CHAPTER

FOUR

Basic Concepts of Graph Theory

In Example 1.2, we discussed the problem of assigning frequencies to radio stations in the situation where stations within 200 miles of each other must broadcast on distinct frequencies. Clearly we would like to use the smallest number of frequencies possible for a given layouts of transmitters, but how can we determine what that number is?

Suppose three new homes are being built and each of them must be provided with utility connections. The utilites in question are water, electricity, and natural gas. Each provider needs a direct line from their terminal to each house (the line can zig-zag all it wants, but it must go from the terminal to the house without passing through another provider's terminal or another house en route), and the three providers all wish to bury their lines exactly four feet below ground. Can they do this successfully without the lines crossing?

These are just two of many, many examples where the discrete structure known as a *graph* can serve as an enlightening mathematical model. Graphs are perhaps the most basic and widely studied combinatorial structure, and they are prominently featured in this text. Many of the concepts we will study, while presented in a more abstract mathematical sense, have their origins in applications of graphs as models for real-world problems.

4.1 Basic Notation and Terminology for Graphs

A *graph* $\mathbf{G}$ is a pair (V, E) where V is a set (almost always finite) and E is a set of 2-element subsets of V. Elements of V are called *vertices* and elements of E are called *edges*. We call V the *vertex set* of $\mathbf{G}$ and E is the *edge set*. For convenience, it is customary to abbreviate the edge $\{x, y\}$ as just xy. Remember though that $xy \in E$ means exactly the same as $yx \in E$. If x and y are distinct vertices from V, x and y are *adjacent* when

$xy \in E$; otherwise, we say they are *non-adjacent*. We say the edge xy is *incident to* the vertices x and y.

For example, we could define a graph $\mathbf{G} = (V, E)$ with vertex set $V = \{a, b, c, d, e\}$ and edge set $E = \{\{a, b\}, \{c, d\}, \{a, d\}\}$. Notice that no edge is incident to e, which is perfectly permissible based on our definition. It is quite common to identify a graph with a visualization in which we draw a point for each vertex and a line connecting two vertices if they are adjacent. The graph $\mathbf{G}$ we've just defined is shown in Figure 4.1. It's important to remember that while a drawing of a graph is a helpful too, it is not the same as the graph. We could draw $\mathbf{G}$ in any of several different ways without changing what it is as a graph.

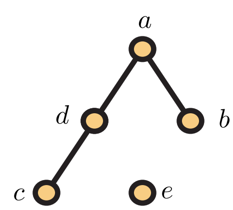


FIGURE 4.1: A GRAPH ON 5 VERTICES

As is often the case in science and mathematics, different authors use slightly different notation and terminology for graphs. As an example, some use *nodes* and *arcs* rather than vertices and edges. Others refer to vertices as *points* and in this case, they often refer to *lines* rather than edges. We will try to stick to vertices and edges but confess that we may occasionally lapse into referring to vertices as points. Also, following the patterns of many others, we will also say that adjacent vertices are *neighbors*. And we will use the more or less standard terminology that the *neighborhood* of a vertex x is the set of vertices adjacent to x. Thus, using the graph $\mathbf{G}$ we have depicted in Figure 4.1, vertices d and a are neighbors, and the neighborhood of d is $\{a, c\}$ while the neighborhood of e is the empty set. Also, the *degree* of a vertex v in a graph $\mathbf{G}$, denoted $\deg_{\mathbf{G}}(v)$, is then the number of vertices in its neighborhood, or equivalently, the number of edges incident to it. For example, we have $\deg_{\mathbf{G}}(d) = \deg_{\mathbf{G}}(a) = 2$, $\deg_{\mathbf{G}}(c) = \deg_{\mathbf{G}}(b) = 1$, and $\deg_{\mathbf{G}}(e) = 0$. If the graph being discussed is clear from context, it is not uncommon to omit the subscript and simply write $\deg(v)$ for the degree of v.

When $\mathbf{G} = (V, E)$ and $\mathbf{H} = (W, F)$ are graphs, we say $\mathbf{H}$ is a *subgraph* of $\mathbf{G}$ when $W \subseteq V$ and $F \subseteq E$. We say $\mathbf{H}$ is an *induced* subgraph when $W \subseteq V$ and $F = \{xy \in E : x, y \in W\}$. In other words, an induced subgraph is defined completely by its vertex set and the original graph $\mathbf{G}$. We say $\mathbf{H}$ is a *spanning* subgraph when $W = V$. In Figure 4.2, we show a graph, a subgraph and an induced subgraph. Neither of these subgraphs is a spanning subgraph

A graph $\mathbf{G} = (V, E)$ is called a *complete* graph when xy is an edge in $\mathbf{G}$ for every distinct pair $x, y \in V$. Conversely, $\mathbf{G}$ is an *independent* graph if $xy \notin E$, for every distinct

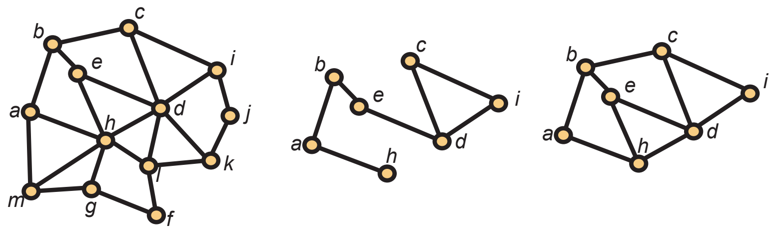


FIGURE 4.2: A GRAPH, A SUBGRAPH AND AN INDUCED SUBGRAPH

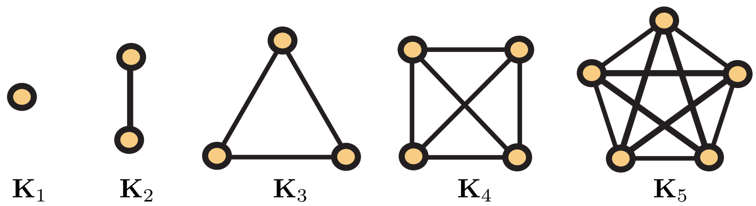


FIGURE 4.3: SMALL COMPLETE GRAPHS

pair $x, y \in V$. It is customary to denote a complete graph on n vertices by $\mathbf{K}_n$ and an independent graph on n vertices by $\mathbf{I}_n$. In Figure 4.3, we show the complete graphs with at most 5 vertices.

A graph $\mathbf{G} = (V, E)$ on $n \geq 1$ vertices is called a *path* when the elements of the vertex set can be labelled as $\{x_1, x_2, \ldots, x_n\}$ so that $E = \{x_i x_{i+1} : 1 \leq i < n\}$. Similarly, if $n \geq 3$, $\mathbf{G}$ is called a *cycle* when $E = \{x_i x_{i+1} : 1 \leq i < n\} \cup \{x_n x_1\}$. It is customary to denote a path on n vertices by $\mathbf{P}_n$, while $\mathbf{C}_n$ denotes a cycle on n vertices. The *length* of a path or a cycle is the number of edges it contains. Therefore, the length of $\mathbf{P}_n$ is $n-1$ and the length of $\mathbf{C}_n$ is n. In Figure 4.4, we show the paths of length at most 4, and in Figure 4.5, we show the cycles of length at most 5.

If $\mathbf{G} = (V, E)$ and $\mathbf{H} = (W, F)$ are graphs, we say $\mathbf{G}$ is *isomorphic* to $\mathbf{H}$ and write $\mathbf{G} \cong \mathbf{H}$ when there exists a bijection $f : V \xrightarrow[\text{onto}]{1\text{–}1} W$ so that x is adjacent to y in $\mathbf{G}$ if and only if $f(x)$ is adjacent to $f(y)$ in $\mathbf{H}$. Often writers will say that $\mathbf{G}$ "contains" $\mathbf{H}$ when there is a subgraph of $\mathbf{G}$ which is isomorphic to $\mathbf{H}$. In particular, it is customary to say that $\mathbf{G}$ contains the cycle $\mathbf{C}_n$ (same for $\mathbf{P}_n$ and $\mathbf{K}_n$) when $\mathbf{G}$ contains a subgraph isomorphic to $\mathbf{C}_n$. The graphs in Figure 4.6 are isomorphic. An isomorphism between these graphs is given by

$$f(a) = 5, \quad f(b) = 3, \quad f(c) = 1, \quad f(d) = 6, \quad f(e) = 2, \quad f(h) = 4.$$

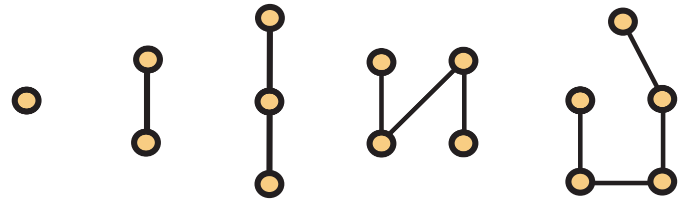


FIGURE 4.4: SHORT PATHS

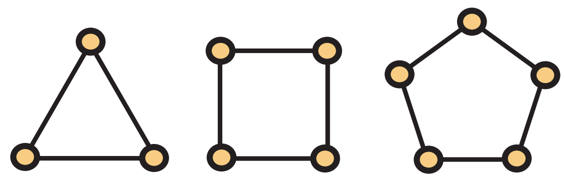


FIGURE 4.5: SMALL CYCLES

On the other hand, the graphs shown in Figure 4.7 are *not* isomorphic, even though they have the same number of vertices and the same number of edges. Can you tell why?

When x and y are vertices in a graph $\mathbf{G} = (V, E)$, we call a sequence $(u_0, u_1, \ldots, u_t)$ of distinct vertices a *path from x to y in* $\mathbf{G}$ when $u_0 = x$, $u_t = y$ and $u_i u_{i+1} \in E$ for all $i = 0, 1, \ldots, t-1$. A graph $\mathbf{G}$ is *connected* when there is a path from x to y in $\mathbf{G}$, for every $x, y \in V$; otherwise, we say $\mathbf{G}$ is *disconnected*. The graph of Figure 4.1 is disconnected (a sufficient justification for this is that there is no path from e to c), while thsoe in Figure 4.6 are connected.

A graph is *acyclic* when it does not contain any cycle on three or more vertices. Acyclic graphs are also called *forests*. A connected acyclic graph is called a *tree*. When $\mathbf{G} = (V, E)$ is a connected graph, a subgraph $\mathbf{H} = (W, F)$ of $\mathbf{G}$ is called a *spanning tree* if $\mathbf{H}$ is

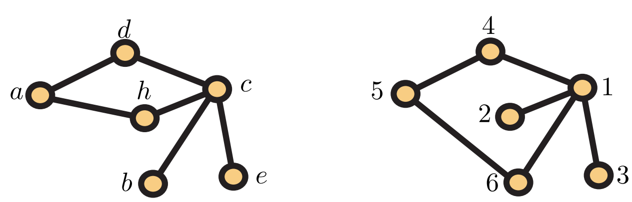


FIGURE 4.6: A PAIR OF ISOMORPHIC GRAPHS

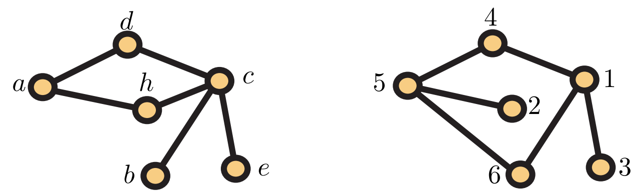


FIGURE 4.7: A PAIR OF NONISOMORPHIC GRAPHS

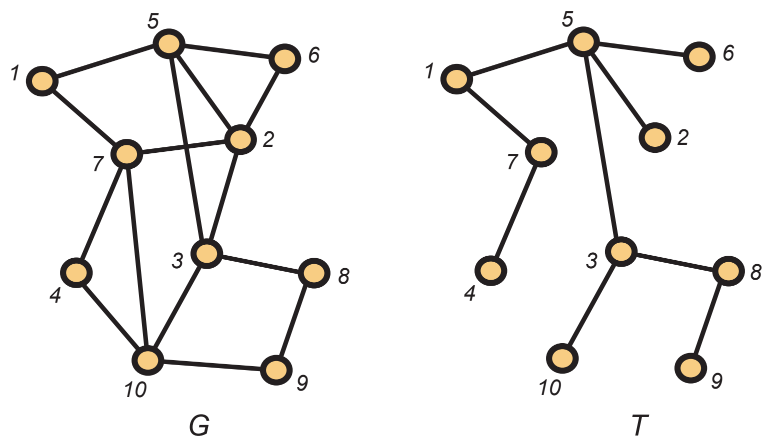


FIGURE 4.8: A GRAPH AND A SPANNING TREE

both a spanning subgraph of **G** and a tree. In Figure 4.8, we show a graph and one of its spanning trees. We will return to the subject of spanning trees in chapter 10.

The following theorem is very elementary, and some authors refer to it as the "first theorem of graph theory". However, this basic result can be surprisingly useful.

Theorem 4.1. *Let* $\deg_{\mathbf{G}}(v)$ *denote the degree of vertex* v *in graph* $\mathbf{G} = (V, E)$*. Then*

$$\sum_{v \in V} \deg_{\mathbf{G}}(v) = 2|E|. \qquad (*)$$

Proof. We consider how many times an edge $e = vw \in E$ contributes to each side of (*). The $\deg_{\mathbf{G}}(x)$ and $\deg_{\mathbf{G}}(y)$ terms on the left hand side each count e once, so e is counted twice on that side. On the right hand side, e is clearly counted twice. Therefore, we have the equality claimed. □

Corollary 4.2. *For any graph, the number of vertices of odd degree is even.* □

We will return to the topic of trees later, but before moving on, let us prove one elementary proposition about trees. First, a *leaf* in a tree $\mathbf{T}$ is a vertex v with $\deg_{\mathbf{T}}(v) = 1$.

Proposition 4.3. *Every tree on $n \geq 2$ vertices has at least two leaves.*

Proof. Our proof is by induction on n. For $n = 2$, there is precisely one tree, which is isomorphic to $\mathbf{K}_2$. Both vertices in this graph are leaves, so the proposition holds for $n = 2$. Now suppose that every tree on m vertices has at least two leaves and let $\mathbf{T}$ be a tree on $m + 1$ vertices. First note that $\mathbf{T}$ must have a leaf, as if all vertices had degree at least 2, you could start at a vertex v and walk from v to another vertex v' and from v' to another vertex and so on but would eventually double back on the vertices you'd already visited, since any time you visit a vertex for the first time there is a way to exit the vertex to another vertex. This would create a cycle, contrary to the fact that $\mathbf{T}$ is a tree. Let v_0 be the leaf that we know must exist and let $\mathbf{T}'$ be the subgraph of $\mathbf{T}$ induced by all the vertices *except* v_0. Then $\mathbf{T}'$ is also a tree (removing v_0 cannot create a disconnected graph) and has m vertices, so by induction $\mathbf{T}'$ has at least two leaves, v_1 and v_2. At least one of v_1 and v_2 is also a leaf in $\mathbf{T}$, since v_0 can be adjacent to at most one of them, and therefore $\mathbf{T}$ has at least two leaves. $\square$

4.2 Multigraphs: Loops and Multiple Edges

Consider a graph in which the vertices represent cities and the edges represent highways. Certain pairs of cities are joined by an edge while other pairs are not. The graph may or may not be connected (although a disconnected graph is likely to result in disgruntled commuters). However, certain aspects of real highway networks are not captured by this model. First, between two nearby cities, there can actually been several interconnecting highways, and traveling on one of them is fundamentally different from traveling on another. This leads to the concept of *multiple edges*, i.e., allowing for more than one edge between two adjacent vertices. Also, we could have a highway which leaves a city, goes through the nearby countryside and the returns to the same city where it originated. This leads to the concept of a *loop*, i.e., an edge with both end points being the same vertex. Also, we can allow for more than one loop with the same end point.

Accordingly, authors frequently lead off a discussion on a graph theory topic with a sentence or two like:

1. In this paper, all graphs will be *simple*, i.e., we will not allow loops or multiple edges.

2. In this paper, graphs can have loops and multiple edges.

The terminology is far from standard, but in this tex, a graph will always be a *simple* graph, i.e., no loops or multiple edges. When we want to allow for loops and multiple

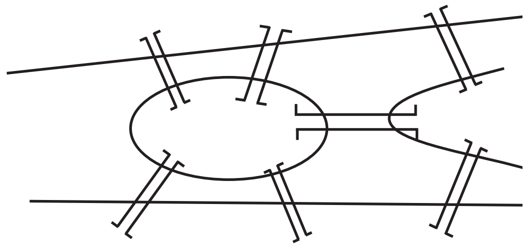


FIGURE 4.9: THE BRIDGES OF KÖNIGSBERG

edges, we will use the term *multigraph*. This begs the question of what we would call a graph if it is allowed to have loops but not multiple edges, or if multiple edges are allowed but not loops. If we *really* needed to talk about such graphs, then the English language comes to our rescue, and we just state the restriction explcitly!

4.3 Eulerian and Hamiltonian Graphs

Graph theory traces its origins to a problem in Königsberg, Prussia (now Kaliningrad, Russia). The river Pregel passes through the city, and there are two large islands in the middle of the channel. These islands were connected to the mainland by seven bridges as indicated in Figure 4.9. It is said that the citizens of Königsberg often wondered if it was possible for one to leave his home, walk through the city in such a way that he crossed each bridge precisely one time, and end up at home again. Leonhard Euler settled this problem in 1736 by using graph theory in the form of Theorem 4.4.

A graph $\mathbf{G}$ is *eulerian* if there is a sequence $(x_0, x_1, x_2, \ldots, x_t)$ of vertices from $\mathbf{G}$, with repetition allowed, so that

1. $x_0 = x_t$;
2. for every $i = 0, 1, \ldots t-1$, $x_i x_{i+1}$ is an edge of $\mathbf{G}$;
3. for every edge $e \in E$, there is a unique integer i with $0 \leq i < t$ for which $e = x_i x_{i+1}$.

When $\mathbf{G}$ is eulerian, a sequence satisfying these three conditions is called an *eulerian circuit*. A sequence of vertices $(x_0, x_1, \ldots, x_t)$ is called a *circuit* when it satisfies only the first two of these conditions. Note that a sequence consisting of a single vertex is a circuit. The following elementary theorem completely characterizes eulerian graphs. It comes with an algorithmic proof, one that is easily implemented.

Theorem 4.4. *A connected graph* $\mathbf{G}$ *is eulerian if and only if every vertex has even degree.*

Proof. We give a deterministic procedure, starting from a specified vertex x_0. We assume that the vertices have been labeled with positive integers (or some other linear order) so that we can consider the neighbors according to a fixed order.

We launch our algorithm with a trivial circuit C consisting of just the vertex x_0. Thereafter suppose that we have a partial circuit C defined by a sequence $(x_0, x_1, \ldots, x_t)$. The edges of the form $x_i x_{i+1}$ have been *traversed*, while the remaining edges in $\mathbf{G}$ (if any) have not. If the third condition for an euler circuit is satisfied, we are done, so we assume it does not hold.

We then choose the least integer i for which there is an edge incident with x_i that has not already been traversed. If there is no such integer, then it is easy to see that the graph is disconnected. On the other hand, if there is such an edge and it is incident with x_i, we simply follow this edge and at the other endpoint, choose another edge that we have not already traversed. If there is no such edge, then we have found a vertex of odd degree, since we have previously used an even number of edges incident with the vertex (one to "enter" it and one to "exit" it). Repeat until you come to a vertex where there are no additional edges to traverse. This must occur at x_i. Then expand the single vertex x_i in the old circuit C by replacing x_i with the string of vertices encountered in this new walk. □

Consider the graph $\mathbf{G}$ shown in Figure 4.10. Evidently, this graph is connected and all vertices have even degree. Here is the sequence of circuits starting with the trivial circuit C consisting only of the vertex 1.

$$\begin{aligned} C &= (1) \\ &= (1,2,4,3,1) \quad \text{start next from 2} \\ &= (1,2,5,8,2,4,3,1) \quad \text{start next from } 4 \\ &= (1,2,5,8,2,4,6,7,4,9,6,10,4,3,1) \quad \text{start next from 7} \\ &= (1,2,5,8,2,4,6,7,9,11,7,4,9,6,10,4,3,1) \quad \text{Done!!} \end{aligned}$$

You should note that Theorem 4.4 holds for loopless graphs in which multiple edges are allowed. Euler used his theorem to show that the multigraph of Königsberg shown in Figure 4.11, in which each land mass is a vertex and each bridge is an edge, is *not* eulerian, and thus the citizens could not find the route they desired. (Note that in Figure 4.11 there are multiple edges between the same pair of vertices.)

A graph $\mathbf{G} = (V, E)$ is said to be *hamiltonian* if there exists a sequence $(x_1, x_2, \ldots, x_n)$ so that

1. every vertex of $\mathbf{G}$ appears exactly once in the sequence;
2. $x_1 x_n$ is an edge of $\mathbf{G}$; and
3. for each $i = 1, 2, \ldots, n-1$, $x_i x_{i+1}$ is an edge in $\mathbf{G}$.

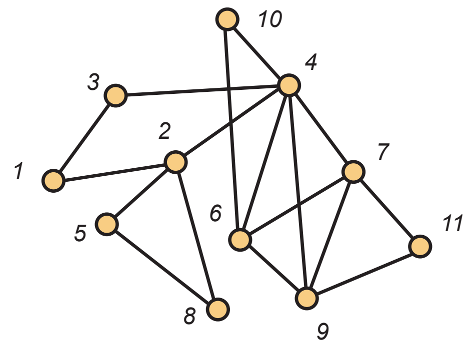


FIGURE 4.10: AN EULERIAN GRAPH

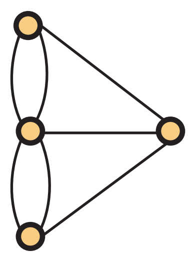


FIGURE 4.11: THE MULTIGRAPH OF KÖNIGSBERG'S BRIDGES

The first graph shown in Figure 4.12 both eulerian and hamiltonian. The second is hamiltonian but not eulerian.

In Figure 4.13, we show a famous graph known as the Petersen graph. It is not hamiltonian.

Unlike the situation with eulerian circuits, there is no known method for quickly determining whether a graph is hamiltonian. However, there are a number of interesting conditions which are sufficient. Here is one quite well known example, due to Dirac.

Theorem 4.5. *If* $\mathbf{G}$ *is a graph on* n *vertices and each vertex in* $\mathbf{G}$ *has at least* $\lceil \frac{n}{2} \rceil$ *neighbors, then* $\mathbf{G}$ *is hamiltonian.*

4.4 Graph Coloring

Let's return now to the subject of Example 1.2, assigning frequencies to radio stations so that they don't interfere. The first thing that we will need to do is to turn the map

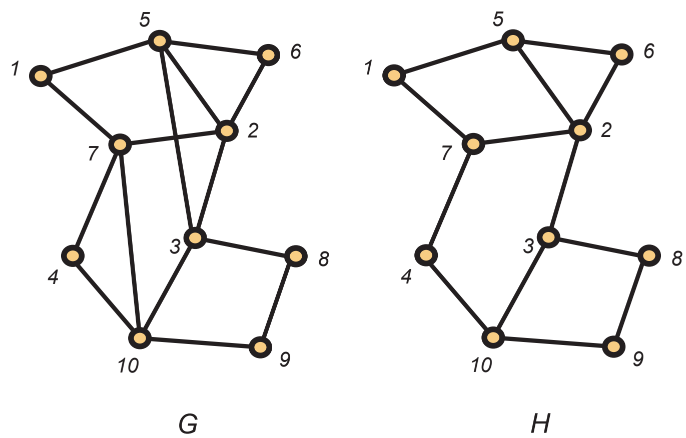


FIGURE 4.12: EULERIAN AND HAMILTONIAN GRAPHS

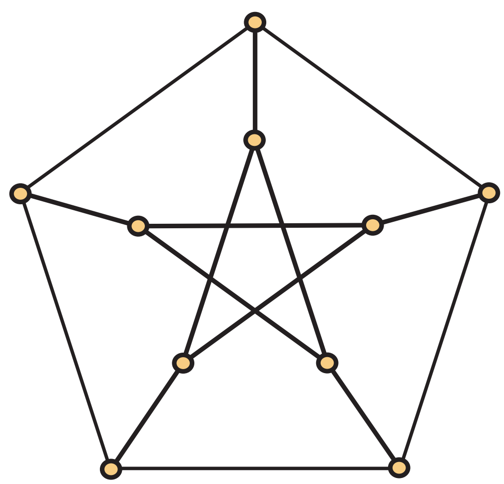


FIGURE 4.13: THE PETERSEN GRAPH

of radio stations into a suitable graph, which should be pretty natural at this juncture. We define a graph $\mathbf{G} = (V, E)$ in which V is the set of radio stations and $xy \in E$ if and only if radio station x and radio station y are within 200 miles of each other. With this as our model, then we need to assign different frequencies to two stations if their corresponding vertices are joined by an edge. This leads us to our next topic, coloring graphs.

When $\mathbf{G} = (V, E)$ is a graph and C is a set of elements called *colors*, a map $\phi : V \to C$ is called a *proper coloring* of $\mathbf{G}$ if $\phi(x) \neq \phi(y)$ whenever xy is an edge in $\mathbf{G}$. The least t for which $\mathbf{G}$ has a proper coloring using a set C of t colors is called the *chromatic number* of $\mathbf{G}$ and is denoted $\chi(\mathbf{G})$. In Figure 4.14, we show a proper coloring of a graph using 5 colors. However, it is easy to see that this graph has chromatic number at most 4. Now

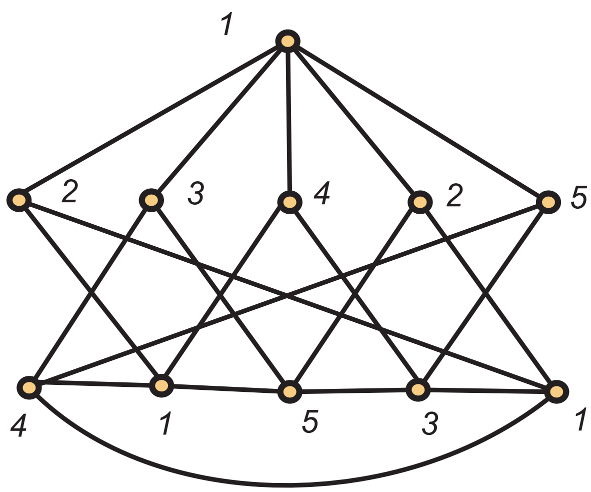


FIGURE 4.14: A GRAPH WITH CHROMATIC NUMBER 4

we can see that our radio frequency assignment problem is the much-studied question of finding the chromatic number of an appropriate graph.

A graph $\mathbf{G} = (V, E)$ with $\chi(\mathbf{G}) = 2$ is called *bipartite* since the coloring function $\phi\colon V \to \{1, 2\}$ induces a partition of V into two sets A and B with $A = \phi^{-1}(1)$ and $B = \phi^{-1}(2)$ having the property that the subgraphs induced by A and B are isomorphic to independent graphs, i.e., no edge has both of its endpoints in A or in B. Clearly the cycles $\mathbf{C}_{2n}$ on an even number of vertices are bipartite, while $\chi(\mathbf{C}_{2n+1}) = 3$ for $n \geq 1$. As the following proposition shows, the existence of an odd cycle (i.e., $\mathbf{C}_{2n+1}$ for $n \geq 1$)

is the only impediment to a graph being bipartite.

Proposition 4.6. *A graph is bipartite if and only if it does not contain an odd cycle.*

Proof. Let $\mathbf{G} = (V, E)$ be a bipartite graph whose coloring function partitions V as $A \cup B$. Since there are no edges between vertices on the same side of the partition, any cycle in $\mathbf{G}$ must alternate vertices between A and B. In order to complete the cycle, therefore, the number of vertices in the cycle from A must be the same as the number from B, implying that the cycle has even length.

Now suppose that $\mathbf{G}$ does not contain an odd cycle. Note that we may assume that $\mathbf{G}$ is connected, as each component may be colored individually. The *distance* $d(u, v)$ between vertices $u, v \in V$ is the length of a shortest path from u to v, and of course $d(u, u) = 0$. Fix a vertex $v_0 \in V$ and define

$$A = \{v \in V : d(v_0, v) \text{ is even}\} \quad \text{and} \quad B = \{v \in V : d(v_0, v) \text{ is odd}\}.$$

We claim that coloring the vertices of A with color 1 and the vertices of B with color 2 is a proper coloring. Suppose not. Then without loss of generality, there are vertices $x, y \in A$ such that $xy \in E$. Since $x, y \in A$, $d(v_0, x)$ and $d(v_0, y)$ are both even. Let

$$v_0, x_1, x_2, \ldots, x_n = x$$

and

$$v_0, y_1, y_2, \ldots, y_m = y$$

be shortest paths from v_0 to x and y, respectively. If $x_i \neq y_j$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$, then

$$v_0, x_1, x_2, \ldots, x_n = x, y = y_m, y_{m-1}, \ldots, y_2, y_1, v_0$$

is an odd cycle in $\mathbf{G}$, which is a contradiction. Thus, there must be i, j such that $x_i = y_j$, and we may take i, j as large as possible. (That is, after $x_i = y_j$, the two paths do not intersect again.) Thus

$$x_i, x_{i+1}, \ldots, x_n = x, y = y_m, y_{m-1}, \ldots, y_j = x_i$$

is a cycle in $\mathbf{G}$. How many vertices are there in this cycle? A quick count shows that it has

$$n - (i - 1) + m - (j - 1) - 1 = n + m - (i + j) + 1$$

vertices. We know that n and m are even, and notice that i and j are either both even or both odd, since $x_i = y_j$ and the odd-subscripted vertices of our path belong to B while those with even subscripts belong to A. Thus $i + j$ is even, so $n + m - (i + j) + 1$ is odd, giving a contradiction. $\square$

A *clique* in a graph $\mathbf{G} = (V, E)$ is a set $K \subseteq V$ such that the subgraph induced by K is isomorphic to the complete graph $\mathbf{K}_{|K|}$. Equivalently, we can say that every pair of vertices in K are adjacent. The *maximum clique size* or *clique number* of a graph $\mathbf{G}$, denoted $\omega(\mathbf{G})$ is the largest t for which there exists a clique K with $|K| = t$. For example, the graph in Figure 4.14 has maximum clique size 2.

For every graph $\mathbf{G}$, it is obvious that $\chi(\mathbf{G}) \geq \omega(\mathbf{G})$. On the other hand, the inequality may be far from tight. In fact, we have the following elementary result.

Proposition 4.7. *For every $t \geq 3$, there exists a graph $\mathbf{G}_t$ so that $\chi(\mathbf{G}_t) = t$ and $\omega(\mathbf{G}_t) = 2$*

Proof. We proceed by induction on t. For $t = 3$, we take $\mathbf{G}_3$ to be the cycle $\mathbf{C}_5$ on five vertices. Now assume that for some $t \geq 3$, we have determined the graph $\mathbf{G}_t$. Suppose that $\mathbf{G}_t$ has n_t vertices. Label the vertices of $\mathbf{G}_t$ as $x_1, x_2, \ldots, x_{n_t}$. Construct $\mathbf{G}_{t+1}$ as follows. Begin with an independent set I of cardinality $t(n_t - 1) + 1$. For every subset S of I with $|S| = n_t$, label the elements of S as $y_1, y_2, \ldots, y_{n_t}$. For this particular n_t-element subset attach a copy of $\mathbf{G}_t$ with y_i adjacent to x_i for $i = 1, 2, \ldots, n_t$. Vertices in copies of $\mathbf{G}_t$ for distinct n_t-element subsets of I are nonadjacent, and a vertex in I has at most one neighbor in a particular copy of $\mathbf{G}_t$.

To see that $\omega(\mathbf{G}_{t+1}) = 2$, it will suffice to argue that $\mathbf{G}_{t+1}$ contains no triangle ($\mathbf{K}_3$). Since $\mathbf{G}_t$ is triangle-free, any triangle in $\mathbf{G}_{t+1}$ must contain a vertex of I. Since none of the vertices of I are adjacent, any triangle in $\mathbf{G}_{t+1}$ contains only one point of I. Since each vertex of I is adjacent to at most one vertex of any fixed copy of $\mathbf{G}_t$, if $y \in I$ is part of a triangle, the other two vertices must come from distinct copies of $\mathbf{G}_t$. However, vertices in different copies of $\mathbf{G}_t$ are not adjacent, so $\omega(\mathbf{G}_{t+1}) = 2$. Notice that $\chi(\mathbf{G}_{t+1}) \geq t$ since $\mathbf{G}_{t+1}$ contains $\mathbf{G}_t$. On the other hand, $\chi(\mathbf{G}_{t+1}) \leq t + 1$ since we may use t colors on the copies of $\mathbf{G}_t$ and a new color on the independent set I. To see that $\chi(\mathbf{G}_{t+1}) = t + 1$, observe that if we use only t colors, then by the Pigeon Hole Principle, there is an n_t-element subset of I in which all vertices have the same color. Then this color cannot be used in the copy of $\mathbf{G}_t$ which is attached to that n_t-element subset. $\square$

Here is another argument for the same result.

Proof. We again start with $\mathbf{G}_3$ as the cycle $\mathbf{C}_5$. As before we assume that we have constructed for some $t \geq 3$ a graph $\mathbf{G}_t$ with $\omega(\mathbf{G}_t) = 2$ and $\chi(\mathbf{G}_t) = t$. Again, label the vertices of $\mathbf{G}_t$ as $x_1, x_2, \ldots, x_{n_t}$. To construct $\mathbf{G}_{t+1}$, we now start with an independent set I, but now I has only n_t points, which we label as $y_1, y_2, \ldots, y_{n_t}$. We then add a copy of $\mathbf{G}_t$ with y_i adjacent to x_j if and only if x_i is adjacent to x_j. Finally, attach a new vertex z adjacent to all vertices in I.

By Exercise 12, $\chi(\mathbf{G}_{t+1}) \leq t + 1$. We claim that in fact $\chi(\mathbf{G}_{t+1}) = t + 1$. Suppose not. Then since $\mathbf{G}_{t+1}$ contains $\mathbf{G}_t$, $\chi(\mathbf{G}_{t+1}) = t$. Let ϕ be a proper coloring of $\mathbf{G}_{t+1}$. Without loss of generality, ϕ uses the colors in $\{1, 2, \ldots, t\}$ and ϕ assigns color t to z. Then consider the nonempty set S of vertices in the copy of $\mathbf{G}_t$ to which ϕ assigns color t. For each x_i in S, change the color on x_i so that it matches the color assigned to y_i by

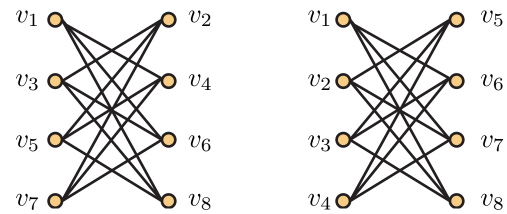


FIGURE 4.15: TWO ORDERINGS OF THE VERTICES OF A BIPARTITE GRAPH.

ϕ, which cannot be t, as$z$ is colored t. What results is a proper coloring of the copy of $\mathbf{G}_t$ with only $t-1$ colors since x_i and y_i are adjacent to the same vertices of the copy of $\mathbf{G}_t$. The contradiction shows that $\chi(\mathbf{G}_{t+1}) = t+1$, as claimed. □

Since a 3-clique looks like a triangle, Proposition 4.7 is often stated as "There exist triangle-free graphs with large chromatic number."

As an illustration of the construction in the second proof above, we again refer to Figure 4.14. The graph shown is $\mathbf{G}_4$.

In general, determining either the maximum clique size of a graph or its chromatic number are (apparently) very difficult problems. A very naïve algorithmic way to approach graph coloring is the First Fit, or "greedy", algorithm. For this algorithm, we take a fixed ordering of the vertex set $V = \{v_1, v_2, \ldots v_n\}$ and define the coloring function ϕ one vertex at a time in increasing order of subscript. We begin with $\phi(v_1) = 1$ and then we define $\phi(v_{i+1})$ (assuming vertices $v_1, v_2, \ldots, v_i$ have been colored) to be the least positive integer color that has not already been used on any of its neighbors in the set $\{v_1, \ldots v_i\}$.

Figure 4.15 shows two different orderings of the same graph. Exercise 15 demonstrates that the ordering of V is vital to the ability of the First Fit algorithm to color $\mathbf{G}$ using $\chi(\mathbf{G})$ colors. In general, finding an optimal ordering is just as difficult as coloring $\mathbf{G}$. However, for some classes of graphs, there is a "natural" ordering that leads to optimal performance of First Fit. Here is one such example—one that we will study again in the next chapter in a different context.

Given an indexed family of sets $\mathcal{F} = \{S_\alpha : \alpha \in V\}$, we associate with $\mathcal{F}$ a graph $\mathbf{G}$ defined as follows. The vertex set of $\mathbf{G}$ is the set V and vertices x and y in V are adjacent in $\mathbf{G}$ if and only if $S_x \cap S_y \neq \emptyset$. We call $\mathbf{G}$ an *intersection graph*. It is easy to see that every graph is an intersection graph (*why?*), so it makes sense to restrict the sets which belong to $\mathcal{F}$. For example, we call $\mathbf{G}$ an *interval graph* if it is the intersection graph of a family of closed intervals of the real line $\mathbb{R}$.

Theorem 4.8. *If* $\mathbf{G} = (V, E)$ *is an interval graph, then* $\chi(\mathbf{G}) = \omega(\mathbf{G})$.

Proof. For each $v \in V$, let $I(v) = [a_v, b_v]$ be a closed interval of the real line so that uv is an edge in $\mathbf{G}$ if and only if $I(u) \cap I(v) \neq \emptyset$. Order the vertex set V as $\{v_1, v_2, \ldots v_n\}$

such that $a_1 \leq a_2 \leq \cdots \leq a_n$. (Ties may be broken arbitrarily.) Apply the First Fit coloring algorithm to $\mathbf{G}$ with this ordering on V. Each vertex is adjacent to at most $\omega(\mathbf{G}) - 1$ other vertices, so when the algorithm colors vertex v_i, there will be a color from $\{1, 2, \ldots, \omega(\mathbf{G})\}$ not already in use on its neighbors, and it will receive the smallest such color. Thus, we never need to use more than $\omega(\mathbf{G})$ colors, so $\chi(\mathbf{G}) = \omega(\mathbf{G})$. $\square$

A graph $\mathbf{G}$ is said to be *perfect* if $\chi(\mathbf{H}) = \omega(\mathbf{H})$ for every induced subgraph $\mathbf{H}$. Since an induced subgraph of an interval graph is an interval graph, Theorem 4.8 shows interval graphs are perfect. The study of perfect graphs originated in connection with the theory of communications networks and has proved to be a major area of research in graph theory for many years now.

4.5 Planar Graphs

Let's return to the problem of providing lines for water, electricity, and natural gas to three homes which we discussed in the introduction to this chapter. How can we model this problem using a graph? The best way is to have a vertex for each utility and a vertex for each of the three homes. Then what we're asking is if we can draw the graph that has an edge from each utility to each home so that none of the edges cross. This graph is shown in Figure 4.16 and is denoted $\mathbf{K}_{3,3}$. The notation is related to our use of $\mathbf{K}_n$ for

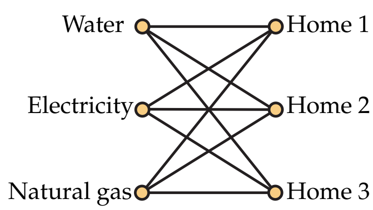


FIGURE 4.16: A GRAPH OF CONNECTING HOMES TO UTILITIES

the complete graph with n vertices. In general, we call $\mathbf{K}_{n,m}$ the *complete bipartite graph with part sizes m and n*, and it consists of a bipartite graph with vertex set $V = A \cup B$ in which $|A| = n$ and $|B| = m$ and edge set $E = \{xy\colon x \in A, y \in B\}$.

While this example of utility lines might seem a bit contrived, since there's really no good reason that the providers can't bury their lines at different depths, the question of whether a graph can be drawn in the plane such that edges intersect only at vertices is a long-studied question in mathematics that does have useful applications. One area where it arises is in the design of microchips and circuit boards. In those contexts, the material is so thin that the option of placing connections at different depths either does not exist or is severely restricted. There is much deep mathematics that underlies this area, and this section is intended to introduce a few of the key concepts.

By a *drawing* of a graph, we mean a way of associating its vertices with points in the Cartesian plane $\mathbb{R}^2$ and its edges with simple polygonal arcs whose endpoints are the points associated to the vertices that are the endpoints of the edge. You can think of a polygonal arc as just a finite sequence of line segments such that the endpoint of one line segment is the starting point of the next line segment, and a simple polygonal arc is one that does not cross itself. (Our choice of polygonal arcs rather than arbitrary curves actually doesn't cause an impediment, since by taking very, very, very short line segments we can approximate any curve.) A *planar drawing* of a graph is one in which the polygonal arcs corresponding to two edges intersect only at a point corresponding to a vertex to which they are both incident. A graph is *planar* if it has a planar drawing. A *face* of a planar drawing of a graph is a region bounded by edges and vertices and not containing any other vertices or edges.

Figure 4.17 shows a planar drawing of a graph with 6 vertices and 9 edges. Notice how one of the edges is drawn as a true polygonal arc rather than a straight line segment. This drawing determines 5 regions, since we also count the unbounded region that surrounds the drawing. Figure 4.18 shows a planar drawing of the complete graph $\mathbf{K}_4$. There are 4 vertices, 6 edges, and 4 faces in the drawing. What happens if we compute the number of vertices minus the number of edges plus the number of faces for these drawings? We have

$$6 - 9 + 5 = 2$$
$$4 - 6 + 8 = 2$$

While it might seem like a coincidence that this computation results in 2 for these planar drawings, there's a more general principle at work here, and in fact it holds for *any* planar drawing of *any* planar graph.

Theorem 4.9 (Euler)**.** *Let* $\mathbf{G}$ *be a connected planar graph with* n *vertices and* m *edges. Every planar drawing of* $\mathbf{G}$ *has* f *faces, where* f *satisfies*

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

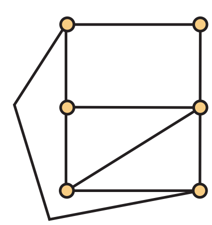


FIGURE 4.17: A PLANAR DRAWING OF A GRAPH

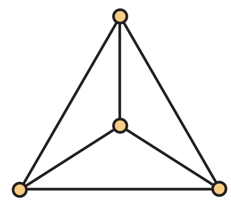


FIGURE 4.18: A PLANAR DRAWING OF $\mathbf{K}_4$

The number 2 here actually results from a fundamental property of the plane, and there are a corresponding theorems for other surfaces. However, we only need the result as stated above.

Proof. Our proof is by induction on the number m of edges. If $m = 0$, then since $\mathbf{G}$ is connected, our graph has a single vertex, and so there is one face. Thus $n - m + f = 1 - 0 + 1 = 2$ as needed. Now suppose that we have proven Euler's formula for all graphs with less than m edges and let $\mathbf{G}$ have m edges. Pick an edge e of $\mathbf{G}$. What happens if we form a new graph $\mathbf{G}'$ by deleting e from $\mathbf{G}$? If $\mathbf{G}'$ is connected, our inductive hypothesis applies. Say that $\mathbf{G}'$ has n' vertices, m' edges, and f' faces. Then by induction, these numbers satisfy

$$n' - m' + f' = 2.$$

Since we only deleted one edge, $n' = n$ and $m' = m - 1$. What did the removal of e do to the number of faces? In $\mathbf{G}'$ there's a new face that was formerly two faces divided by e in $\mathbf{G}$. Thus, $f' = f - 1$. Substituting these into $n' - m' + f' = 2$, we have

$$n - (m - 1) + (f - 1) = 2 \iff n - m + f = 2.$$

Thus, if $\mathbf{G}'$ is connected, we are done. If $\mathbf{G}'$ is disconnected, however, we cannot apply the inductive assumption to $\mathbf{G}'$ directly. Fortunately, since we removed only one edge, $\mathbf{G}'$ has two components, which we can view as two connected graphs $\mathbf{G}'_1$ and $\mathbf{G}'_2$. Each of these has fewer than m edges, so we may apply the inductive hypothesis to them. For $i = 1, 2$, let n'_i be the number of vertices of $\mathbf{G}'_i$, m'_i the number of edges of $\mathbf{G}'_i$, and f'_i the number of faces of $\mathbf{G}'_i$. Then by induction we have

$$n'_1 - m'_1 + f'_1 = 2 \quad \text{and} \quad n'_2 - m'_2 + f'_2 = 2.$$

Adding these together, we have

$$(n'_1 + n'_2) - (m'_1 + m'_2) + (f'_1 + f'_2) = 4.$$

But now $n = n_1' + n_2'$, and $m_1' + m_2' = m - 1$, so the equality becomes

$$n - (m-1) + (f_1' + f_2') = 4 \iff n - m + (f_1' + f_2') = 3.$$

The only thing we have yet to figure out is how $f_1' + f_2'$ relates to f, and we have to hope that it will allow us to knock the 3 down to a 2. Every face of $\mathbf{G}_1'$ and $\mathbf{G}_2'$ is a face of $\mathbf{G}$, since the fact that removing e disconnects $\mathbf{G}$ means that e must be part of the boundary of the unbounded face. Further, the unbounded face is counted twice in the sum $f_1' + f_2'$, so $f = f_1' + f_2' - 1$. This gives exactly what we need to complete the proof. □

Taken by itself, Euler's formula doesn't seem that useful, since it requires counting the number of faces in a planar embedding. However, we can use this formula to get a quick way to determine that a graph is not planar. We consider pairs (e, F) where e is an edge of $\mathbf{G}$ and F is a face that has e as part of its boundary. How many such pairs are there? Let's call the number of pairs p. Each edge can bound either one or two faces, so we have that $p \leq 2m$. We can also bound p by counting the number of pairs in which a face F appears. Each face is bounded by at least 3 edges, so it appears in at least 3 pairs, and so $p \geq 3f$. Thus $3f \leq 2m$ or $f \leq 2m/3$. Now, utilizing Euler's formula, we have

$$m = n + f - 2 \leq n + \frac{2m}{3} - 2 \iff \frac{m}{3} \leq n - 2.$$

Thus, we've proven the following theorem.

Theorem 4.10. *A planar graph on n vertices has at most $3n - 6$ edges.*

The contrapositive of this theorem, namely that an n-vertex graph with more than $3n - 6$ edges is not planar, is usually the most useful formulation of this result. For instance, we've seen (Figure 4.18) that $\mathbf{K}_4$ is planar. What about $\mathbf{K}_5$? It has 5 vertices and $C(5,2) = 10 > 9 = 3 \cdot 5 - 6$ edges, so it is not planar, and thus for $n \geq 5$, $\mathbf{K}_n$ is not planar, since it contains $\mathbf{K}_5$. It's important to note that Theorem 4.10 is not the be-all, end-all of determining if a graph is planar. To see this, let's return to the subject of drawing $\mathbf{K}_{3,3}$ in the plane. This graph has 6 vertices and 9 edges, so it passes the test of Theorem 4.10. However, if you spend a couple minutes trying to find a way to draw $\mathbf{K}_{3,3}$ in the plane without any crossing edges, you'll pretty quickly begin to believe that it can't be done—and you'd be right!

To see why $\mathbf{K}_{3,3}$ is not planar, we'll have to return to Euler's formula, and we again work with edge-face pairs. For $\mathbf{K}_{3,3}$, we see that every edge would have to be part of the boundary of two faces, and faces are bounded by cycles. Also, since the graph is bipartite, there are no odd cycles. Thus, counting edge-face pairs from the edge perspective, we see that there are $2m = 18$ pairs. If we let f_k be the number of faces bounded by a cycle of length k, then $f = f_4 + f_6$. Thus, counting edge-face pairs from the face perspective, there are $4f_4 + 6f_6$ pairs. From Euler's formula, we see that the number of faces f must be 5, so then $4f_4 + 6f_6 \geq 20$. But from our count of edge-face pairs, we have $2m = 4f_4 + 6f_6$, giving $18 \geq 20$, which is clearly absurd. Thus, $\mathbf{K}_{3,3}$ is not planar.

At this point, you're probably asking yourself "So what?" We've invested a fair amount of effort to establish that $\mathbf{K}_5$ and $\mathbf{K}_{3,3}$ are nonplanar. Clearly any graph that contains them is also nonplanar, but there are a lot of graphs, so you might think that we could be at this forever. Fortunately, we won't be, since at its core, planarity really comes down to just these two graphs. To see how, we first need to discuss the concept of a subdivision. If $\mathbf{G}$ is a graph and uv is an edge of $\mathbf{G}$, we can form a new graph $\mathbf{G}'$ called a *subdivision* of $\mathbf{G}$ by adding a new vertex v' and replacing the edge uv by edges uv' and $v'v$. What happens to $\mathbf{K}_5$ if we subdivide any of its edges? Clearly it remains nonplanar. If you take any nonplanar graph and subdivide one of its edges, the resulting graph is nonplanar. The following very deep theorem was proved by the Polish mathematician Kazimierz Kuratowski in 1930. Its proof is beyond the scope of this text.

Theorem 4.11 (Kuratowski)**.** *A graph is planar if and only if it does not contain a subgraph isomorphic to a subdivision of* $\mathbf{K}_5$ *or* $\mathbf{K}_{3,3}$.

Kuratowski's Theorem gives a useful way for checking if a graph is planar. Although it's not always easy to find a subdivision of $\mathbf{K}_5$ or $\mathbf{K}_{3,3}$ by hand, there are efficient algorithms for planarity testing that make use of this characterization. To see this theorem at work, let's consider the Peterson graph shown in Figure 4.13. The Petersen graph has 10 vertices and 15 edges, so it passes the test of Theorem 4.10, and our argument using Euler's formula to prove that $\mathbf{K}_{3,3}$ is nonplanar was complex enough, we probably don't want to try it for the Petersen graph. To use Kuratowski's Theorem here, we need to make a decision regarding which graph, $\mathbf{K}_5$ or $\mathbf{K}_{3,3}$, we want to try to find a subdivision of. Although the Petersen graph looks very similar to $\mathbf{K}_5$, it's actually simultaneously *too* similar and too different for us to be able to find a subdivision of $\mathbf{K}_5$, since each vertex has degree 3. Thus, we set out to find a subgraph of the Petersen graph isomorphic to a subdivision of $\mathbf{K}_{3,3}$. To do so, note that $\mathbf{K}_{3,3}$ contains a cycle of length 6 and three edges that are in place between vertices opposite each other on the cycle. We identify a six-cycle in the Petersen graph and draw it as a hexagon and place the remaining four vertices inside the cycle. such a drawing is shown in Figure 4.19. The subdivision of $\mathbf{K}_{3,3}$ is found by deleting the black vertex, as then the white vertices have degree two, and we can replace each of them and their two incident edges (shown in bold) by a single edge.

We close this section with a problem that brings the current section together with the topic of graph coloring. In 1852 Francis Guthrie, an Englishman who was at the time studying to be lawyer but subsequently became a professor of mathematics in South Africa, was trying to color a map of the counties of England so that any two counties that shared a boundary segment (meaning they touched in more than a single point) were colored with different colors. He noticed that he only needed four colors to do this, and was unable to draw any sort of map that would require five colors. (He was able to find a map that required four colors, an example of which is shown in Figure 4.20.) Could it possibly be true that *every* map could be colored with only four colors? He asked his brother Frederick Guthrie, who was a mathematics student at University College,

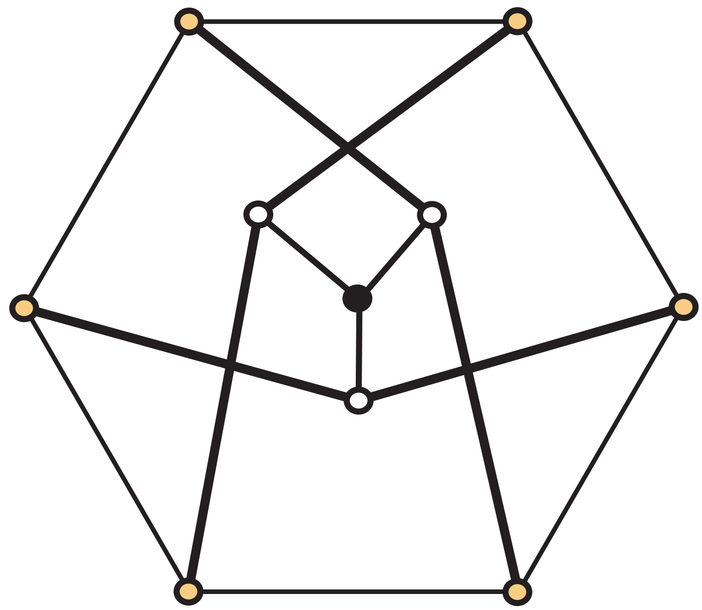


FIGURE 4.19: A MORE ILLUSTRATIVE DRAWING OF THE PETERSEN GRAPH

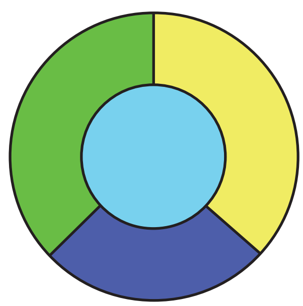


FIGURE 4.20: A MAP THAT REQUIRES FOUR COLORS

London, about the problem, and Frederick eventually communicated the problem to Augustus de Morgan (of de Morgan's laws fame), one of his teachers. It was in this way that one of the most famous (or infamous) problems, known for a century as the Four Color Problem and now the Four Color Theorem, in graph theory was born. De Morgan was very interested in the Four Color Problem, and communicated it to Sir William Rowan Hamilton, another prominent British mathematician and the one for home hamiltonian cycles are named, but Hamilton did not find the problem interesting. Hamilton is one of the few people who considered the Four Color Problem but did not become captivated by it.

We'll continue our discussion of the history of the Four Color Theorem in a moment, but first, we must consider how we can turn the problem of coloring a map into a graph theory question. Well, it seems natural that each region should be assigned a corresponding vertex. We want to force regions that share a boundary to have different colors, so this suggests that we should place an edge between two vertices if and only if their corresponding regions have a common boundary. (As an example, the map in Figure 4.20 corresponds to the graph $\mathbf{K}_4$.) It is not difficult to see that this produces a planar graph, since we may draw the edges through the common boundary segment. Furthermore, with a little bit of thought, you should see that given a planar drawing of a graph, you can create a map in which each vertex leads to a region and edges lead to common boundary segments. Thus, the Four Color Problem could be stated as "Does every planar graph have chromatic number at most four?"

Interest in the Four Color Problem languished until 1877, when the British mathematician Arthur Cayley worte a letter to the Royal Society asking if the problem had been resolved. This brought the problem to the attention of many more people, and the first "proof" of the Four Color Theorem, due to Alfred Bray Kempe, was completed in 1878 and published a year later. It took 11 years before Percy John Heawood found a flaw in the proof but was able to salvage enough of it to show that every planar graph has chromatic number at most five. In 1880, Peter Guthrie Tait, a British physicist best known for his book *Treatise on Natural Philosophy* with Sir William Thomson (Lord Kelvin), made an announcement that suggested he had a proof of the Four Color Theorem utilizing hamiltonian cycles in certain planar graphs. However, consistent with the way Tait approached some conjectures in the mathematical theory of knots, it appears that he subsequently realized around 1883 that he could not prove that the hamiltonian cycles he was using actually existed and so Tait likely only believed he had a proof of the Four Color Theorem for a short time, if at all. However, it would take until 1946 to find a counterexample to the conjecture Tait had used in his attempt to prove the Four Color Theorem.

In the first half of the twentieth century, some incremental progress toward resolving the Four Color Problem was made, but few prominent mathematicians took a serious interest in it. The final push to prove the Four Color Theorem came with about at the same time that the first electronic computers were coming into widespread use in industry and research. In 1976, two mathematicians at the University of Illinois announced their computer-assisted proof of the Four Color Theorem. The proof by Kenneth Appel and Wolfgang Haken led the University of Illinois to add the phrase "FOUR COLORS SUFFICE" to its postage meter's imprint.[1]

Theorem 4.12 (Four Color Theorem)**.** *Every planar graph has chromatic number at most four.*

Appel and Haken's proof of the Four Color Theorem was at a minimum unsatisfactory for many mathematicians, and to some it simply wasn't a proof. These mathematicians felt that the using a computer to check various cases was simply too uncertain; how could you be certain that the code that checked the 1,482 "unavoidable configurations" didn't contain any logic errors? In fact, there were several mistakes found in the cases analyzed, but none were found to be fatal flaws. In 1989, Appel and Haken published a 741-page tome entitled *Every Planar Map is Four Colorable* which provided corrections to all known flaws in their original argument. This still didn't satisfy many, and in the early 1990's a team consisting of Neil Robertson from The Ohio State University; Daniel P. Sanders, a graduate student at the Georgia Institute of Technology; Paul Seymour of Bellcore; and Robin Thomas from Georgia Tech announced a new proof of the Four Color Theorem. However, it still required the use of computers. The proof did gain more widespread acceptance than that of Appel and Haken, in part because the new proof used fewer than half (633) of the number of configurations the Appel-Haken proof used and the computer code was provided online for anyone to verify. While still unsatisfactory to many, the proof by Robertson, et al. was generally accepted, and today the issue of the Four Color Theorem has largely been put to rest. However, many still wonder if anyone will ever find a proof of this simple statement that does not require the assistance of a computer.

4.6 Counting Labeled Trees

How many trees are there with vertex set $[n] = \{1, 2, \ldots, n\}$? Let T_n be this number. For $n = 1$, there is clearly only one tree. Also, for $n = 2$, there is only one tree, which is isomorphic to $\mathbf{K}_2$. In determining T_3, we finally have some work to do; however, there's not much, since all trees on 3 vertices are isomorphic to $\mathbf{P}_3$. Thus, there are $T_3 = 3$ *labeled* trees on 3 vertices, corresponding to which vertex is the one of degree 2. When $n = 4$, we can begin by counting the number of nonisomorphic trees and consider two cases depending on whether the tree has a vertex of degree 3. If there is a vertex of degree 3, the tree is isomorphic to $\mathbf{K}_{1,3}$ or it does not have a vertex of degree three, in which case it is isomorphic to $\mathbf{P}_4$, since there must be precisely two vertices of degree 2 in such a graph. There are four labelings by $[4]$ for $\mathbf{K}_{1,3}$ (choose the vertex of degree three). How

[1] A photograph of an envelope with such a meter mark on it can be found in the book *The Four-Color Theorem: History, Topological Foundations, and Idea of Proof* by Rudolf and Gerda Fritsch. (Springer, 1998)

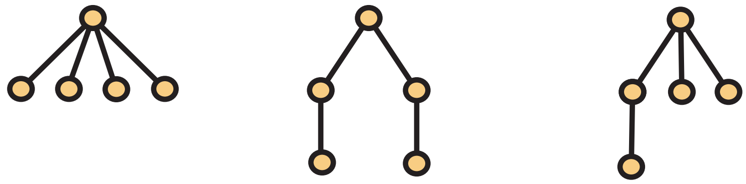


FIGURE 4.21: THE NONISOMORPHIC TREES ON $n = 5$ VERTICES

many labelings by $[4]$ are there for $\mathbf{P}_4$? There are $C(4,2)$ ways to choose the labels i, j given to the vertices of degree 2 and two ways to select one of the remaining labels to be made adjacent to i. Thus, there are 12 ways to label $\mathbf{P}_4$ by $[4]$ and so $T_4 = 16$.

To this point, it looks like maybe there's a pattern forming. Perhaps it is the case that for all $n \geq 1$, $T_n = n^{n-2}$. This is in fact the case, but let's see how it works out for $n = 5$ before proving the result in general. What are the nonisomorphic trees on five vertices? Well, there's $\mathbf{K}_{1,4}$ and $\mathbf{P}_5$ for sure, and there's also the third tree shown in Figure 4.21. After thinking for a minute or two, you should be able to convince yourself that this is all of the possibilities. How many labelings by $[5]$ does each of these have? There are 5 for $\mathbf{K}_{1,4}$ since there are 5 ways to choose the vertex of degree 4. For $\mathbf{P}_5$, there are 5 ways to choose the middle vertex of the path, $C(4,2) = 6$ ways to label the two remaining vertices of degree 2 once the middle vertex is labeled, and then 2 ways to label the vertices of degree 1. This gives 60 labelings. For the last tree, there are 5 ways to label the vertex of degree 3, $C(4,2) = 6$ ways to label the two leaves adjacent to the vertex of degree 3, and 2 ways to label the remaining two vertices, giving 60 labelings. Therefore, $T_5 = 125 = 5^3 = 5^{5-2}$.

It turns out that we are in fact on the right track, and we will now set out to prove the following:

Theorem 4.13 (Cayley's Formula)**.** *The number T_n of labeled trees on n vertices is n^{n-2}.*

This result is usually referred to as Cayley's Formula, although equivalent results were proven earlier by James J. Sylvester (1857) and Carl W. Borchardt (1860). The reason that Cayley's name is most often affixed to this result is that he was the first to state and prove it in graph theoretic terminology (in 1889). (Although one could argue that Cayley really only proved it for $n = 6$ and then claimed that it could easily be extended for all other values of n, and whether such an extension can actually happen is open to some debate.) Cayley's Formula has many different proofs, most of which are quite elegant. If you're interested in presentations of several proofs, we encourage you to read the chapter on Cayley's Formula in *Proofs from THE BOOK* by Aigner, Ziegler, and Hofmann, which contains four different proofs, all using different proof techniques. Here we give a fifth proof, due to Prüfer and published in 1918. Interestingly, even though Prüfer's proof came after much of the terminology of graph theory was established, he seemed unaware of it and worked in the context of permutations and his own terminology, even though his approach clearly includes the ideas of graph theory. We will use a recursive technique in order to find a bijection between the set of labeled trees on n vertices and a natural set of size n^{n-2}, the set of strings of length $n-2$ where the symbols in the string come from $[n]$.

We define a recursive algorithm that takes a tree $\mathbf{T}$ on $k \geq 2$ vertices labeled by elements of a set S of positive integers of size k and returns a string of length $k-2$ whose symbols are elements of S. (The set S will usually be $[k]$, but in order to define a recursive procedure, we need to allow that it be an arbitrary set of k positive integers.) This string is called the *Prüfer code* of the tree $\mathbf{T}$. Let prüfer$(\mathbf{T})$ denote the Prüfer code of the tree $\mathbf{T}$, and if v is a leaf of $\mathbf{T}$, let $\mathbf{T}-v$ denote the tree obtained from $\mathbf{T}$ by removing v (i.e., the subgraph induced by all the other vertices). We can then define prüfer$(\mathbf{T})$ recursively by

1. If $\mathbf{T} \cong \mathbf{K}_2$, return the empty string.
2. Else, let v be the leaf of $\mathbf{T}$ with the smallest label and let u be its unique neighbor. Let i be the label of u. Return $(i, \text{prüfer}(\mathbf{T}-v))$.

Example 4.14. Before using Prüfer codes to prove Cayley's Formula, let's take a moment to make sure we understand how they are computed given a tree. Consider the 9-vertex tree $\mathbf{T}$ in Figure 4.22. How do we compute prüfer$(\mathbf{T})$? Since $\mathbf{T}$ has more than

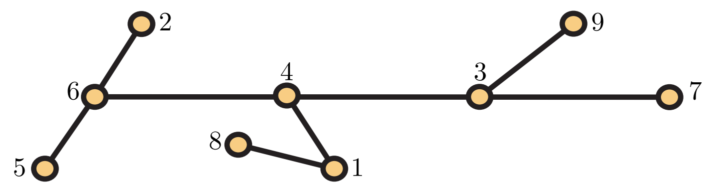


FIGURE 4.22: A LABELED 9-VERTEX TREE

two vertices, we use the second step and find that v is the vertex with label 2 and u is the vertex with label 6, so prüfer$(\mathbf{T}) = (6, \text{prüfer}(\mathbf{T}-v))$. The graph $\mathbf{T}-v$ is shown in Figure 4.23. The recursive call prüfer$(\mathbf{T}-v)$ returns $(6, \text{prüfer}(\mathbf{T}-v-v'))$, where v' is the vertex labeled 5. Continuing recursively, the next vertex deleted is 6, which appends a 4 to the string. Then 7 is deleted, appending 3. Next 8 is deleted, appending 1. This is followed by the deletion of 1, appending 4. Finally 4 is deleted, appending 3, and the final recursive call has the subtree isomorphic to $\mathbf{K}_2$ with vertices labeled 3 and 9, and an empty string is returned. Thus, prüfer$(\mathbf{T}) = 6643143$.

We're now prepared to give a proof of Cayley's Formula.

Proof. It is clear that prüfer$(\mathbf{T})$ takes an n-vertex labeled tree with labels from $[n]$ and returns a string of length $n-2$ whose symbols are elements of $[n]$. What we have yet

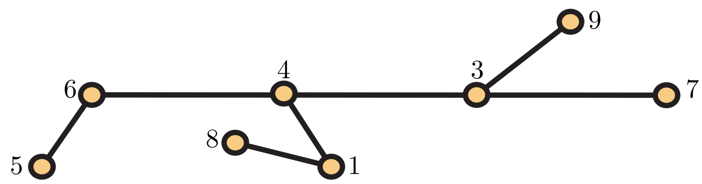


FIGURE 4.23: THE TREE $\mathbf{T} - v$

to do is determine a way to take such a string and construct an n-vertex labeled tree from it. If we can find such a construction, we will have a bijection between the set $\mathcal{T}_n$ of labeled trees on n vertices and the set of strings of length $n-2$ whose symbols come from $[n]$, which will imply that $T_n = n^{n-2}$.

First, let's look at how prüfer($\mathbf{T}$) behaves. What numbers actually appear in the Prüfer code? The numbers that appear in the Prüfer code are the labels of the *non-leaf* vertices of $\mathbf{T}$. The label of a leaf simply cannot appear, since we always record the label of the *neighbor* of the leaf we are deleting, and the only way we would delete the neighbor of a leaf is if that neighbor were also a leaf, which can only happen $\mathbf{T} \cong \mathbf{K}_2$, in which case prüfer($\mathbf{T}$) simply returns the empty string. Thus if $I \subset [n]$ is the set of symbols that appear in prüfer($\mathbf{T}$), the labels of the leaves of $\mathbf{T}$ are precisely the elements of $[n] - I$.

With the knowledge of which labels belong to the leaves of $\mathbf{T}$ in hand, we are ready to use induction to complete the proof. Our goal is to show that if given a string $\mathsf{s} = s_1 s_2 \cdots s_{n-2}$ whose symbols come from a set S of n elements, there is a unique tree $\mathbf{T}$ with prüfer($\mathbf{T}$) $= \mathsf{s}$. If $n = 2$, the only such string is the empty string, so 1 and 2 both label leaves and we can construct only $\mathbf{K}_2$. Now suppose we have the result for some $m \geq 2$, and we try to prove it for $m+1$. We have a string $\mathsf{s} = s_1 s_2 \cdots s_{m-1}$ with symbols from $[m+1]$. Let I be the set of symbols appearing in s and let k be the least element of $[m+1] - I$. By the previous paragraph, we know that k is the label of a leaf of $\mathbf{T}$ and that is unique neighbor is the vertex labeled s_1. The string $\mathsf{s}' = s_2 s_3 \cdots s_{m-1}$ has length $m-2$ and since k does not appear in s, its symbols come from $S = [m+1] - \{k\}$, which has size m. Thus, by induction, there is a unique tree $\mathbf{T}'$ whose Prüfer code is s'. We form $\mathbf{T}$ from $\mathbf{T}'$ by attaching a leaf with label k to the vertex of $\mathbf{T}'$ with label s_1 and have a tree of the desired type. □

Example 4.15. We close this section with an example of how to take a Prüfer code and use it to construct a labeled tree. Consider the string $\mathsf{s} = 75531$ as a Prüfer code. Then the tree $\mathbf{T}$ corresponding to s has 7 vertices, and its leaves are labeled 2, 4, and 6. The inductive step in our proof attaches the vertex labeled 2 to the vertex labeled 7 in the tree $\mathbf{T}'$ with Prüfer code 5531 and vertex labels $\{1, 3, 4, 5, 6, 7\}$, since 2 is used to label the last vertex added. What are the leaves of $\mathbf{T}'$? The symbols in $\{4, 6, 7\}$ do not appear in 5531, so they must be the labels of leaves, and the construction says that we would attach the vertex labeled 4 to the vertex labeled 5 in the tree we get by induction. In Table 4.1, we show how this recursive process continues. We form each row from the row above it

Prüfer code	Label set	Edge added
75531	$\{1,2,3,4,5,6,7\}$	2–7
5531	$\{1,3,4,5,6,7\}$	4–5
531	$\{1,3,5,6,7\}$	6–5
31	$\{1,3,5,7\}$	5–1
1	$\{1,3,7\}$	3–1
(empty string)	$\{1,7\}$	1–7

TABLE 4.1: TURNING THE PRÜFER CODE 75531 INTO A LABELED TREE

by removing the first label used on the edge added from the label set and removing the first symbol from the Prüfer code. Once the Prüfer code becomes the empty string, we know that the two remaining labels must be the labels we place on the ends of $\mathbf{K}_2$ to start building $\mathbf{T}$. We then work back up the edge added column, adding a new vertex and the edge indicated. The tree we construct in this manner is shown in Figure 4.24.

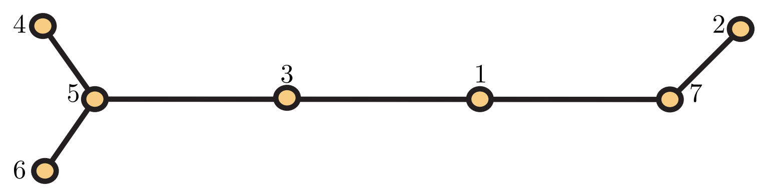


FIGURE 4.24: THE LABELED TREE WITH PRÜFER CODE 75531

4.7 A Digression into Complexity Theory

In this section, we give a *very* informal discussion of some relevant topics from computational complexity. You might call it *Complexity for Dummies.*

Some combinatorial problems can be solved in a number of steps which can be bounded by a polynomial of the input size. For example, consider the following situation. You are given a graph on n vertices and asked whether or not the graph is connected. Notice that in this case, a positive answer can be justified by providing a spanning tree. On the other hand, a negative answer can be justified by providing a partition of the vertex sets $V = V_1 \cup V_2$ with V_1 and V_2 non-empty subsets and having no edges with one end-point in V_1 and the other in V_2.

Here is another example. If you are asked whether a connected graph is eulerian, then a positive answer can be justified by producing the appropriate sequence. A negative answer can be justified by producing a vertex of odd degree.

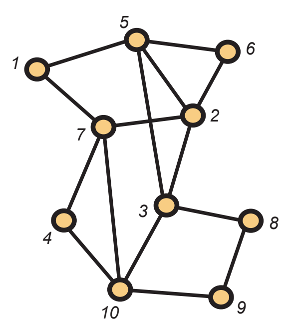
On the other hand, if you are asked whether a graph is hamiltonian, the situation appears to be quite different. A positive answer can be justified by providing the sequence of vertices which forms a hamiltonian cycle. However no one knows how to justify a negative answer—at least not in the general case.

Informally, we consider the class **P** of problems, which when posed as a yes–no question, can be answered in time which is polynomial in the input size. Also, we consider the class **NP** which consists of those problems for which it is possible to justify a "yes" answer with a " certificate", and the correctness of the certificate can be checked in polynomial time. For example, if you give a yes answer to whether a graph is hamiltonian, then you can simply give the sequence of vertices which forms the cycle. The correctness of your cycle can be readily checked. However, you do not have to explain how you were able to find this cycle.

Perhaps the most famous problem in all of combinatorial mathematics and theoretical computer science is to resolve whether the classes **P** and **NP** are the same. It is easy to see that **P** is a subset of **NP**, and most experts believe that there are problems in **NP** which are not in **P**. But—no one has been able to settle this problem, and many, many top researchers have worked very hard on it.

4.8 Exercises

1. Find a cycle of length 8 in the graph **G** below. What is the length of the shortest path from 3 to 4? How about from 8 to 7?



G

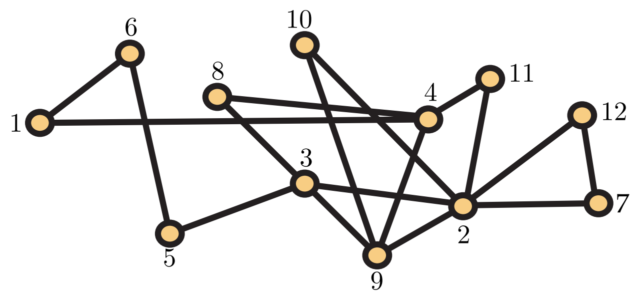
2. Draw a graph with 8 vertices, all of odd degree, that does not contain a path of length 3 or explain why such a graph does not exist.

3. Draw a graph with 6 vertices having degrees 5, 4, 4, 2, 1, and 1 or explain why such a graph does not exist.

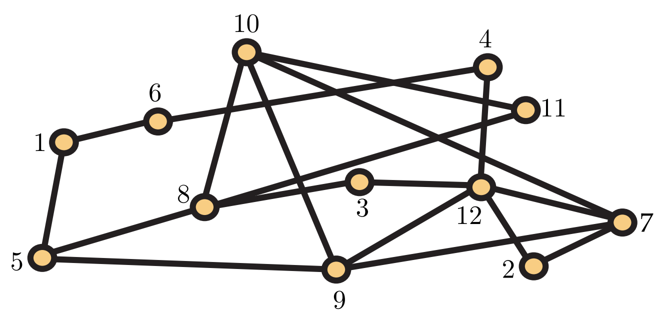
4. For the next Olympic Winter Games, the organizers wish to expand the number of teams competing in curling. They wish to have 14 teams enter, divided into two pools of seven teams each. Right now, they're thinking of requiring that in preliminary play each team will play seven games against distinct opponents. Five of the opponents will come from their own pool and two of the opponents will come from the other pool. They're having trouble setting up such a schedule, so they've come to you. By using an appropriate graph-theoretic model, either argue that they cannot use their current plan or devise a way for them to do so.

5. Prove that every tree on n vertices has exactly $n-1$ edges.

6. Find an eulerian circuit in the graph below or explain why one does not exist.



7. Explain why the graph below does not have an eulerian circuit, but show that by adding a single edge, you can make it eulerian.



8. An *eulerian trail* is defined in the same manner as an euler circuit (see page 4-7) except that we drop the condition that $x_0 = x_t$. Prove that a connected graph has an eulerian trail if and only if it has precisely two vertices of odd degree.

9. Alice and Bob are discussing a graph that has 17 vertices and 129 edges. Bob argues that the graph is Hamiltonian, while Alice says that he's wrong. Without knowing anything more about the graph, must one of them be right? If so, who and why, and if not, why not?

10. A pharmaceutical manufacturer is building a new warehouse to store its supply of 10 chemicals it uses in production. However, some of the chemicals cannot be stored in the same room due to undesirable reactions that will occur. The matrix below has a 1 in position (i, j) if and only if chemical i and chemical j cannot be stored in the same room. Develop an appropriate graph theoretic model and determine the smallest number of rooms into which they can divide their warehouse so that they can safely store all 10 chemicals in the warehouse.

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

11. All trees with more than one vertex have the same chromatic number. What is it, and why?

12. Find a proper $(t+1)$-coloring of the graph $\mathbf{G}_{t+1}$ in the second proof of Proposition 4.7. This establishes that $\chi(\mathbf{G}_{t+1}) \leq t+1$.

13. Construct and draw the graph $\mathbf{G}_4$ from the *first* proof of Proposition 4.7.

14. Construct and draw the graph $\mathbf{G}_5$ from the *second* proof of Proposition 4.7.

15. Use the First Fit algorithm to color the graph in Figure 4.15 using the two different orderings of the vertex set shown there.

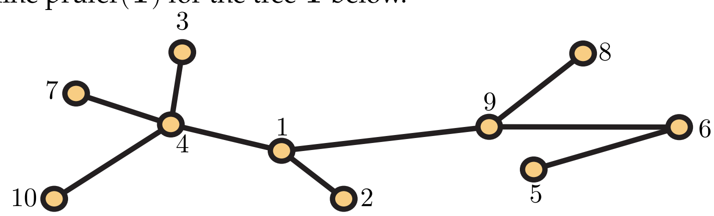
16. Find a planar drawing of the graph $\mathbf{K}_5 - e$, by which we mean the graph formed from the complete graph on 5 vertices by deleting any edge.

17. Draw a planar drawing of an eulerian planar graph with 10 vertices and 21 edges.

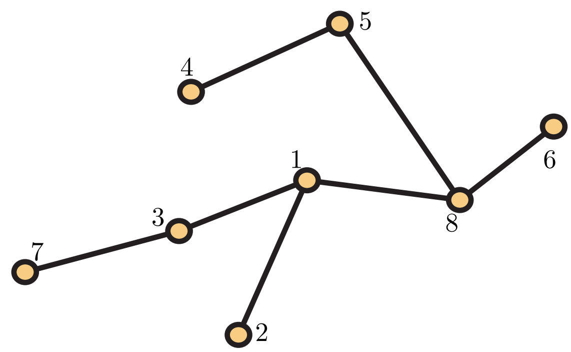
18. Show that every planar graph has a vertex that is incident to at most five edges.

19. Draw the 16 labeled trees on 4 vertices.

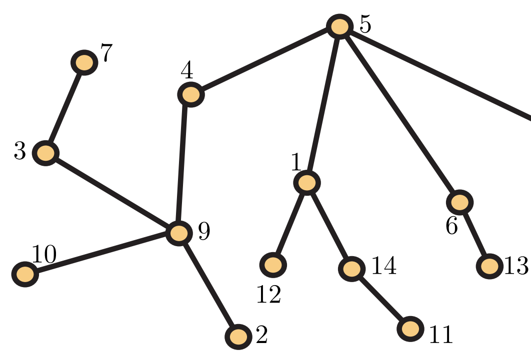
20. Determine prüfer($\mathbf{T}$) for the tree $\mathbf{T}$ below.



21. Determine prüfer($\mathbf{T}$) for the tree $\mathbf{T}$ below.



22. Determine prüfer($\mathbf{T}$) for the tree $\mathbf{T}$ below.



23. Construct the labeled tree $\mathbf{T}$ with Prüfer code 96113473.

24. Construct the labeled tree $\mathbf{T}$ with Prüfer code 23134.

25. Construct the labeled tree $\mathbf{T}$ with Prüfer code (using commas to separate symbols in the string, since we have labels greater than 9) 10, 1, 7, 4, 3, 4, 10, 2, 2, 8.