1.2.35. *Tucker's Algorithm.* Let G be a connected even graph. At each vertex, partition the incident edges into pairs (each edge appears in a pair for each of its endpoints). Starting along a given edge e, form a trail by leaving each vertex along the edge paired with the edge just used to enter it, ending with the edge paired with e. This decomposes G into closed trails. As long as there is more than one trail in the decomposition, find two trails with a common vertex and combine them into a longer trail by changing the pairing at a common vertex. Prove that this procedure works and produces an Eulerian circuit as its final trail. (Tucker [1976])

1.2.36. (+) *Alternative characterization of Eulerian graphs.*

a) Prove that if G is Eulerian and $G' = G - uv$, then G' has an odd number of u, v-trails that visit v only at the end. Prove also that the number of the trails in this list that are not paths is even. (Toida [1973])

b) Let v be a vertex of odd degree in a graph. For each edge e incident to v, let $c(e)$ be the number of cycles containing e. Use $\sum_e c(e)$ to prove that $c(e)$ is even for some e incident to v. (McKee [1984])

c) Use part (a) and part (b) to conclude that a nontrivial connected graph is Eulerian if and only if every edge belongs to an odd number of cycles.

1.2.37. (!) Use extremality to prove that the connection relation is transitive. (Hint: Given a u, v-path P and a v, w-path Q, consider the first vertex of P in Q.)

1.2.38. (!) Prove that every n-vertex graph with at least n edges contains a cycle.

1.2.39. Suppose that every vertex of a loopless graph G has degree at least 3. Prove that G has a cycle of even length. (Hint: Consider a maximal path.) (P. Kwok)

1.2.40. (!) Let P and Q be paths of maximum length in a connected graph G. Prove that P and Q have a common vertex.

1.2.41. Let G be a connected graph with at least three vertices. Prove that G has two vertices x, y such that 1) $G - \{x, y\}$ is connected and 2) x, y are adjacent or have a common neighbor. (Hint: Consider a longest path.) (Chung [1978a])

1.2.42. Let G be a connected simple graph that does not have P_4 or C_4 as an induced subgraph. Prove that G has a vertex adjacent to all other vertices. (Hint: Consider a vertex of maximum degree.) (Wolk [1965])

1.2.43. (+) Use induction on k to prove that every connected simple graph with an even number of edges decomposes into paths of length 2. Does the conclusion remain true if the hypothesis of connectedness is omitted?

1.3. Vertex Degrees and Counting

The degrees of the vertices are fundamental parameters of a graph. We repeat the definition in order to introduce important notation.

1.3.1. Definition. The **degree** of vertex v in a graph G, written $d_G(v)$ or $d(v)$, is the number of edges incident to v, except that each loop at v counts twice. The maximum degree is $\Delta(G)$, the minimum degree is $\delta(G)$, and G is **regular** if $\Delta(G) = \delta(G)$. It is **k-regular** if the common degree is k. The **neighborhood** of v, written $N_G(v)$ or $N(v)$, is the set of vertices adjacent to v.

1.3.2. Definition. The **order** of a graph G, written $n(G)$, is the number of vertices in G. An **n-vertex graph** is a graph of order n. The **size** of a graph G, written $e(G)$, is the number of edges in G. For $n \in \mathbb{N}$, the notation $[n]$ indicates the set $\{1, \ldots, n\}$.

Since our graphs are finite, $n(G)$ and $e(G)$ are well-defined nonnegative integers. We also often use "e" by itself to denote an edge. When e denotes a particular edge, it is not followed by the name of a graph in parentheses, so the context indicates the usage. We have used "n-cycle" to denote a cycle with n vertices; this is consistent with "n-vertex graph".

COUNTING AND BIJECTIONS

We begin with counting problems about subgraphs in a graph. The first such problem is to count the edges; we do this using the vertex degrees. The resulting formula is an essential tool of graph theory, sometimes called the "First Theorem of Graph Theory" or the "Handshaking Lemma".

1.3.3. Proposition. (Degree-Sum Formula) If G is a graph, then

$$\sum_{v \in V(G)} d(v) = 2e(G).$$

Proof: Summing the degrees counts each edge twice, since each edge has two ends and contributes to the degree at each endpoint. ■

The proof holds even when G has loops, since a loop contributes 2 to the degree of its endpoint. For a loopless graph, the two sides of the formula count the set of pairs (v, e) such that v is an endpoint of e, grouped by vertices or grouped by edges. "Counting two ways" is an elegant technique for proving integer identities (see Exercise 31 and Appendix A).

The degree-sum formula has several immediate corollaries. Corollary 1.3.5 applies in Exercises 9–13 and in many arguments of later chapters.

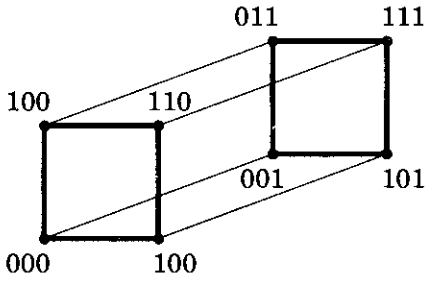
1.3.4. Corollary. In a graph G, the average vertex degree is $\frac{2e(G)}{n(G)}$, and hence $\delta(G) \le \frac{2e(G)}{n(G)} \le \Delta(G)$. ■

1.3.5. Corollary. Every graph has an even number of vertices of odd degree. No graph of odd order is regular with odd degree. ■

1.3.6. Corollary. A k-regular graph with n vertices has $nk/2$ edges. ■

We next introduce an important family of graphs.

1.3.7. Definition. The **k-dimensional cube** or **hypercube** Q_k is the simple graph whose vertices are the k-tuples with entries in $\{0, 1\}$ and whose edges are the pairs of k-tuples that differ in exactly one position. A j-**dimensional subcube** of Q_k is a subgraph of Q_k isomorphic to Q_j.



Above we show Q_3. The hypercube is a natural computer architecture. Processors can communicate directly if they correspond to adjacent vertices in Q_k. The k-tuples that name the vertices serve as addresses for the processors.

1.3.8. Example. *Structure of hypercubes.* The *parity* of a vertex in Q_k is the parity of the number of 1s in its name, even or odd. Each edge of Q_k has an even vertex and an odd vertex as endpoints. Hence the even vertices form an independent set, as do the odd vertices, and Q_k is bipartite.

Each position in the k-tuples can be specified in two ways, so $n(Q_k) = 2^k$. A neighbor of a vertex is obtained by changing one of the k positions in its name to the other value. Thus Q_k is k-regular. By Corollary 1.3.6, $e(Q_k) = k2^{k-1}$.

The bold edges above show two subgraphs of Q_3 isomorphic to Q_2, formed by keeping the last coordinate fixed at 0 or at 1. We can form a j-dimensional subcube by keeping any $k - j$ coordinates fixed and letting the values in the remaining j coordinates range over all 2^j possible j-tuples. The subgraph induced by such a set of vertices is isomorphic to Q_j. Since there are $\binom{k}{j}$ ways to pick j coordinates to vary and 2^{k-j} ways to specify the values in the fixed coordinates, this specifies $\binom{k}{j}2^{k-j}$ such subcubes. In fact, there are no other j-dimensional subcubes (Exercise 29).

The copies of Q_1 are simply the edges in Q_k. Our formula reduces to $k2^{k-1}$ when $j = 1$, so we have found another counting argument to compute $e(Q_k)$.

When $j = k - 1$, our discussion suggests a recursive description of Q_k. Append 0 to the vertex names in a copy of Q_{k-1}; append 1 in another copy. Add edges joining vertices from the two copies whose first $k - 1$ coordinates are equal. The result is Q_k. The basis of the construction is the 1-vertex graph Q_0. This description leads to inductive proofs for many properties of hypercubes, including $e(Q_k) = k2^{k-1}$ (Exercise 23). ■

A hypercube is a regular bipartite graph. A simple counting argument proves a fundamental observation about such graphs.

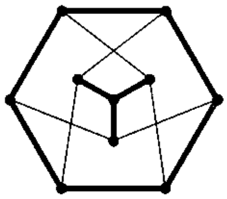
1.3.9. Proposition. If $k > 0$, then a k-regular bipartite graph has the same number of vertices in each partite set.

Proof: Let G be an X, Y-bigraph. Counting the edges according to their endpoints in X yields $e(G) = k\,|X|$. Counting them by their endpoints in Y yields $e(G) = k\,|Y|$. Thus $k\,|X| = k\,|Y|$, which yields $|X| = |Y|$ when $k > 0$. ■

Another technique for counting a set is to establish a bijection from it to a set of known size. Our next example uses this approach. Other examples of combinatorial arguments for counting problems appear in Appendix A. Exercises 18–35 involve counting.

1.3.10. Example. *The Petersen graph has ten 6-cycles.* Let G be the Petersen graph. Being 3-regular, G has ten claws (copies of $K_{1,3}$). We establish a one-to-one correspondence between the 6-cycles and the claws.

Since G has girth 5, every 6-cycle F is an induced subgraph. Each vertex of F has one neighbor outside F. Since nonadjacent vertices have exactly one common neighbor (Proposition 1.1.38), opposite vertices on F have a common neighbor outside F. Since G is 3-regular, the resulting three vertices outside F are distinct. Thus deleting $V(F)$ leaves a subgraph with three vertices of degree 1 and one vertex of degree 3; it is a claw.



We show that each claw H in G arises exactly once in this way. Let S be the set of vertices with degree 1 in H; S is an independent set. The central vertex of H is already a common neighbor, so the six other edges from S reach distinct vertices. Thus $G - V(H)$ is 2-regular. Since G has girth 5, $G - V(H)$ must be a 6-cycle. This 6-cycle yields H when its vertices are deleted. ■

We present one more counting argument related to a long-standing conjecture. Subgraphs obtained by deleting a single vertex are called **vertex-deleted subgraphs**. These subgraphs need not all be distinct; for example, the n vertex-deleted subgraphs of C_n are all isomorphic to P_{n-1}.

1.3.11.* Proposition. For a simple graph G with vertices $v_1, \ldots, v_n$ and $n \geq 3$,

$$e(G) = \frac{\Sigma e(G - v_i)}{n-2} \qquad \text{and} \qquad d_G(v_j) = \frac{\Sigma e(G - v_i)}{n-2} - e(G - v_j).$$

Proof: An edge e of G appears in $G - v_i$ if and only if v_i is not an endpoint of e. Thus $\sum(G - v_i)$ counts each edge exactly $n - 2$ times.

Once we know $e(G)$, the degree of v_j can be computed as the number of edges lost when deleting v_j to form $G - v_j$. ■

Typically, we are given the vertex-deleted subgraphs as unlabeled graphs; we know only the list of isomorphism classes, not which vertex of $G - v_i$ corresponds to which vertex in G. This can make it very difficult to tell what G is. For example, K_2 and its complement have the same list of vertex-deleted subgraphs. For larger graphs we have the **Reconstruction Conjecture**, formulated in 1942 by Kelly and Ulam.

1.3.12.* Conjecture. (Reconstruction Conjecture) If G is a simple graph with at least three vertices, then G is uniquely determined by the list of (isomorphism classes of) its vertex-deleted subgraphs. ■

The list of vertex-deleted subgraphs of G has $n(G)$ items. Proposition 1.3.11 shows that $e(G)$ and the list of vertex degrees can be reconstructed. The latter implies that regular graphs can be reconstructed (Exercise 37). We can also determine whether G is connected (Exercise 38); using this, disconnected graphs can be reconstructed (Exercise 39). Other sufficient conditions for reconstructibility are known, but the general conjecture remains open.

EXTREMAL PROBLEMS

An **extremal problem** asks for the maximum or minimum value of a function over a class of objects. For example, the maximum number of edges in a simple graph with n vertices is $\binom{n}{2}$.

1.3.13. Proposition. The minimum number of edges in a connected graph with n vertices is $n - 1$.

Proof: By Proposition 1.2.11, every graph with n vertices and k edges has at least $n - k$ components. Hence every n-vertex graph with fewer than $n - 1$ edges has at least two components and is disconnected. The contrapositive of this is that every connected n-vertex graph has at least $n - 1$ edges. This lower bound is achieved by the path P_n. ■

1.3.14. Remark. Proving that β is the minimum of $f(G)$ for graphs in a class **G** requires showing two things:

1) $f(G) \geq \beta$ for all $G \in \mathbf{G}$.
2) $f(G) = \beta$ for some $G \in \mathbf{G}$.

The proof of the bound must apply to every $G \in \mathbf{G}$. For equality it suffices to obtain an example in **G** with the desired value of f.

Changing "$\geq$" to "$\leq$" yields the criteria for a maximum. ■

Next we solve a maximization problem that is not initially phrased as such.

1.3.15. Proposition. If G is a simple n-vertex graph with $\delta(G) \geq (n - 1)/2$, then G is connected.

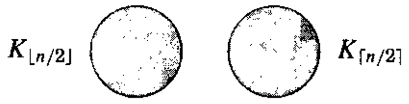
Proof: Choose $u, v \in V(G)$. It suffices to show that u, v have a common neighbor if they are not adjacent. Since G is simple, we have $|N(u)| \ge \delta(G) \ge (n-1)/2$, and similarly for v. When $u \not\leftrightarrow v$, we have $|N(u) \cup N(v)| \le n-2$, since u and v are not in the union. Using Remark A.13 of Appendix A, we thus compute

$$|N(u) \cap N(v)| = |N(u)| + |N(v)| - |N(u) \cup N(v)| \ge \tfrac{n-1}{2} + \tfrac{n-1}{2} - (n-2) = 1. \blacksquare$$

We say that a result is **best possible** or **sharp** when there is some aspect of it that cannot be strengthened without the statement becoming false. As shown by the next example, this holds for Proposition 1.3.15; when $\delta(G)$ is smaller than $(n(G)-1)/2$, we cannot still conclude that G must be connected.

1.3.16. Example. Let G be the n-vertex graph with components isomorphic to $K_{\lfloor n/2 \rfloor}$ and $K_{\lceil n/2 \rceil}$, where the **floor** $\lfloor x \rfloor$ of x is the largest integer at most x and the **ceiling** $\lceil x \rceil$ of x is the smallest integer at least x. Since $\delta(G) = \lfloor n/2 \rfloor - 1$ and G is disconnected, the inequality in Proposition 1.3.15 is sharp.

We use the floor and ceiling functions here in order to describe a single family of graphs providing an example for each n. $\blacksquare$



By providing a family of examples to show that the bound is best possible, we have solved an extremal problem. Together, Proposition 1.3.15 and Example 1.3.16 prove "The minimum value of $\delta(G)$ that forces an n-vertex simple graph G to be connected is $\lfloor n/2 \rfloor$," or "The maximum value of $\delta(G)$ among disconnected n-vertex simple graphs is $\lfloor n/2 \rfloor - 1$."

We introduce compact notation to describe the graph of Example 1.3.16.

1.3.17. Definition. The graph obtained by taking the union of graphs G and H with disjoint vertex sets is the **disjoint union** or **sum**, written $G + H$. In general, mG is the graph consisting of m pairwise disjoint copies of G.

1.3.18. Example. If G and H are connected, then $G+H$ has components G and H, so the graph in Example 1.3.16 is $K_{\lfloor n/2 \rfloor} + K_{\lceil n/2 \rceil}$. This notation is convenient when we have not named the vertices. Note that $K_m + K_n = \overline{K_{m,n}}$.

The graph mK_2 consists of m pairwise disjoint edges. $\blacksquare$

In graph theory, we use "extremal problem" for finding an optimum over a class of graphs. When seeking extremes in a single graph, such as the maximum size of an independent set, or maximum size of a bipartite subgraph, we have a different problem for each graph. To distinguish these from the earlier type of problem, we call them **optimization problems**.

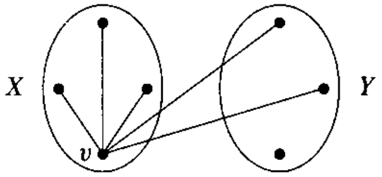
Since an optimization problem has an instance for each graph, we usually can't list all solutions. We may seek a solution procedure or bounds on the answer in terms of other aspects of the input graph. In this light we consider the problem of finding a large bipartite subgraph. It allows us to introduce the technique of constructive or "algorithmic" proof. (An **algorithm** is a procedure for performing some task.)

One way to prove that something exists is to build it. Such proofs can be viewed as algorithms. To complete an algorithmic proof, we must prove that the algorithm terminates and yields the desired result. This may involve induction,contradiction, finiteness, etc. We prove that every graph has a large bipartitesubgraph by providing an algorithm to find one. Exercises 45–49 arerelated to finding large bipartite subgraphs.

1.3.19. Theorem. Every loopless graph G has a bipartite subgraph with at least $e(G)/2$ edges.

Proof: We start with any partition of $V(G)$ into two sets X, Y. Using the edges having one endpoint in each set yields a bipartite subgraph H with bipartition X, Y. If H contains fewer than half the edges of G incident to a vertex v, then v has more edges to vertices in its own class than in the other class, as illustrated below. Moving v to the other class gains more edges of G than it loses.

We move one vertex in this way as long as the current bipartite subgraph captures less than half of the edges at some vertex. Each such switch increases the size of the subgraph, so the process must terminate. When it terminates, we have $d_H(v) \geq d_G(v)/2$ for every $v \in V(G)$. Summing this and applying the degree-sum formula yields $e(H) \geq e(G)/2$. ∎

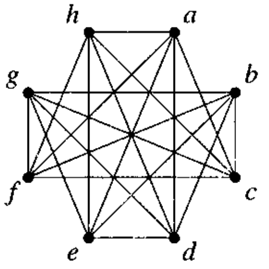


Algorithmic proofs often correspond to proofs by induction or extremality. Such proofs are shorter and may be easier to find, so we may seek such a proof and later convert it to an algorithm. For example, here is the proof of Theorem 1.3.19 in the language of extremality and contradiction; in effect, the extremal choice of H goes directly to the end of the algorithm:

> Let H be the bipartite subgraph of G that has the most edges. If $d_H(v) \geq d_G(v)/2$ for all $v \in V(G)$, then the degree-sum formula yields $e(H) \geq e(G)/2$. Otherwise, $d_H(v) < d_G(v)/2$ for some $v \in V(G)$, and then switching v in the bipartition contradicts the choice of H.

1.3.20. Example. *Local maximum.* The algorithm in Theorem 1.3.19 need not produce a bipartite subgraph with the most edges, merely one with at least half the edges. The graph below is 5-regular with 8 vertices and hence has 20 edges. The bipartition $X = \{a, b, c, d\}$ and $Y = \{e, f, g, h\}$ yields a 3-regular bipartite subgraph with 12 edges. The algorithm terminates here; switching one vertex would pick up two edges but lose three.

Nevertheless, the bipartition $X = \{a, b, g, h\}$ and $Y = \{c, d, e, f\}$ yields a 4-regular bipartite subgraph with 16 edges. An algorithm seeking the maximum by local changes may get stuck in a local maximum. ■



1.3.21. Remark. In a graph G, the (global) maximum number of edges in a bipartite subgraph is $e(G)$ minus the minimum number of edges needed to obtain at least one edge from every odd cycle. ■

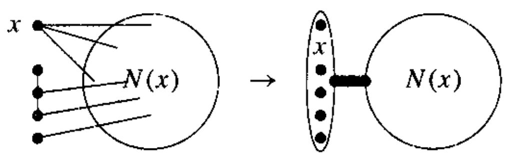
Our next extremal problem doesn't start with bipartite graphs, but it winds up there. In politics and warfare, seldom do two enemies have a common enemy; usually two of the three combine against the third. Given n factions, how many pairs of enemies can there be if no two enemies have a common enemy?

In the language of graphs, we are asking for the maximum number of edges in a simple n-vertex graph with no triangle. Bipartite graphs have no triangles, but also many non-bipartite graphs (such as the Petersen graph) have no triangles. Using extremality (by choosing a vertex of maximum degree), we will prove that the maximum is indeed achieved by a complete bipartite graph.

1.3.22. Definition. A graph G is H**-free** if G has no induced subgraph isomorphic to H.

1.3.23. Theorem. (Mantel [1907]) The maximum number of edges in an n-vertex triangle-free simple graph is $\lfloor n^2/4 \rfloor$.

Proof: Let G be an n-vertex triangle-free simple graph. Let x be a vertex of maximum degree, with $k = d(x)$. Since G has no triangles, there are no edges among neighbors of x. Hence summing the degrees of x and its nonneighbors counts at least one endpoint of every edge: $\sum_{v \notin N(x)} d(v) \ge e(G)$. We sum over $n - k$ vertices, each having degree at most k, so $e(G) \le (n - k)k$.



Since $(n - k)k$ counts the edges in $K_{n-k,k}$, we have now proved that $e(G)$ is bounded by the size of some biclique with n vertices. Moving a vertex of $K_{n,k,k}$

from the set of size k to the set of size $n-k$ gains $k-1$ edges and loses $n-k$ edges. The net gain is $2k-1-n$, which is positive for $2k > n+1$ and negative for $2k < n+1$. Thus $e(K_{n-k,k})$ is maximized when k is $\lceil n/2 \rceil$ or $\lfloor n/2 \rfloor$. The product is then $n^2/4$ for even n and $(n^2-1)/4$ for odd n. Thus $e(G) \le \lfloor n^2/4 \rfloor$.

To prove that the bound is best possible, we exhibit a triangle-free graph with $\lfloor n^2/4 \rfloor$ edges: $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$. ∎

Although $(n-k)k$ can be maximized over k using calculus, the discrete approach is preferable in some ways. It directly restricts k to be an integer and generalizes easily to more variables. The switching idea used is that of Theorem 1.3.19; here we have used it to find the largest bipartite subgraph of K_n. In Theorem 5.2.9 we generalize Theorem 1.3.23 to K_{r+1}-free graphs.

Mantel's result leads us to another reason for phrasing inductive proofs in the format that we have used. The reason is safety.

1.3.24. Example. *A failed proof.* Let us try to prove Theorem 1.3.23 by induction on n. Basis step: $n \le 2$. Here the complete graph K_n has the most edges and has no triangles.

Induction step: $n > 2$. We try "Suppose that the claim is true when $n = k$, so $K_{\lfloor k/2 \rfloor, \lceil k/2 \rceil}$ is the largest triangle-free graph with k vertices. We add a new vertex x to form a triangle-free graph with $k+1$ vertices. Making x adjacent to vertices from both partite sets would create a triangle. Hence we add the most edges by making x adjacent to all the vertices in the larger partite set of $K_{\lfloor k/2 \rfloor, \lceil k/2 \rceil}$. Doing so creates $K_{\lfloor (k+1)/2 \rfloor, \lceil (k+1)/2 \rceil}$. This completes the proof."

This argument is wrong, because we did not consider all triangle-free graphs with $k+1$ vertices. We considered only those containing the extremal k-vertex graph as an induced subgraph. This graph does appear in the extremal graph with $k+1$ vertices, but we cannot use that fact before proving it. It remains possible that the largest example with $k+1$ vertices arises by adding a new vertex of high degree to a non-maximal example with k vertices.

Exercise 51 develops a correct proof by induction on n. ∎

The error in Example 1.3.24 was that our induction step did not consider all instances of the statement for the new larger value of the parameter. We call this error the **induction trap**. If the induction step grows an instance with the new value of the parameter from a smaller instance, then we must prove that all instances with the new value have been considered.

When there is only one instance for each value of the induction parameter (as in summation formulas), this does not cause trouble. With more than one instance, it is safer and simpler to start with an arbitrary instance for the larger parameter value. This explicitly considers each instance G for the larger value, so we don't need to prove that we have generated them all.

However, when we obtain from G a smaller instance, we must confirm that the induction hypothesis applies to it. For example, in the inductive proof of the characterization of Eulerian circuits (Theorem 1.2.26), we must apply the induction hypothesis to each component of the graph obtained by deleting the edges of a cycle, not to the entire graph at once.

1.3.25. Remark. *A template for induction.* Often the statement we want to prove by induction on n is an implication: $A(n) \Rightarrow B(n)$. We must prove that every instance G satisfying $A(n)$ also satisfies $B(n)$. Our induction step follows a typical format. From G we obtain some (smaller) G'. If we show that G' satisfies $A(n-1)$ (for ordinary induction), then the induction hypothesis implies that G' satisfies $B(n-1)$. Now we use the information that G' satisfies $B(n-1)$ to prove that G satisfies $B(n)$.

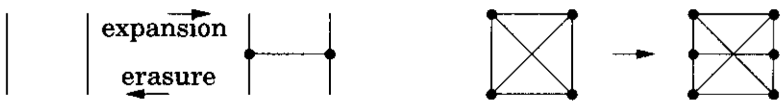
$$\begin{tikzcd}
G \text{ satisfies } A(n) \arrow[d, Rightarrow] & G \text{ satisfies } B(n) \\
G' \text{ satisfies } A(n-1) \arrow[r, Rightarrow] & G' \text{ satisfies } B(n-1) \arrow[u, Rightarrow]
\end{tikzcd}$$

Here the central implication is the statement of the induction hypothesis, and the others are the work we must do. Our induction proofs have followed this format. ■

1.3.26.* Example. *The induction trap.* The induction trap can lead to a *false* conclusion. Let us try to prove by induction on the number of vertices that every 3-regular connected simple graph has no cut-edge.

By the degree-sum formula, every regular graph with odd degree has even order, so we consider graphs with $2m$ vertices. The smallest 3-regular simple graph, K_4, is connected and has no cut-edge; this proves the basis step with $m = 2$. Now consider the induction step.

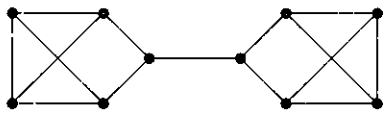
Given a simple 3-regular graph G with $2k$ vertices, we can obtain a simple 3-regular graph G' with $2(k+1)$ vertices (the next larger possible order) by "expansion": take two edges of G, replace them by paths of length 2 through new vertices, and add an edge joining the two new vertices. As illustrated below, $K_{3,3}$ arises from K_4 by one expansion on two disjoint edges.



If G is connected, then the expanded graph G' is also connected: a path between old vertices that traversed a replaced edge has merely lengthened, and a path to a new vertex in G' is obtained from a path in G to a neighbor.

If G has no cut-edge, then every edge lies on a cycle (Theorem 1.2.14). These cycles remain in G' (those using replaced edges become longer). The edge joining the two new vertices in G' also lies on a cycle using a path in G between the edges that were replaced. Theorem 1.2.14 now implies that G' has no cut-edge.

We have proved that if G is connected and has no cut-edge, then the same holds for G'. We might think we have proved by induction on m that every 3-regular simple connected graph with $2m$ vertices has no cut-edge, but the graph below is a counterexample. The proof fails because we cannot build every 3-regular simple connected graph from K_4 by expansions. We cannot even obtain all those without cut-edges, as shown in Exercise 66. ■



Appendix A presents another example of the induction trap.

GRAPHIC SEQUENCES

Next we consider all the vertex degrees together.

1.3.27. Definition. The **degree sequence** of a graph is the list of vertex degrees, *usually* written in nonincreasing order, as $d_1 \geq \cdots \geq d_n$.

Every graph has a degree sequence, but which sequences occur? That is, given nonnegative integers $d_1, \dots, d_n$, is there a graph with these as the vertex degrees? The degree-sum formula implies that $\sum d_i$ must be even. When we allow loops and multiple edges, TONCAS.

1.3.28. Proposition. The nonnegative integers $d_1, \dots, d_n$ are the vertex degrees of some graph if and only if $\sum d_i$ is even.

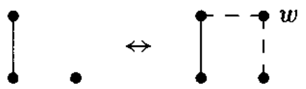
Proof: *Necessity.* When some graph G has these numbers as its vertex degrees, the degree-sum formula implies that $\sum d_i = 2e(G)$, which is even.

Sufficiency. Suppose that $\sum d_i$ is even. We construct a graph with vertex set $v_1, \dots, v_n$ and $d(v_i) = d_i$ for all i. Since $\sum d_i$ is even, the number of odd values is even. First form an arbitrary pairing of the vertices in $\{v_i : d_i \text{ is odd}\}$. For each resulting pair, form an edge having these two vertices as its endpoints. The remaining degree needed at each vertex is even and nonnegative; satisfy this for each i by placing $\lfloor d_i/2 \rfloor$ loops at v_i. ■

This proof is constructive; we could also use induction (Exercise 56). The construction is easy with loops available. Without them, $(2, 0, 0)$ is not realizable and the condition is not sufficient. Exercise 63 characterizes the degree sequences of loopless graphs. We next characterize degree sequences of simple graphs by a recursive condition that readily yields an algorithm. Many other characterizations are known; Sierksma–Hoogeveen [1991] lists seven.

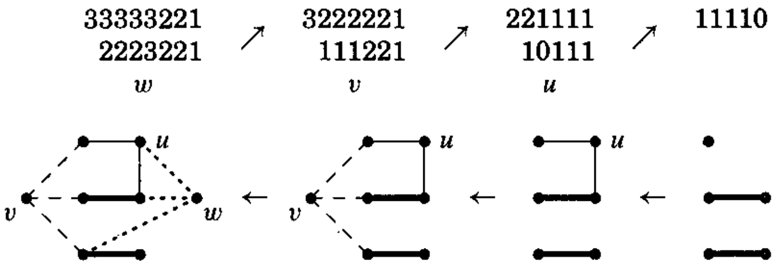
1.3.29. Definition. A **graphic sequence** is a list of nonnegative numbers that is the degree sequence of some simple graph. A simple graph with degree sequence d "realizes" d.

1.3.30. Example. *A recursive condition.* The lists $2, 2, 1, 1$ and $1, 0, 1$ are graphic. The graph $K_2 + K_1$ realizes $1, 0, 1$. Adding a new vertex adjacent to vertices of degrees 1 and 0 yields a graph with degree sequence $2, 2, 1, 1$, as shown below. Conversely, if a graph realizing $2, 2, 1, 1$ has a vertex w with neighbors of degrees 2 and 1, then deleting w yields a graph with degrees $1, 0, 1$.



Similarly, to test 33333221, we seek a realization with a vertex w of degree 3 having three neighbors of degree 3. This exists if and only if 2223221 is graphic. We reorder this and test 3222221. We continue deleting and reordering until we can tell whether the remaining list is realizable. If it is, then we insert vertices with the desired neighbors to work back to a realization of the original list. The realization is not unique.

The next theorem implies that this recursive test works. ∎



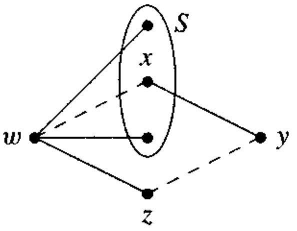
1.3.31. Theorem. (Havel [1955], Hakimi [1962]) For $n > 1$, an integer list d of size n is graphic if and only if d' is graphic, where d' is obtained from d by deleting its largest element Δ and subtracting 1 from its Δ next largest elements. The only 1-element graphic sequence is $d_1 = 0$.

Proof: For $n = 1$, the statement is trivial. For $n > 1$, we first prove that the condition is sufficient. Given d with $d_1 \geq \cdots \geq d_n$ and a simple graph G' with degree sequence d', we add a new vertex adjacent to vertices in G' with degrees $d_2 - 1, \dots, d_{\Delta+1} - 1$. These d_i are the Δ largest elements of d after (one copy of) Δ itself, but $d_2 - 1, \dots, d_{\Delta+1} - 1$ need not be the Δ largest numbers in d'.

To prove necessity, we begin with a simple graph G realizing d and produce a simple graph G' realizing d'. Let w be a vertex of degree Δ in G. Let S be a set of Δ vertices in G having the "desired degrees" $d_2, \dots, d_{\Delta+1}$. If $N(w) = S$, then we delete w to obtain G'.

Otherwise, some vertex of S is missing from $N(w)$. In this case, we modify G to increase $|N(w) \cap S|$ without changing any vertex degree. Since $|N(w) \cap S|$ can increase at most Δ times, repeating this converts G into another graph G^* that realizes d and has S as the neighborhood of w. From G^* we then delete w to obtain the desired graph G' realizing d'.

To find the modification when $N(w) \neq S$, we choose $x \in S$ and $z \notin S$ so that $w \leftrightarrow z$ and $w \not\leftrightarrow x$. We want to add wx and delete wz, but we must preserve vertex degrees. Since $d(x) \geq d(z)$ and already w is a neighbor of z but not x, there must be a vertex y adjacent to x but not to z. Now we delete $\{wz, xy\}$ and add $\{wx, yz\}$ to increase $|N(w) \cap S|$. ■

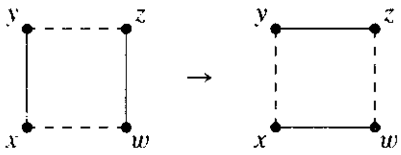


Theorem 1.3.31 tests a list of n numbers by testing a list of $n-1$ numbers; it yields a recursive algorithm to test whether d is graphic. The necessary condition "$\sum d_i$ even" holds implicitly: $\sum d_i' = (\sum d_i) - 2\Delta$ implies that $\sum d_i'$ and $\sum d_i$ have the same parity.

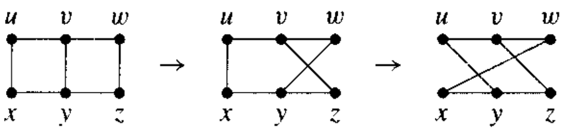
An algorithmic proof using "local change" pushes an object toward a desired condition. This can be phrased as proof by induction, where the induction parameter is the "distance" from the desired condition. In the proof of Theorem 1.3.31, this distance is the number of vertices in S that are missing from $N(w)$.

We used edge switches to transform an arbitrary graph with degree sequence d into a graph satisfying the desired condition. Next we will show that every simple graph with degree sequence d can be transformed by such switches into every other.

1.3.32. Definition. A **2-switch** is the replacement of a pair of edges xy and zw in a simple graph by the edges yz and wx, given that yz and wx did not appear in the graph originally.



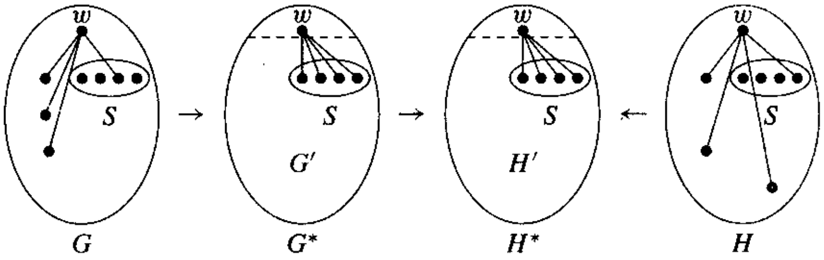
The dashed lines above indicate nonadjacent pairs. If $y \leftrightarrow z$ or $w \leftrightarrow x$, then the 2-switch cannot be performed, because the resulting graph would not be simple. A 2-switch preserves all vertex degrees. If some 2-switch turns H into H^*, then a 2-switch on the same four vertices turns H^* into H. Below we illustrate two successive 2-switches.



1.3.33.* Theorem. (Berge [1973, p153–154]) If G and H are two simple graphs with vertex set V, then $d_G(v) = d_H(v)$ for every $v \in V$ if and only if there is a sequence of 2-switches that transforms G into H.

Proof: Every 2-switch preserves vertex degrees, so the condition is sufficient. Conversely, when $d_G(v) = d_H(v)$ for all $v \in V$, we obtain an appropriate sequence of 2-switches by induction on the number of vertices, n. If $n \le 3$, then for each $d_1, \dots, d_n$ there is at most one simple graph with $d(v_i) = d_i$. Hence we can use $n = 3$ as the basis step.

Consider $n \ge 4$, and let w be a vertex of maximum degree, Δ. Let $S = \{v_1, \dots, v_\Delta\}$ be a fixed set of vertices with the Δ highest degrees other than w. As in the proof of Theorem 1.3.31, some sequence of 2-switches transforms G to a graph G^* such that $N_{G^*}(w) = S$, and some such sequence transforms H to a graph H^* such that $N_{H^*}(w) = S$.



Since $N_{G^*}(w) = N_{H^*}(w)$, deleting w leaves simple graphs $G' = G^* - w$ and $H' = H^* - w$ with $d_{G'}(v) = d_{H'}(v)$ for every vertex v. By the induction hypothesis, some sequence of 2-switches transforms G' to H'. Since these do not involve w, and w has the same neighbors in G^* and H^*, applying this sequence transforms G^* to H^*. Hence we can transform G to H by transforming G to G^*, then G^* to H^*, then (in reverse order) the transformation of H to H^*. ■

We could also phrase this using induction on the number of edges appearing in exactly one of G and H, which is 0 if and only if they are already the same. In this approach, it suffices to find a 2-switch in G that makes it closer to H or a 2-switch in H that makes it closer to G.

EXERCISES

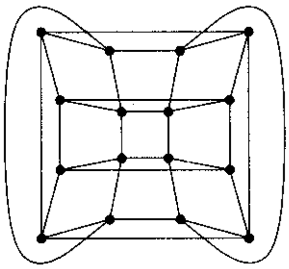
A statement with a parameter must be proved for all values of the parameter; it cannot be proved by giving examples. Counting a set includes providing proof.

1.3.1. (−) Prove or disprove: If u and v are the only vertices of odd degree in a graph G, then G contains a u, v-path.

1.3.2. (−) In a class with nine students, each student sends valentine cards to three others. Determine whether it is possible that each student receives cards from the same three students to whom he or she sent cards.

1.3.3. (−) Let u and v be adjacent vertices in a simple graph G. Prove that uv belongs to at least $d(u) + d(v) - n(G)$ triangles in G.

1.3.4. (−) Prove that the graph below is isomorphic to Q_4.



1.3.5. (−) Count the copies of P_3 and C_4 in Q_k.

1.3.6. (−) Given graphs G and H, determine the number of components and maximum degree of $G + H$ in terms of the those parameters for G and H.

1.3.7. (−) Determine the maximum number of edges in a bipartite subgraph of P_n, of C_n, and of K_n.

1.3.8. (−) Which of the following are graphic sequences? Provide a construction or a proof of impossibility for each.

a) (5,5,4,3,2,2,2,1), c) (5,5,5,3,2,2,1,1),
b) (5,5,4,4,2,2,1,1), d) (5,5,5,4,2,1,1,1).

• • • • •

1.3.9. In a league with two divisions of 13 teams each, determine whether it is possible to schedule a season with each team playing nine games against teams within its division and four games against teams in the other division.

1.3.10. Let l, m, n be nonnegative integers with $l + m = n$. Find necessary and sufficient conditions on l, m, n such that there exists a connected simple n-vertex graph with l vertices of even degree and m vertices of odd degree.

1.3.11. Let W be a closed walk in a graph G. Let H be the subgraph of G consisting of edges used an odd number of times in W. Prove that $d_H(v)$ is even for every $v \in V(G)$.

1.3.12. (!) Prove that an even graph has no cut-edge. For each $k \geq 1$, construct a $2k + 1$-regular simple graph having a cut-edge.

1.3.13. (+) A **mountain range** is a polygonal curve from $(a, 0)$ to $(b, 0)$ in the upper half-plane. Hikers A and B begin at $(a, 0)$ and $(b, 0)$, respectively. Prove that A and B can meet by traveling on the mountain range in such a way that at all times their heights above the horizontal axis are the same. (Hint: Define a graph to model the movements, and use Corollary 1.3.5.) (Communicated by D.G. Hoffman.)