7
Graphs

7.1 Even and Odd Degrees

We start with the following exercise (admittedly of no practical significance).

> *Prove that at a party with 51 people, there is always a person who knows an even number of others.*

(We assume that acquaintance is mutual. There may be people who don't know each other. There may even be people who don't know anybody else. Of course, such people know an even number of others, so the assertion is true if there is such a person.)

If you don't have any idea how to begin a solution, you should try to experiment. But how to experiment with such a problem? Should we find 51 names for the participants, then create, for each person, a list of those people he or she knows? This would be very tedious, and we would be lost among the data. It would be good to experiment with smaller numbers. But which number can we take instead of 51? It is easy to see that 50, for example, would not do: If, say, we have 50 people who all know each other, then everybody knows 49 others, so there is no person with an even number of acquaintances. For the same reason, we could not replace 51 by 48, or 30, or any even number. Let's hope that this is all; let's try to prove that

> *at a party with an odd number of people, there is always a person who knows an even number of others.*

L. Lovász et al., *Discrete Mathematics*
© Springer Science+Business Media, LLC 2003

Now we can at least experiment with smaller numbers. Let us have, say, 5 people: Alice, Bob, Carl, Diane, and Eve. When they first met, Alice knew everybody else; Bob and Carl knew each other, and Carl also knew Eve. So the numbers of acquaintances are: Alice 4, Bob 2, Carl 3, Diane 1, and Eve 2. We have not only one but three people with an even number of acquaintances.

It is still rather tedious to consider examples by listing people and listing pairs knowing each other, and it is quite easy to make mistakes. We can, however, find a graphic illustration that helps a lot. We represent each person by a point in the plane (well, by a small circle, to make the picture nicer), and we connect two of these points by a segment if the people know each other. This simple drawing contains all the information we need (Figure 7.1).

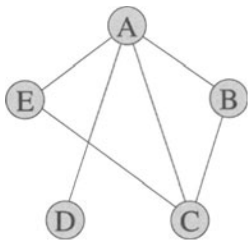

FIGURE 7.1. The graph depicting acquaintance between our friends

A picture of this kind is called a *graph*. More exactly, a graph consists of a set of *nodes* (also known as *points*, or *vertices*) with some pairs of these (not necessarily all pairs) connected by *edges*. It does not matter whether these edges are straight or curvy; all that is important is which pairs of nodes they connect. The set of nodes of a graph G is usually denoted by V; the set of edges, by E. Thus we write $G = (V, E)$ to indicate that the graph G has node set V and edge set E.

The only thing that matters about an edge is the pair of nodes it connects; hence the edges can be considered as 2-element subsets of V. This means that the edge connecting nodes u and v is just the set $\{u, v\}$. We'll further simplify notation and denote this edge by uv.

Can two edges connect the same pair of nodes (parallel edges)? Can an edge connect a node to itself (loop)? The answer to these questions is, of course, our decision. In some applications it is advantageous to allow such edges; in others, they must be excluded. In this book, we generally assume that a pair of nodes is connected by at most one edge, and no node is connected to itself. Such graphs are often called *simple graphs*. If parallel edges are allowed, the graph is often called a *multigraph* to emphasize this fact.

If two nodes are connected by an edge, then they are called *adjacent*. Nodes adjacent to a given node v are called its *neighbors*.

Coming back to our problem, we see that we can represent the party by a graph very conveniently. Our concern is the number of people known by a given person. We can read this off the graph by counting the number of edges leaving a given node. This number is called the *degree* of the node. The degree of node v is denoted by $d(v)$. So A has degree 4, B has degree 2, etc. If Frank now arrives, and he does not know anybody, then we add a new node that is not connected to any other node. So this new node has degree 0.

In the language of graph theory, we want to prove the following:

> *If a graph has an odd number of nodes, then it has a node with even degree.*

Since it is much easier to experiment with graphs than with tables of acquaintances, we can draw many graphs with an odd number of nodes, and count the number of nodes with even degree (Figure 7.2). We find that they contain $5, 1, 1, 7, 3, 3$ such nodes (the last one is a single graph on 7 nodes, not two graphs). So we observe that not only is there always such a node, but the number of such nodes is odd.

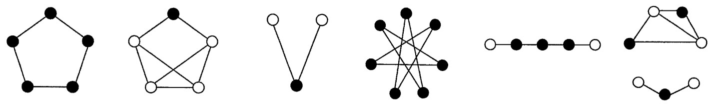

FIGURE 7.2. Some graphs with an odd number of nodes. Black circles mark nodes of even degree.

Now, this is a case in which it is easier to prove more: If we formulate the following stronger statement,

> *If a graph has an odd number of nodes, then the number of nodes with even degree is odd,*

then we made an important step towards the solution! (Why is this statement stronger? Because 0 is not an odd number.) Let's try to find an even stronger statement by looking also at graphs with an even number of nodes. Experimenting on several small graphs again (Figure 7.3), we find that the number of nodes with even degree is $2, 4, 0, 6, 2, 4$. So we conjecture the following:

> *if a graph has an even number of nodes, then the number of nodes with even degree is even.*

This is nicely parallel to the statement about graphs with an odd number of nodes, but it would be better to have a single common statement for the odd and even case. We get such a version if we look at the number of nodes with *odd*, rather than *even*, degree. This number is obtained by subtracting

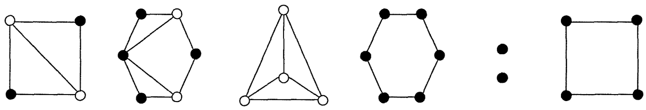

FIGURE 7.3. Some graphs with an even number of nodes. Black circles mark nodes of even degree.

the number of nodes with even degree from the total number of nodes, and hence both statements will be implied by the following:

Theorem 7.1.1 *In every graph, the number of nodes with odd degree is even.*

So what we have to prove is this theorem. It seems that having made the statement stronger and more general in several steps, we have made our task harder and harder. But in fact, we have gotten closer to the solution.

Proof. One way of proving the theorem is to build up the graph one edge at a time, and observe how the parities of the degrees change. An example is shown in Figure 7.4. We start with a graph with no edge, in which every degree is 0, and so the number of nodes with odd degree is 0, which is an even number.

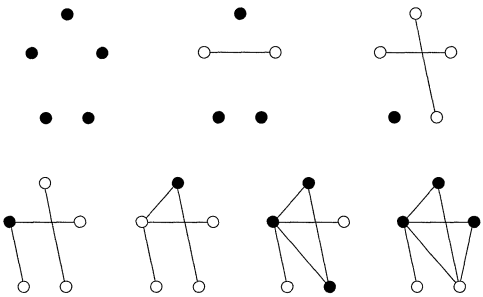

FIGURE 7.4. Building up a graph one edge at a time. Black circles mark nodes of even degree.

Now if we connect two nodes by a new edge, we change the parity of the degrees at these nodes. In particular,

— if both endpoints of the new edge had even degree, we increase the number of nodes with odd degree by 2;

— if both endpoints of the new edge had odd degree, we decrease the number of nodes with odd degree by 2;

— if one endpoint of the new edge had even degree and the other had odd degree, then we don't change the number of nodes with odd degree.

Thus if the number of nodes with odd degree was even before adding the new edge, it remained even after this step. This proves the theorem. (Note that this is a proof by induction on the number of edges.) □

Graphs are very handy in representing a large variety of situations, not only parties. It is quite natural to consider the graph whose nodes are towns and whose edges are highways (or railroads, or telephone lines) between these towns. We can use a graph to describe an electrical network, say the printed circuit on a card in your computer.

In fact, graphs can be used in any situation where a "relation" between certain objects is defined. Graphs are used to describe bonds between atoms in a molecule, connections between cells in the brain, descent between species, etc. Sometimes the nodes represent more abstract things: For example, they may represent stages of a large construction project, and an edge between two stages means that one arises from the other in a single phase of work. Or the nodes can represent all possible positions in a game (say, chess, although you don't really want to draw this graph), where we connect two nodes by an edge if one can be obtained from the other in a single move.

7.1.1 Find all graphs with $2, 3$, and 4 nodes.

7.1.2 (a) Is there a graph on 6 nodes with degrees $2, 3, 3, 3, 3, 3$?

(b) Is there a graph on 6 nodes with degrees $0, 1, 2, 3, 4, 5$?

(c) How many graphs are there on 4 nodes with degrees $1, 1, 2, 2$?

(d) How many graphs are there on 10 nodes with degrees $1, 1, 1, 1, 1, 1, 1, 1, 1, 1$?

7.1.3 At the end of the party with n people, everybody knows everybody else. Draw the graph representing this situation. How many edges does it have?

7.1.4 (a) Draw a graph with nodes representing the numbers $1, 2, \dots, 10$, in which two nodes are connected by an edge if and only if one is a divisor of the other.

(b) Draw a graph with nodes representing the numbers $1, 2, \dots, 10$, in which two nodes are connected by an edge if and only if they have no common divisor larger than 1.[1]

(c) Find the number of edges and the degrees in these graphs, and check that Theorem 7.1.1 holds.

7.1.5 What is the largest number of edges a graph with 10 nodes can have?

[1] This is an example where *loops* could play a role: Since $\gcd(1, 1) = 1$ but $\gcd(k, k) > 1$ for $k > 1$, we could connect 1 to itself by a loop, if we allowed loops at all.

7.1.6 How many graphs are there on 20 nodes? (To make this question precise, we have to make sure we know what it means that two graphs are the same. For the purpose of this exercise, we consider the nodes given, and labeled, say, as Alice, Bob, ... The graph consisting of a single edge connecting Alice and Bob is different from the graph consisting of a single edge connecting Eve and Frank.)

7.1.7 Formulate the following assertion as a theorem about graphs, and prove it: At every party one can find two people who know the same number of other people (like Bob and Eve in our first example).

It will be instructive to give another proof of the theorem formulated in the last section. This will hinge on the answer to the following question: How many edges does a graph have? This can be answered easily if we think back to the problem of counting handshakes: For each node, we count the edges that leave that node (this is the degree of the node). If we sum these numbers, we count every edge twice. So dividing the sum by two, we get the number of edges. Let us formulate this observation as a theorem:

Theorem 7.1.2 *The sum of degrees of all nodes in a graph is twice the number of edges.*

In particular, we see that the sum of degrees in any graph is an even number. If we omit the even terms from this sum, we still get an even number. So the sum of odd degrees is even. But this is possible only if the number of odd degrees is even (since the sum of an odd number of odd numbers is odd). Thus we have obtained a new proof of Theorem 7.1.1.

7.2 Paths, Cycles, and Connectivity

Let us get acquainted with some special kinds of graphs. The simplest graphs are the *edgeless graphs*, having any number of nodes but no edges.

We get another very simple kind of graphs if we take n nodes and connect any two of them by an edge. Such a graph is called a *complete graph* (or a *clique*). A complete graph with n nodes is denoted by K_n. It has $\binom{n}{2}$ edges (recall Exercise 7.1.3).

If we think of a graph as representing some kind of relation, then it is clear that we could just as well represent the relation by connecting two nodes if they are not related. So for every graph G, we can construct another graph $\overline{G}$ that has the same node set but in which two nodes are connected precisely if they are *not* connected in the original graph G. The graph $\overline{G}$ is called the *complement* of G.

If we take n nodes and connect one of them to all the others, we get a *star*. This star has $n-1$ edges.

Let us draw n nodes in a row and connect the consecutive ones by an edge. This way we obtain a graph with $n-1$ edges, which is called a *path*.

The first and last nodes in the row are called the *endpoints* of the path. If we also connect the last node to the first, we obtain a *cycle* (or *circuit*). The number of edges in a path or cycle is called its *length.* A cycle of length k is often called a k-cycle. Of course, we can draw the same graph in many other ways, placing the nodes elsewhere, and we may get edges that intersect (Figure 7.5).

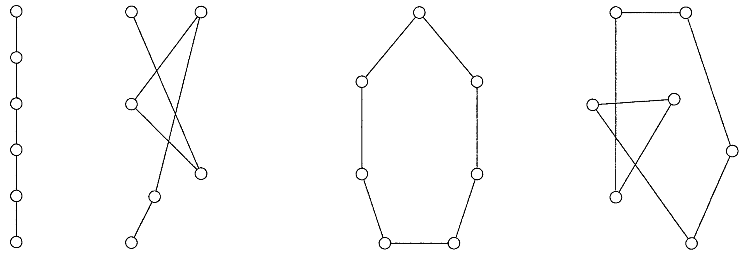

FIGURE 7.5. Two paths and two cycles

A graph H is called a *subgraph* of a graph G if it can be obtained from G by deleting some of its edges and nodes (of course, if we delete a node we automatically delete all the edges that connect it to other nodes).

7.2.1 Find all complete graphs, paths, and cycles among the graphs in Figures 7.1–7.5.

7.2.2 How many subgraphs does an edgeless graph on n nodes have? How many subgraphs does a triangle have?

7.2.3 Find all graphs that are paths or cycles and whose complements are also paths or cycles.

A key notion in graph theory is that of a *connected* graph. It is intuitively clear what this should mean, but it is also easy to formulate the property as follows: A graph G is connected if every two nodes of the graph are connected by a path in G. To be more precise: A graph G is connected if for every two nodes u and v, there exists a path with endpoints u and v that is a subgraph of G (Figure 7.6).

It will be useful to include a little discussion of this notion. Suppose that nodes a and b are connected by a path P in our graph. Also suppose that nodes b and c are connected by a path Q. Can a and c be connected by a path? The answer seems to be obviously "yes," since we can just go from a to b and then from b to c. But there is a difficulty: Concatenating (joining together) the two paths may not yield a path from a to c, since P and Q may intersect each other (Figure 7.7). But we can construct a path from a to c easily: Let us follow the path P to its first common node d with Q; then let us follow Q to c. Then the nodes we traversed are all distinct. Indeed,

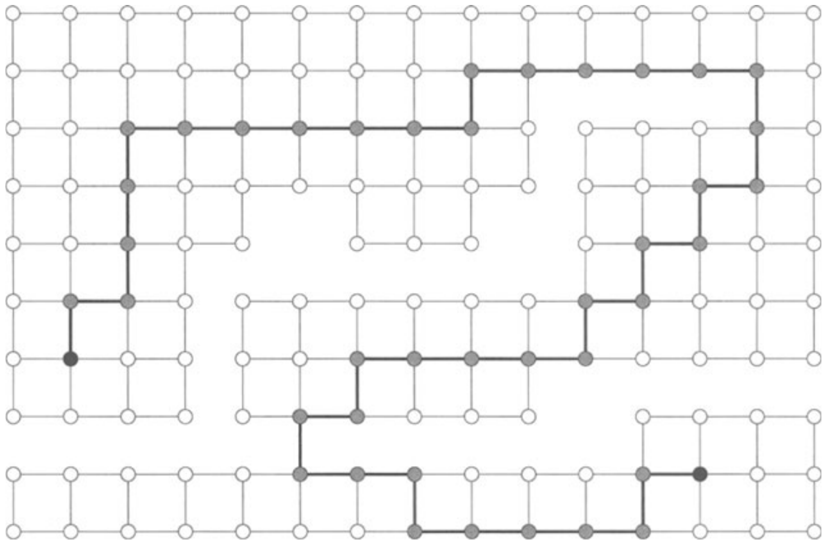

FIGURE 7.6. A path in a graph connecting two nodes

the nodes on the first part of our walk are distinct because they are nodes of the path P; similarly, the nodes on the second part are distinct because they are nodes of the path Q; finally, any node of the first part must be distinct from any node of the second part (except, of course, the node d), because d is the *first* common node of the two paths and so the nodes of P that we passed through before d are not nodes of Q at all. Hence the nodes and edges we have traversed form a path from a to c as claimed.[2]

A *walk* in a graph G is a sequence of nodes $v_0, v_1, \dots, v_k$ such that v_0 is adjacent to v_1, which is adjacent to v_2, which is adjacent to v_3, etc.; any two consecutive nodes in the sequence must be connected by an edge. This sounds almost like a path: The difference is that a walk may pass through the same node several times, while a path must go through different nodes. Informally, a walk is a "path with repetition"; more correctly, a path is a walk without repetition. Even the first and last nodes of the walk may be the same; in this case, we call it a *closed walk*. The shortest possible walk consists of a single node v_0 (this is closed). If the first node v_0 is different from the last node v_k, then we say that this walk *connects* nodes v_0 and v_k.

[2]We have given more details of this proof than was perhaps necessary. One should note, however, that when arguing about paths and cycles in graphs, it is easy to draw pictures (on the paper or mentally) that make implicit assumptions and are therefore misleading. For example, when joining together two paths, one's first mental image is a single (longer) path, which may not be the case.

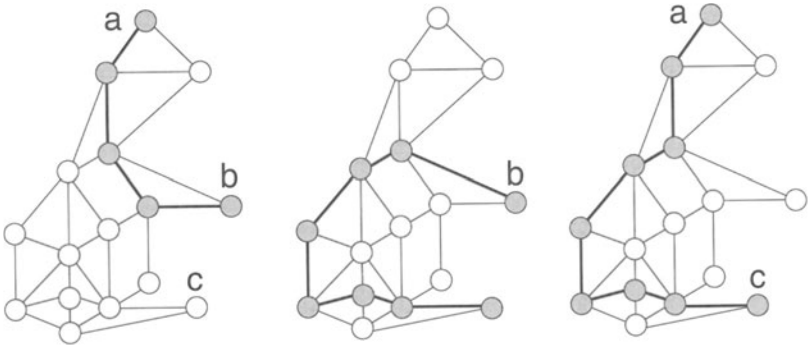

FIGURE 7.7. Selecting a path from a to c, given a path from a to b and a path from b to c.

Is there a difference between connecting two nodes by a walk and connecting them by a path? Not really: If two nodes can be connected by a walk, then they can also be connected by a path. Sometimes it is more convenient to use paths, sometimes, to use walks (see Exercise 7.2.6).

Let G be a graph that is not necessarily connected. G will have connected subgraphs; for example, the subgraph consisting of a single node (and no edge) is connected. A *connected component* H is a maximal subgraph that is connected; in other words, H is a connected component if it is connected but every other subgraph of G that contains H is disconnected. It is clear that every node of G belongs to some connected component. It follows by Exercise 7.2.7 that different connected components of G have no node in common (otherwise, their union would be a connected subgraph containing both of them). In other words, every node of G is contained in a unique connected component.

7.2.4 Is the proof as given above valid if (a) the node a lies on the path Q; (b) the paths P and Q have no node in common except b?

7.2.5 (a) We delete an edge e from a connected graph G. Show by an example that the remaining graph may not be connected.

(b) Prove that if we assume that the deleted edge e belongs to a cycle that is a subgraph of G, then the remaining graph is connected.

7.2.6 Let G be a graph and let u and v be two nodes of G.

(a) Prove that if there is a walk in G from u to v, then G contains a path connecting u and v.

(b) Use part (a) to give another proof of the fact that if G contains a path connecting a and b, and also a path connecting b and c, then it contains a path connecting a and c.

7.2.7 Let G be a graph, and let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two subgraphs of G that are connected. Assume that H_1 and H_2 have at least one node in common. Form their union, i.e., the subgraph $H = (V', E')$, where $V' = V_1 \cup V_2$ and $E' = E_1 \cup E_2$. Prove that H is connected.

7.2.8 Determine the connected components of the graphs constructed in Exercise 7.1.4.

7.2.9 Prove that no edge of G can connect nodes in different connected components.

7.2.10 Prove that a node v is a node of the connected component of G containing node u if and only if g contains a path connecting u to v.

7.2.11 Prove that a graph with n nodes and more than $\binom{n-1}{2}$ edges is always connected.

7.3 Eulerian Walks and Hamiltonian Cycles

Perhaps the oldest result in graph theory was discovered by Leonhard Euler, the greatest mathematician of the eighteenth century.

FIGURE 7.8. Leonhard Euler 1707–1783

It started with a recreational challenge that the citizens of Königsberg (today, Kaliningrad) raised. The city was divided into four districts by branches of the river Pregel (Figure 7.9), which were connected by seven bridges. It was nice to walk around, crossing these bridges, and so the question arose, is it possible to take a walk so that one crosses every bridge exactly once?

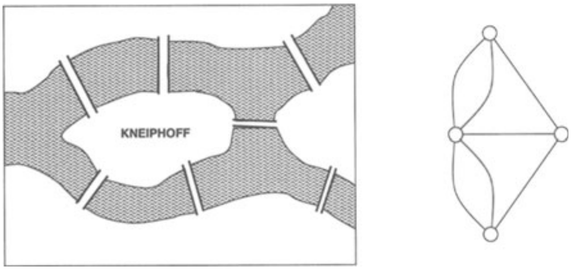

FIGURE 7.9. The bridges of Königsberg in Euler's time, and the graph modeling them.

Euler published a paper in 1736 in which he proved that such a walk was impossible. The argument is quite simple. Suppose that there is such a walk. Consider any of the four parts of the town, say the island Kneiphoff, and suppose that our walk does not start here. Then at some point in time, we enter the island by crossing a bridge; somewhat later, we leave it through another bridge (by the rules of the walk). Then we enter it again through a third bridge, then leave it through a fourth, then enter it through the fifth, then.... We cannot leave the island (at least not as part of the walk), since we have used up all the bridges that lead to it. We must end our walk on the island.

So we must either start or end our walk on the island. This is OK—the rules don't forbid it. The trouble is that we can draw the same conclusion for any of the other three districts of the town. The only difference is that instead of five bridges, these districts are connected to the rest of the town by only three bridges; so if we don't start there, we get stuck there at the second visit, not the third.

But now we are in trouble: we cannot start or end the walk in each of the four districts! This proves that no walk can cross every bridge exactly once.

Euler remarked that one could reach this conclusion by making an exhaustive list of all possible routes, and checking that none of them can be completed as required; but this would be impractical due to the large number of possibilities. More significantly, he formulated a general criterion by which one could decide for every city (no matter how many islands and bridges it had) whether one could take a walk crossing every bridge exactly once.

Euler's result is generally regarded as the first theorem of graph theory. Of course, Euler did not have the terminology of graphs (which was not to appear for more than a century), but we can use it to state Euler's theorem.

Let G be a graph; for the following discussion, we allow parallel edges, i.e., several edges connecting the same pair of nodes. A *walk* in such a graph is a bit more difficult to define. It consists of a sequence of nodes again such that any two consecutive nodes are connected by an edge; but if there are several edges connecting these consecutive nodes, we also have to specify which of these edges is used to move from one node to the next. So formally, a walk in a graph with parallel edges is a sequence $v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k$, where $v_0, v_1, \dots, v_k$ are nodes, $e_1, e_2, \dots, e_k$ are edges, and edge e_i connects nodes v_{i-1} and v_i $(i = 1, 2, \dots, k)$.

An *Eulerian walk* is a walk that goes through every edge exactly once (the walk may or may not be closed; see Figure 7.10). To see how to cast the problem of the Königsberg bridges into this language, let us represent each district by a node and draw an edge connecting two nodes for every bridge connecting the two corresponding districts. We get the little graph on the right hand side of Figure 7.9. A walk in the town corresponds to a walk in this graph (at least, if only crossing the bridges matters), and a walk that crosses every bridge exactly once corresponds to an Eulerian walk.

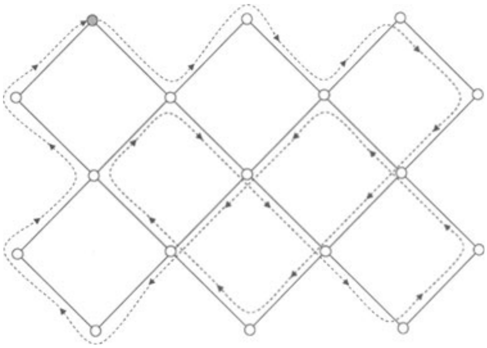

FIGURE 7.10. An Eulerian walk in a graph.

Recast in this language, Euler's criteria are stated in the following theorem.

Theorem 7.3.1 (a) *If a connected graph has more than two nodes with odd degree, then it has no Eulerian walk.*

(b) *If a connected graph has exactly two nodes with odd degree, then it has an Eulerian walk. Every Eulerian walk must start at one of these and end at the other one.*

(c) *If a connected graph has no nodes with odd degree, then it has an Eulerian walk. Every Eulerian walk is closed.*

Proof. Euler's argument above gives the following: *If a node v has odd degree, then every Eulerian walk must either start or end at v.* Similarly, we can see that *if a node v has even degree, then every Eulerian walk either starts and ends at v, or starts and ends somewhere else.* This observation immediately implies (a), as well as the second assertions in (b) and (c).

To finish the proof, we have to show that if a connected graph has 0 or 2 nodes with odd degree, then it has an Eulerian walk. We describe the proof in the case where there is no node of odd degree (part (c)); the other case is left to the reader as Exercise 7.3.14.

Let v be any node. Consider a closed walk starting and ending at v that uses every edge at most once. Such a walk exists. For example, we can take the walk consisting of the node v only. But we don't want this very short walk; instead, we consider a longest closed walk W starting at v, using every edge at most once.

We want to show that this walk W is Eulerian. Suppose not. Then there is at least one edge e that is not used by W. We claim that we can choose this edge so that W passes through at least one of its endpoints. Indeed, if

p and q are the endpoints of e and W does not pass through them, then we take a path from p to v (such a path exists since the graph is connected), and look at the first node r on this path that is also on the walk W (Figure 7.11(a)). Let $e' = sr$ be the edge of the path just before r. Then W does

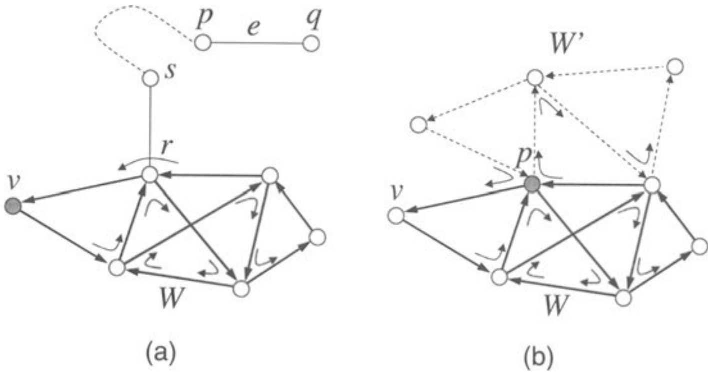

FIGURE 7.11. (a) Finding an edge not in W but meeting W. (b) Combining W and W'.

not pass through e (because it does not pass through s), so we can replace e by e', which has one endpoint on W.

So let e be an edge that is not used by W but has an endpoint p that is used by W. Then we start a new walk W' at p. We start through e, and continue walking as we please, only taking care that (i) we don't use the edges of W, and (ii) we don't use any edge twice.

Sooner of later we get stuck, but where? Let u be the node where we get stuck, and suppose that $u \neq p$. Node u has even degree; W uses up an even number of edges incident with u; every previous visit of the new walk to this node used up two edges (in and out); our last entrance used up one edge; so we have an odd number of edges that are edges neither of W nor of W'. But this means that we can continue our walk!

So the only node we can get stuck in is node p. This means that W' is a closed walk. Now we take a walk as follows. Starting at v, we follow W to p; then follow W' all the way through, so that eventually we get back to p; then follow W to its end at v (Figure 7.11(b)). This new walk starts and ends at v, uses every edge at most once, and is longer than W, which is a contradiction. □

Euler's result above is often formulated as follows: *A connected graph has a closed Eulerian walk if and only if every node has even degree.*

7.3.1 Which of the graphs in Figure 7.12 have an Eulerian walk? Which of them have a closed Eulerian walk? Find an Eulerian walk if it exists.

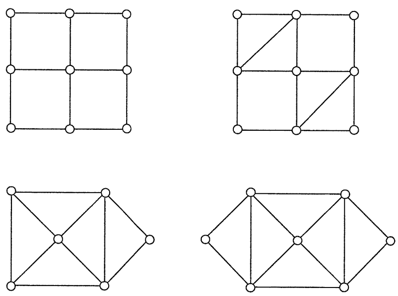

FIGURE 7.12. Which of these graphs has an Eulerian walk?

7.3.2 When does a connected graph contain two walks such that every edge is used by exactly one of them, exactly once?

A question similar to the problem of the Bridges of Königsberg was raised by another famous mathematician, the Irish William R. Hamilton, in 1856. A *Hamiltonian cycle* is a cycle that contains all nodes of a graph. The Hamilton cycle problem is the problem of deciding whether or not a given graph has a Hamiltonian cycle.

Hamiltonian cycles sound quite similar to Eulerian walks: Instead of requiring that every edge be used exactly once, we require that every node be used exactly once. But much less is known about them than about Eulerian walks. Euler told us how to decide whether a given graph has an Eulerian walk; but no efficient way is known to check whether a given graph has a Hamiltonian cycle, and no useful necessary and sufficient condition for the existence of a Hamiltonian cycle is known. If you solve Exercise 7.3.3, you'll get a feeling about the difficulty of the Hamiltonian cycle problem.

7.3.3 Decide whether the graphs in Figure 7.13 have a Hamiltonian cycle.

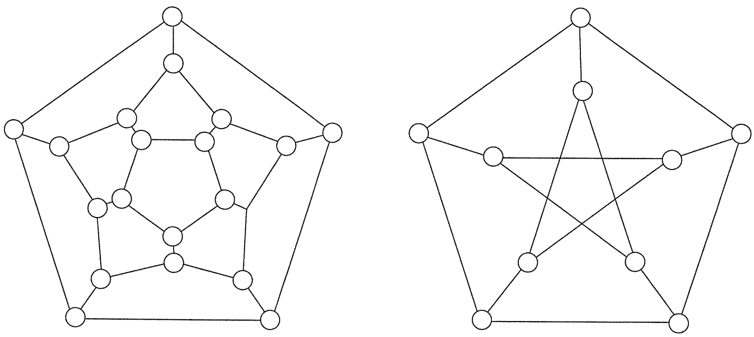

FIGURE 7.13. Two famous graphs: the dodecahedron graph (cf. Chapter 12) and the Petersen graph.

Review Exercises

7.3.4 Draw all graphs on 5 nodes in which every node has degree at most 2.

7.3.5 Does there exists a graph with the following degrees: (a) $0, 2, 2, 2, 4, 4, 6$; (b) $2, 2, 3, 3, 4, 4, 5$.

7.3.6 Draw the graphs representing the bonds between atoms in (a) a water molecule; (b) a methane molecule; (c) two water molecules.

7.3.7 At a party there were 7 boys and 6 girls. Every boy danced with every girl. Draw the graph representing the dancing. How many edges does it have? What are its degrees?

7.3.8 How many subgraphs does a 4-cycle have?

7.3.9 Prove that at least one of G and $\overline{G}$ is connected.

7.3.10 Let G be a connected graph with at least two nodes. Prove that it has a node such that if this node is removed (along with all edges incident with it), the remaining graph is connected.

7.3.11 Let G be a connected graph that is not a path. Prove that it has at least three vertices such that if any of them is removed, the remaining graph is still connected.

7.3.12 Let G be a connected graph in which every pair of edges have an endpoint in common. Show that G is either a star or a K_3.

7.3.13 There are $(m-1)n+1$ people in a room. Show that either there are m people who mutually do not know each other, or there is a person who knows at least n others.

7.3.14 Prove part (b) of Theorem 7.3.1.

7.3.15 Theorem 7.3.1 talks about connected graphs. Which disconnected graphs have an Eulerian walk?