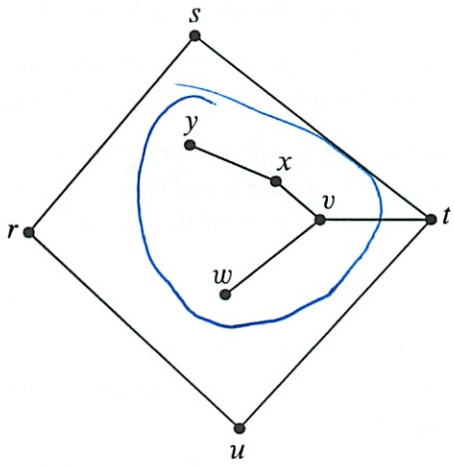


Figure 12.8 A planar drawing with a *dongle*.

The planar drawing in Figure 12.8 illustrates another complication. This drawing has what we will call a *dongle*, namely, the nodes v, x, y, and w, and the edges incident to them. The sequence of vertices along the boundary of the inner region is

$$rstvxyxvwvtur.$$

This sequence defines a closed walk, but once again does not define a cycle, because the walk traverses *every* edge of the dongle twice—once "coming" and once "going."

It turns out that bridges and dongles are the only complications, at least for connected graphs. In particular, every continuous face in a planar drawing corresponds to a closed walk in the graph. These closed walks will be called the *discrete faces* of the drawing, and we'll define them next.

12.2.2 A Recursive Definition for Planar Embeddings

The association between the continuous faces of a planar drawing and closed walks will allow us to characterize a planar drawing in terms of the closed walks that bound the continuous faces. In particular, it leads us to the discrete data type of *planar embeddings* that we can use in place of continuous planar drawings. Namely, we'll define a planar embedding recursively to be the set of boundary-tracing closed walks that we could get by drawing one edge after another.

Definition 12.2.2. A *planar embedding* of a *connected* graph consists of a nonempty set of closed walks of the graph called the *discrete faces* of the embedding. Planar embeddings are defined recursively as follows:

Base case: If G is a graph consisting of a single vertex, v, then a planar embedding of G has one discrete face, namely, the length zero closed walk, v.

Constructor case: (split a face): Suppose G is a connected graph with a planar embedding, and suppose a and b are distinct, nonadjacent vertices of G that appear on some discrete face, γ, of the planar embedding. That is, γ is a closed walk of the form

$$a \ldots b \cdots a.$$

Then the graph obtained by adding the edge a—b to the edges of G has a planar embedding with the same discrete faces as G, except that face γ is replaced by the two discrete faces[4]

$$a \ldots ba \quad \text{and} \quad ab \cdots a,$$

as illustrated in Figure 12.9.

Constructor case (add a bridge): Suppose G and H are connected graphs with planar embeddings and disjoint sets of vertices. Let a be a vertex on a discrete face, γ, in the embedding of G. That is, γ is of the form

$$a \ldots a.$$

Similarly, let b be a vertex on a discrete face, δ, in the embedding of H. So δ is of the form

$$b \cdots b.$$

[4] There is a minor exception to this definition of embedding in the special case when G is a line graph beginning with a and ending with b. In this case the cycles into which γ splits are actually the same. That's because adding edge a—b creates a cycle that divides the plane into "inner" and "outer" continuous faces that are both bordered by this cycle. In order to maintain the correspondence between continuous faces and discrete faces in this case, we define the two discrete faces of the embedding to be two "copies" of this same cycle.

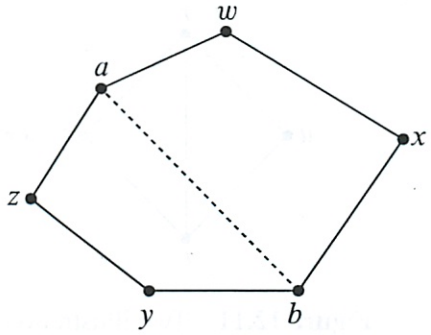


Figure 12.9 The "split a face" case.

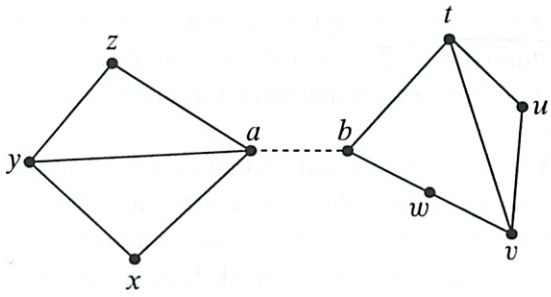


Figure 12.10 The "add a bridge" case.

Then the graph obtained by connecting G and H with a new edge, a—b, has a planar embedding whose discrete faces are the union of the discrete faces of G and H, except that faces γ and δ are replaced by one new face

$$a \ldots ab \cdots ba.$$

This is illustrated in Figure 12.10, where the faces of G and H are:

$$G : \{axyza,\ axya,\ ayza\} \qquad H : \{btuvwb,\ btvwb,\ tuvt\},$$

and after adding the bridge a—b, there is a single connected graph with faces

$$\{axyzabtuvwba,\ axya,\ ayza,\ btvwb,\ tuvt\}.$$

12.2.3 Does It Work?

Yes! In general, a graph is planar if and only if each of its connected components has a planar embedding as defined in Definition 12.2.2. Unfortunately, proving this

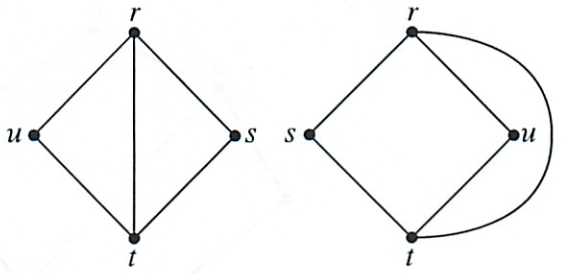


Figure 12.11 Two illustrations of the same embedding.

fact requires a bunch of mathematics that we don't cover in this text—stuff like geometry and topology. Of course, that is why we went to the trouble of including Definition 12.2.2—we don't want to deal with that stuff in this text and now that we have a recursive definition for planar graphs, we won't need to. That's the good news.

The bad news is that Definition 12.2.2 looks a lot more complicated than the intuitively simple notion of a drawing where edges don't cross. It seems like it would be easier to stick to the simple notion and give proofs using pictures. Perhaps so, but your proofs are more likely to be complete and correct if you work from the discrete Definition 12.2.2 instead of the continuous Definition 12.2.1.

12.2.4 Where Did the Outer Face Go?

Every planar drawing has an immediately-recognizable outer face—its the one that goes to infinity in all directions. But where is the outer face in a planar embedding?

There isn't one! That's because there really isn't any need to distinguish one. In fact, a planar embedding could be drawn with any given face on the outside. An intuitive explanation of this is to think of drawing the embedding on a *sphere* instead of the plane. Then any face can be made the outside face by "puncturing" that face of the sphere, stretching the puncture hole to a circle around the rest of the faces, and flattening the circular drawing onto the plane.

So pictures that show different "outside" boundaries may actually be illustrations of the same planar embedding. For example, the two embeddings shown in Figure 12.11 are really the same.

This is what justifies the "add bridge" case in Definition 12.2.2: whatever face is chosen in the embeddings of each of the disjoint planar graphs, we can draw a bridge between them without needing to cross any other edges in the drawing, because we can assume the bridge connects two "outer" faces.

12.3 Euler's Formula

The value of the recursive definition is that it provides a powerful technique for proving properties of planar graphs, namely, structural induction. For example, we will now use Definition 12.2.2 and structural induction to establish one of the most basic properties of a connected planar graph; namely, the number of vertices and edges completely determines the number of faces in every possible planar embedding of the graph.

Theorem 12.3.1 (Euler's Formula). *If a connected graph has a planar embedding, then*

$$v - e + f = 2$$

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

For example, in Figure 12.5, $v = 4$, $e = 6$, and $f = 4$. Sure enough, $4-6+4 = 2$, as Euler's Formula claims.

Proof. The proof is by structural induction on the definition of planar embeddings. Let $P(\mathcal{E})$ be the proposition that $v - e + f = 2$ for an embedding, $\mathcal{E}$.

Base case ($\mathcal{E}$ is the one-vertex planar embedding): By definition, $v = 1$, $e = 0$, and $f = 1$, so $P(\mathcal{E})$ indeed holds.

Constructor case (split a face): Suppose G is a connected graph with a planar embedding, and suppose a and b are distinct, nonadjacent vertices of G that appear on some discrete face, $\gamma = a \ldots b \cdots a$, of the planar embedding.

Then the graph obtained by adding the edge a—b to the edges of G has a planar embedding with one more face and one more edge than G. So the quantity $v-e+f$ will remain the same for both graphs, and since by structural induction this quantity is 2 for G's embedding, it's also 2 for the embedding of G with the added edge. So P holds for the constructed embedding.

Constructor case (add bridge): Suppose G and H are connected graphs with planar embeddings and disjoint sets of vertices. Then connecting these two graphs with a bridge merges the two bridged faces into a single face, and leaves all other faces unchanged. So the bridge operation yields a planar embedding of a connected

graph with $v_G + v_H$ vertices, $e_G + e_H + 1$ edges, and $f_G + f_H - 1$ faces. Since

$$\begin{aligned}(v_G + v_H) - (e_G + e_H + 1) + (f_G + f_H - 1) \\ &= (v_G - e_G + f_G) + (v_H - e_H + f_H) - 2 \\ &= (2) + (2) - 2 \quad \text{(by structural induction hypothesis)} \\ &= 2,\end{aligned}$$

$v - e + f$ remains equal to 2 for the constructed embedding. That is, $P(e)$ also holds in this case.

This completes the proof of the constructor cases, and the theorem follows by structural induction. ■

12.4 Bounding the Number of Edges in a Planar Graph

Like Euler's formula, the following lemmas follow by structural induction directly from Definition 12.2.2.

Lemma 12.4.1. *In a planar embedding of a connected graph, each edge is traversed once by each of two different faces, or is traversed exactly twice by one face.*

Lemma 12.4.2. *In a planar embedding of a connected graph with at least three vertices, each face is of length at least three.*

Combining Lemmas 12.4.1 and 12.4.2 with Euler's Formula, we can now prove that planar graphs have a limited number of edges:

Theorem 12.4.3. *Suppose a connected planar graph has $v \geq 3$ vertices and e edges. Then*

$$e \leq 3v - 6. \tag{12.2}$$

Proof. By definition, a connected graph is planar iff it has a planar embedding. So suppose a connected graph with v vertices and e edges has a planar embedding with f faces. By Lemma 12.4.1, every edge is traversed exactly twice by the face boundaries. So the sum of the lengths of the face boundaries is exactly $2e$. Also by Lemma 12.4.2, when $v \geq 3$, each face boundary is of length at least three, so this sum is at least $3f$. This implies that

$$3f \leq 2e. \tag{12.3}$$

But $f = e - v + 2$ by Euler's formula, and substituting into (12.3) gives

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

■

12.5 Returning to K_5 and $K_{3,3}$

Finally we have a simple way to answer the quadrapi question at the begiing of this chapter: the five quadrapi can't all shake hands without crossing. The reason is that we know the quadrupi question is the same as asking whether a complete graph K_5 is planar, and Theorem 12.4.3 has the immediate:

Corollary 12.5.1. *K_5 is not planar.*

Proof. K_5 is connected and has 5 vertices and 10 edges. But since $10 > 3 \cdot 5 - 6$, K_5 does not satisfy the inequality (12.2) that holds in all planar graphs. ■

Oh just saying inequality does not hold!

We can also use Euler's Formula to show that $K_{3,3}$ is not planar. The proof is similar to that of Theorem 12.2 except that we use the additional fact that $K_{3,3}$ is a bipartite graph.

Lemma 12.5.2. *In a planar embedding of a connected bipartite graph with at least 3 vertices, each face has length at least 4.*

Proof. By Lemma 12.4.2, every face of a planar embedding of the graph has length at least 3. But by Lemma 11.7.2 and Theorem 11.10.1.3, a bipartite graph can't have odd length closed walks. Since the faces of a planar embedding are closed walks, there can't be any faces of length 3 in a bipartite embedding. So every face must have length at least 4. ■

Theorem 12.5.3. *Suppose a connected bipartite graph with $v \geq 3$ vertices and e edges is planar. Then*

$$e \leq 2v - 4. \tag{12.4}$$

Proof. Lemma 12.5.2 implies that all the faces of an embedding of the graph have length at least 4. Now arguing as in the proof of Theorem 12.4.3, we find that the sum of the lengths of the face boundaries is exactly $2e$ and at least $4f$. Hence,

$$4f \leq 2e \tag{12.5}$$

for any embedding of a planar bipartite graph. By Euler's theorem, $f = 2 - v + e$. Substituting $2 - v + e$ for f in (12.5), we have

$$4(2 - v + e) \leq 2e,$$

which simplies to (12.4). ■

Corollary 12.5.4. *$K_{3,3}$ is not planar.*

Proof. $K_{3,3}$ is connected, bipartite and has 6 vertices and 9 edges. But since $9 > 2 \cdot 6 - 4$, $K_{3.3}$ does not satisfy the inequality (12.2) that holds in all bipartite planar graphs. ■

12.6 Another Characterization for Planar Graphs

We did not pick K_5 and $K_{3,3}$ as examples because of their application to dog houses or quadrapi shaking hands. We really picked them because they provide another, famous, discrete characterizarion of planar graphs:

Theorem 12.6.1 (Kuratowski). *A graph is not planar if and only if it contains K_5 or $K_{3,3}$ as a minor.*

Definition 12.6.2. A *minor* of a graph G is a graph that can be obtained by repeatedly[5] deleting vertices, deleting edges, and merging *adjacent* vertices of G. *Merging* two adjacent vertices, n_1 and n_2 of a graph means deleting the two vertices and then replacing them by a new "merged" vertex, m, adjacent to all the vertices that were adjacent to either of n_1 or n_2, as illustrated in Figure 12.12.

like editing

For example, Figure 12.13 illustrates why C_3 is a minor of the graph in Figure 12.13(a). In fact C_3 is a minor of a connected graph G if and only if G is not a tree.

The known proofs of Kuratowski's Theorem 12.6.1 are a little too long to include in an introductory text, so we won't prove it. There are two further facts about planarity that we will need in our proof planar graphs are 5-colorable.

Lemma 12.6.3. *Any subgraph of a planar graph is planar.*

Lemma 12.6.4. *Merging two adjacent vertices of a planar graph leaves another planar graph.*

[5]The three operations can each be performed any number of times in any order.

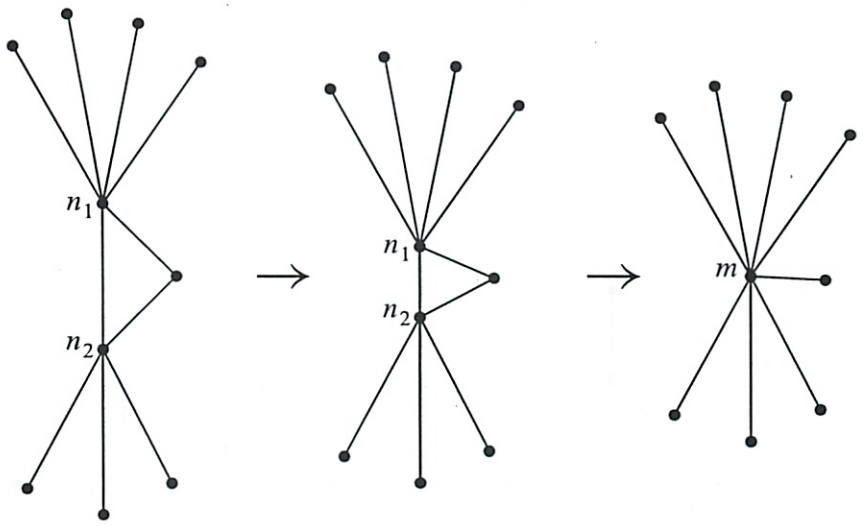


Figure 12.12 Merging adjacent vertices n_1 and n_2 into new vertex, m.

Many authors take Lemmas 12.6.3 and 12.6.4 for granted for continuous drawings of planar graphs described by Definition 12.2.1. With the recursive Definition 12.2.2 both Lemmas can actually by proved using structural induction, but those proofs are better left to some homework problems 12.6.

~~Sam~~ actually might be fun

12.7 Coloring Planar Graphs

We've covered a lot of ground with planar graphs, but not nearly enough to prove the famous 4-color theorem. But we can get awfully close. Indeed, we have done almost enough work to prove that every planar graph can be colored using only 5 colors. We need only one more lemma:

Lemma 12.7.1. *Every planar graph has a vertex of degree at most five.*

Proof. By contradiction. If every vertex had degree at least 6, then the sum of the vertex degrees is at least $6v$, but since the sum of the vertex degrees equals $2e$, by the Handshake Lemma 11.2.1, we have $e \geq 3v$ contradicting the fact that $e \leq 3v - 6 < 3v$ by Theorem 12.4.3. ■

Theorem 12.7.2. *Every planar graph is five-colorable.*

Proof. The proof will be by strong induction on the number, v, of vertices, with induction hypothesis:

(by # colors for vertex)

my attitude is changing

fun putting the pieces together

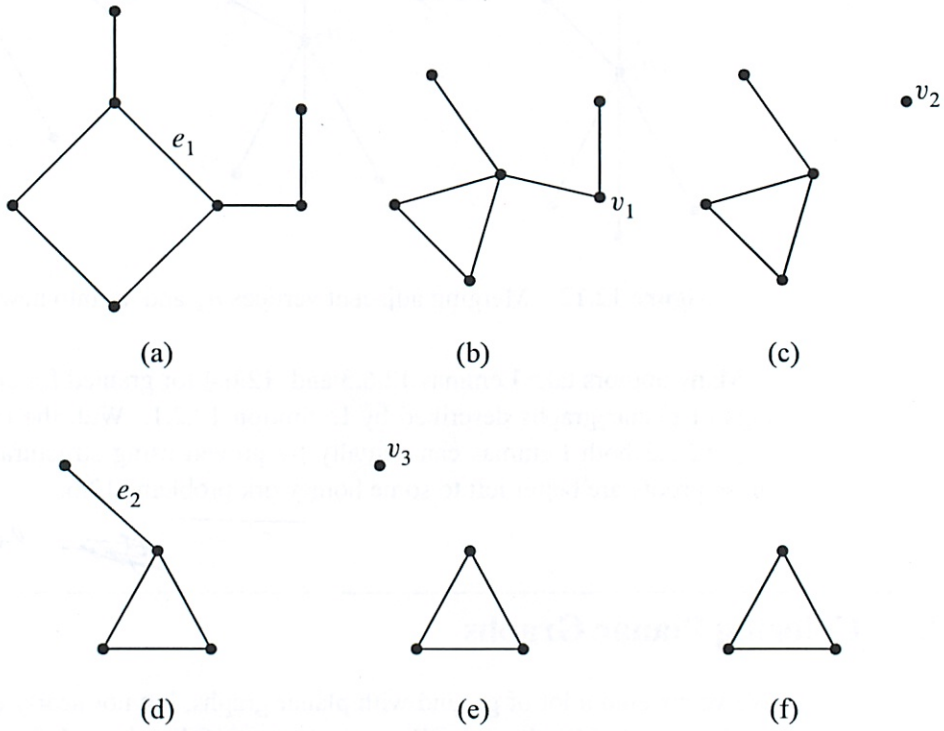


Figure 12.13 One method by which the graph in (a) can be reduced to C_3 (f), thereby showing that C_3 is a minor of the graph. The steps are: merging the nodes incident to e_1 (b), deleting v_1 and all edges incident to it (c), deleting v_2 (d), deleting e_2, and deleting v_3 (f).

Every planar graph with v vertices is five-colorable.

Base cases ($v \leq 5$): immediate.

Inductive case: Suppose G is a planar graph with $v + 1$ vertices. We will describe a five-coloring of G.

First, choose a vertex, g, of G with degree at most 5; Lemma 12.7.1 guarantees there will be such a vertex.

Case 1: ($\deg(g) < 5$): Deleting g from G leaves a graph, H, that is planar by Lemma 12.6.3, and, since H has v vertices, it is five-colorable by induction hypothesis. Now define a five coloring of G as follows: use the five-coloring of H for all the vertices besides g, and assign one of the five colors to g that is not the same as the color assigned to any of its neighbors. Since there are fewer than 5 neighbors, there will always be such a color available for g.

Case 2: ($\deg(g) = 5$): If the five neighbors of g in G were all adjacent to each other, then these five vertices would form a nonplanar subgraph isomorphic to K_5, contradicting Lemma 12.6.3 (since K_5 is not planar). So there must be two neighbors, n_1 and n_2, of g that are not adjacent. Now merge n_1 and g into a new vertex, m. In this new graph, n_2 is adjacent to m, and the graph is planar by Lemma 12.6.4. So we can then merge m and n_2 into a another new vertex, m', resulting in a new graph, G', which by Lemma 12.6.4 is also planar. Since G' has $v - 1$ vertices, it is five-colorable by the induction hypothesis.

Now define a five coloring of G as follows: use the five-coloring of G' for all the vertices besides g, n_1 and n_2. Next assign the color of m' in G' to be the color of the neighbors n_1 and n_2. Since n_1 and n_2 are not adjacent in G, this defines a proper five-coloring of G except for vertex g. But since these two neighbors of g have the same color, the neighbors of g have been colored using fewer than five colors altogether. So complete the five-coloring of G by assigning one of the five colors to g that is not the same as any of the colors assigned to its neighbors.

■

12.8 Classifying Polyhedra

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

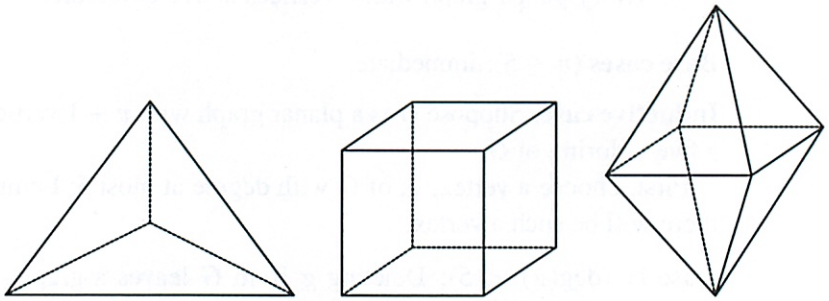


Figure 12.14 The tetrahedron (a), cube (b), and octahedron (c).

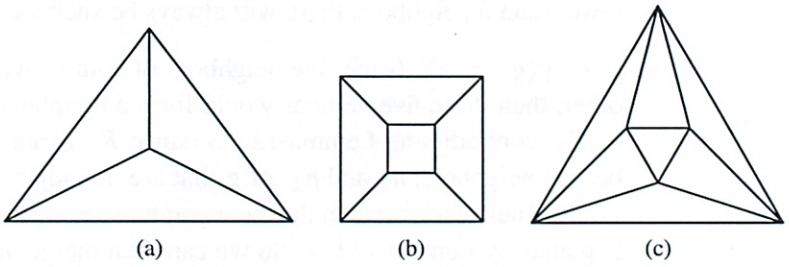


Figure 12.15 Planar embeddings of the tetrahedron (a), cube (b, and octahedron (c).

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 in Figure 12.14: the tetrahedron, the cube, and the octahedron.

We can determine how many more regular polyhedra there are by thinking about planarity. Suppose we took *any* polyhedron and placed a sphere inside it. Then we could project the polyhedron face boundaries onto the sphere, which would give an image that was a planar graph embedded on the sphere, with the images of the corners of the polyhedron corresponding to vertices of the graph. We've already observed that embeddings on a sphere are the same as embeddings on the plane, so Euler's formula for planar graphs can help guide our search for regular polyhedra.

For example, planar embeddings of the three polyhedra in Figure 12.1 are shown in Figure 12.15.

Let m be the number of faces that meet at each corner of a polyhedron, and let n

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

Figure 12.16 The only possible regular polyhedra.

be the number of edges on each face. In the corresponding planar graph, there are m edges incident to each of the v vertices. By the Handshake Lemma 11.2.1, 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:

$$\frac{2e}{m} - e + \frac{2e}{n} = 2$$

which simplifies to

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{e} + \frac{1}{2} \tag{12.6}$$

Equation 12.6 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 $1/3 + 1/6 = 1/2$, which is less than the right side. Checking the finitely-many cases that remain turns up only five solutions, as shown in Figure 12.16. 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. The largest polyhedron, the dodecahedron, was the other great mathematical secret of the Pythagorean sect.

The 5 polyhedra in Figure 12.16 are the only possible regular polyhedra. So if you want to put more than 20 geocentric satellites in orbit so that they *uniformly* blanket the globe—tough luck!

Problems for Section 12.2

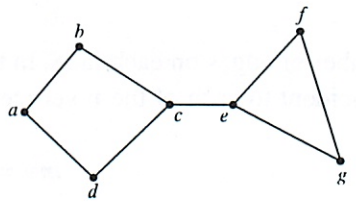
Practice Problems

Problem 12.1.

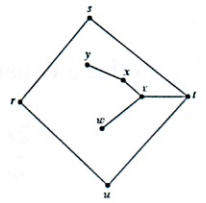
What are the discrete faces of the following two graphs?

Write each cycle as a sequence of letters without spaces, starting with the alphabetically earliest letter in the clockwise direction, for example "`adbfa`." Separate the sequences with spaces.

(a)



(b)



Problems for Section 12.8

Exam Problems

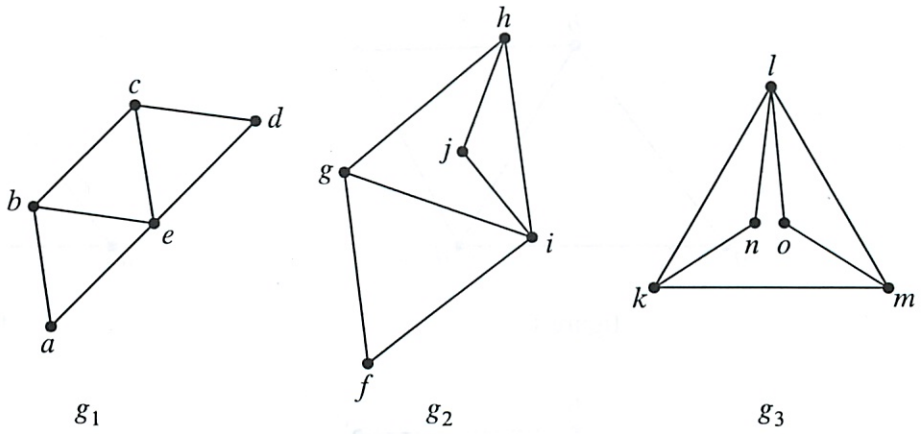
Problem 12.2.

(a) Describe an isomorphism between graphs G_1 and G_2, and another isomorphism between G_2 and G_3.

(b) Why does part (a) imply that there is an isomorphism between graphs G_1 and G_3?

Let G and H be planar graphs. An embedding E_G of G is isomorphic to an embedding E_H of H iff there is an isomorphism from G to H that also maps each face of E_G to a face of E_H.

(c) One of the embeddings pictured above is not isomorphic to either of the others. Which one? Briefly explain why.



g_1 g_2 g_3

(d) Explain why all embeddings of two isomorphic planar graphs must have the same number of faces.

Class Problems

Problem 12.3.

Figures 1–4 show different pictures of planar graphs.

(a) For each picture, describe its discrete faces (simple cycles that define the region borders).

(b) Which of the pictured graphs are isomorphic? Which pictures represent the same *planar embedding*?—that is, they have the same discrete faces.

(c) Describe a way to construct the embedding in Figure 4 according to the recursive Definition 12.2.2 of planar embedding. For each application of a constructor rule, be sure to indicate the faces (cycles) to which the rule was applied and the cycles which result from the application.

Problem 12.4.

Prove the following assertions by structural induction on the definition of planar embedding.

(a) In a planar embedding of a graph, each edge is traversed a total of two times by the faces of the embedding.

(b) In a planar embedding of a connected graph with at least three vertices, each face is of length at least three.

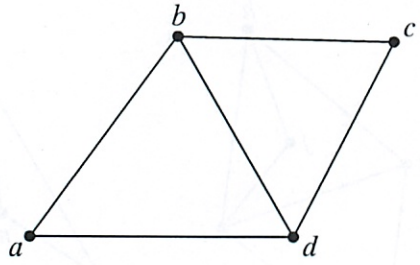


figure 1

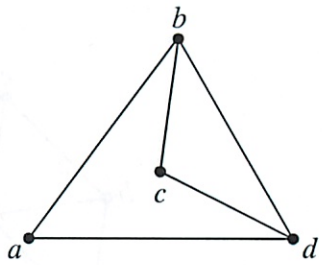


figure 2

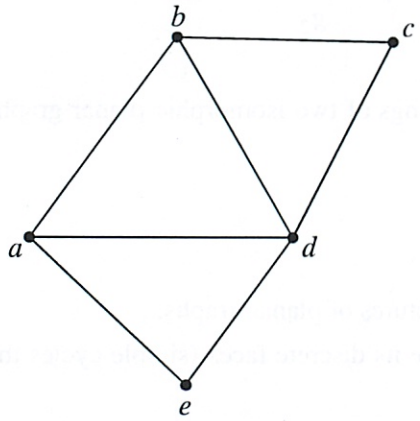


figure 3

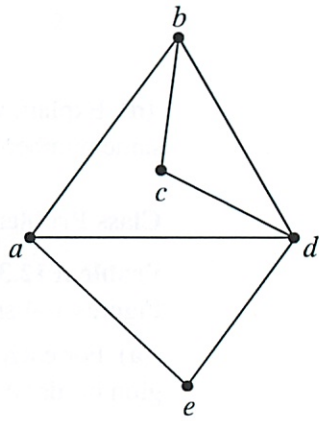


figure 4

Homework Problems

Problem 12.5.

A simple graph is *triangle-free* when it has no simple cycle of length three.

(a) Prove for any connected triangle-free planar graph with $v > 2$ vertices and e edges, $e \leq 2v - 4$.

Hint: Similar to the proof that $e \leq 3v - 6$. Use Problem 12.4.

(b) Show that any connected triangle-free planar graph has at least one vertex of degree three or less.

(c) Prove by induction on the number of vertices that any connected triangle-free planar graph is 4-colorable.

Hint: use part (b).

Problem 12.6. **(a)** Prove

Lemma (Switch Edges)**.** *Suppose that, starting from some embeddings of planar graphs with disjoint sets of vertices, it is possible by two successive applications of constructor operations to add edges e and then f to obtain a planar embedding, $\mathcal{F}$. Then starting from the same embeddings, it is also possible to obtain $\mathcal{F}$ by adding f and then e with two successive applications of constructor operations.*

Hint: There are four cases to analyze, depending on which two constructor operations are applied to add e and then f. Structural induction is not needed.

(b) Prove

Corollary (Permute Edges)**.** *Suppose that, starting from some embeddings of planar graphs with disjoint sets of vertices, it is possible to add a sequence of edges $e_0, e_1, \ldots, e_n$ by successive applications of constructor operations to obtain a planar embedding, $\mathcal{F}$. Then starting from the same embeddings, it is also possible to obtain $\mathcal{F}$ by applications of constructor operations that successively add any permutation[6] of the edges $e_0, e_1, \ldots, e_n$.*

Hint: By induction on the number of switches of adjacent elements needed to convert the sequence 0,1,...,n into a permutation $\pi(0), \pi(1), \ldots, \pi(n)$.

(c) Prove

Corollary (Delete Edge)**.** *Deleting an edge from a planar graph leaves a planar graph.*

[6] If $\pi : \{0, 1, \ldots, n\} \to \{0, 1, \ldots, n\}$ is a bijection, then the sequence $e_{\pi(0)}, e_{\pi(1)}, \ldots, e_{\pi(n)}$ is called a *permutation* of the sequence $e_0, e_1, \ldots, e_n$.