- G is a tree.
- G is connected and acyclic.
- G is connected and every edge of G is a cut edge.
- Between any two vertices of G there is a unique path.
- G is connected and $|E(G)| = |V(G)| - 1$.

We also introduced the concept of spanning tree and proved that a graph has a spanning tree if and only if it is connected.

50 Exercises

50.1. Let G be a graph in which every vertex has degree 2. Is G necessarily a cycle?

50.2. Let T be a tree. Prove that the average degree of a vertex in T is less than 2.

50.3. There are exactly three trees with vertex set $\{1, 2, 3\}$. Note that all these trees are paths; the only difference is which vertex has degree 2.

How many trees have vertex set $\{1, 2, 3, 4\}$?

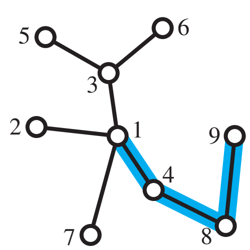
50.4. Let $d_1, d_2, \ldots, d_n$ be $n \geq 2$ positive integers (not necessarily distinct). Prove that $d_1, \ldots, d_n$ are the degrees of the vertices of a tree on n vertices if and only if $\sum_{i=1}^{n} d_i = 2n - 2$.

50.5. Let e be an edge of a graph G. Prove that e is not a cut edge if and only if e is in a cycle of G.

50.6. Complete the proof of Theorem 50.4. That is, prove that if G is a graph in which any two vertices are joined by a unique path, then G must be a tree.

50.7. Prove the following converse to Proposition 50.8:

Let T be a tree with at least two vertices and let $v \in V(T)$. If $T - v$ is a tree, then v is a leaf.



50.8. Let T be a tree whose vertices are the integers 1 through n. We call T a *recursive tree* if it has the following special property. Let P be any path in T starting at vertex 1. Then, as we move along the path P, the vertices we encounter come in increasing numerical order. The tree in the figure is an example of a recursive tree. Notice that all paths starting at vertex 1 encounter the vertices in increasing order. For example, the highlighted path encounters the vertices $1 < 4 < 8 < 9$.

Please do the following:

a. Prove: If T is a recursive tree on n vertices, then vertex n is a leaf (provided $n > 1$).

b. Prove: If T is a recursive tree on $n > 1$ vertices, then $T - n$ (the tree T with vertex n deleted) is also a recursive tree (on $n - 1$ vertices).

c. Prove: If T is a recursive tree on n vertices and a new vertex $n + 1$ is attached as a leaf to any vertex of T to form a new tree T', then T' is also a recursive tree.

d. How many different recursive trees on n vertices are there? Prove your answer.

50.9. Let G be a forest with n vertices and c components. Find and prove a formula for the number of edges in G.

50.10. Prove that a graph is a forest if and only if all of its edges are cut edges.

50.11. In this problem, you will develop a new proof that every tree with two or more vertices has a leaf. Here is an outline for your proof.

a. First prove, using strong induction and the fact that every edge of a tree is a cut edge (Theorem 50.5), that a tree with n vertices has exactly $n - 1$ edges.

Please note that our previous proof of this fact (Theorem 50.9) used the fact that trees have leaves; that is why we need an alternative proof.

b. Use (a) to prove that the average degree of a vertex in a tree is less than 2.

c. Use (b) to prove that every tree (with at least two vertices) has a leaf.

50.12. Let T be a tree with $u, v \in V(T)$, $u \neq v$, and $uv \notin E(T)$. Prove that if we add the edge uv to T, the resulting graph has exactly one cycle.

50.13. Let G be a connected graph with $|V(G)| = |E(G)|$. Prove that G contains exactly one cycle.

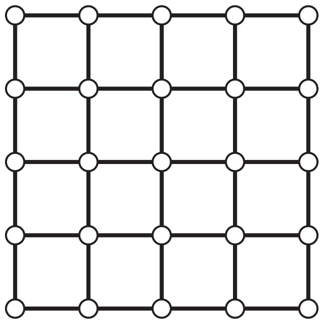
50.14. Prove:

a. Every cycle is connected.

b. Every cycle is 2-regular.

c. Conversely, every connected, 2-regular graph must be a cycle.

50.15. Let e be an edge of a graph G. Prove that e is a cut edge if and only if e is not in any cycle of G.



50.16. Let G be a graph. A cycle of G that contains all the vertices in G is called a *Hamiltonian cycle*.

a. Show that if $n \geq 5$, then $\overline{C_n}$ has a Hamiltonian cycle.

b. Prove that the graph in the figure does not have a Hamiltonian cycle.

50.17. Consider the following algorithm.

- **Input**: A connected graph G.
- **Output**: A spanning tree of G.

(1) Let T be a graph with the same vertices as G, but with no edges.

(2) Let $e_1, e_2, \ldots, e_m$ be the edges of G.

(3) For $k = 1, 2, \ldots, m$, do:

(3a) **If** adding edge e_k to T does not form a cycle with edges already in T, **then** add edge e_k to T.

(4) Output T.

Prove that this algorithm is correct. In other words, prove that whenever the input to this algorithm is a connected graph, the output of this algorithm is a spanning tree of G.

50.18. Consider the following algorithm.

- **Input**: A connected graph G.
- **Output**: A spanning tree of G.

(1) Let T be a copy of G.

(2) Let $e_1, e_2, \ldots, e_m$ be the edges of G.

(3) For $k = 1, 2, \ldots, m$, do:

(3a) **If** edge e_k is not a cut edge of T, **then** delete e_k from T

(4) Output T.

Prove that this algorithm is correct. In other words, prove that whenever the input to this algorithm is a connected graph, the output of this algorithm is a spanning tree of G.

50.19. Let G be a connected graph. The *Weiner index* of G, denoted $W(G)$, is the sum of the distances between all pairs of vertices in G. In other words, if $V(G) = \{1, 2, 3, \ldots, n\}$, then

$$W(G) = \sum_{1 \leq i < j \leq n} d(i, j)$$

where $d(i, j)$ is the distance between vertices i and j (see Exercise 49.13). For example, for a path on four vertices we have

$$W(P_4) = (1 + 2 + 3) + (1 + 2) + 1 = 10.$$

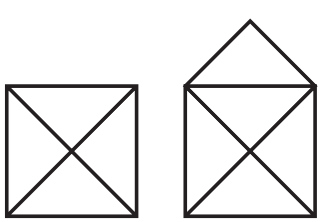
In this problem we ask that you show that a star (a tree with one vertex adjacent to all the other vertices which are, consequently, leaves) is the tree with the smallest Weiner index of all trees. Just for this problem, let S_n denote the star with n vertices.

a. Calculate $W(S_n)$ in simplest possible terms.

b. Prove that if T is any tree on n vertices, then $W(T) \geq W(S_n)$.

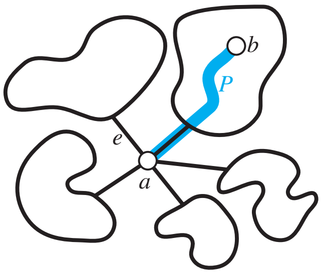
c. Prove that if T is any tree on n vertices and $W(T) = W(S_n)$ then T must be a star.

51 Eulerian Graphs



Earlier (in Section 47) we presented the classic Seven Bridges of Königsburg problem. We explained that it is impossible to walk all seven bridges without retracing a bridge (or taking a swim across the river) because the multigraph that represents the bridges has more than two vertices of odd degree.

Consider the two figures shown. The figure on the left has four corners where an odd number of lines meet. Therefore, it is impossible to draw this figure without lifting your pencil or redrawing a line. The odd corners must be the first or last points on such a drawing.



Proof. Suppose, for the sake of contradiction, that all edges incident at a are cut edges. Let b be the other vertex of odd degree in G.

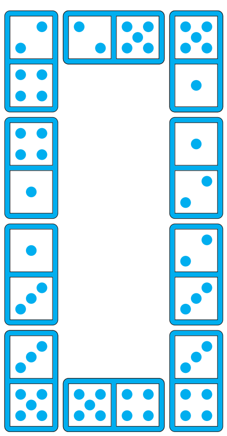
Since G is connected, there is an (a, b)-path P in G. Exactly one edge incident at a is traversed by P. Let e be any other edge incident at a.

Now consider the graph $G' = G - e$. This graph has exactly two components (Theorem 49.12). Since the path P does not use the edge e, vertices a and b are in the same component. Notice also that, in G', vertex a has even degree, and all other vertices in its component have not changed degree. This means that, in G', the component containing vertex a has exactly one vertex of odd degree, contradicting Exercise 47.15. ■

Recap

Motivated by the Seven Bridges of Königsburg problem, we defined Eulerian trails and tours in graphs. We showed that every connected graph with at most two vertices of odd degree has an Eulerian trail. If there are no vertices of odd degree, it has an Eulerian tour.

51 Exercises



51.1. For which values of n is the complete graph K_n Eulerian?

51.2. We noticed that a graph with more than two vertices of odd degree cannot have an Eulerian trail, but connected graphs with zero or two vertices of odd degree do have Eulerian trails. The missing case is connected graphs with exactly one vertex of odd degree. What can you say about those graphs?

51.3. A *domino* is a 2×1 rectangular piece of wood. On each half of the domino is a number, denoted by dots. (See the cover of this book.) In the figure, we show all $\binom{5}{2} = 10$ dominoes we can make where the numbers on the dominoes are all pairs of values chosen from $\{1, 2, 3, 4, 5\}$ (we do not include dominoes where the two numbers are the same). Notice that we have arranged the ten dominoes in a ring such that, where two dominoes meet, they show the same number.

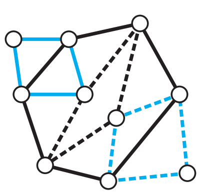
For what values of $n \geq 2$ is it possible to form a domino ring using all $\binom{n}{2}$ dominoes formed by taking all pairs of values from $\{1, 2, 3, \ldots, n\}$? Prove your answer.

Note: In a conventional box of dominoes, there are also dominoes both of whose squares have the same number of dots. You may either ignore these "doubles" or explain how they can easily be inserted into a ring made with the other dominoes.

51.4. Let G be a connected graph that is not Eulerian. Prove that it is possible to add a single vertex to G, together with some edges from this new vertex to some old vertices such that the new graph is Eulerian.

51.5. Let G be a connected graph that is not Eulerian. In G there must be an even number of odd-degree vertices (see Exercise 47.15). Let $a_1, b_1, a_2, b_2, \ldots, a_t, b_t$ be the vertices of odd degree in G.

If we add edges $a_1b_1, a_2b_2, \ldots, a_tb_t$ to G, does this give an Eulerian graph?



51.6. Let G be an Eulerian graph. Prove that it is possible to partition the edge set of G such that the edges in each part of the partition form a cycle of G.

The figure shows such a partition in which the edges from different parts of the partition are drawn in different colors and line styles.

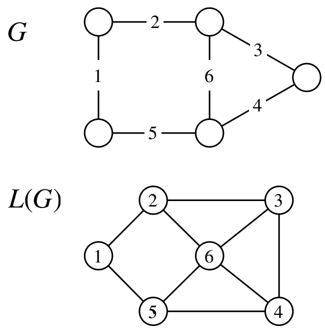
Note: A standard chess board is an 8×8 grid of squares.

51.7. A *rook* is a chess piece that may, on a single turn, move any number of squares horizontally or any number of squares vertically on the board. That is, if squares A and B are in the same row [or same column] then we are permitted to move the rook from A to B. But if A and B are in neither the same row nor the same column, a move between these squares is illegal. Thus in every row and every column there are $\binom{8}{2}$ pairs of squares between which the rook may move. This gives a total of $16\binom{8}{2} = 448$ such pairs.

Suppose a rook is placed on an empty chess board. Can we repeatedly move the rook so that it moves exactly once between each pair of squares in the same row and once between each pair of squares in the same column?

Note: When the rook travels between squares A and B, it should traverse either from A to B or from B to A, but not both.

51.8. Is it possible to walk the seven bridges of Königsburg so that you cross every bridge exactly twice, once in each direction?



51.9. Let G be a graph. The *line graph* of G is a new graph $L(G)$ whose vertices are the edges of G; two vertices of $L(G)$ are adjacent if, as edges of G, they share a common end point. In symbols:

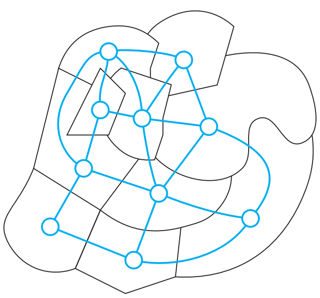
$$V[L(G)] = E(G) \quad \text{and} \quad E[L(G)] = \{e_1 e_2 : |e_1 \cap e_2| = 1\}.$$

Prove or disprove the following statements about the relationship between a graph G and its line graph $L(G)$:

a. If G is Eulerian, the $L(G)$ is also Eulerian.
b. If G has a Hamiltonian cycle, then $L(G)$ is Eulerian. (See Exercise 50.16 for the definition of a Hamiltonian cycle.)
c. If $L(G)$ is Eulerian, then G is also Eulerian.
d. If $L(G)$ is Eulerian, then G has a Hamiltonian cycle.

52 Coloring

The four color map problem and the exam-scheduling problem are both examples of *graph-coloring* problems. The general problem is as follows: Let G be a graph. To each vertex of G, we wish to assign a color such that adjacent vertices receive different colors. Of course, we could give every vertex its own color, but this is not terribly interesting and not relevant to applications. The objective is to use as few colors as possible.



For example, consider the map-coloring problem from Section 47. We can convert this problem into a graph-coloring problem by representing each country as a vertex of a graph. Two vertices in this graph are adjacent exactly when the countries they represent share a common border. Thus coloring the countries on the map corresponds exactly to coloring the vertices of the graph.

We can also convert the exam-scheduling problem into a graph-coloring problem. The vertices of this graph represent the courses at the university. Two vertices are adjacent when the courses they represent have a common student enrolled. The colors on the vertices represent the different examination time slots. Minimizing the number of colors assigned to the vertices corresponds to minimizing the number of exam periods.

Core Concepts

Colors are phenomena of the physical world and graphs are mathematical objects. It is mildly illogical to speak of applying colors (physical pigments) to vertices (abstract elements).

The careful way to define graph coloring is to give a mathematical definition of *coloring*.

Definition 52.1 **(Graph coloring)** Let G be a graph and let k be a positive integer. A k-*coloring* of G is a function

$$f : V(G) \to \{1, 2, \ldots, k\}.$$

We call this coloring *proper* provided

$$\forall xy \in E(G),\ f(x) \neq f(y).$$

If a graph has a proper k-coloring, we call it k-*colorable*.

The central idea in the definition is the function f. To each vertex $v \in V(G)$, the function f associates a value $f(v)$. The value $f(v)$ is the color of v. The palette of colors we use is the set $\{1, 2, \ldots, k\}$; we are using positive integers as "colors." Thus $f(v) = 3$ means that vertex v is assigned color 3 by the coloring f.

The condition $\forall xy \in E(G),\ f(x) \neq f(y)$ means that whenever vertices x and y are adjacent (form an edge of G), then $f(x) \neq f(y)$ (the vertices must get different colors). In a proper coloring, adjacent vertices are not assigned the same color.

Notice what the definition does not require: It does not say that all the colors must be used; that is, it does not require f to be onto. The number k refers to the size of the palette of