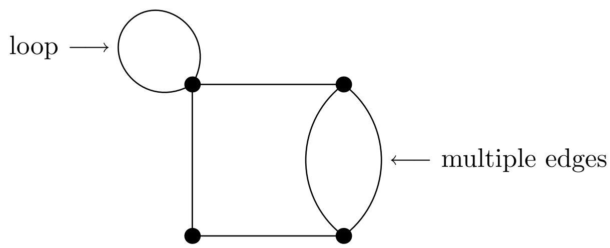
21-110: Problem Solving in Recreational Mathematics

Homework assignment 8 solutions

Problem 1. A *loop* is an edge that joins a vertex to itself. *Multiple edges* are two or more edges between the same pair of vertices. (See the picture below.) A *simple graph* is a graph with no loops and no multiple edges. Draw a simple graph with at least two vertices such that no two vertices have the same degree, or explain why this is impossible.



Solution. It is impossible to draw such a graph. We can prove that it is impossible by the method of *proof by contradiction*, in which we assume that it is possible to do and show that this assumption leads to a logical contradiction.

Suppose that we have a simple graph $\mathscr{G}$ with n vertices (where $n \geq 2$) such that no two vertices of $\mathscr{G}$ have the same degree. The smallest possible degree for a vertex of $\mathscr{G}$ is 0, which can happen if the vertex does not have any edges leading to it. The greatest possible degree for a vertex of $\mathscr{G}$ is $n-1$, which can happen if it is joined by an edge to every other vertex in the graph. (Since $\mathscr{G}$ is a simple graph, two vertices cannot have more than one edge between them, and a vertex cannot have an edge joining it to itself, so a vertex cannot have a degree greater than $n-1$.) So the possible degrees for a vertex of $\mathscr{G}$ are 0, 1, 2, 3, …, $n-1$.

Therefore there are n different possible degrees for a vertex of $\mathscr{G}$. If every one of the n vertices of $\mathscr{G}$ is to have a different degree, then each of these possible degrees must be used exactly once. In particular, we must have a vertex A of degree 0 and another vertex B of degree $n-1$. (Also, A and B cannot be the same vertex, because $0 \neq n-1$; this is where we use the fact that $n \geq 2$.)

Now, is there an edge between the vertices A and B? Since the degree of A is 0, we know that A is not joined to any other vertex by an edge, so there cannot be an edge between A and B. On the other hand, since the degree of B is $n-1$, we know that B must be joined to every other vertex in $\mathscr{G}$, including A, so there must be an edge between A and B. This is a logical contradiction. Therefore the assumption that led us to this contradiction—namely, the assumption that it is possible to have a simple graph $\mathscr{G}$, having at least two vertices, such that no two vertices have the same degree—must be false. □

Problem 2. Draw an example of each of the following:

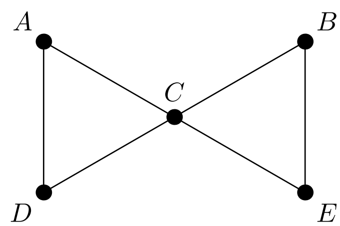
(a) A graph that has an Eulerian circuit but no Hamiltonian circuit.

(b) A graph that has a Hamiltonian circuit but no Eulerian circuit.

In both cases, justify that your graph satisfies the required conditions.

Solution.

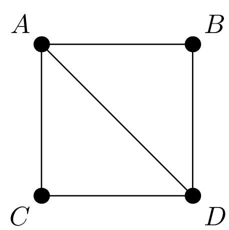
(a) The following graph, which is sometimes called the "bowtie" graph, has an Eulerian circuit but no Hamiltonian circuit.



An example of an Eulerian circuit in this graph is C–A–D–C–B–E–C; this circuit passes through each edge exactly once.

However, this graph does not have a Hamiltonian circuit. If it did, we could start at, say, the vertex A, travel along the Hamiltonian circuit to visit every other vertex exactly once, and finally return to A. (Note that we can choose to start at any vertex we like, since a Hamiltonian circuit passes through every vertex in the graph—we can just "hop on" the circuit at any vertex and follow it around until it brings us back to our starting point.) But we must pass through vertex C to get from vertex A to vertex B, and then we must pass through C again to get back to A, so we would need to visit vertex C twice, which is not allowed in a Hamiltonian circuit.

(b) The following graph has a Hamiltonian circuit but no Eulerian circuit.



An example of a Hamiltonian circuit in this graph is A–B–D–C–A; this circuit passes through every vertex exactly once and returns to its starting vertex.

We saw in class that a (connected) graph has an Eulerian circuit if and only if the degree of every vertex is even. The graph above has two vertices of odd degree (vertices A and D have degree 3), so it does not have an Eulerian circuit. (It *does* have an Eulerian path, for example A–B–D–A–C–D, but the Eulerian paths in this graph do not end where they begin. If we want to rule out Eulerian paths too, we can add another edge joining the vertices B and C.) □

Problem 3. Agnes hosts a party. Over the course of the evening some people at the party shake hands. Near the end of the party, but before any of the guests have left, Agnes gets everyone's attention and asks how many people shook hands an odd number of times. Exactly 11 guests say they have done so. Assuming that all the guests remember how many hands they shook, did Agnes shake an even number or an odd number of hands? Why?

Solution. Imagine drawing a graph $\mathscr{G}$ in which the vertices represent the people at the party (including one vertex A for Agnes) and in which two vertices are joined by an edge if the corresponding people shook hands. Then the degree of a vertex in this graph is the number of times the corresponding person shook hands.

We don't know much about the structure of this graph—we don't even know how many vertices or edges it has—but we do know that exactly 11 of the vertices representing guests at the party (i.e., exactly 11 of the vertices which are not A) have odd degrees.

Every graph must have an even number of odd vertices. (This is Theorem 6.5 in *Problem Solving Through Recreational Mathematics*, which is often called the "handshaking lemma," because of its application to puzzles like this one.) In particular, whatever the graph $\mathscr{G}$ looks like, it must have an even number of vertices of odd degree. Therefore the vertex A must also have an odd degree, in order to bring the total number of odd vertices up to 12, an even number.

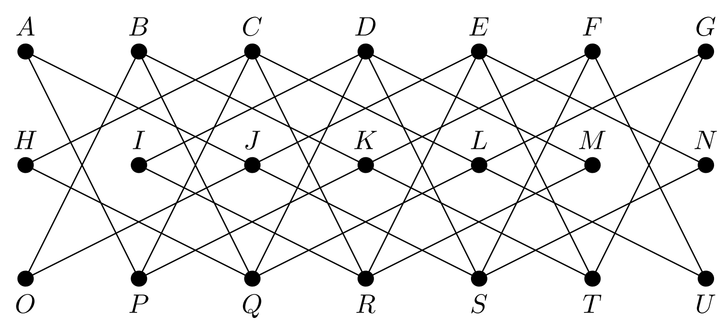
So Agnes must have shaken an odd number of hands. □

Problem 4. (Problem 6.21 from *Problem Solving Through Recreational Mathematics*.) Describe a knight's tour on a 3×7 chessboard.

Solution. We shall label the squares of the chessboard as they are labeled in the book:

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U

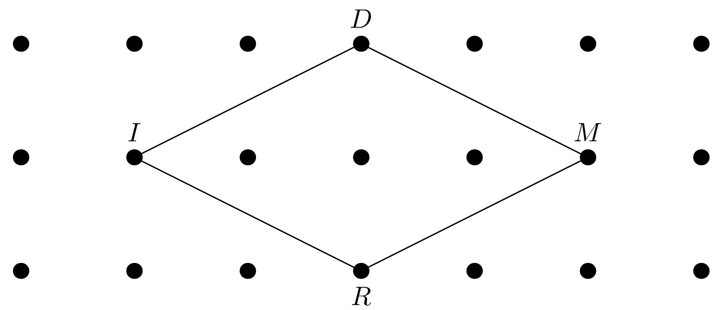
We can draw a graph in which the vertices represent the squares of the chessboard and in which two vertices are joined by an edge if and only if the knight can move directly from one square to the other. This graph, which we'll creatively call $\mathscr{G}$, is shown below.



A knight's tour of the chessboard corresponds to a Hamiltonian path in this graph, because we want to visit every square of the chessboard (that is, every vertex of the graph) exactly once.

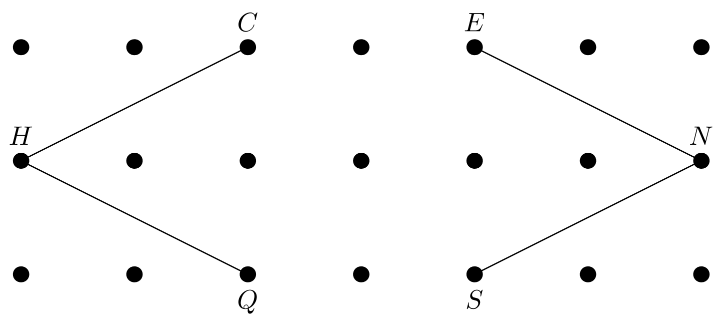
Since we are trying to find a Hamiltonian path in this graph, it is helpful to find vertices of degree 2. We see that the vertices A, G, H, I, M, N, O, and U all have degree 2. A vertex of degree 2 that appears in the middle of a Hamiltonian path must have both of its adjacent edges used in the path—one of the edges must be used to arrive at the vertex and the other adjacent edge must be used to leave. Identifying "forced" edges in this way will help us to find our Hamiltonian path.

To simplify the picture, let's draw only the edges adjacent to the two vertices I and M, which are of degree 2. These edges are shown below.



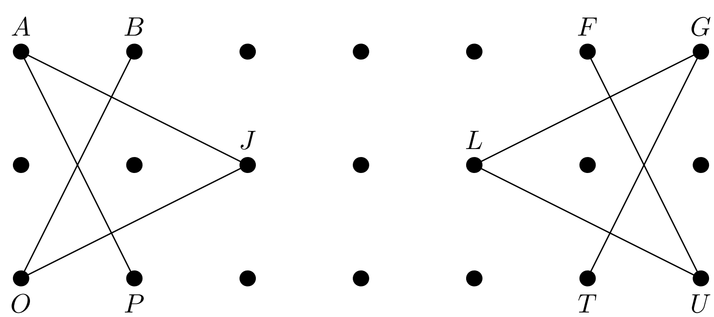
Note that these four edges form a diamond, a circuit in the graph $\mathscr{G}$. By our reasoning in the previous paragraph, if the vertices I and M appear in the middle of the Hamiltonian path, then we must use all four of these edges. But we cannot do this, because if we did we would come back to where we started before we had traveled to all the vertices. So at least one of the vertices I or M must be at the beginning or end of the Hamiltonian path.

Now let's draw only the edges adjacent to the two vertices H and N, which are also of degree 2. These edges are shown below.



Unless H or N is the other end of the Hamiltonian path, we will need to use all four of these edges. So we can think of the vertex H as simply a "bridge" between the vertices C and Q—when our Hamiltonian path first reaches one of the vertices C or Q, we will be forced to travel next to vertex H and then to the other of C or Q, for otherwise we will not be able to reach H at all. In other words, our Hamiltonian path will need to contain C–H–Q (or Q–H–C) as a subpath. Likewise, the vertex N will act as a "bridge" between the vertices E and S, which we will be required to take.

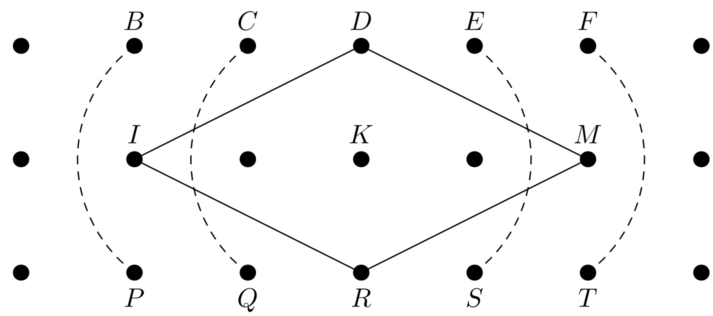
Finally, let's draw only the edges adjacent to the four "corner" vertices A, G, O, and U, which are the remaining vertices of degree 2. These edges are shown below.



These edges form more interesting "bridges." Assuming that the vertices A, G, O, and U are not the other end of our Hamiltonian path, we will need to travel along each of the edges in this picture.

So, for example, vertex A must be used as a "bridge" between vertices J and P, and vertex O must be a "bridge" between vertices B and J. But also vertex J must be a "bridge" between vertices A and O, even though vertex J actually has degree 4 in the graph $\mathscr{G}$, because we are forced to use the two edges A–J and J–O. (Since our Hamiltonian path can visit the vertex J only once, the two edges E–J and J–S cannot be used.) So our Hamiltonian path will need to travel along the edges B–O–J–A–P (or the reverse order, P–A–J–O–B). In other words, we can ignore the vertices A, J, and O except as a way to get from B to P (or P to B), because that is how they must be used. Similarly, from the pattern on the right side of the grid we see that our Hamiltonian path will be required to go F–U–L–G–T (or T–G–L–U–F).

At this point we have identified several structures in the graph G. These structures are summarized in the drawing below; dashed lines connect the endpoints of the necessary "bridges" we have found, and the vertices that must appear in the middle of these "bridges" have been left unlabeled. These "bridges," together with the diamond formed by the vertices D, I, M, and R, cover all the vertices of the graph $\mathscr{G}$ except the central vertex K.



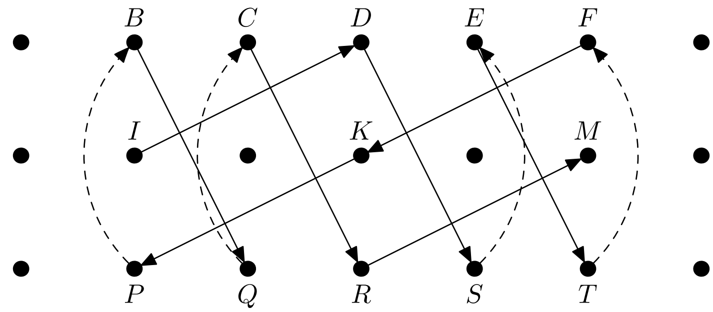
Our goal now is to find a way to connect all these pieces together into a single Hamiltonian path. We know that every Hamiltonian path must have at least one of the vertices I or M as its beginning or ending vertex. Due to the evident symmetry here, the choice is arbitrary; let's decide to start our Hamiltonian path at the vertex I. For purely aesthetic reasons, it would be beautiful if our path ended at the symmetrically opposite vertex M, so let's try to do that if possible. (This is not a forced requirement; it is possible to find a Hamiltonian path in $\mathscr{G}$ that begins at I and ends at some vertex other than M.) A little bit of trial and error gives us a solution, which we will describe next.

We seek a Hamiltonian path in $\mathscr{G}$ that begins at I and ends at M. Suppose we start by taking the edge I–D. Then the only way to get to the vertex M at the end is to take the edge R–M, so that will be our last step. This takes care of the diamond; let's figure out the middle of our Hamiltonian path.

From the vertex D we can go to the vertex S, since these squares are a knight's move apart. Now that we are at S, we are forced to travel along the "bridge" to vertex E. From E we can go to the vertex T, after which we must travel along the "bridge" to vertex F. These steps, which give us a path from the starting vertex I to the vertex F, cover all of the vertices to the right of the center column of the chessboard.

To cover the vertices to the left of the center, we can take the same journey, but mirrored and going in reverse from the vertex R. So we will get to R from the vertex C, which must appear at the end of the "bridge" from the vertex Q. We will get to Q from the vertex B, which comes at the end of the "bridge" from P. We worked backward, so these steps actually give us a path from the vertex P to the ending vertex M.

Now, can we connect these two halves? At this point the only unused vertex is the center vertex K, which is exactly one knight's move from both the vertex F and the vertex P, so the vertex K will serve as the link between the two halves of our Hamiltonian path. The Hamiltonian path we have found is summarized in the drawing below, again using dashed lines to represent the "bridges."



Finally we can list the vertices in order along the Hamiltonian path we found, remembering that each of the "bridges" in the picture above is really a sequence of several vertices between the two endpoints. So our Hamiltonian path in $\mathscr{G}$, which represents a knight's tour on the 3×7 chessboard, is

$$I\text{–}D\text{–}S\text{–}N\text{–}E\text{–}T\text{–}G\text{–}L\text{–}U\text{–}F\text{–}K\text{–}P\text{–}A\text{–}J\text{–}O\text{–}B\text{–}Q\text{–}H\text{–}C\text{–}R\text{–}M. \qquad \square$$

Problem 5. You decide that you want to visit the six largest cities in Pennsylvania (according to the 2007 population estimates from the U.S. Census Bureau). The shortest distances between these cities, according to Google Maps, are given in the distance table below.

Distances in miles	**A.**	**Erie**	**Phila.**	**Pgh.**	**R.**	**S.**
Allentown	—	371	55	279	39	75
Erie	371	—	401	128	358	297
Philadelphia	55	401	—	301	57	124
Pittsburgh	279	128	301	—	258	282
Reading	39	358	57	258	—	97
Scranton	75	297	124	282	97	—

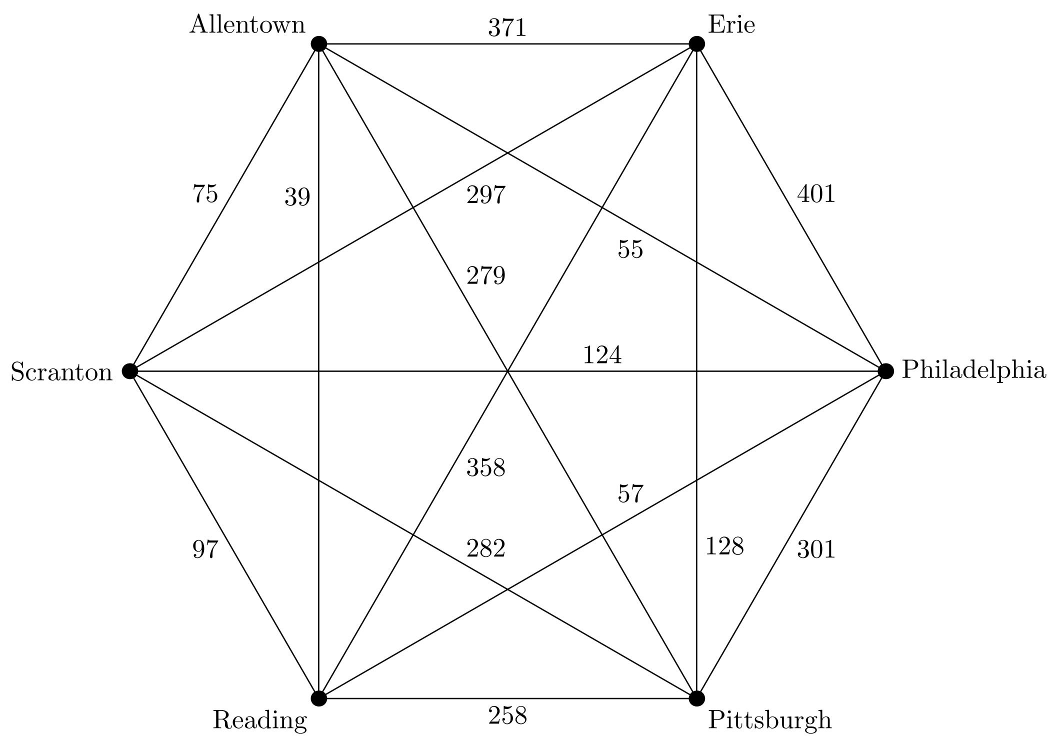
(a) Draw a weighted graph to represent this information. (Be sure to label the vertices of the graph.)

(b) You want to start in Pittsburgh, visit all of these cities, and return to your starting point. In mathematical terms, what are you trying to find? Why?

(c) You would like to keep your total travel distance low. Use your graph from part (a) to find an efficient way to tour these six cities. Explain the method you are using and show all of your steps. What is the total distance you travel if you go this way?

Solution.

(a) A weighted graph representing the distances between these six cities is given below. The vertices represent the cities, and the weight on each edge represents the distance between the corresponding cities.



(b) We want to find a low-cost Hamiltonian circuit in this weighted graph. We would like to visit every vertex in the graph (i.e., every city) exactly once and return to our starting point; such a path is a Hamiltonian circuit. Moreover, our aim is to have a small total travel distance, so we seek a Hamiltonian circuit that has a low total cost for the edges it uses.

(c) Let's use the nearest-neighbor algorithm to find a low-cost Hamiltonian circuit in the weighted graph from part (a). We'll start at Pittsburgh. From Pittsburgh the nearest neighbor is Erie, 128 miles away, so we'll go there next. Erie's nearest neighbor (other than Pittsburgh, which we've already visited) is Scranton, 297 miles away, so Scranton will be our next stop. From Scranton the nearest city is Allentown, 75 miles away, and then from Allentown we will go to Reading, which is a distance of just 39 miles. Now Philadelphia is the only city we have not yet visited, so we will travel the 57 miles from Reading to Philadelphia before finally returning to Pittsburgh, which is 301 miles from Philadelphia. The total distance we will have traveled if we use this route is 897 miles. □

[Note that the nearest-neighbor algorithm is only an *approximation* algorithm, so the route we have chosen may not actually be the shortest possible tour through these six cities. Using a different algorithm, such as the cheapest-link algorithm, or even just starting at a different city with the nearest-neighbor algorithm, may produce a different and possibly shorter tour.

If we start at Allentown and use the nearest-neighbor algorithm, the tour produced is Allentown–Reading–Philadelphia–Scranton–Pittsburgh–Erie–Allentown, for a total distance of 1,001 miles. Using Erie as our starting point for the nearest-neighbor algorithm, we get Erie–Pittsburgh–Reading–Allentown–Philadelphia–Scranton–Erie, a total distance of 901 miles. With Philadelphia as our starting point, the nearest-neighbor algorithm produces Philadelphia–Allentown–Reading–Scranton–Pittsburgh–Erie–Philadelphia, a total distance of 1,002 miles. If we start at Reading and use the nearest-neighbor algorithm we get Reading–Allentown–Philadelphia–Scranton–Pittsburgh–Erie–Reading, which is a total of 986 miles. Starting at Scranton and using the nearest-neighbor algorithm, we get Scranton–Allentown–Reading–Philadelphia–Pittsburgh–Erie–Scranton, for a total of 897 miles; this is essentially the same tour as we found by starting at Pittsburgh.

If we use the cheapest-link algorithm, we select the edges in the following order: Allentown–Reading, Allentown–Philadelphia, Reading–Scranton, Erie–Pittsburgh, Pittsburgh–Scranton, and Erie–Philadelphia. So we get the tour Pittsburgh–Erie–Philadelphia–Allentown–Reading–Scranton–Pittsburgh, for a total distance of 1,002 miles; this is essentially the same tour as we found with the nearest-neighbor algorithm when we started at Philadelphia.

But in fact none of these tours is actually the best we can do. The tour that goes Pittsburgh–Erie–Scranton–Allentown–Philadelphia–Reading–Pittsburgh is 870 miles long. This is the shortest possible tour through these six cities, but it is not produced by either the nearest-neighbor algorithm or the cheapest-link algorithm.]

Problem 6. (Problem 6.25 from *Problem Solving Through Recreational Mathematics.*) Three married couples want to cross a river. The only boat available is capable of holding two people at a time. This would present no difficulty were it not for the fact that the women are all very jealous, so that each woman refuses to allow her husband to be in the presence of another woman unless she herself is also present.

How should they cross the river with the least amount of rowing?

Solution. We can use a state diagram to help us solve this problem, as in the solution to Sample Problem 6.4 on pages 196–197 of *Problem Solving Through Recreational Mathematics.*

First we must decide what constitutes a "state" in this problem. A natural choice is to consider a state to be defined by the sets of people on either side of the river and the location of the boat. It turns out that we need only keep track of how many men and women are on either side of the river; we do not need to keep track of exactly which of the couples are where. (For example, it is clear that the state in which the Andersons and the Browns are on the near shore and the Clarks are on the far shore is equivalent to the state in which the Andersons and the Clarks are on the near shore and the Browns are on the far shore, so we will consider both of these states to be the same—two men and two women on the near shore and one man and one woman on the far shore.)

As a shorthand for describing states, we shall use a vertical bar ('|') to represent the river; everyone begins on the left side of the bar, with the boat, and our goal is to shuttle everyone to the right side of the bar. We shall use the letter 'f' to represent a female, the letter 'm' to represent a male, and the letter 'b' to represent the boat.

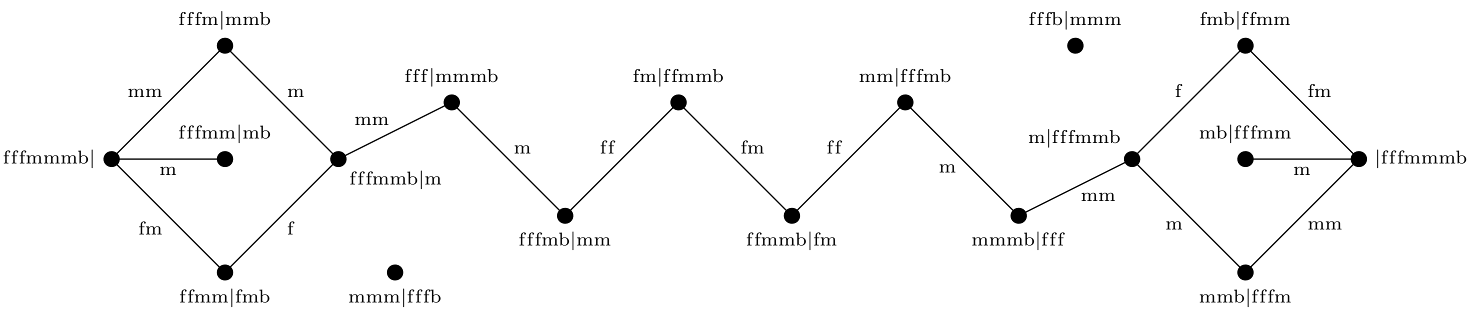
The conditions of the problem require that men should not outnumber women on either shore at any time (except when there are men but no women on one shore). Additionally, the boat cannot be on one shore of the river by itself. This means that there are only 18 allowable states:

fffmmmb|	fffmb|mm	fff|mmmb	fmb|ffmm	mmm|fffb	mb|fffmm
fffmmb|m	fffm|mmb	ffmmb|fm	fm|ffmmb	mmb|fffm	m|fffmmb
fffmm|mb	fffb|mmm	ffmm|fmb	mmmb|fff	mm|fffmb	|fffmmmb

The initial state is fffmmmb| (with all three couples and the boat on the left side of the bar), and our goal is to reach the state |fffmmmb (with all three couples and the boat on the right side of the bar).

We shall represent the possible states as the vertices of a graph. We shall join two vertices with an edge if it is possible to move from one of the states to the other in a single transition, where a single transition consists of one boat trip across the river. To help us remember how to get from one state to another, we shall also label the edges of the graph with one or two letters describing the people who row across the river. (So, for example, an edge labeled 'fm' indicates a transition in which a woman and a man row across—necessarily this will be a woman and her husband.)

The graph we obtain is shown below. (Note that the vertices representing the states mmm|fffb and fffb|mmm are isolated—they are not joined by edges to any other vertices. This shows that it is impossible to reach these states through a legal sequence of moves, even though these states themselves are allowable.)



Now that we have this state diagram, we need only identify a path from our initial state (fffmmmb|, the leftmost vertex above) to our goal (|fffmmmb, the rightmost vertex). Since we want to minimize the amount of rowing required, we should choose a shortest path, though this is not an issue here since all paths from fffmmmb| to |fffmmmb have the same length.

Therefore, one solution to the problem (there are several) is as follows:

	On the near shore	**On the far shore**
start	Mr. & Mrs. A, Mr. & Mrs. B, Mr. & Mrs. C, boat	
Mr. A & Mr. B row across	Mrs. A, Mrs. B, Mr. & Mrs. C	Mr. A, Mr. B, boat
Mr. A rows back	Mr. & Mrs. A, Mrs. B, Mr. & Mrs. C, boat	Mr. B
Mr. A & Mr. C row across	Mrs. A, Mrs. B, Mrs. C	Mr. A, Mr. B, Mr. C, boat
Mr. A rows back	Mr. & Mrs. A, Mrs. B, Mrs. C, boat	Mr. B, Mr. C
Mrs. B & Mrs. C row across	Mr. & Mrs. A	Mr. & Mrs. B, Mr. & Mrs. C, boat
Mr. & Mrs. B row back	Mr. & Mrs. A, Mr. & Mrs. B, boat	Mr. & Mrs. C
Mrs. A & Mrs. C row across	Mr. A, Mr. B	Mrs. A, Mrs. B, Mr. & Mrs. C, boat
Mr. C rows back	Mr. A, Mr. B, Mr. C, boat	Mrs. A, Mrs. B, Mrs. C
Mr. A & Mr. B row across	Mr. C	Mr. & Mrs. A, Mr. & Mrs. B, Mrs. C, boat
Mrs. C rows back	Mr. & Mrs. C, boat	Mr. & Mrs. A, Mr. & Mrs. B
Mr. & Mrs. C row across		Mr. & Mrs. A, Mr. & Mrs. B, Mr. & Mrs. C, boat

In particular, there can be no fewer than 11 river crossings. □

Problem 7. Eight radio stations, one in each of the eight cities in the distance table below, are to be assigned frequencies. Two stations cannot be assigned the same frequency if they are within 200 miles of each other; otherwise their signals will interfere.

Distances in miles	**Ashland**	**Bristol**	**Clinton**	**Douglas**	**Empire**	**Franklin**	**Greene**	**Huron**
Ashland	—	292	221	366	357	127	240	298
Bristol	292	—	93	195	366	329	247	419
Clinton	221	93	—	171	290	239	162	329
Douglas	366	195	171	—	223	330	171	330
Empire	357	366	290	223	—	252	136	142
Franklin	127	329	239	330	252	—	167	172
Greene	240	247	162	171	136	167	—	176
Huron	298	419	329	330	142	172	176	—

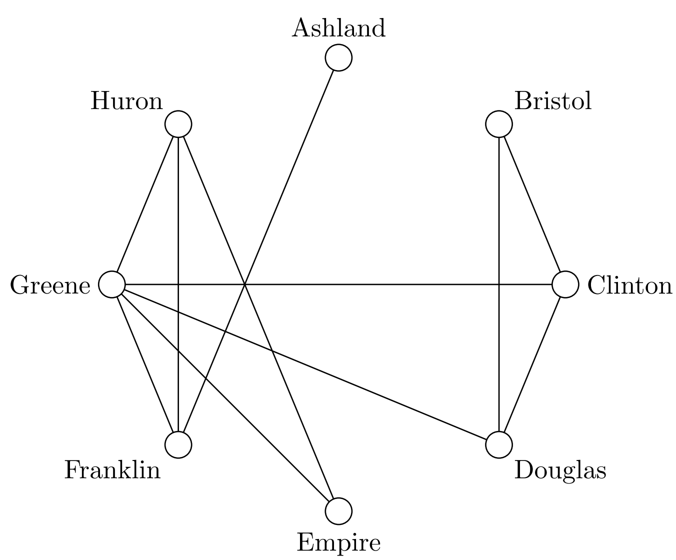
(a) Draw a conflict graph for this scenario. Explain what the vertices represent and what causes a conflict.

(b) Color the vertices of the conflict graph using as few colors as possible, according to the rules of graph coloring.

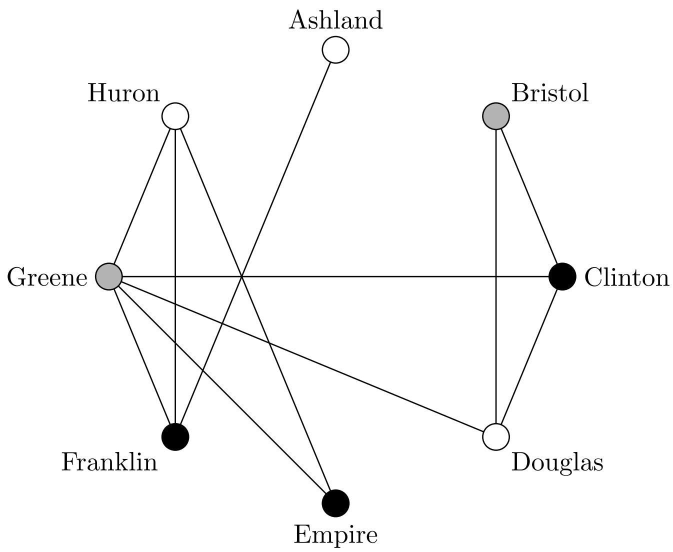
(c) Interpret your coloring from part (b). How many different frequencies are needed? How should they be assigned?

Solution.

(a) A conflict graph for this scenario is shown below. (The vertices have been drawn as circles rather than dots, because we will be coloring them later.) The vertices of the conflict graph represent the eight cities. There is a conflict between two cities, and hence an edge between the corresponding vertices, if they are within 200 miles of each other, because in this case the radio stations in these cities cannot be assigned the same frequency.



(b) We must color the vertices of the conflict graph in such a way that no two adjacent vertices (that is, no two vertices that are joined by an edge) receive the same color. The cities of Bristol, Clinton, and Douglas form a triangle in the graph—each of these three cities is adjacent to the other two—so we will certainly require at least three colors to color the vertices of the conflict graph (because we need three colors simply to color Bristol, Clinton, and Douglas). As it turns out, three colors are enough. Shown below is a valid coloring of the vertices of the conflict graph with three colors (black, white, and gray).



(c) Three different frequencies will be needed for these eight radio stations. One frequency can be assigned to the three stations in Ashland, Douglas, and Huron; the second can be assigned to the stations in Bristol and Greene; and the third can be assigned to the stations in Clinton, Empire, and Franklin. □

Problem 8. A map of South America requires four colors if no two countries which share a common border are to receive the same color; three colors are not enough. An easy way to see this is to consider Paraguay and its three neighbors, as shown in the map on the left below. Each of these four countries (Argentina, Bolivia, Brazil, and Paraguay) borders all of the other three. (In other words, the dual graph of this map is the complete graph on four vertices.) So four colors are necessary.

This situation does not arise in a map of the United States—there are no four states such that each of the four borders all of the other three. So, no matter which four states you choose, there will always be two of them that do not border each other. (Verify this for yourself.)

Does this mean that a map of the United States can be colored with just three colors? If so, show how it can be done by coloring the map of the 48 contiguous United States (shown on the right below, not to the same scale as the other map) using just three colors in such a way that no two bordering states receive the same color. If it cannot be done, explain, as carefully and as thoroughly as you can, why four colors are necessary.

(Note that Utah and New Mexico touch at only a single point, so they are not considered to be bordering states. The same is true for Colorado and Arizona.)

Solution. A map of the United States requires four colors if no two bordering states are to receive the same color; three colors are not enough.

One way to see this is to consider the state of Nevada and its five neighboring states (Oregon, Idaho, Utah, Arizona, and California). We can prove that three colors are not enough by assuming that these six states can be colored with just three colors and finding a logical contradiction.

So we assume that these states can be colored with just three colors. Nevada must be colored with some color; call it color 1. Since Nevada borders each of the other five states, color 1 cannot be used again. Oregon must be colored with some color, and it cannot be color 1; call it color 2. Now Idaho needs a color, and it can be neither color 1 nor color 2, because Idaho borders both Nevada and Oregon. So Idaho must be colored with color 3. (These could be any three colors. We are giving them numbers to emphasize that the particular colors used are not important.)

Next we must color Utah. We cannot color Utah with color 1, because Utah borders Nevada, and we cannot color it with color 3, because Utah borders Idaho. Since we are assuming that these states can be colored with just three colors, we cannot introduce a fourth color, so we are forced to use color 2 for Utah. Similarly, Arizona cannot be colored with colors 1 or 2, because it borders both Nevada and Utah, so Arizona must have color 3.

But now California poses a problem. We cannot color California with color 1, because it borders Nevada; we cannot color it with color 2, because it borders Oregon; and we cannot color it with color 3, because it borders Arizona. So we have no color for California, which is a contradiction of our assumption that we can color these six states with just three colors. (Note that we did not make any *choices* when we colored the other states, so we cannot go back and change things to make California work.)

Since we ran into a contradiction when we assumed these six states could be colored with three colors, our assumption must have been wrong. Therefore we conclude that at least four colors are necessary to color a map of the United States. (Of course, we know from the four-color theorem that four colors will be enough, so four is exactly the number of colors that are needed.) □

[Three colors are also not enough to color Kentucky and its neighbors, or West Virginia and its neighbors—do you see why?

This problem illustrates a very interesting and subtle property of graph coloring. Somehow the number of colors needed to color a particular graph is a *global* property of the graph, not a *local* property. In other words, it may be possible to color any *portion* of the graph with, say, three colors, yet the graph as a whole requires more. In this case, any group of four states, considered in isolation, can be colored with just three colors, but the 48 states taken as a whole require four colors. This is part of what makes the analysis of graph coloring so challenging—the whole is more than the sum of its parts.]