Theory is the first term in the Taylor series of practice.
— THOMAS M COVER (1992)

The number of systems of terminology presently used in graph theory is equal, to a close approximation, to the number of graph theorists.
— RICHARD P STANLEY (1986)

Graph theory: The basics. A graph G consists of a set V of vertices together with a set E of edges, which are pairs of distinct vertices. We will assume that V and E are *finite* sets unless otherwise specified. We write $u \mathbin{—} v$ if u and v are vertices with $\{u,v\} \in E$, and $u \not\mathbin{—} v$ if u and v are vertices with $\{u,v\} \notin E$. Vertices with $u \mathbin{—} v$ are called "neighbors," and they're also said to be "adjacent" in G. One consequence of this definition is that we have $u \mathbin{—} v$ if and only if $v \mathbin{—} u$. Another consequence is that $v \not\mathbin{—} v$, for all $v \in V$; that is, no vertex is adjacent to itself. (We shall, however, discuss multigraphs below, in which loops from a vertex to itself are permitted.)

The graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. It's a *spanning* subgraph of G if, in fact, $V' = V$. And it's an *induced* subgraph of G if E' has as many edges as possible, when V' is a given subset of the vertices. In other words, when $V' \subseteq V$ the subgraph of $G = (V, E)$ induced by V' is $G' = (V', E')$, where

$$E' \;=\; \bigl\{\, \{u,v\} \bigm| u \in V',\ v \in V',\ \text{and } \{u,v\} \in E \,\bigr\}. \tag{15}$$

This subgraph G' is denoted by $G \mid V'$, and often called "G restricted to V'." In the common case where $V' = V \setminus \{v\}$, we write simply $G \setminus v$ ("G minus vertex v") as an abbreviation for $G \mid (V \setminus \{v\})$. The similar notation $G \setminus e$ is used when $e \in E$ to denote the subgraph $G' = (V, E \setminus \{e\})$, obtained by removing an edge instead of a vertex. Notice that all of the SGB graphs known as $\mathit{words}(n, l, t, s)$, described earlier, are induced subgraphs of the main graph $\mathit{words}(5757, 0, 0, 0)$; only the vocabulary changes in those graphs, not the rule for adjacency.

A graph with n vertices and e edges is said to have *order* n and *size* e. The simplest and most important graphs of order n are the *complete graph* K_n, the *path* P_n, and the *cycle* C_n. Suppose the vertices are $V = \{1, 2, \ldots, n\}$. Then

- K_n has $\binom{n}{2} = \frac{1}{2}n(n-1)$ edges $u \mathbin{—} v$ for $1 \le u < v \le n$; every n-vertex graph is a spanning subgraph of K_n.
- P_n has $n-1$ edges $v \mathbin{—} (v{+}1)$ for $1 \le v < n$, when $n \ge 1$; it is a path of length $n{-}1$ from 1 to n.
- C_n has n edges $v \mathbin{—} ((v \bmod n){+}1)$ for $1 \le v \le n$, when $n \ge 1$; it is a graph only when $n \ge 3$ (but C_1 and C_2 are multigraphs).

We could actually have defined K_n, P_n, and C_n on the vertices $\{0, 1, \ldots, n{-}1\}$, or on *any* n-element set V instead of $\{1, 2, \ldots, n\}$, because two graphs that differ only in the names of their vertices but not in the structure of their edges are combinatorially equivalent.

Formally, we say that graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if there is a one-to-one correspondence φ from V to V' such that $u \mathbin{—} v$ in G if and only if $\varphi(u) \longrightarrow \varphi(v)$ in G'. The notation $G \cong G'$ is often used to indicate that G and G' are isomorphic; but we shall often be less precise, by treating isomorphic graphs as if they were equal, and by occasionally writing $G = G'$ even when the vertex sets of G and G' aren't strictly identical.

Small graphs can be defined by simply drawing a diagram, in which the vertices are small circles and the edges are lines between them. Figure 2 illustrates several important examples, whose properties we will be studying later. The Petersen graph in Figure 2(e) is named after Julius Petersen, an early graph theorist who used it to disprove a plausible conjecture [L'*Intermédiaire des Mathématiciens* **5** (1898), 225–227]; it is, in fact, a remarkable configuration that serves as a counterexample to many optimistic predictions about what might be true for graphs in general. The Chvátal graph, Figure 2(f), was introduced by Václav Chvátal in *J. Combinatorial Theory* **9** (1970), 93–94.

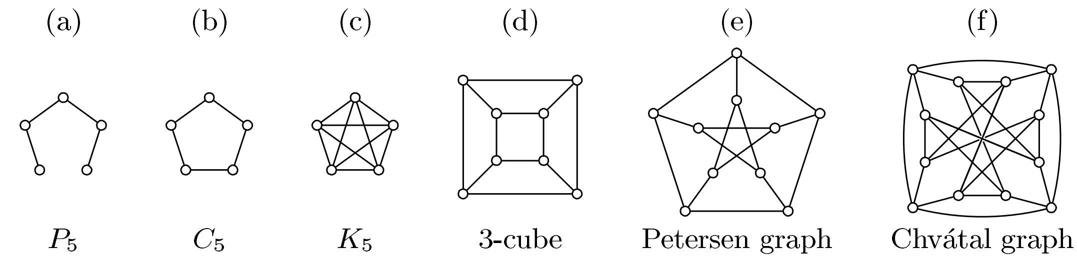
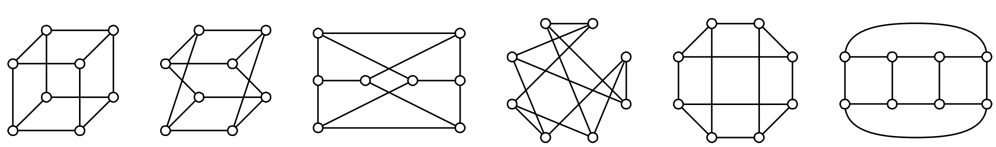


Fig. 2. Six example graphs, which have respectively $(5, 5, 5, 8, 10, 12)$ vertices and $(4, 5, 10, 12, 15, 24)$ edges.

The lines of a graph diagram are allowed to cross each other at points that aren't vertices. For example, the center point of Fig. 2(f) is *not* a vertex of Chvátal's graph. A graph is called *planar* if there's a way to draw it without any crossings. Clearly P_n and C_n are always planar; Fig. 2(d) shows that the 3-cube is also planar. But K_5 has too many edges to be planar (see exercise 46).

The *degree* of a vertex is the number of neighbors that it has. If all vertices have the same degree, the graph is said to be *regular*. In Fig. 2, for example, P_5 is irregular because it has two vertices of degree 1 and three of degree 2. But the other five graphs are regular, of degrees $(2, 4, 3, 3, 4)$ respectively. A regular graph of degree 3 is often called "cubic" or "trivalent."

There are many ways to draw a given graph, some of which are much more perspicuous than others. For example, each of the six diagrams

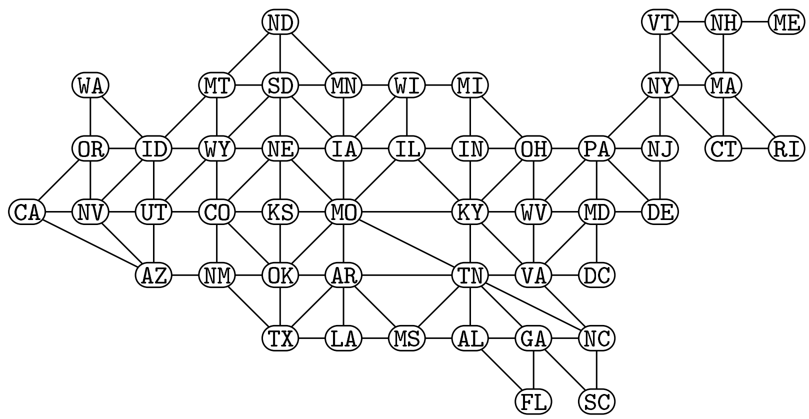


(16)

is isomorphic to the 3-cube, Fig. 2(d). The layout of Chvátal's graph that appears in Fig. 2(f) was discovered by Adrian Bondy many years after Chvátal's paper was published, thereby revealing unexpected symmetries.

The symmetries of a graph, also known as its *automorphisms*, are the permutations of its vertices that preserve adjacency. In other words, the permutation φ is an automorphism of G if we have $\varphi(u) \longrightarrow \varphi(v)$ whenever $u \longrightarrow v$ in G. A well-chosen drawing like Fig. 2(f) can reveal underlying symmetry, but a single diagram isn't always able to display all the symmetries that exist. For example, the 3-cube has 48 automorphisms, and the Petersen graph has 120. We'll study algorithms that deal with isomorphisms and automorphisms in Section 7.2.3. Symmetries can often be exploited to avoid unnecessary computations, making an algorithm almost k times faster when it operates on a graph that has k automorphisms.

Graphs that have evolved in the real world tend to be rather different from the mathematically pristine graphs of Figure 2. For example, here's a familiar graph that has no symmetry whatsoever, although it does have the virtue of being planar:



(17)

It represents the contiguous United States of America, and we'll be using it later in several examples. The 49 vertices of this diagram have been labeled with two-letter postal codes for convenience, instead of being reduced to empty circles.

Paths and cycles. A spanning path of a graph is called a *Hamiltonian path*, and a spanning cycle is called a *Hamiltonian cycle*, because W. R. Hamilton invented a puzzle in 1856 whose goal was to find such paths and cycles on the edges of a dodecahedron. T. P. Kirkman had independently studied the problem for polyhedra in general, in *Philosophical Transactions* **146** (1856), 413–418; **148** (1858), 145–161. [See *Graph Theory 1736–1936* by N. L. Biggs, E. K. Lloyd, and R. J. Wilson (1998), Chapter 2.] The task of finding a spanning path or cycle is, however, much older — indeed, we can legitimately consider it to be the oldest combinatorial problem of all, because paths and tours of a knight on a chessboard have a continuous history going back to ninth-century India (see Section 7.3.3). A graph is called *Hamiltonian* if it has a Hamiltonian cycle. (The Petersen graph, incidentally, is the smallest 3-regular graph that is neither planar nor Hamiltonian; see C. de Polignac, *Bull. Soc. Math. de France* **27** (1899), 142–145.)

The *girth* of a graph is the length of its shortest cycle; the girth is infinite if the graph is acyclic (containing no cycles). For example, the six graphs of Fig. 2 have girths $(\infty, 5, 3, 4, 5, 4)$, respectively. It's not difficult to prove that a graph of minimum degree k and girth 5 must have at least $k^2 + 1$ vertices. Further analysis shows in fact that this minimum value is achievable only if $k = 2$ (C_5), $k = 3$ (Petersen), $k = 7$, or perhaps $k = 57$. (See exercises 63 and 65.)

The *distance* $d(u,v)$ between two vertices u and v is the minimum length of a path from u to v in the graph; it is infinite if there's no such path. Clearly $d(v,v)=0$, and $d(u,v)=d(v,u)$. We also have the triangle inequality

$$d(u,v)+d(v,w) \;\geq\; d(u,w). \tag{18}$$

For if $d(u,v)=p$ and $d(v,w)=q$ and $p<\infty$ and $q<\infty$, there are paths

$$u=u_0 \mathbin{—} u_1 \mathbin{—} \cdots \mathbin{—} u_p=v \quad\text{and}\quad v=v_0 \mathbin{—} v_1 \mathbin{—} \cdots \mathbin{—} v_q=w, \tag{19}$$

and we can find the least subscript r such that $u_r=v_s$ for some s. Then

$$u_0 \mathbin{—} u_1 \mathbin{—} \cdots \mathbin{—} u_{r-1} \mathbin{—} v_s \mathbin{—} v_{s+1} \mathbin{—} \cdots \mathbin{—} v_q \tag{20}$$

is a path of length $\leq p+q$ from u to w.

The *diameter* of a graph is the maximum of $d(u,v)$, over all vertices u and v. The graph is *connected* if its diameter is finite. The vertices of a graph can always be partitioned into connected *components*, where two vertices u and v belong to the same component if and only if $d(u,v)<\infty$.

In the graph *words*$(5757,0,0,0)$, for example, we have $d(\texttt{tears},\texttt{smile})=6$, because (11) is a shortest path from `tears` to `smile`. Also $d(\texttt{tears},\texttt{happy})=6$, and $d(\texttt{smile},\texttt{happy})=10$, and $d(\texttt{world},\texttt{court})=6$. But $d(\texttt{world},\texttt{happy})=\infty$; the graph isn't connected. In fact, it contains 671 words like `aloof`, which have no neighbors and form connected components of order 1 all by themselves. Word pairs such as `alpha` — `aloha`, `droid` — `druid`, and `opium` — `odium` account for 103 further components of order 2. Some components of order 3, like `chain` — `chair` — `choir`, are paths; others, like $\{\texttt{getup},\texttt{letup},\texttt{setup}\}$, are cycles. A few more small components are also present, like the curious path

$$\texttt{login} \mathbin{—} \texttt{logic} \mathbin{—} \texttt{yogic} \mathbin{—} \texttt{yogis} \mathbin{—} \texttt{yogas} \mathbin{—} \texttt{togas}, \tag{21}$$

whose words have no other neighbors. But the vast majority of all five-letter words belong to a giant component of order 4493. If you can go two steps away from a given word, the odds are better than 15 to 1 that your word is connected to everything in the giant component.

Similarly, the graph *words*$(n,0,0,0)$ has a giant component of order $(3825, 2986, 2056, 1198, 224)$ when $n=(5000,4000,3000,2000,1000)$, respectively. But if n is small, there aren't enough edges to provide much connectivity. For example, *words*$(500,0,0,0)$ has 327 different components, none of order 15 or more.

The concept of distance can be generalized to $d(v_1,v_2,\ldots,v_k)$ for any value of k, meaning the minimum number of edges in a connected subgraph that contains the vertices $\{v_1,v_2,\ldots,v_k\}$. For example, $d(\texttt{blood},\texttt{sweat},\texttt{tears})$ turns out be 15, because the subgraph

$$\begin{array}{ccccccccccccc}
\texttt{blood} & — & \texttt{brood} & — & \texttt{broad} & — & \texttt{bread} & — & \texttt{tread} & — & \texttt{treed} & — & \texttt{tweed} \\
 & & & & & & & & & & | & & | \\
\texttt{tears} & — & \texttt{teams} & — & \texttt{trams} & — & \texttt{trims} & — & \texttt{tries} & — & \texttt{trees} & & \texttt{tweet} \\
 & & & & & & & & & & & & | \\
 & & & & & & & & & & \texttt{sweat} & — & \texttt{sweet}
\end{array} \tag{22}$$

has 15 edges, and there's no suitable 14-edge subgraph.

We noted in Section 2.3.4.1 that a connected graph with fewest edges is called a *free tree*. A subgraph that corresponds to the generalized distance $d(v_1, \ldots, v_k)$ will always be a free tree. It is misleadingly called a *Steiner tree,* because Jacob Steiner once mentioned the case $k = 3$ for points $\{v_1, v_2, v_3\}$ in the Euclidean plane [*Crelle* **13** (1835), 362–363]. Franz Heinen had solved that problem in *Über Systeme von Kräften* (1834); Gauss extended the analysis to $k = 4$ in a letter to Schumacher (21 March 1836).

Coloring. A graph is said to be *k-partite* or *k-colorable* if its vertices can be partitioned into k or fewer parts, with the endpoints of each edge belonging to different parts — or equivalently, if there's a way to paint its vertices with at most k different colors, never assigning the same color to two adjacent vertices. The famous Four Color Theorem, conjectured by F. Guthrie in 1852 and finally proved with massive computer aid by K. Appel, W. Haken, and J. Koch [*Illinois J. Math.* **21** (1977), 429–567], states that *every planar graph is 4-colorable*. No simple proof is known, but special cases like (17) can be colored at sight (see exercise 45); and $O(n^2)$ steps suffice to 4-color a planar graph in general [N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas, *STOC* **28** (1996), 571–575].

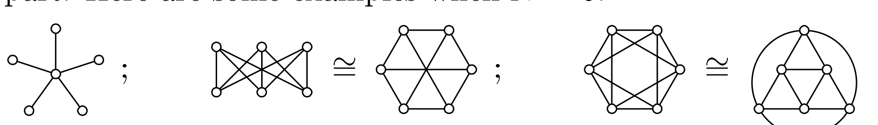
The case of 2-colorable graphs is especially important in practice. A 2-partite graph is generally called *bipartite*, or simply a "bigraph"; every edge of such a graph has one endpoint in each part.

Theorem B. *A graph is bipartite if and only if it contains no cycle of odd length.*

Proof. [See D. König, *Math. Annalen* **77** (1916), 453–454.] Every subgraph of a k-partite graph is k-partite. Therefore the cycle C_n can be a subgraph of a bipartite graph only if C_n itself is a bigraph, in which case n must be even.

Conversely, if a graph contains no odd cycles we can color its vertices with the two colors $\{0, 1\}$ by carrying out the following procedure: Begin with all vertices uncolored. If all neighbors of colored vertices are already colored, choose an uncolored vertex w, and color it 0. Otherwise choose a colored vertex u that has an uncolored neighbor v; assign to v the opposite color. Exercise 48 proves that a valid 2-coloring is eventually obtained. ∎

The *complete bipartite graph* $K_{m,n}$ is the largest bipartite graph whose vertices have two parts of sizes m and n. We can define it on the vertex set $\{1, 2, \ldots, m+n\}$ by saying that $u \frac{\quad}{} v$ whenever $1 \le u \le m < v \le m + n$. In other words, $K_{m,n}$ has mn edges, one for each way to choose one vertex in the first part and another in the second part. Similarly, the *complete k-partite graph* $K_{n_1,\ldots,n_k}$ has $N = n_1 + \cdots + n_k$ vertices partitioned into parts of sizes $\{n_1, \ldots, n_k\}$, and it has edges between any two vertices that don't belong to the same part. Here are some examples when $N = 6$:



(23)

$K_{1,5}$ $K_{3,3}$ $K_{2,2,2}$

Notice that $K_{1,n}$ is a free tree; it is popularly called the *star graph* of order $n+1$.

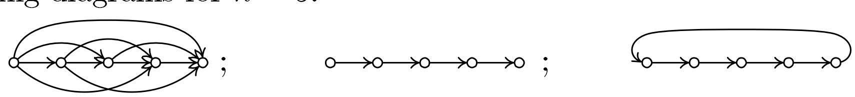
From now on say "digraph" instead of "directed graph."
It is clear and short and it will catch on.
— GEORGE POLYA, letter to Frank Harary (c 1954)

Directed graphs. In Section 2.3.4.2 we defined *directed graphs* (or *digraphs*), which are very much like graphs except that they have *arcs* instead of edges. An arc $u \longrightarrow v$ runs from one vertex to another, while an edge $u \mathrel{-\!\!-} v$ joins two vertices without distinguishing between them. Furthermore, digraphs are allowed to have self-loops $v \longrightarrow v$ from a vertex to itself, and more than one arc $u \longrightarrow v$ may be present between the same vertices u and v.

Formally, a digraph $D = (V, A)$ of order n and size m is a set V of n vertices and a multiset A of m ordered pairs (u, v), where $u \in V$ and $v \in V$. The ordered pairs are called arcs, and we write $u \longrightarrow v$ when $(u, v) \in A$. The digraph is called *simple* if A is actually a set instead of a general multiset—namely, if there's at most one arc (u, v) for all u and v. Each arc (u, v) has an initial vertex u and a final vertex v, also called its "tip." Each vertex has an *out-degree* $d^+(v)$, the number of arcs for which v is the initial vertex, and an *in-degree* $d^-(v)$, the number of arcs for which v is the tip. A vertex with in-degree 0 is called a "source"; a vertex with out-degree 0 is called a "sink." Notice that $\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v)$, because both sums are equal to m, the total number of arcs.

Most of the notions we've defined for graphs carry over to digraphs in a natural way, if we just insert the word "directed" or "oriented" (or the syllable "di") when it's necessary to distinguish between edges and arcs. For example, digraphs have subdigraphs, which can be spanning or induced or neither. An isomorphism between digraphs $D = (V, A)$ and $D' = (V', A')$ is a one-to-one correspondence φ from V to V' for which the number of arcs $u \longrightarrow v$ in D equals the number of arcs $\varphi(u) \longrightarrow \varphi(v)$ in D', for all $u, v \in V$.

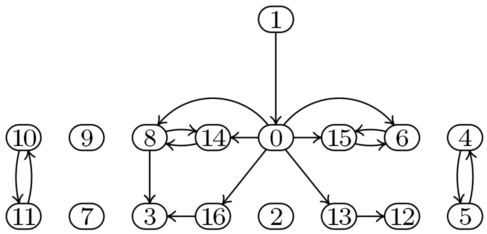
Diagrams for digraphs use arrows between the vertices, instead of unadorned lines. The simplest and most important digraphs of order n are directed variants of the graphs K_n, P_n, and C_n, namely the *transitive tournament* $K_n^{\rightarrow}$, the *oriented path* $P_n^{\rightarrow}$, and the *oriented cycle* $C_n^{\rightarrow}$. They can be schematically indicated by the following diagrams for $n = 5$:



$K_5^{\rightarrow}$; $P_5^{\rightarrow}$; $C_5^{\rightarrow}$. (24)

There's also the *complete digraph* J_n, which is the largest simple digraph on n vertices; it has n^2 arcs $u \longrightarrow v$, one for each choice of u and v.

Figure 3 shows a more elaborate diagram, for a digraph of order 17 that we might call "expressly oriented": It is the directed graph described by Hercule Poirot in Agatha Christie's novel *Murder on the Orient Express* (1934). Vertices correspond to the berths of the Stamboul–Calais coach in that story, and an arc $u \longrightarrow v$ means that the occupant of berth u has corroborated the alibi of the person in berth v. This example has six connected components, namely $\{0, 1, 3, 6, 8, 12, 13, 14, 15, 16\}$, $\{2\}$, $\{4, 5\}$, $\{7\}$, $\{9\}$, and $\{10, 11\}$, because connectivity in a digraph is determined by treating arcs as edges.



LEGEND
0: Pierre Michel, the French conductor
1: Hercule Poirot, the Belgian detective
2: Samuel Edward Ratchett, the deceased American
3: Caroline Martha Hubbard, the American matron
4: Edward Henry Masterman, the British valet
5: Antonio Foscarelli, the Italian automobile salesman
6: Hector MacQueen, the American secretary
7: Harvey Harris, the Englishman who didn't show up
8: Hildegarde Schmidt, the German lady's maid
9: (vacancy)
10: Greta Ohlsson, the Swedish nurse
11: Mary Hermione Debenham, the English governess
12: Helena Maria Andrenyi, the beautiful countess
13: Rudolph Andrenyi, the Hungarian count/diplomat
14: Natalia Dragomiroff, the Russian princess dowager
15: Colonel Arbuthnot, the British officer from India
16: Cyrus Bettman Hardman, the American detective

Fig. 3. A digraph of order 17 and size 18, devised by Agatha Christie.

Two arcs are *consecutive* if the tip of the first is the initial vertex of the second. A sequence of consecutive arcs $(a_1, a_2, \ldots, a_k)$ is called a *walk* of length k; it can be symbolized by showing the vertices as well as the arcs:

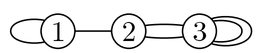
$$v_0 \xrightarrow{a_1} v_1 \xrightarrow{a_2} v_2 \cdots v_{k-1} \xrightarrow{a_k} v_k. \tag{25}$$

In a simple digraph it's sufficient merely to specify the vertices; for example, $1 \longrightarrow 0 \longrightarrow 8 \longrightarrow 14 \longrightarrow 8 \longrightarrow 3$ is a walk in Fig. 3. The walk in (25) is an oriented path when the vertices $\{v_0, v_1, \ldots, v_k\}$ are distinct; it's an oriented cycle when they are distinct except that $v_k = v_0$.

In a digraph, the directed distance $d(u, v)$ is the number of arcs in the shortest *oriented* path from u to v, which is also the length of the shortest walk from u to v. It may differ from $d(v, u)$; but the triangle inequality (18) remains valid.

Every graph can be regarded as a digraph, because an edge u — v is essentially equivalent to a matched pair of arcs, $u \longrightarrow v$ and $v \longrightarrow u$. The digraph obtained in this way retains all the properties of the original graph; for example, the degree of each vertex in the graph becomes its out-degree in the digraph, and also its in-degree in the digraph. Furthermore, distances remain the same.

A *multigraph* (V, E) is like a graph except that its edges E can be any *multiset* of pairs $\{u, v\}$; edges v — v that loop from a vertex to itself, which correspond to "multipairs" $\{v, v\}$, are also permitted. For example,



(26)

is a multigraph of order 3 with six edges, $\{1,1\}$, $\{1,2\}$, $\{2,3\}$, $\{2,3\}$, $\{3,3\}$, and $\{3,3\}$. The vertex degrees in this example are $d(1) = d(2) = 3$ and $d(3) = 6$, because each loop contributes 2 to the degree of its vertex. An edge loop v — v becomes *two* arc loops $v \longrightarrow v$ when a multigraph is regarded as a digraph.

Representation of graphs and digraphs. Any digraph, and therefore any graph or multigraph, is completely described by its *adjacency matrix* $A = (a_{uv})$, which has n rows and n columns when there are n vertices. Each entry a_{uv} of this matrix specifies the number of arcs from u to v. For example, the adjacency matrices for $K_3^{\rightarrow}$, $P_3^{\rightarrow}$, $C_3^{\rightarrow}$, J_3, and (26) are respectively

$$K_3^{\rightarrow} = \begin{pmatrix} 011 \\ 001 \\ 000 \end{pmatrix}, \quad P_3^{\rightarrow} = \begin{pmatrix} 010 \\ 001 \\ 000 \end{pmatrix}, \quad C_3^{\rightarrow} = \begin{pmatrix} 010 \\ 001 \\ 100 \end{pmatrix}, \quad J_3 = \begin{pmatrix} 111 \\ 111 \\ 111 \end{pmatrix}, \quad A = \begin{pmatrix} 210 \\ 102 \\ 024 \end{pmatrix}. \tag{27}$$

The powerful mathematical tools of matrix theory make it possible to prove many nontrivial results about graphs by studying their adjacency matrices; exercise 65 provides a particularly striking example of what can be done. One of the main reasons is that matrix multiplication has a simple interpretation in the context of digraphs. Consider the square of A, where the element in row u and column v is

$$(A^2)_{uv} \;=\; \sum_{w\in V} a_{uw}a_{wv}, \tag{28}$$

by definition. Since a_{uw} is the number of arcs from u to w, we see that $a_{uw}a_{wv}$ is the number of walks of the form $u \longrightarrow w \longrightarrow v$. Therefore $(A^2)_{uv}$ is the total number of walks of length 2 from u to v. Similarly, the entries of A^k tell us the total number of walks of length k between any ordered pair of vertices, for all $k \geq 0$. For example, the matrix A in (27) satisfies

$$A = \begin{pmatrix} 2 & 1 & 0\\ 1 & 0 & 2\\ 0 & 2 & 4 \end{pmatrix}, \qquad A^2 = \begin{pmatrix} 5 & 2 & 2\\ 2 & 5 & 8\\ 2 & 8 & 20 \end{pmatrix}, \qquad A^3 = \begin{pmatrix} 12 & 9 & 12\\ 9 & 18 & 42\\ 12 & 42 & 96 \end{pmatrix}; \tag{29}$$

there are 12 walks of length 3 from the vertex 1 of the multigraph (26) to vertex 3, and 18 such walks from vertex 2 to itself.

Reordering of the vertices changes an adjacency matrix from A to P^-AP, where P is a permutation matrix (a 0–1 matrix with exactly one 1 in each row and column), and $P^- = P^T$ is the matrix for the inverse permutation. Thus

$$\begin{pmatrix}210\\102\\024\end{pmatrix}, \quad \begin{pmatrix}201\\042\\120\end{pmatrix}, \quad \begin{pmatrix}012\\120\\204\end{pmatrix}, \quad \begin{pmatrix}021\\240\\102\end{pmatrix}, \quad \begin{pmatrix}402\\021\\210\end{pmatrix}, \quad \text{and} \quad \begin{pmatrix}420\\201\\012\end{pmatrix} \tag{30}$$

are all adjacency matrices for (26), and there are no others.

There are more than $2^{n(n-1)/2}/n!$ graphs of order n, when $n > 1$, and almost all of them require $\Omega(n^2)$ bits of data in their most economical encoding. Consequently the best way to represent the vast majority of all possible graphs inside a computer, from the standpoint of memory usage, is essentially to work with their adjacency matrices.

But the graphs that actually arise in practical problems have quite different characteristics from graphs that are chosen at random from the set of all possibilities. A real-life graph usually turns out to be "sparse," having say $O(n \log n)$ edges instead of $\Omega(n^2)$, unless n is rather small, because $\Omega(n^2)$ bits of data are difficult to generate. For example, suppose the vertices correspond to people, and the edges correspond to friendships. If we consider 5 billion people, few of them will have more than 10000 friends. But even if everybody had 10000 friends, on average, the graph would still have only 2.5×10^{13} edges, while almost all graphs of order 5 billion have approximately 6.25×10^{18} edges.

Thus the best way to represent a graph inside a machine usually turns out to be rather different than to record n^2 values a_{uv} of adjacency matrix elements. Instead, the algorithms of the Stanford GraphBase were developed with a data structure akin to the linked representation of sparse matrices discussed in Section 2.2.6, though somewhat simplified. That approach has proved to be not only versatile and efficient, but also easy to use.

The SGB representation of a digraph is a combination of sequential and linked allocation, using nodes of two basic types. Some nodes represent vertices, other nodes represent arcs. (There's also a third type of node, which represents an entire graph, for algorithms that deal with several graphs at once. But each graph needs only one graph node, so the vertex and arc nodes predominate.)

Here's how it works: Every SGB digraph of order n and size m is built upon a sequential array of n vertex nodes, making it easy to access vertex k for $0 \le k < n$. The m arc nodes, by contrast, are linked together within a general memory pool that is essentially unstructured. Each vertex node typically occupies 32 bytes, and each arc node occupies 20 (and the graph node occupies 220); but the node sizes can be modified without difficulty. A few fields of each node have a fixed, definite meaning in all cases; the remaining fields can be used for different purposes in different algorithms or in different phases of a single algorithm. The fixed-purpose parts of a node are called its "standard fields," and the multipurpose parts are called its "utility fields."

Every vertex node has two standard fields called `NAME` and `ARCS`. If v is a variable that points to a vertex node, we'll call it a *vertex variable.* Then `NAME`(v) points to a string of characters that can be used to identify the corresponding vertex in human-oriented output; for example, the 49 vertices of graph (17) have names like `CA`, `WA`, `OR`, ..., `RI`. The other standard field, `ARCS`(v), is far more important in algorithms: It points to an arc node, the first in a singly linked list of length $d^+(v)$, with one node for each arc that emanates from vertex v.

Every arc node has two standard fields called `TIP` and `NEXT`; a variable a that points to an arc node is called an *arc variable.* `TIP`(a) points to the vertex node that represents the tip of arc a; `NEXT`(a) points to the arc node that represents the next arc whose initial vertex agrees with that of a.

A vertex v with out-degree 0 is represented by letting `ARCS`$(v) = \Lambda$ (the null pointer). Otherwise if, say, the out-degree is 3, the data structure contains three arc nodes with `ARCS`$(v) = a_1$, `NEXT`$(a_1) = a_2$, `NEXT`$(a_2) = a_3$, and `NEXT`$(a_3) = \Lambda$; and the three arcs from v lead to `TIP`(a_1), `TIP`(a_2), `TIP`(a_3).

Suppose, for example, that we want to compute the out-degree of vertex v, and store it in a utility field called `ODEG`. It's easy:

$$\begin{aligned}&\text{Set } a \leftarrow \texttt{ARCS}(v) \text{ and } d \leftarrow 0.\\ &\text{While } a \ne \Lambda, \text{ set } d \leftarrow d+1 \text{ and } a \leftarrow \texttt{NEXT}(a).\\ &\text{Set } \texttt{ODEG}(v) \leftarrow d.\end{aligned} \qquad (31)$$

When a graph or a multigraph is considered to be a digraph, as mentioned above, its edges $u \mathbin{—} v$ are each equivalent to two arcs, $u \longrightarrow v$ and $v \longrightarrow u$. These arcs are called "mates"; and they occupy two arc nodes, say a and a', where a appears in the list of arcs from u and a' appears in the list of arcs from v. Then `TIP`$(a) = v$ and `TIP`$(a') = u$. We'll also write

$$\texttt{MATE}(a) = a' \qquad \text{and} \qquad \texttt{MATE}(a') = a, \qquad (32)$$

in algorithms that want to move rapidly from one list to another. However, we usually won't need to store an explicit pointer from an arc to its mate, or to have a utility field called MATE within each arc node, because the necessary link can be deduced *implicitly* when the data structure has been constructed cleverly.

The implicit-mate trick works like this: While creating each edge $u \text{ --- } v$ of an undirected graph or multigraph, we introduce *consecutive* arc nodes for $u \longrightarrow v$ and $v \longrightarrow u$. For example, if there are 20 bytes per arc node, we'll reserve 40 consecutive bytes for each new pair. We can also make sure that the memory address of the first byte is a multiple of 8. Then if the arc node a is in memory location α, its mate is in location

$$\left\{\begin{array}{ll}\alpha+20, & \text{if } \alpha \bmod 8 = 0\\ \alpha-20, & \text{if } \alpha \bmod 8 = 4\end{array}\right\} \;=\; \alpha - 20 + \bigl(40 \mathbin{\&} ((\alpha \mathbin{\&} 4) - 1)\bigr). \qquad (33)$$

Such tricks are valuable in combinatorial problems, when operations might be performed a trillion times, because every way to save 3.6 nanoseconds per operation will make such a computation finish an hour sooner. But (33) isn't directly "portable" from one implementation to another. If the size of an arc node were changed from 20 to 24, for example, we would have to change the numbers 40, 20, 8, and 4 in (33) to 48, 24, 16, and 8.

The algorithms in this book will make no assumptions about node sizes. Instead, we'll adopt a convention of the C programming language and its descendants, so that if a points to an arc node, '$a+1$' denotes a pointer to the arc node that follows it in memory. And in general

$$\mathtt{LOC(NODE(}a+k\mathtt{))} \;=\; \mathtt{LOC(NODE(}a\mathtt{))} + kc, \qquad (34)$$

when there are c bytes in each arc node. Similarly, if v is a vertex variable, '$v+k$' will stand for the kth vertex node following node v; the actual memory location of that node will be v plus k times the size of a vertex node.

The standard fields of a graph node g include M(g), the total number of arcs; N(g), the total number of vertices; VERTICES(g), a pointer to the first vertex node in the sequential list of all vertex nodes; ID(g), the graph's identification, which is a string like words(5757,0,0,0); and some other fields needed for the allocation and recycling of memory when the graph grows or shrinks, or for exporting a graph to external formats that interface with other users and other graph-manipulation systems. But we will rarely need to refer to any of these graph node fields, nor will it be necessary to give a complete description of SGB format here, since we shall describe almost all of the graph algorithms in this chapter by sticking to an English-language description at a fairly abstract level instead of descending to the bit level of machine programs.

A simple graph algorithm. To illustrate a medium-high-level algorithm of the kind that will appear later, let's convert the proof of Theorem B into a step-by-step procedure that paints the vertices of a given graph with two colors whenever that graph is bipartite.

Algorithm B (*Bipartiteness testing*). Given a graph represented in SGB format, this algorithm either finds a 2-coloring with COLOR(v) $\in \{0,1\}$ in each vertex v, or it terminates unsuccessfully when no valid 2-coloring is possible. Here COLOR is a utility field in each vertex node. Another vertex utility field, LINK(v), is a vertex pointer used to maintain a stack of all colored vertices whose neighbors have not yet been examined. An auxiliary vertex variable s points to the top of this stack. The algorithm also uses variables u, v, w for vertices and a for arcs. The vertex nodes are assumed to be $v_0 + k$ for $0 \le k < n$.

B1. [Initialize.] Set $\texttt{COLOR}(v_0 + k) \leftarrow -1$ for $0 \le k < n$. (Now all vertices are uncolored.) Then set $w \leftarrow v_0 + n$.

B2. [Done?] (At this point all vertices $\ge w$ have been colored, and so have the neighbors of all colored vertices.) Terminate the algorithm successfully if $w = v_0$. Otherwise set $w \leftarrow w - 1$, the next lower vertex node.

B3. [Color w if necessary.] If $\texttt{COLOR}(w) \ge 0$, return to B2. Otherwise set $\texttt{COLOR}(w) \leftarrow 0$, $\texttt{LINK}(w) \leftarrow \Lambda$, and $s \leftarrow w$.

B4. [Stack $\Rightarrow u$.] Set $u \leftarrow s$, $s \leftarrow \texttt{LINK}(s)$, $a \leftarrow \texttt{ARCS}(u)$. (We will examine all neighbors of the colored vertex u.)

B5. [Done with u?] If $a = \Lambda$, go to B8. Otherwise set $v \leftarrow \texttt{TIP}(a)$.

B6. [Process v.] If $\texttt{COLOR}(v) < 0$, set $\texttt{COLOR}(v) \leftarrow 1 - \texttt{COLOR}(u)$, $\texttt{LINK}(v) \leftarrow s$, and $s \leftarrow v$. Otherwise if $\texttt{COLOR}(v) = \texttt{COLOR}(u)$, terminate unsuccessfully.

B7. [Loop on a.] Set $a \leftarrow \texttt{NEXT}(a)$ and return to B5.

B8. [Stack nonempty?] If $s \ne \Lambda$, return to B4. Otherwise return to B2. ▮

This algorithm is a variant of a general graph traversal procedure called "depth-first search," which we will study in detail in Section 7.4.1. Its running time is $O(m + n)$ when there are m arcs and n vertices (see exercise 70); therefore it is well adapted to the common case of sparse graphs. With small changes we can make it output an odd-length cycle whenever it terminates unsuccessfully, thereby proving the impossibility of a 2-coloring (see exercise 72).

Examples of graphs. The Stanford GraphBase includes a library of more than three dozen generator routines, capable of producing a great variety of graphs and digraphs for use in experiments. We've already discussed *words*; now let's look at a few of the others, in order to get a feeling for some of the possibilities.

• *roget*$(1022, 0, 0, 0)$ is a directed graph with 1022 vertices and 5075 arcs. The vertices represent the categories of words or concepts that P. M. Roget and J. L. Roget included in their famous 19th-century *Thesaurus* (London: Longmans, Green, 1879). The arcs are the cross references between categories, as found in that book. For example, typical arcs are `water` ⟶ `moisture`, `discovery` ⟶ `truth`, `preparation` ⟶ `learning`, `vulgarity` ⟶ `ugliness`, `wit` ⟶ `amusement`.

• *book*(`"jean"`$, 80, 0, 1, 356, 0, 0, 0)$ is a graph with 80 vertices and 254 edges. The vertices represent the characters of Victor Hugo's *Les Misérables*; the edges connect characters who encounter each other in that novel. Typical edges are `Fantine` — `Javert`, `Cosette` — `Thénardier`.

• *bi_book*(`"jean"`$, 80, 0, 1, 356, 0, 0, 0)$ is a bipartite graph with 80+356 vertices and 727 edges. The vertices represent characters or chapters in *Les Misérables*; the edges connect characters with the chapters in which they appear (for instance, `Napoleon` — `2.1.8`, `Marius` — `4.14.4`).

• *plane_miles*$(128,0,0,0,1,0,0)$ is a planar graph with 129 vertices and 381 edges. The vertices represent 128 cities in the United States or Canada, plus a special vertex `INF` for a "point at infinity." The edges define the so-called *Delaunay triangulation* of those cities, based on latitude and longitude in a plane; this means that u — v if and only if there's a circle passing through u and v that does not enclose any other vertex. Edges also run between `INF` and all vertices that lie on the convex hull of all city locations. Typical edges are `Seattle, WA` — `Vancouver, BC` — `INF`; `Toronto, ON` — `Rochester, NY`.

• *plane_lisa*$(360,250,15,0,360,0,250,0,0,2295000)$ is a planar graph that has 3027 vertices and 5967 edges. It is obtained by starting with a digitized image of Leonardo da Vinci's *Mona Lisa*, having 360 rows and 250 columns of pixels, then rounding the pixel intensities to 16 levels of gray from 0 (black) to 15 (white). The resulting 3027 rookwise connected regions of constant brightness are then considered to be neighbors when they share a pixel boundary. (See Fig. 4.)

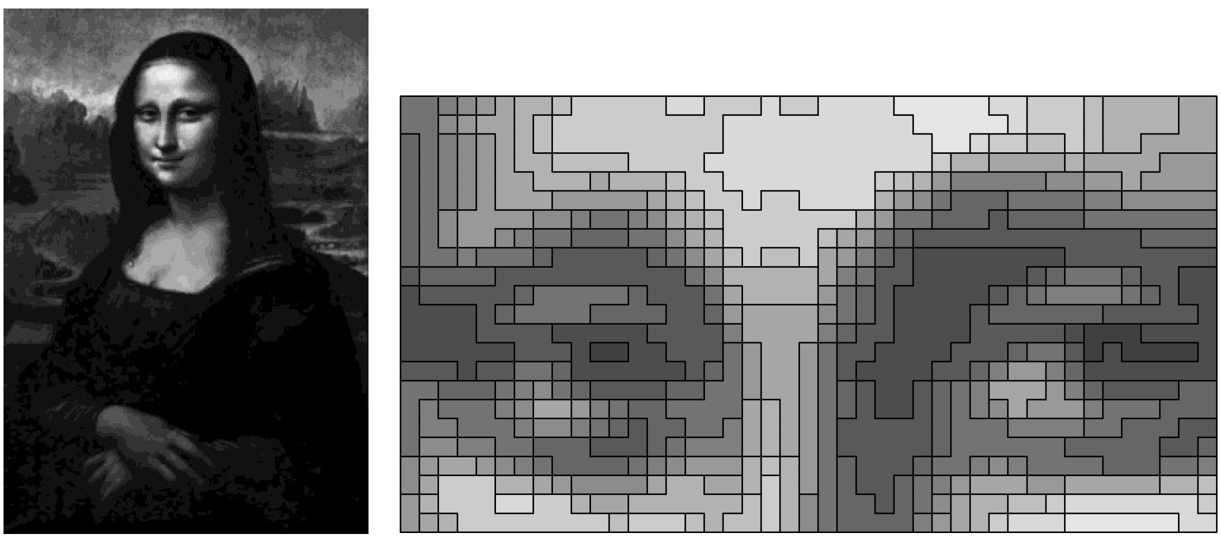


Fig. 4. A digital rendition of *Mona Lisa*, with a closeup detail (best viewed from afar).

• *bi_lisa*$(360,250,0,360,0,250,8192,0)$ is a bipartite graph with $360+250=610$ vertices and 40923 edges. It's another takeoff on Leonardo's famous painting, this time linking rows and columns where the brightness level is at least $1/8$. For example, the edge `r102` — `c113` occurs right in the middle of Lisa's "smile."

• *raman*$(31,23,3,1)$ is a graph with quite a different nature from the SGB graphs in previous examples. Instead of being linked to language, literature, or other outgrowths of human culture, it's a so-called "Ramanujan expander graph," based on strict mathematical principles. Each of its $(23^3-23)/2=6072$ vertices has degree 32; hence it has 97152 edges. The vertices correspond to equivalence classes of 2×2 matrices that are nonsingular modulo 23; a typical edge is `(2,7;1,1)` — `(4,6;1,3)`. Ramanujan graphs are important chiefly because they have unusually high girth and low diameter for their size and degree. This one has girth 4 and diameter 4.

• *raman*$(5, 37, 4, 1)$, similarly, is a regular graph of degree 6 with 50616 vertices and 151848 edges. It has girth 10, diameter 10, and happens also to be bipartite.

• *random_graph*$(1000, 5000, 0, 0, 0, 0, 0, 0, 0, s)$ is a graph with 1000 vertices, 5000 edges, and seed s. It "evolved" by starting with no edges, then by repeatedly choosing pseudorandom vertex numbers $0 \le u, v < 1000$ and adding the edge u — v, unless $u = v$ or that edge was already present. When $s = 0$, all vertices belong to a giant component of order 999, except for the isolated vertex 908.

• *random_graph*$(1000, 5000, 0, 0, 1, 0, 0, 0, 0, 0)$ is a digraph with 1000 vertices and 5000 arcs, obtained via a similar sort of evolution. (In fact, each of its arcs happens to be part also of *random_graph*$(1000, 5000, 0, 0, 0, 0, 0, 0, 0, 0)$.)

• *subsets*$(5, 1, -10, 0, 0, 0, \#\mathtt{1}, 0)$ is a graph with $\binom{11}{5} = 462$ vertices, one for every five-element subset of $\{0, 1, \ldots, 10\}$. Two vertices are adjacent whenever the corresponding subsets are disjoint; thus, the graph is regular of degree 6, and it has 1386 edges. We can consider it to be a generalization of the Petersen graph, which has *subsets*$(2, 1, -4, 0, 0, 0, \#\mathtt{1}, 0)$ as one of its SGB names.

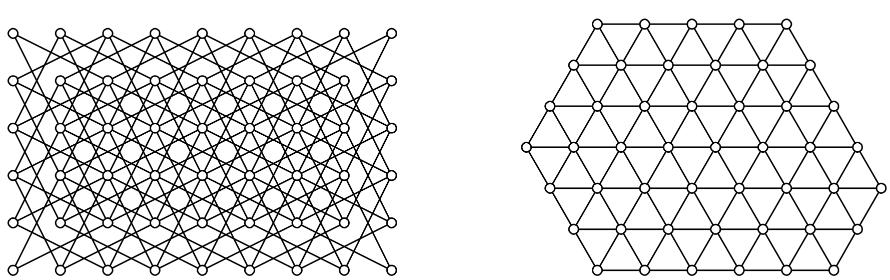
• *subsets*$(5, 1, -10, 0, 0, 0, \#\mathtt{10}, 0)$ has the same 462 vertices, but now they are adjacent if the corresponding subsets have four elements in common. This graph is regular of degree 30, and it has 6930 edges.

• *parts*$(30, 10, 30, 0)$ is another SGB graph with a mathematical basis. It has 3590 vertices, one for each partition of 30 into at most 10 parts. Two partitions are adjacent when one is obtained by subdividing a part of the other; this rule defines 31377 edges. The digraph *parts*$(30, 10, 30, 1)$ is similar, but its 31377 arcs point from shorter to longer partitions (for example, `13+7+7+3` $\longrightarrow$ `7+7+7+6+3`).

• *simplex*$(10, 10, 10, 10, 10, 0, 0)$ is a graph with 286 vertices and 1320 edges. Its vertices are the integer solutions to $x_1 + x_2 + x_3 + x_4 = 10$ with $x_i \ge 0$, namely the "compositions of 10 into four nonnegative parts"; they can also be regarded as barycentric coordinates for points inside a tetrahedron. The edges, such as `3,1,4,2` — `3,0,4,3`, connect compositions that are as close together as possible.

• *board*$(8, 8, 0, 0, 5, 0, 0)$ and *board*$(8, 8, 0, 0, -2, 0, 0)$ are graphs on 64 vertices whose 168 or 280 edges correspond to the moves of a knight or bishop in chess.

And zillions of further examples are obtainable by varying the parameters to the SGB graph generators. For example, Fig. 5 shows two simple variants of *board* and *simplex*; the somewhat arcane rules of *board* are explained in exercise 75.



board$(6, 9, 0, 0, 5, 0, 0)$
(Knight moves on a 6×9 chessboard)

simplex$(10, 8, 7, 6, 0, 0, 0)$
(A truncated triangular grid)

Fig. 5. Samples of SGB graphs related to board games.

Graph algebra. We can also obtain new graphs by operating on the graphs that we already have. For example, if $G = (V, E)$ is any graph, its *complement* $\overline{G} = (V, \overline{E})$ is obtained by letting

$$u \mathrel{-\!\!-} v \text{ in } \overline{G} \qquad \Longleftrightarrow \qquad u \neq v \text{ and } u \not\mathrel{-\!\!-} v \text{ in } G. \tag{35}$$

Thus, non-edges become edges, and vice versa. Notice that $\overline{\overline{G}} = G$, and that $\overline{K_n}$ has no edges. The corresponding adjacency matrices A and $\overline{A}$ satisfy

$$A + \overline{A} \;=\; J - I; \tag{36}$$

here J is the matrix of all 1s, and I is the identity matrix, so J and $J - I$ are respectively the adjacency matrices of J_n and K_n when G has order n.

Furthermore, every graph $G = (V, E)$ leads to a *line graph* $L(G)$, whose vertices are the edges E; two edges are adjacent in $L(G)$ if they have a common vertex. Thus, for example, the line graph $L(K_n)$ has $\binom{n}{2}$ vertices, and it is regular of degree $2n - 4$ when $n \geq 2$ (see exercise 82). A graph is called *k-edge-colorable* when its line graph is k-colorable.

Given two graphs $G = (U, E)$ and $H = (V, F)$, their *union* $G \cup H$ is the graph $(U \cup V, E \cup F)$ obtained by combining the vertices and edges. For example, suppose G and H are the graphs of rook and bishop moves in chess; then $G \cup H$ is the graph of queen moves, and its official SGB name is

$$\textit{gunion}(\textit{board}(8, 8, 0, 0, -1, 0, 0), \textit{board}(8, 8, 0, 0, -2, 0, 0), 0, 0). \tag{37}$$

In the special case where the vertex sets U and V are disjoint, the union $G \cup H$ doesn't require the vertices to be identified in any consistent way for cross-correlation; we get a diagram for $G \cup H$ by simply drawing a diagram of G next to a diagram of H. This special case is called the "juxtaposition" or *direct sum* of G and H, and we shall denote it by $G \oplus H$. For example, it's easy to see that

$$K_m \oplus K_n \;\cong\; \overline{K_{m,n}}, \tag{38}$$

and that every graph is the direct sum of its connected components.

Equation (38) is a special case of the general formula

$$K_{n_1} \oplus K_{n_2} \oplus \cdots \oplus K_{n_k} \;\cong\; \overline{K_{n_1, n_2, \ldots, n_k}}, \tag{39}$$

which holds for complete k-partite graphs whenever $k \geq 2$. But (39) fails when $k = 1$, because of a scandalous fact: The standard graph-theoretic notation for complete graphs is inconsistent! Indeed, $K_{m,n}$ denotes a complete 2-partite graph, but K_n does *not* denote a complete 1-partite graph. Somehow graph theorists have been able to live with this anomaly for decades without going berserk.

Another important way to combine disjoint graphs G and H is to form their *join*, $G \mathrel{-\!\!-} H$, which consists of $G \oplus H$ together with all edges $u \mathrel{-\!\!-} v$ for $u \in U$ and $v \in V$. [See A. A. Zykov, *Mat. Sbornik* **24** (1949), 163–188, §I.3.] And if G and H are disjoint *digraphs*, their *directed join* $G \longrightarrow H$ is similar, but it supplements $G \oplus H$ by adding only the one-way arcs $u \longrightarrow v$ from U to V.

The direct sum of two matrices A and B is obtained by placing B diagonally below and to the right of A:

$$A \oplus B \;=\; \begin{pmatrix} A & O \\ O & B \end{pmatrix}, \tag{40}$$

where each O in this example is a matrix of all zeros, with the proper number of rows and columns to make everything line up correctly. Our notation $G \oplus H$ for the direct sum of graphs is easy to remember because the adjacency matrix for $G \oplus H$ is precisely the direct sum of the respective adjacency matrices A and B for G and H. Similarly, the adjacency matrices for $G \mathbin{-\!\!-} H$, $G \longrightarrow H$, and $G \longleftarrow H$ are

$$A \mathbin{-\!\!-} B = \begin{pmatrix} A & J \\ J & B \end{pmatrix}, \qquad A \longrightarrow B = \begin{pmatrix} A & J \\ O & B \end{pmatrix}, \qquad A \longleftarrow B = \begin{pmatrix} A & O \\ J & B \end{pmatrix}, \tag{41}$$

respectively, where J is an all-1s matrix as in (36). These operations are associative, and related by complementation:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C, \qquad A \mathbin{-\!\!-} (B \mathbin{-\!\!-} C) = (A \mathbin{-\!\!-} B) \mathbin{-\!\!-} C; \tag{42}$$

$$A \longrightarrow (B \longrightarrow C) = (A \longrightarrow B) \longrightarrow C, \qquad A \longleftarrow (B \longleftarrow C) = (A \longleftarrow B) \longleftarrow C; \tag{43}$$

$$\overline{A \oplus B} = \overline{A} \mathbin{-\!\!-} \overline{B}, \qquad \overline{A \mathbin{-\!\!-} B} = \overline{A} \oplus \overline{B}; \tag{44}$$

$$\overline{A \longrightarrow B} = \overline{A} \longleftarrow \overline{B}, \qquad \overline{A \longleftarrow B} = \overline{A} \longrightarrow \overline{B}; \tag{45}$$

$$(A \oplus B) + (A \mathbin{-\!\!-} B) \;=\; (A \longrightarrow B) + (A \longleftarrow B). \tag{46}$$

Notice that, by combining (39) with (42) and (44), we have

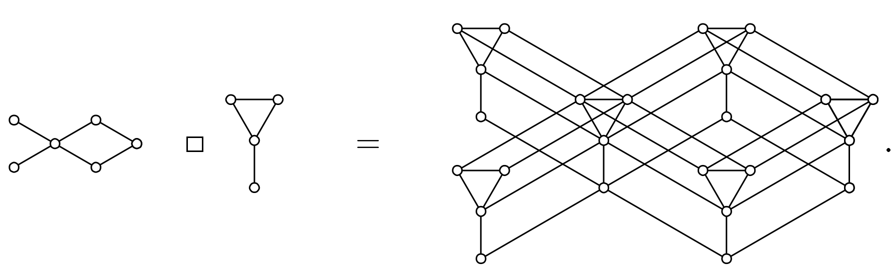
$$K_{n_1,n_2,\ldots,n_k} \;=\; \overline{K_{n_1}} \mathbin{-\!\!-} \overline{K_{n_2}} \mathbin{-\!\!-} \cdots \mathbin{-\!\!-} \overline{K_{n_k}} \tag{47}$$

when $k \ge 2$. Also

$$K_n = K_1 \mathbin{-\!\!-} K_1 \mathbin{-\!\!-} \cdots \mathbin{-\!\!-} K_1 \quad \text{and} \quad K_n^{\rightarrow} = K_1 \longrightarrow K_1 \longrightarrow \cdots \longrightarrow K_1\,, \tag{48}$$

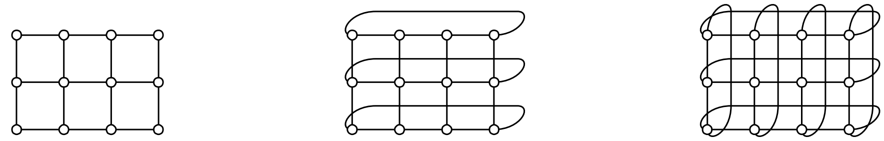
with n copies of K_1, showing that $K_n = K_{1,1,\ldots,1}$ is a complete n-partite graph.

Direct sums and joins are analogous to addition, because we have $\overline{K_m} \oplus \overline{K_n} = \overline{K_{m+n}}$ and $K_m \mathbin{-\!\!-} K_n = K_{m+n}$. We can also combine graphs with algebraic operations that are analogous to multiplication. For example, the *Cartesian product* operation forms a graph $G \,\square\, H$ of order mn from a graph $G = (U, E)$ of order m and a graph $H = (V, F)$ of order n. The vertices of $G \,\square\, H$ are ordered pairs (u, v), where $u \in U$ and $v \in V$; the edges are $(u, v) \mathbin{-\!\!-} (u', v)$ when $u \mathbin{-\!\!-} u'$ in G, together with $(u, v) \mathbin{-\!\!-} (u, v')$ when $v \mathbin{-\!\!-} v'$ in H. In other words, $G \,\square\, H$ is formed by replacing each vertex of G by a copy of H, and replacing each edge of G by edges between corresponding vertices of the appropriate copies:



(49)

As usual, the simplest special cases of this general construction turn out to be especially important in practice. When both G and H are paths or cycles, we get "graph-paper graphs," namely the $m \times n$ *grid* $P_m \,\square\, P_n$, the $m \times n$ *cylinder* $P_m \,\square\, C_n$, and the $m \times n$ *torus* $C_m \,\square\, C_n$, illustrated here for $m = 3$ and $n = 4$:



(50)

$P_3 \,\square\, P_4$ (3 × 4 grid) $\qquad$ $P_3 \,\square\, C_4$ (3 × 4 cylinder) $\qquad$ $C_3 \,\square\, C_4$ (3 × 4 torus)

Four other noteworthy ways to define products of graphs have also proved to be useful. In each case the vertices of the product graph are ordered pairs (u, v).

- The *direct product* $G \otimes H$, also called the "conjunction" of G and H, or their "categorical product," has $(u, v) -\!\!- (u', v')$ when $u -\!\!- u'$ in G and $v -\!\!- v'$ in H.
- The *strong product* $G \boxtimes H$ combines the edges of $G \,\square\, H$ with those of $G \otimes H$.
- The *odd product* $G \triangle H$ has $(u, v) -\!\!- (u', v')$ when we have either $u -\!\!- u'$ in G or $v -\!\!- v'$ in H, but not both.
- The *lexicographic product* $G \circ H$, also called the "composition" of G and H, has $(u, v) -\!\!- (u', v')$ when $u -\!\!- u'$ in G, and $(u, v) -\!\!- (u, v')$ when $v -\!\!- v'$ in H.

All five of these operations extend naturally to products of $k \geq 2$ graphs $G_1 = (V_1, E_1)$, ..., $G_k = (V_k, E_k)$, whose vertices are the ordered k-tuples $(v_1, \ldots, v_k)$ with $v_j \in V_j$ for $1 \leq j \leq k$. For example, when $k = 3$, the Cartesian products $G_1 \,\square\, (G_2 \,\square\, G_3)$ and $(G_1 \,\square\, G_2) \,\square\, G_3$ are isomorphic, if we consider the compound vertices $(v_1, (v_2, v_3))$ and $((v_1, v_2), v_3)$ to be the same as (v_1, v_2, v_3). Therefore we can write this Cartesian product without parentheses, as $G_1 \,\square\, G_2 \,\square\, G_3$. The most important example of a Cartesian product with k factors is the k-cube,

$$P_2 \,\square\, P_2 \,\square\, \cdots \,\square\, P_2; \tag{51}$$

its 2^k vertices $(v_1, \ldots, v_k)$ are adjacent when their Hamming distance is 1.

In general, suppose $v = (v_1, \ldots, v_k)$ and $v' = (v'_1, \ldots, v'_k)$ are k-tuples of vertices, where we have $v_j -\!\!- v'_j$ in G_j for exactly a of the subscripts j, and $v_j = v'_j$ for exactly b of the subscripts. Then we have:

- $v -\!\!- v'$ in $G_1 \,\square\, \cdots \,\square\, G_k$ if and only if $a = 1$ and $b = k - 1$;
- $v -\!\!- v'$ in $G_1 \otimes \cdots \otimes G_k$ if and only if $a = k$ and $b = 0$;
- $v -\!\!- v'$ in $G_1 \boxtimes \cdots \boxtimes G_k$ if and only if $a + b = k$ and $a > 0$;
- $v -\!\!- v'$ in $G_1 \triangle \cdots \triangle G_k$ if and only if a is odd.

The lexicographic product is somewhat different, because it isn't commutative; in $G_1 \circ \cdots \circ G_k$ we have $v -\!\!- v'$ for $v \neq v'$ if and only if $v_j -\!\!- v'_j$, where j is the minimum subscript with $v_j \neq v'_j$.

Exercises 91–102 explore some of the basic properties of graph products. See also the book *Product Graphs* by Wilfried Imrich and Sandi Klavzar (2000), which contains a comprehensive introduction to the general theory, including algorithms for factorization of a given graph into "prime" subgraphs.

***Graphical degree sequences.** A sequence $d_1d_2\ldots d_n$ of nonnegative integers is called *graphical* if there's at least one graph on vertices $\{1,2,\ldots,n\}$ such that vertex k has degree d_k. We can assume that $d_1 \ge d_2 \ge \cdots \ge d_n$. Clearly $d_1 < n$ in any such graph; and the sum $m = d_1 + d_2 + \cdots + d_n$ of any graphical sequence is always even, because it is twice the number of edges. Furthermore, it's easy to see that the sequence 3311 is not graphical; therefore graphical sequences must also satisfy additional conditions. What are they?

A simple way to decide if a given sequence $d_1d_2\ldots d_n$ is graphical, and to construct such a graph if one exists, was discovered by V. Havel [*Časopis pro Pěstování Matematiky* **80** (1955), 477–479]. We begin with an empty tableau, having d_k cells in row k; these cells represent "slots" into which we'll place the neighbors of vertex k in the constructed graph. Let c_j be the number of cells in column j; thus $c_1 \ge c_2 \ge \cdots$, and when $1 \le k \le n$ we have $c_j \ge k$ if and only if $d_k \ge j$. For example, suppose $n = 8$ and $d_1 \ldots d_8 = 55544322$; then

1					
2					
3					
4					
5					
6					
7					
8					

(52)

is the initial tableau, and we have $c_1 \ldots c_5 = 88653$. Havel's idea is to pair up vertex n with d_n of the highest-degree vertices. In this case, for example, we create the two edges $8 — 3$ and $8 — 2$, and the tableau takes the following form:

1					
2					8
3					8
4					
5					
6					
7					
8	2	3			

. (53)

(We don't want $8 — 1$, because the empty slots should continue to form a tableau shape; the cells of each column must be filled from the bottom up.) Next we set $n \leftarrow 7$ and create two further edges, $7 — 1$ and $7 — 5$. And then come three more, $6 — 4$, $6 — 3$, $6 — 2$, making the tableau almost half full:

1					7
2				6	8
3				6	8
4				6	
5				7	
6	2	3	4		
7	5	1			
8	2	3			

. (54)

We've reduced the problem to finding a graph with degree sequence $d_1 \ldots d_5 = 43333$; at this point we also have $c_1 \ldots c_4 = 5551$. The reader is encouraged to fill in the remaining blanks, before looking at the answer in exercise 103.

Algorithm H (*Graph generator for specified degrees*). Given $d_1 \geq \cdots \geq d_n \geq d_{n+1} = 0$, this algorithm creates edges between the vertices $\{1, \ldots, n\}$ in such a way that exactly d_k edges touch vertex k, for $1 \leq k \leq n$, unless the sequence $d_1 \ldots d_n$ isn't graphical. An array $c_1 \ldots c_{d_1}$ is used for auxiliary storage.

H1. [Set the c's.] Start with $k \leftarrow d_1$ and $j \leftarrow 0$. Then while $k > 0$ do the following operations: Set $j \leftarrow j + 1$; while $k > d_{j+1}$, set $c_k \leftarrow j$ and $k \leftarrow k - 1$. Terminate successfully if $j = 0$ (all d's are zero).

H2. [Find n.] Set $n \leftarrow c_1$. Terminate successfully if $n = 0$; terminate unsuccessfully if $d_1 \geq n > 0$.

H3. [Begin loop on j.] Set $i \leftarrow 1$, $t \leftarrow d_1$, $r \leftarrow c_t$, and $j \leftarrow d_n$.

H4. [Generate a new edge.] Set $c_j \leftarrow c_j - 1$ and $m \leftarrow c_t$. Create the edge $n \mathrel{-\!\!-} m$, and set $d_m \leftarrow d_m - 1$, $c_t \leftarrow m - 1$, $j \leftarrow j - 1$. If $j = 0$, return to step H2. Otherwise, if $m = i$, set $i \leftarrow r + 1$, $t \leftarrow d_i$, and $r \leftarrow c_t$ (see exercise 104); repeat step H4. ▮

When Algorithm H succeeds, it certainly has constructed a graph with the desired degrees. But when it fails, how can we be sure that its mission was impossible? The key fact is based on an important concept called "majorization": If $d_1 \ldots d_n$ and $d'_1 \ldots d'_n$ are two partitions of the same integer (that is, if $d_1 \geq \cdots \geq d_n$ and $d'_1 \geq \cdots \geq d'_n$ and $d_1 + \cdots + d_n = d'_1 + \cdots + d'_n$), we say that $d_1 \ldots d_n$ *majorizes* $d'_1 \ldots d'_n$ if $d_1 + \cdots + d_k \geq d'_1 + \cdots + d'_k$ for $1 \leq k \leq n$.

Lemma M. *If $d_1 \ldots d_n$ is graphical and $d_1 \ldots d_n$ majorizes $d'_1 \ldots d'_n$, then $d'_1 \ldots d'_n$ is also graphical.*

Proof. It is sufficient to prove the claim when $d_1 \ldots d_n$ and $d'_1 \ldots d'_n$ differ in only two places,

$$d'_k \;=\; d_k - [k\!=\!i] + [k\!=\!j] \qquad \text{where } i < j, \tag{55}$$

because any sequence majorized by $d_1 \ldots d_n$ can be obtained by repeatedly performing mini-majorizations such as this. (Exercise 7.2.1.4–55 discusses majorization in detail.)

Condition (55) implies that $d_i > d'_i \geq d'_{i+1} \geq d'_j > d_j$. So any graph with degree sequence $d_1 \ldots d_n$ contains a vertex v such that $v \mathrel{-\!\!-} i$ and $v \not\mathrel{-\!\!-} j$. Deleting the edge $v \mathrel{-\!\!-} i$ and adding the edge $v \mathrel{-\!\!-} j$ yields a graph with degree sequence $d'_1 \ldots d'_n$, as desired. ▮

Corollary H. *Algorithm H succeeds whenever $d_1 \ldots d_n$ is graphical.*

Proof. We may assume that $n > 1$. Suppose G is any graph on $\{1, \ldots, n\}$ with degree sequence $d_1 \ldots d_n$, and let G' be the subgraph induced by $\{1, \ldots, n-1\}$; in other words, obtain G' by removing vertex n and the d_n edges that it touches. The degree sequence $d'_1 \ldots d'_{n-1}$ of G' is obtained from $d_1 \ldots d_{n-1}$ by reducing some d_n of the entries by 1 and sorting them into nonincreasing order. By definition, $d'_1 \dots d'_{n-1}$ is graphical. The new degree sequence $d''_1 \dots d''_{n-1}$ produced by the strategy of steps H3 and H4 is designed to be majorized by every such $d'_1 \dots d'_{n-1}$, because it reduces the largest possible d_n entries by 1. Thus the new $d''_1 \dots d''_{n-1}$ is graphical. Algorithm H, which sets $d_1 \dots d_{n-1} \leftarrow d''_1 \dots d''_{n-1}$, will therefore succeed by induction on n. ▮

The running time of Algorithm H is roughly proportional to the number of edges generated, which can be of order n^2. Exercise 105 presents a faster method, which decides in $O(n)$ steps whether or not a given sequence $d_1 \dots d_n$ is graphical (without constructing any graph).

Beyond graphs. When the vertices and/or arcs of a graph or digraph are decorated with additional data, we call it a *network*. For example, every vertex of *words*$(5757, 0, 0, 0)$ has an associated rank, which corresponds to the popularity of the corresponding five-letter word. Every vertex of *plane_lisa*$(360, 250, 15, 0, 360, 0, 250, 0, 0, 2295000)$ has an associated pixel density, between 0 and 15. Every arc of *board*$(8, 8, 0, 0, -2, 0, 0)$ has an associated length, which reflects the distance of a piece's motion on the board: A bishop's move from corner to corner has length 7. The Stanford GraphBase includes several further generators that were not mentioned above, because they are primarily used to generate interesting networks, rather than to generate graphs with interesting structure:

• *miles*$(128, 0, 0, 0, 0, 127, 0)$ is a network with 128 vertices, corresponding to the same North American cities as the graph *plane_miles* described earlier. But *miles*, unlike *plane_miles*, is a complete graph with $\binom{128}{2}$ edges. Every edge has an integer length, which represents the distance that a car or truck would have needed to travel in 1949 when going from one given city to another. For example, '`Vancouver, BC`' is 3496 miles from '`West Palm Beach, FL`' in the *miles* network.

• *econ*$(81, 0, 0, 0)$ is a network with 81 vertices and 4902 arcs. Its vertices represent sectors of the United States economy, and its arcs represent the flow of money from one sector to another during the year 1985, measured in millions of dollars. For example, the flow value from `Apparel` to `Household furniture` is 44, meaning that the furniture industry paid \$44,000,000 to the apparel industry in that year. The sum of flows coming into each vertex is equal to the sum of flows going out. An arc appears only when the flow is nonzero. A special vertex called `Users` receives the flows that represent total demand for a product; a few of these end-user flows are negative, because of the way imported goods are treated by government economists.

• *games*$(120, 0, 0, 0, 0, 0, 0, 128, 0)$ is a network with 120 vertices and 1276 arcs. Its vertices represent football teams at American colleges and universities. Arcs run between teams that played each other during the exciting 1990 season, and they are labeled with the number of points scored. For example, the arc `Stanford` $\longrightarrow$ `California` has value 27, and the arc `California` $\longrightarrow$ `Stanford` has value 25, because the Stanford Cardinal defeated the U. C. Berkeley Golden Bears by a score of 27–25 on 17 November 1990.

• *risc*(16) is a network of an entirely different kind. It has 3240 vertices and 7878 arcs, which define a *directed acyclic graph* or "dag" —namely, a digraph that contains no oriented cycles. The vertices represent gates that have Boolean values; an arc such as `Z45` $\longrightarrow$ `R0:7~` means that the value of gate `Z45` is an input to gate `R0:7~`. Each gate has a type code (AND, OR, XOR, NOT, latch, or external input); each arc has a length, denoting an amount of delay. The network contains the complete logic for a miniature RISC chip that is able to obey simple commands governing sixteen registers, each 16 bits wide.

Complete details about all the SGB generators can be found in the author's book *The Stanford GraphBase* (New York: ACM Press, 1994), together with dozens of short example programs that explain how to manipulate the graphs and networks that the generators produce. For example, a program called LADDERS shows how to find a shortest path between one five-letter word and another. A program called TAKE_RISC demonstrates how to put a nanocomputer through its paces by simulating the actions of a network built from the gates of $risc(16)$.

Hypergraphs. Graphs and networks can be utterly fascinating, but they aren't the end of the story by any means. Lots of important combinatorial algorithms are designed to work with *hypergraphs*, which are more general than graphs because their edges are allowed to be *arbitrary* subsets of the vertices.

For example, we might have seven vertices, identified by nonzero binary strings $v = a_1a_2a_3$, together with seven edges, identified by bracketed nonzero binary strings $e = [b_1b_2b_3]$, with $v \in e$ if and only if $(a_1b_1+a_2b_2+a_3b_3) \bmod 2 = 0$. Each of these edges contains exactly three vertices:

$$\begin{gathered}[001] = \{010, 100, 110\}; \quad [010] = \{001, 100, 101\}; \quad [011] = \{011, 100, 111\}; \\ [100] = \{001, 010, 011\}; \quad [101] = \{010, 101, 111\}; \\ [110] = \{001, 110, 111\}; \quad [111] = \{011, 101, 110\}. \end{gathered} \tag{56}$$

And by symmetry, each vertex belongs to exactly three edges. (Edges that contain three or more vertices are sometimes called "hyperedges," to distinguish them from the edges of an ordinary graph. But it's OK to call them just "edges.")

A hypergraph is said to be *r-uniform* if every edge contains exactly r vertices. Thus (56) is a 3-uniform hypergraph, and a 2-uniform hypergraph is an ordinary graph. The complete r-uniform hypergraph $K_n^{(r)}$ has n vertices and $\binom{n}{r}$ edges.

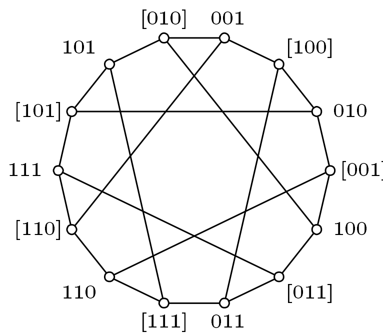
Most of the basic concepts of graph theory can be extended to hypergraphs in a natural way. For example, if $H = (V, E)$ is a hypergraph and if $U \subseteq V$, the subhypergraph $H \mid U$ induced by U has the edges $\{e \mid e \in E \text{ and } e \subseteq U\}$. The complement $\overline{H}$ of an r-uniform hypergraph has the edges of $K_n^{(r)}$ that aren't edges of H. A k-coloring of a hypergraph is an assignment of colors to the vertices so that no edge is monochromatic. And so on.

Hypergraphs go by many other names, because the same properties can be formulated in many different ways. For example, every hypergraph $H = (V, E)$ is essentially a *family of sets*, because each edge is a subset of V. A 3-uniform hypergraph is also called a *triple system*. A hypergraph is also equivalent to a matrix B of 0s and 1s, with one row for each vertex v and one column for each edge e; row v and column e of this matrix contains the value $b_{ve} = [v \in e]$.

Matrix B is called the *incidence matrix* of H, and we say that "v is incident with e" when $v \in e$. Furthermore, a hypergraph is equivalent to a *bipartite graph*, with vertex set $V \cup E$ and with the edge v — e whenever v is incident with e. The hypergraph is said to be *connected* if and only if the corresponding bipartite graph is connected. A *cycle* of length k in a hypergraph is defined to be a cycle of length $2k$ in the corresponding bipartite graph.

For example, the hypergraph (56) can be defined by an equivalent incidence matrix or an equivalent bipartite graph as follows:

$$\begin{array}{c} \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \begin{array}{c} \begin{array}{ccccccc} [001] & [010] & [011] & [100] & [101] & [110] & [111] \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$



(57)

It contains 28 cycles of length 3, such as

$$[101] — 101 — [010] — 001 — [100] — 010 — [101]. \tag{58}$$

The *dual* H^T of a hypergraph H is obtained by interchanging the roles of vertices and edges, but retaining the incidence relation. In other words, it corresponds to transposing the incidence matrix. Notice, for example, that the dual of an r-regular graph is an r-uniform hypergraph.

Incidence matrices and bipartite graphs might correspond to hypergraphs in which some edges occur more than once, because distinct columns of the matrix might be equal. When a hypergraph $H = (V, E)$ does not have any repeated edges, it corresponds also to yet another combinatorial object, namely a *Boolean function.* For if, say, the vertex set V is $\{1, 2, \ldots, n\}$, the function

$$h(x_1, x_2, \ldots, x_n) \;=\; \bigl[\{j \mid x_j = 1\} \in E\bigr] \tag{59}$$

characterizes the edges of H. For example, the Boolean formula

$$\begin{aligned} &(x_1 \oplus x_2 \oplus x_3) \wedge (x_2 \oplus x_4 \oplus x_6) \wedge (x_3 \oplus x_4 \oplus x_7) \\ &\qquad\qquad\qquad\qquad \wedge (x_3 \oplus x_5 \oplus x_6) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4) \end{aligned} \tag{60}$$

is another way to describe the hypergraph of (56) and (57).

The fact that combinatorial objects can be viewed in so many ways can be mind-boggling. But it's also extremely helpful, because it suggests different ways to solve equivalent problems. When we look at a problem from different perspectives, our brains naturally think of different ways to attack it. Sometimes we get the best insights by thinking about how to manipulate rows and columns in a matrix. Sometimes we make progress by imagining vertices and paths, or by visualizing clusters of points in space. Sometimes Boolean algebra is just the thing. If we're stuck in one domain, another might come to our rescue.

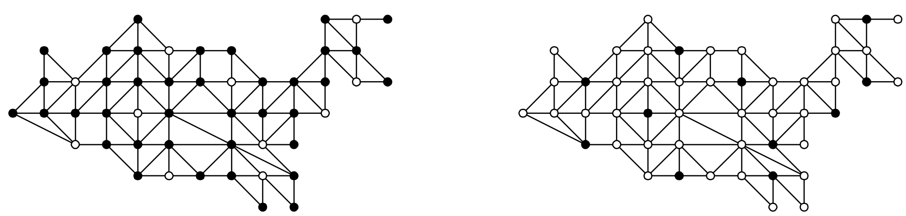
Covering and independence. If $H = (V, E)$ is a graph or hypergraph, a set U of vertices is said to *cover* H if every edge contains at least one member of U. A set W of vertices is said to be *independent* (or "stable") in H if no edge is completely contained in W.

From the standpoint of the incidence matrix, a covering is a set of rows whose sum is nonzero in every column. And in the special case that H is a graph, every column of the matrix contains just two 1s; hence an independent set in a graph corresponds to a set of rows that are mutually orthogonal — that is, a set for which the dot product of any two different rows is zero.

These concepts are opposite sides of the same coin. If U covers H, then $W = V \setminus U$ is independent in H; conversely, if W is independent in H, then $U = V \setminus W$ covers H. Both statements are equivalent to saying that the induced hypergraph $H \mid W$ has no edges.

This dual relationship between covering and independence, which was perhaps first noted by Claude Berge [*Proc. National Acad. Sci.* **43** (1957), 842–844], is somewhat paradoxical. Although it's logically obvious and easy to verify, it's also intuitively surprising. When we look at a graph and try to find a large independent set, we tend to have rather different thoughts from when we look at the same graph and try to find a small vertex cover; yet both goals are the same.

A covering set U is *minimal* if $U \setminus u$ fails to be a cover for all $u \in U$. Similarly, an independent set W is *maximal* if $W \cup w$ fails to be independent for all $w \notin W$. Here, for example, is a minimal cover of the 49-vertex graph of the contiguous United States, (17), and the corresponding maximal independent set:

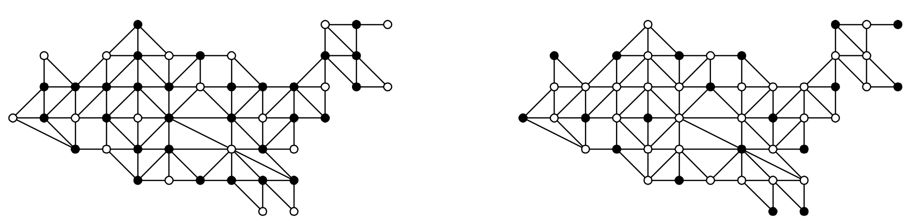


(61)

Minimal vertex cover, with 38 vertices

Maximal independent set, with 11 vertices

A covering is called *minimum* if it has the smallest possible size, and an independent set is called *maximum* if it has the largest possible size. For example, with graph (17) we can do much better than (61):



(62)

Minimum vertex cover, with 30 vertices

Maximum independent set, with 19 vertices

Notice the subtle distinction between "minimal" and "minimum" here: In general (but in contrast to most dictionaries of English), people who work with combinatorial algorithms use '-al' words like "minimal" or "optimal" to refer to combinatorial configurations that are *locally* best, in the sense that small changes don't improve them. The corresponding '-um' words, "minimum" or "optimum," are reserved for configurations that are *globally* best, considered over all possibilities. It's easy to find solutions to any optimization problem that are merely optimal, in the weak local sense, by climbing repeatedly until reaching the top of a hill. But it's usually much harder to find solutions that are truly optimum. For example, we'll see in Section 7.9 that the problem of finding a maximum independent set in a given graph belongs to a class of difficult problems that are called *NP-complete*.

Even when a problem is NP-complete, we needn't despair. We'll discuss techniques for finding minimum covers in several parts of this chapter, and those methods work fine on smallish problems; the optimum solution in (62) was found in less than a second, after examining only a tiny fraction of the 2^{49} possibilities. Furthermore, special cases of NP-complete problems often turn out to be simpler than the general case. In Section 7.5.1 we'll see that a minimum vertex cover can be discovered quickly in any bipartite graph, or in any hypergraph that is the dual of a graph. And in Section 7.5.5 we'll study efficient ways to discover a maximum *matching*, which is a maximum independent set in the line graph of a given graph.

The problem of maximizing the size of an independent set occurs sufficiently often that it has acquired a special notation: If H is any hypergraph, the number

$$\alpha(H) \;=\; \max\bigl\{|W| \bigm| W \text{ is an independent set of vertices in } H\bigr\} \qquad (63)$$

is called the *independence number* (or the stability number) of H. Similarly,

$$\chi(H) \;=\; \min\{k \mid H \text{ is } k\text{-colorable}\} \qquad (64)$$

is called the *chromatic number* of H. Notice that $\chi(H)$ is the size of a minimum covering of H by independent sets, because the vertices that receive any particular color must be independent according to our definitions.

These definitions of $\alpha(H)$ and $\chi(H)$ apply in particular to the case when H is an ordinary graph, but of course we usually write $\alpha(G)$ and $\chi(G)$ in such situations. Graphs have another important number called their *clique number*,

$$\omega(G) \;=\; \max\bigl\{|X| \bigm| X \text{ is a clique in } G\bigr\}, \qquad (65)$$

where a "clique" is a set of mutually adjacent vertices. Clearly

$$\omega(G) \;=\; \alpha(\overline{G}), \qquad (66)$$

because a clique in G is an independent set in the complementary graph. Similarly we can see that $\chi(\overline{G})$ is the minimum size of a "clique cover," which is a set of cliques that exactly covers all of the vertices.

Several instances of "exact cover problems" were mentioned earlier in this section, without an explanation of exactly what such a problem really signifies. Finally we're ready for the definition: Given the incidence matrix of a hypergraph H, an *exact cover* of H is a set of rows whose sum is $(1\,1\,\ldots\,1)$. In other words, an exact cover is a set of vertices that touches each hyperedge exactly once; an ordinary cover is only required to touch each hyperedge *at least* once.

EXERCISES

1. [*25*] Suppose $n = 4m - 1$. Construct arrangements of Langford pairs for the numbers $\{1, 1, \ldots, n, n\}$, with the property that we also obtain a solution for $n = 4m$ by changing the first '$2m{-}1$' to '$4m$' and appending '$2m{-}1\ 4m$' at the right. *Hint:* Put the $m - 1$ even numbers $4m{-}4$, $4m{-}6$, $\ldots$, $2m$ at the left.

2. [*20*] For which n can $\{0, 0, 1, 1, \ldots, n{-}1, n{-}1\}$ be arranged as Langford pairs?

3. [*22*] Suppose we arrange the numbers $\{0, 0, 1, 1, \ldots, n{-}1, n{-}1\}$ in a *circle*, instead of a straight line, with distance k between the two k's. Do we get solutions that are essentially distinct from those of exercise 2?

4. [*M20*] (T. Skolem, 1957.) Show that the Fibonacci string $S_\infty = babbababbabba\ldots$ of exercise 1.2.8–36 leads directly to an infinite sequence $0012132453674\ldots$ of Langford pairs for the set of *all* nonnegative integers, if we simply replace the a's and b's independently by 0, 1, 2, etc., from left to right.

▶ **5.** [*HM22*] If a permutation of $\{1, 1, 2, 2, \ldots, n, n\}$ is chosen at random, what is the probability that the two k's are exactly k positions apart, given k? Use this formula to guess the size of the Langford numbers L_n in (1).

▶ **6.** [*M28*] (M. Godfrey, 2002.) Let $f(x_1, \ldots, x_{2n}) = \prod_{k=1}^{n} \bigl(x_k x_{n+k} \sum_{j=1}^{2n-k-1} x_j x_{j+k+1}\bigr)$.

a) Prove that $\sum_{x_1,\ldots,x_{2n} \in \{-1,+1\}} f(x_1, \ldots, x_{2n}) = 2^{2n+1} L_n$.

b) Explain how to evaluate this sum in $O(4^n n)$ steps. How many bits of precision are needed for the arithmetic?

c) Gain a factor of eight by exploiting the identities

$$f(x_1, \ldots, x_{2n}) = f(-x_1, \ldots, -x_{2n}) = f(x_{2n}, \ldots, x_1) = f(x_1, -x_2, \ldots, x_{2n-1}, -x_{2n}).$$

7. [*M22*] Prove that every Langford pairing of $\{1, 1, \ldots, 16, 16\}$ must have seven uncompleted pairs at some point, when read from left to right.

8. [*23*] The simplest Langford sequence is not only well-balanced; it's *planar*, in the sense that its pairs can be connected up without crossing lines as in (2):

$$2\ 3\ 1\ 2\ 1\ 3\,.$$

Find all of the planar Langford pairings for which $n \le 8$.

9. [*24*] (*Langford triples.*) In how many ways can $\{1, 1, 1, 2, 2, 2, \ldots, 9, 9, 9\}$ be arranged in a row so that consecutive k's are k apart, for $1 \le k \le 9$?

10. [*M20*] Explain how to construct a *magic square* directly from Fig. 1. (Convert each card into a number between 1 and 16, in such a way that the rows, columns, and main diagonals all sum to 34.)

11. [*20*] Extend (5) to a "Hebraic-Græco-Latin" square by appending one of the letters {א, ב, ג, ד} to the two-letter string in each compartment. No letter pair (Latin, Greek), (Latin, Hebrew), or (Greek, Hebrew) should appear in more than one place.

▶ **12.** [*M21*] (L. Euler.) Let $L_{ij} = (i{+}j) \bmod n$ for $0 \le i, j < n$ be the addition table for integers mod n. Prove that a latin square orthogonal to L exists if and only if n is odd.

13. [*M25*] A 10×10 square can be divided into four quarters of size 5×5. A 10×10 latin square formed from the digits $\{0, 1, \ldots, 9\}$ has k "intruders" if its upper left quarter has exactly k elements ≥ 5. (See exercise 14(e) for an example with $k = 3$.) Prove that the square has no orthogonal mate unless there are at least three intruders.

14. [*29*] Find all orthogonal mates of the following latin squares:

(a)	(b)	(c)	(d)	(e)
3145926870	2718459036	0572164938	1680397425	7823456019
2819763504	0287135649	6051298473	8346512097	8234067195
9452307168	7524093168	4867039215	9805761342	2340178956
6208451793	1435962780	1439807652	2754689130	3401289567
8364095217	6390718425	8324756091	0538976214	4012395678
5981274036	4069271853	7203941586	4963820571	5678912340
4627530981	3102684597	5610473829	7192034658	6789523401
0576148329	9871546302	9148625307	6219405783	0195634782
1730689452	8956307214	2795380164	3471258906	1956740823
7093812645	5643820971	3986512740	5027143869	9567801234

15. [*50*] Find three 10×10 latin squares that are mutually orthogonal to each other.

16. [*48*] (H. J. Ryser, 1967.) A latin square is said to be of "order n" if it has n rows, n columns, and n symbols. Does every latin square of odd order have a transversal?

17. [*25*] Let L be a latin square with elements L_{ij} for $0 \le i, j < n$. Show that the problems of (a) finding all the transversals of L, and (b) finding all the orthogonal mates of L, are special cases of the general exact cover problem.

18. [*M26*] The string $x_1 x_2 \ldots x_N$ is called "n-ary" if each element x_j belongs to the set $\{0, 1, \ldots, n-1\}$ of n-ary digits. Two strings $x_1 x_2 \ldots x_N$ and $y_1 y_2 \ldots y_N$ are said to be *orthogonal* if the N pairs (x_j, y_j) are distinct for $1 \le j \le N$. (Consequently, two n-ary strings cannot be orthogonal if their length N exceeds n^2.) An n-ary matrix with m rows and n^2 columns whose rows are orthogonal to each other is called an *orthogonal array* of order n and depth m.

Find a correspondence between orthogonal arrays of depth m and lists of $m - 2$ mutually orthogonal latin squares. What orthogonal array corresponds to exercise 11?

▸ **19.** [*M25*] Continuing exercise 18, prove that an orthogonal array of order $n > 1$ and depth m is possible only if $m \le n + 1$. Show that this upper limit is achievable when n is a prime number p. Write out an example when $p = 5$.

20. [*HM20*] Show that if each element k in an orthogonal array is replaced by $e^{2\pi k i/n}$, the rows become orthogonal vectors in the usual sense (their dot product is zero).

▸ **21.** [*M21*] A *geometric net* is a system of points and lines that obeys three axioms:

i) Each line is a set of points.
ii) Distinct lines have at most one point in common.
iii) If p is a point and L is a line with $p \notin L$, then there is exactly one line M such that $p \in M$ and $L \cap M = \emptyset$.

If $L \cap M = \emptyset$ we say that L is *parallel* to M, and write $L \parallel M$.

a) Prove that the lines of a geometric net can be partitioned into equivalence classes, with two lines in the same class if and only if they are equal or parallel.
b) Show that if there are at least two classes of parallel lines, every line contains the same number of points as the other lines in its class.
c) Furthermore, if there are at least three classes, there are numbers m and n such that all points belong to exactly m lines and all lines contain exactly n points.

▸ **22.** [*M22*] Show that every orthogonal array can be regarded as a geometric net. Is the converse also true?

23. [*M23*] (*Error-correcting codes.*) The "Hamming distance" $d(x, y)$ between two strings $x = x_1 \ldots x_N$ and $y = y_1 \ldots y_N$ is the number of positions j where $x_j \ne y_j$. A *b-ary code with n information digits and r check digits* is a set $C(b,n,r)$ of b^n strings $x = x_1 \ldots x_{n+r}$, where $0 \le x_j < b$ for $1 \le j \le n+r$. When a codeword x is transmitted and the message y is received, $d(x,y)$ is the number of transmission errors. The code is called *t-error correcting* if we can reconstruct the value of x whenever a message y is received with $d(x,y) \le t$. The *distance* of the code is the minimum value of $d(x,x')$, taken over all pairs of codewords $x \ne x'$.

a) Prove that a code is t-error correcting if and only if its distance exceeds $2t$.
b) Prove that a single-error correcting b-ary code with 2 information digits and 2 check digits is equivalent to a pair of orthogonal latin squares of order b.
c) Furthermore, a code $C(b,2,r)$ with distance $r{+}1$ is equivalent to a set of r mutually orthogonal latin squares of order b.

▸ **24.** [*M30*] A geometric net with N points and R lines leads naturally to the binary code $C(2,N,R)$ with codewords $x_1 \ldots x_N x_{N+1} \ldots x_{N+R}$ defined by the parity bits

$$x_{N+k} = f_k(x_1,\ldots,x_N) = \bigl(\sum\{x_j \mid \text{point } j \text{ lies on line } k\}\bigr) \bmod 2.$$

a) If the net has m classes of parallel lines, prove that this code has distance $m+1$.
b) Find an efficient way to correct up to t errors with this code, assuming that $m = 2t$. Illustrate the decoding process in the case $N = 25$, $R = 30$, $t = 3$.

25. [*27*] Find a latin square whose rows and columns are five-letter words. (For this exercise you'll need to dig out the big dictionaries.)

▸ **26.** [*25*] Compose a meaningful English sentence that contains only five-letter words.

27. [*20*] How many SGB words contain exactly k distinct letters, for $1 \le k \le 5$?

28. [*20*] Are there any pairs of SGB word vectors that differ by ± 1 in each component?

29. [*20*] Find all SGB words that are *palindromes* (equal to their reflection), or mirror pairs (like `regal` `lager`).

▸ **30.** [*20*] The letters of `first` are in alphabetic order from left to right. What is the lexicographically *first* such five-letter word? What is the last?

31. [*21*] (C. McManus.) Find all sets of three SGB words that are in arithmetic progression but have no common letters in any fixed position. (One such example is $\{$`power`, `slugs`, `visit`$\}$.)

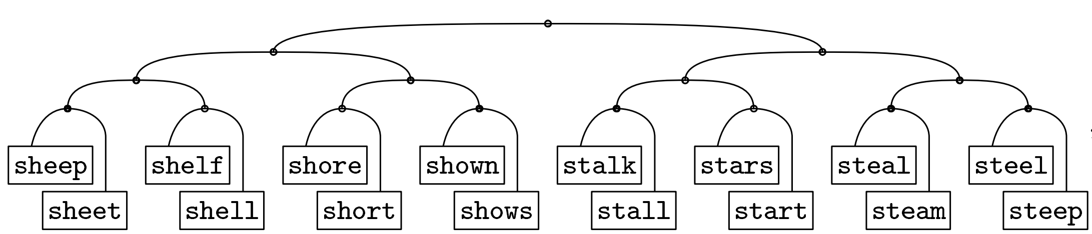
32. [*23*] Does the English language contain any 10-letter words $a_0 a_1 \ldots a_9$ for which both $a_0 a_2 a_4 a_6 a_8$ and $a_1 a_3 a_5 a_7 a_9$ are SGB words?

33. [*20*] (Scot Morris.) Complete the following list of 26 interesting SGB words:

`about`, `bacon`, `faced`, `under`, `chief`, ..., `pizza`.

▸ **34.** [*21*] For each SGB word that doesn't include the letter `y`, obtain a 5-bit binary number by changing the vowels $\{$`a`, `e`, `i`, `o`, `u`$\}$ to 1 and the other letters to 0. What are the most common words for each of the 32 binary outcomes?

▸ **35.** [*26*] Sixteen well-chosen elements of `WORDS`(1000) lead to the branching pattern



,

which is a complete binary trie of words that begin with the letter `s`. But there's no such pattern of words beginning with `a`, even if we consider the full collection `WORDS`(5757).

What letters of the alphabet can be used as the starting letter of sixteen words that form a complete binary trie within `WORDS`(n), given n?

36. [*M17*] Explain the symmetries that appear in the word cube (10). Also show that two more such cubes can be obtained by changing only the two words {`stove`, `event`}.

37. [*20*] Which vertices of the graph *words*(5757, 0, 0, 0) have maximum degree?

38. [*22*] Using the digraph rule in (14), change `tears` to `smile` in just three steps, *without computer assistance.*

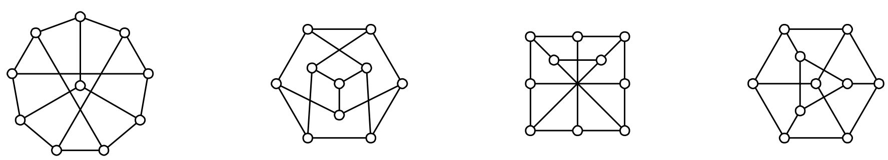
39. [*M00*] Is $G \setminus e$ an induced subgraph of G? Is it a spanning subgraph?

40. [*M15*] How many (a) spanning (b) induced subgraphs does a graph $G = (V, E)$ have, when $|V| = n$ and $|E| = e$?

41. [*M10*] For which integers n do we have (a) $K_n = P_n$? (b) $K_n = C_n$?

42. [*15*] (D. H. Lehmer.) Let G be a graph with 13 vertices, in which every vertex has degree 5. Make a nontrivial statement about G.

43. [*23*] Are any of the following graphs the same as the Petersen graph?



44. [*M23*] How many symmetries does Chvátal's graph have? (See Fig. 2(f).)

45. [*20*] Find an easy way to 4-color the planar graph (17). Would 3 colors suffice?

46. [*M25*] Let G be a graph with $n \ge 3$ vertices, defined by a planar diagram that is "maximal," in the sense that no additional lines can be drawn between nonadjacent vertices without crossing an existing edge.

a) Prove that the diagram partitions the plane into regions that each have exactly three vertices on their boundary. (One of these regions is the set of all points that lie outside the diagram.)

b) Therefore G has exactly $3n - 6$ edges.

47. [*M22*] Prove that the complete bigraph $K_{3,3}$ isn't planar.

48. [*M25*] Complete the proof of Theorem B by showing that the stated procedure never gives the same color to two adjacent vertices.

49. [*18*] Draw diagrams of all the cubic graphs with at most 6 vertices.

50. [*M24*] Find all bipartite graphs that can be 3-colored in exactly 24 ways.

▸ **51.** [*M22*] Given a geometric net as described in exercise 21, construct the bipartite graph whose vertices are the points p and the lines L of the net, with $p \longrightarrow L$ if and only if $p \in L$. What is the *girth* of this graph?

52. [*M16*] Find a simple inequality that relates the diameter of a graph to its girth. (How small can the diameter be, if the girth is large?)

53. [*15*] Which of the words `world` and `happy` belongs to the giant component of the graph *words*(5757, 0, 0, 0)?

▶ **54.** [*21*] The 49 postal codes in graph (17) are `AL`, `AR`, `AZ`, `CA`, `CO`, `CT`, `DC`, `DE`, `FL`, `GA`, `IA`, `ID`, `IL`, `IN`, `KS`, `KY`, `LA`, `MA`, `MD`, `ME`, `MI`, `MN`, `MO`, `MS`, `MT`, `NC`, `ND`, `NE`, `NH`, `NJ`, `NM`, `NV`, `NY`, `OH`, `OK`, `OR`, `PA`, `RI`, `SC`, `SD`, `TN`, `TX`, `UT`, `VA`, `VT`, `WA`, `WI`, `WV`, `WY`, in alphabetical order.

a) Suppose we consider two states to be adjacent if their postal codes agree in one place (namely `AL` — `AR` — `OR` — `OH`, etc.). What are the components of this graph?

b) Now form a directed graph with `XY` $\longrightarrow$ `YZ` (for example, `AL` $\longrightarrow$ `LA` $\longrightarrow$ `AR`, etc.). What are the *strongly connected components* of this digraph? (See Section 2.3.4.2.)

c) The United States has additional postal codes `AA`, `AE`, `AK`, `AP`, `AS`, `FM`, `GU`, `HI`, `MH`, `MP`, `PW`, `PR`, `VI`, besides those in (17). Reconsider question (b), using all 62 codes.

55. [*M20*] How many edges are in the complete k-partite graph $K_{n_1,\ldots,n_k}$?

▶ **56.** [*M10*] True or false: A multigraph is a graph if and only if the corresponding digraph is simple.

57. [*M10*] True or false: Vertices u and v are in the same connected component of a directed graph if and only if either $d(u,v) < \infty$ or $d(v,u) < \infty$.

58. [*M17*] Describe all (a) graphs (b) multigraphs that are regular of degree 2.

▶ **59.** [*M23*] A *tournament* of order n is a digraph on n vertices that has exactly $\binom{n}{2}$ arcs, either $u \longrightarrow v$ or $v \longrightarrow u$ for every pair of distinct vertices $\{u, v\}$.

a) Prove that every tournament contains an oriented spanning path $v_1 \longrightarrow \quad \longrightarrow v_n$.

b) Consider the tournament on vertices $\{0, 1, 2, 3, 4\}$ for which $u \longrightarrow v$ if and only if $(u - v) \bmod 5 \geq 3$. How many oriented spanning paths does it have?

c) Is $K_n^{\to}$ the only tournament of order n that has a unique oriented spanning path?

▶ **60.** [*M22*] Let u be a vertex of greatest out-degree in a tournament, and let v be any other vertex. Prove that $d(u, v) \leq 2$.

61. [*M16*] Construct a digraph that has k walks of length k from vertex 1 to vertex 2.

62. [*M21*] A *permutation digraph* is a directed graph in which every vertex has out-degree 1 and in-degree 1; therefore its components are oriented cycles. If it has n vertices and k components, we call it *even* if $n - k$ is even, *odd* if $n - k$ is odd.

a) Let G be a directed graph with adjacency matrix A. Prove that the number of spanning permutation digraphs of G is $\operatorname{per} A$, the permanent of A.

b) Interpret the determinant, $\det A$, in terms of spanning permutation digraphs.

63. [*M23*] Let G be a graph of girth g in which every vertex has at least d neighbors. Prove that G has at least N vertices, where

$$N = \begin{cases} 1 + \sum_{0 \leq k < t} d(d-1)^k, & \text{if } g = 2t + 1; \\ 1 + (d-1)^t + \sum_{0 \leq k < t} d(d-1)^k, & \text{if } g = 2t + 2. \end{cases}$$

▶ **64.** [*M21*] Continuing exercise 63, show that there's a *unique* graph of girth 4, minimum degree d, and order $2d$, for each $d \geq 2$.

▶ **65.** [*HM31*] Suppose graph G has girth 5, minimum degree d, and $N = d^2 + 1$ vertices.

a) Prove that the adjacency matrix A of G satisfies the equation $A^2 + A = (d-1)I + J$.

b) Since A is a symmetric matrix, it has N orthogonal eigenvectors x_j, with corresponding eigenvalues λ_j, such that $Ax_j = \lambda_j x_j$ for $1 \leq j \leq N$. Prove that each λ_j is either d or $(-1 \pm \sqrt{4d-3})/2$.

c) Show that if $\sqrt{4d-3}$ is irrational, then $d = 2$. *Hint:* $\lambda_1 + \quad + \lambda_N = \operatorname{trace}(A) = 0$.

d) And if $\sqrt{4d-3}$ is rational, $d \in \{3, 7, 57\}$.

66. [*M30*] Continuing exercise 65, construct such a graph when $d = 7$.

67. [*M48*] Is there a regular graph of degree 57, order 3250, and girth 5?

68. [*M20*] How many different adjacency matrices does a graph G on n vertices have?

▸ **69.** [*20*] Extending (31), explain how to calculate both out-degree `ODEG(`v`)` and in-degree `IDEG(`v`)` for *all* vertices v in a graph that has been represented in SGB format.

▸ **70.** [*M20*] How often is each step of Algorithm B performed, when that algorithm successfully 2-colors a graph with m arcs and n vertices?

71. [*26*] Implement Algorithm B for the `MMIX` computer, using the `MMIXAL` assembly language. Assume that, when your program begins, register `v0` points to the first vertex node and register `n` contains the number of vertices.

▸ **72.** [*M22*] When `COLOR(`v`)` is set in step B6, call u the *parent* of v; but when `COLOR(`w`)` is set in step B3, say that w has no parent. Define the (inclusive) *ancestors* of vertex v, recursively, to be v together with the ancestors of v's parent (if any).

a) Prove that if v is below u in the stack during Algorithm B, the parent of v is an ancestor of u.

b) Furthermore, if `COLOR(`v`)` $=$ `COLOR(`u`)` in step B6, v is currently in the stack.

c) Use these facts to extend Algorithm B so that, if the given graph is not bipartite, the names of vertices in a cycle of odd length are output.

73. [*15*] What's another name for $\mathit{random\_graph}(10,45,0,0,0,0,0,0,0,0)$?

74. [*21*] What vertex of $\mathit{roget}(1022,0,0,0)$ has the largest out-degree?

75. [*22*] The SGB graph generator $\mathit{board}(n_1,n_2,n_3,n_4,p,w,o)$ creates a graph whose vertices are the t-dimensional integer vectors $(x_1,\ldots,x_t)$ for $0 \le x_i < b_i$, determined by the first four parameters (n_1,n_2,n_3,n_4) as follows: Set $n_5 \leftarrow 0$ and let $j \ge 0$ be minimum such that $n_{j+1} \le 0$. If $j=0$, set $b_1 \leftarrow b_2 \leftarrow 8$ and $t \leftarrow 2$; this is the default 8×8 board. Otherwise if $n_{j+1}=0$, set $b_i \leftarrow n_i$ for $1 \le i \le j$ and $t \leftarrow j$. Finally, if $n_{j+1}<0$, set $t \leftarrow |n_{j+1}|$, and set b_i to the ith element of the periodic sequence $(n_1,\ldots,n_j, n_1,\ldots,n_j,n_1,\ldots)$. (For example, the specification $(n_1,n_2,n_3,n_4)=(2,3,5,-7)$ is about as tricky as you can get; it produces a 7-dimensional board with $(b_1,\ldots,b_7)=(2,3,5,2,3,5,2)$, hence a graph with $2\ \ 3\ \ 5\ \ 2\ \ 3\ \ 5\ \ 2=1800$ vertices.)

The remaining parameters (p,w,o), for "piece, wrap, and orientation," determine the arcs of the graph. Suppose first that $w=o=0$. If $p>0$, we have $(x_1,\ldots,x_t)\longrightarrow(y_1,\ldots,y_t)$ if and only if $y_i = x_i+\delta_i$ for $1\le i\le t$, where $(\delta_1,\ldots,\delta_t)$ is an integer solution to the equation $\delta_1^2 + \quad + \delta_t^2 = |p|$. And if $p<0$, we allow also $y_i = x_i + k\delta_i$ for $k \ge 1$, corresponding to k moves in the same direction.

If $w \ne 0$, let $w=(w_t\ldots w_1)_2$ in binary notation. Then we allow "wraparound," $y_i=(x_i+\delta_i) \bmod b_i$ or $y_i = (x_i+k\delta_i) \bmod b_i$, in each coordinate i for which $w_i=1$.

If $o \ne 0$, the graph is directed; offsets $(\delta_1,\ldots,\delta_t)$ produce arcs only when they are lexicographically greater than $(0,\ldots,0)$. But if $o=0$, the graph is undirected.

Find settings of (n_1,n_2,n_3,n_4,p,w,o) for which *board* will produce the following fundamental graphs: (a) the complete graph K_n; (b) the path P_n; (c) the cycle C_n; (d) the transitive tournament $\vec{K_n}$; (e) the oriented path $\vec{P_n}$; (f) the oriented cycle $\vec{C_n}$; (g) the $m\times n$ grid $P_m \,\square\, P_n$; (h) the $m\times n$ cylinder $P_m \,\square\, C_n$; (i) the $m\times n$ torus $C_m \,\square\, C_n$; (j) the $m\times n$ rook graph $K_m \,\square\, K_n$; (k) the $m\times n$ directed torus $\vec{C_m} \,\square\, \vec{C_n}$; (l) the null graph $\overline{K_n}$; (m) the n-cube $P_2 \,\square \quad \square\, P_2$ with 2^n vertices.

76. [*20*] Can $\mathit{board}(n_1,n_2,n_3,n_4,p,w,o)$ produce loops, or parallel (repeated) edges?

77. [*M20*] If graph G has diameter ≥ 3, prove that $\overline{G}$ has diameter ≤ 3.

78. [*M27*] Let $G = (V, E)$ be a graph with $|V| = n$ and $G \cong \overline{G}$. (In other words, G is *self-complementary*: There's a permutation φ of V such that $u \longrightarrow v$ if and only if $\varphi(u) \not\longrightarrow \varphi(v)$ and $u \neq v$. We can imagine that the edges of K_n have been painted black or white; the white edges define a graph that's isomorphic to the graph of black edges.)

a) Prove that $n \bmod 4 = 0$ or 1. Draw diagrams for all such graphs with $n < 8$.
b) Prove that if $n \bmod 4 = 0$, every cycle of the permutation φ has a length that is a multiple of 4.
c) Conversely, every permutation φ with such cycles arises in some such graph G.
d) Extend these results to the case $n \bmod 4 = 1$.

▶ **79.** [*M22*] Given $k \geq 0$, construct a graph on the vertices $\{0, 1, \ldots, 4k\}$ that is both regular and self-complementary.

▶ **80.** [*M22*] A self-complementary graph must have diameter 2 or 3, by exercise 77. Given $k \geq 2$, construct self-complementary graphs of both possible diameters, when (a) $V = \{1, 2, \ldots, 4k\}$; (b) $V = \{0, 1, 2, \ldots, 4k\}$.

81. [*20*] The complement of a simple digraph without loops is defined by extending (35) and (36), so that we have $u \to v$ in $\overline{D}$ if and only if $u \neq v$ and $u \not\to v$ in D. What are the self-complementary digraphs of order 3?

82. [*M21*] Are the following statements about line graphs true or false?

a) If G is contained in G', then $L(G)$ is an induced subgraph of $L(G')$.
b) If G is a regular graph, so is $L(G)$.
c) $L(K_{m,n})$ is regular, for all $m, n > 0$.
d) $L(K_{m,n,r})$ is regular, for all $m, n, r > 0$.
e) $L(K_{m,n}) \cong K_m \,\square\, K_n$.
f) $L(K_4) \cong K_{2,2,2}$.
g) $L(P_{n+1}) \cong P_n$.
h) The graphs G and $L(G)$ both have the same number of components.

83. [*16*] Draw the graph $\overline{L(K_5)}$.

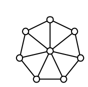
▶ **84.** [*M21*] Is $L(K_{3,3})$ self-complementary?

85. [*M22*] (O. Ore, 1962.) For which graphs G do we have $G \cong L(G)$?

86. [*M20*] (R. J. Wilson.) Find a graph G of order 6 for which $\overline{G} \cong L(G)$.

87. [*20*] Is the Petersen graph (a) 3-colorable? (b) 3-edge-colorable?

88. [*M20*] The graph $W_n = K_1 \longrightarrow C_{n-1}$ is called the *wheel* of order n, when $n \geq 4$. How many cycles does it contain as subgraphs?



W_8

89. [*M20*] Prove the associative laws, (42) and (43).

▶ **90.** [*M24*] A graph is called a *cograph* if it can be constructed algebraically from 1-element graphs by means of complementation and/or direct sum operations. For example, there are four nonisomorphic graphs of order 3, and they all are cographs: $\overline{K_3} = K_1 \oplus K_1 \oplus K_1$ and its complement, K_3; $\overline{K_{1,2}} = K_1 \oplus K_2$ and its complement, $K_{1,2}$, where $K_2 = \overline{K_1 \oplus K_1}$.

Exhaustive enumeration shows that there are 11 nonisomorphic graphs of order 4. Give algebraic formulas to prove that 10 of them are cographs. Which one isn't?

▶ **91.** [*20*] Draw diagrams for the 4-vertex graphs (a) $K_2 \square K_2$; (b) $K_2 \otimes K_2$; (c) $K_2 \boxtimes K_2$; (d) $K_2 \triangle K_2$; (e) $K_2 \circ K_2$; (f) $\overline{K_2} \circ K_2$; (g) $K_2 \circ \overline{K_2}$.

92. [*21*] The five types of graph products defined in the text work fine for simple digraphs as well as for ordinary graphs. Draw diagrams for the 4-vertex digraphs (a) $\vec{K}_2 \,\square\, \vec{K}_2$; (b) $\vec{K}_2 \otimes \vec{K}_2$; (c) $\vec{K}_2 \boxtimes \vec{K}_2$; (d) $\vec{K}_2 \triangle \vec{K}_2$; (e) $\vec{K}_2 \circ \vec{K}_2$.

93. [*15*] Which of the five graph products takes K_m and K_n into K_{mn}?

94. [*10*] Are the SGB *words* graphs induced subgraphs of $P_{26} \mathbin{\square} P_{26} \mathbin{\square} P_{26} \mathbin{\square} P_{26} \mathbin{\square} P_{26}$?

95. [*M20*] If vertex u of G has degree d_u and vertex v of H has degree d_v, what is the degree of vertex (u, v) in (a) $G \mathbin{\square} H$? (b) $G \otimes H$? (c) $G \boxtimes H$? (d) $G \mathbin{\triangle} H$? (e) $G \circ H$?

▶ **96.** [*M22*] Let A be an $m \times m'$ matrix with $a_{uu'}$ in row u and column u'; let B be an $n \times n'$ matrix with $b_{vv'}$ in row v and column v'. The *direct product* $A \otimes B$ is an $mn \times m'n'$ matrix with $a_{uu'}b_{vv'}$ in row (u, v) and column (u', v'). Thus $A \otimes B$ is the adjacency matrix of $G \otimes H$, if A and B are the adjacency matrices of G and H.

Find analogous formulas for the adjacency matrices of (a) $G \mathbin{\square} H$; (b) $G \boxtimes H$; (c) $G \mathbin{\triangle} H$; (d) $G \circ H$.

97. [*M25*] Find as many interesting algebraic relations between graph sums and products as you can. (For example, the distributive law $(A \oplus B) \otimes C = (A \otimes C) \oplus (B \otimes C)$ for direct sums and products of matrices implies that $(G \oplus G') \otimes H = (G \otimes H) \oplus (G' \otimes H)$. We also have $\overline{K_m} \mathbin{\square} H = H \oplus \quad \oplus H$, with m copies of H, etc.)

98. [*M20*] If the graph G has k components and the graph H has l components, how many components are in the graphs $G \mathbin{\square} H$ and $G \boxtimes H$?

99. [*M20*] Let $d_G(u, u')$ be the distance from vertex u to vertex u' in graph G. Prove that $d_{G \square H}((u, v), (u', v')) = d_G(u, u') + d_H(v, v')$, and find a similar formula for $d_{G \boxtimes H}((u, v), (u', v'))$.

100. [*M21*] For which connected graphs is $G \otimes H$ connected?

▶ **101.** [*M25*] Find all connected graphs G and H such that $G \mathbin{\square} H \cong G \otimes H$.

102. [*M20*] What's a simple algebraic formula for the graph of *king moves* (which take one step horizontally, vertically, or diagonally) on an $m \times n$ board?

103. [*20*] Complete tableau (54). Also apply Algorithm H to the sequence 866444444.

104. [*18*] Explain the manipulation of variables i, t, and r in steps H3 and H4.

105. [*M38*] Suppose $d_1 \geq \quad \geq d_n \geq 0$, and let $c_1 \geq \quad \geq c_{d_1}$ be its conjugate as in Algorithm H. Prove that $d_1 \ldots d_n$ is graphical if and only if $d_1 + \quad + d_n$ is even and $d_1 + \quad + d_k \leq c_1 + \quad + c_k - k$ for $1 \leq k \leq s$, where s is maximal such that $d_s \geq s$.

106. [*20*] True or false: If $d_1 = \quad = d_n = d < n$ and nd is even, Algorithm H constructs a *connected* graph.

107. [*M21*] Prove that the degree sequence $d_1 \ldots d_n$ of a self-complementary graph satisfies $d_j + d_{n+1-j} = n - 1$ and $d_{2j-1} = d_{2j}$ for $1 \leq j \leq n/2$.

▶ **108.** [*M23*] Design an algorithm analogous to Algorithm H that constructs a *simple directed graph* on vertices $\{1, \ldots, n\}$, having specified values d_k^- and d_k^+ for the in-degree and out-degree of each vertex k, whenever at least one such graph exists.

109. [*M20*] Design an algorithm analogous to Algorithm H that constructs a *bipartite graph* on vertices $\{1, \ldots, m + n\}$, having specified degrees d_k for each vertex k when possible; all edges $j \text{---} k$ should have $j \leq m$ and $k > m$.

110. [*M22*] Without using Algorithm H, show by a direct construction that the sequence $d_1 \ldots d_n$ is graphical when $n > d_1 \geq \quad \geq d_n \geq d_1 - 1$ and $d_1 + \quad + d_n$ is even.

▶ **111.** [*25*] Let G be a graph on vertices $V = \{1, \ldots, n\}$, with d_k the degree of k and $\max(d_1, \ldots, d_n) = d$. Prove that there's an integer N with $n \leq N \leq 2n$ and a graph H on vertices $\{1, \ldots, N\}$, such that H is regular of degree d and $H \mid V = G$. Explain how to construct such a regular graph with N as small as possible.

▶ **112.** [*20*] Does the network *miles*$(128,0,0,0,0,127,0)$ have three equidistant cities? If not, what three cities come closest to an equilateral triangle?

113. [*05*] When H is a hypergraph with m edges and n vertices, how many rows and columns does its incidence matrix have?

114. [*M20*] Suppose the multigraph (26) is regarded as a hypergraph. What is the corresponding incidence matrix? What is the corresponding bipartite graph?

▶ **115.** [*M20*] When B is the incidence matrix of a graph G, explain the significance of the symmetric matrices B^TB and BB^T.

116. [*M17*] Describe the edges of the complete bipartite r-uniform hypergraph $K_{m,n}^{(r)}$.

117. [*M22*] How many nonisomorphic 1-uniform hypergraphs have m edges and n vertices? (Edges may be repeated.) List them all when $m=4$ and $n=3$.

118. [*M20*] A "hyperforest" is a hypergraph that contains no cycles. If a hyperforest has m edges, n vertices, and p components, what's the sum of the degrees of its vertices?

119. [*M18*] What hypergraph corresponds to (60) without the final term $(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4)$?

120. [*M20*] Define *directed hypergraphs*, by generalizing the concept of directed graphs.

121. [*M19*] Given a hypergraph $H=(V,E)$, let $I(H)=(V,F)$, where F is the family of all maximal independent sets of H. Express $\chi(H)$ in terms of $|V|$, $|F|$, and $\alpha(I(H)^T)$.

▶ **122.** [*M24*] Find a maximum independent set and a minimum coloring of the following triple systems: (a) the hypergraph (56); (b) the dual of the Petersen graph.

123. [*17*] Show that the optimum colorings of $K_n \,\square\, K_n$ are equivalent to the solutions of a famous combinatorial problem.

124. [*M22*] What is the chromatic number of the Chvátal graph, Fig. 2(f)?

125. [*M48*] For what values of g is there a 4-regular, 4-chromatic graph of girth g?

▶ **126.** [*M22*] Find optimum colorings of the "kingwise torus," $C_m \boxtimes C_n$, when $m,n \geq 3$.

127. [*M22*] Prove that (a) $\chi(G)+\chi(\overline{G}) \leq n+1$ and (b) $\chi(G)\chi(\overline{G}) \geq n$ when G is a graph of order n, and find graphs for which equality holds.

128. [*M18*] Express $\chi(G \,\square\, H)$ in terms of $\chi(G)$ and $\chi(H)$, when G and H are graphs.

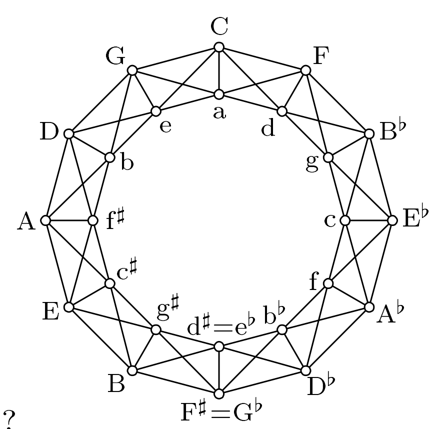
129. [*23*] Describe the maximal cliques of the 8×8 queen graph (37).

130. [*M20*] How many maximal cliques are in a complete k-partite graph?

131. [*M30*] Let $N(n)$ be the largest number of maximal cliques that an n-vertex graph can have. Prove that $3^{\lfloor n/3 \rfloor} \leq N(n) \leq 3^{\lceil n/3 \rceil}$.

▶ **132.** [*M20*] We call G *tightly colorable* if $\chi(G)=\omega(G)$. Prove that $\chi(G \boxtimes H) = \chi(G)\chi(H)$ whenever G and H are tightly colorable.

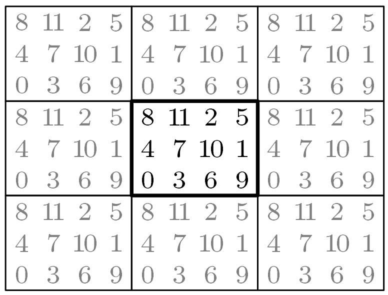
133. [*21*] The "musical graph" illustrated here provides a nice way to review numerous definitions that were given in this section, because its properties are easily analyzed. Determine its (a) order; (b) size; (c) girth; (d) diameter; (e) independence number, $\alpha(G)$; (f) chromatic number, $\chi(G)$; (g) edge-chromatic number, $\chi(L(G))$; (h) clique number, $\omega(G)$; (i) algebraic formula as a product of well-known smaller graphs. What is the size of (j) a minimum vertex cover? (k) a maximum matching? Is G (l) regular? (m) planar? (n) connected? (o) directed? (p) a free tree? (q) Hamiltonian?

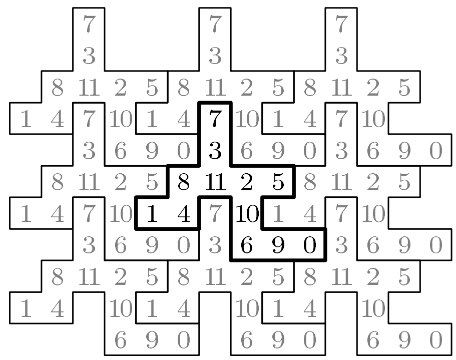


134. [*M22*] How many automorphisms does the musical graph have?

▸ **135.** [*HM26*] Suppose a composer takes a random walk in the musical graph, starting at vertex C and then making five equally likely choices at each step. Show that after an even number of steps, the walk is more likely to end at vertex C than at any other vertex. What is the exact probability of going from C to C in a 12-step walk?

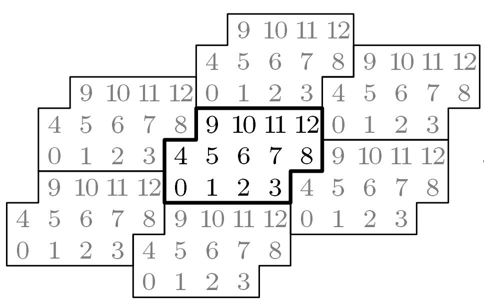
136. [*HM23*] A *Cayley digraph* is a directed graph whose vertices V are the elements of a group and whose arcs are $v \longrightarrow v\alpha_j$ for $1 \le j \le d$ and all vertices v, where $(\alpha_1, \ldots, \alpha_d)$ are fixed elements of the group. A *Cayley graph* is a Cayley digraph that is also a graph. Is the Petersen graph a Cayley graph?





▸ **137.** [*M25*] (*Generalized toruses.*) An $m \times n$ torus can be regarded as a tiling of the plane. For example, we can imagine that infinitely many copies of the 3×4 torus in (50) have been placed together gridwise, as indicated in the left-hand illustration above; from each vertex we can move north, south, east, or west to another vertex of the torus. The vertices have been numbered here so that a northward move from v goes to $(v+4) \bmod 12$, and an eastward move to $(v+3) \bmod 12$, etc. The right-hand illustration shows the same torus, but with a differently shaped tile; *any* way to choose twelve cells numbered $\{0, 1, \ldots, 11\}$ will tile the plane, with exactly the same underlying graph.

Shifted copies of a single shape will also tile the plane if they form a *generalized* torus, in which cell (x, y) corresponds to the same vertex as cells $(x + a, y + b)$ and $(x + c, y + d)$, where (a, b) and (c, d) are integer vectors and $n = ad - bc > 0$. The generalized torus will then have n points. These vectors (a, b) and (c, d) are $(4, 0)$ and $(0, 3)$ in the 3×4 example above; and when they are respectively $(5, 2)$ and $(1, 3)$ we get



.

Here $n = 13$, and a northward move from v goes to $(v + 4) \bmod 13$; an eastward move goes to $(v + 1) \bmod 13$.

Prove that if $\gcd(a, b, c, d) = 1$, the vertices of such a generalized torus can always be assigned integer labels $\{0, 1, \ldots, n-1\}$ in such a way that the neighbors of v are $(v \pm p) \bmod n$ and $(v \pm q) \bmod n$, for some integers p and q.

138. [*HM27*] Continuing exercise 137, what is a good way to label k-dimensional vertices $x = (x_1, \ldots, x_k)$, when integer vectors α_j are given such that each vector x is equivalent to $x + \alpha_j$ for $1 \le j \le k$? Illustrate your method in the case $k = 3$, $\alpha_1 = (3,1,1)$, $\alpha_2 = (1,3,1)$, $\alpha_3 = (1,1,3)$.

▶ **139.** [*M22*] Let H be a fixed graph of order h, and let $\#(H{:}G)$ be the number of times that H occurs as an induced subgraph of a given graph G. If G is chosen at random from the set of all $2^{n(n-1)/2}$ graphs on the vertices $V = \{1, 2, \ldots, n\}$, what is the average value of $\#(H{:}G)$ when H is (a) K_h; (b) P_h, for $h > 1$; (c) C_h, for $h > 2$; (d) arbitrary?

140. [*M30*] A graph G is called *proportional* if its induced subgraph counts $\#(K_3{:}G)$, $\#(\overline{K_3}{:}G)$, and $\#(P_3{:}G)$ each agree with the expected values derived in exercise 139.

a) Show that the wheel graph W_8 of exercise 88 is proportional in this sense.

b) Prove that G is proportional if and only if $\#(K_3{:}G) = \frac{1}{8}\binom{n}{3}$ and the degree sequence $d_1 \ldots d_n$ of its vertices satisfies the identities

$$d_1 + \quad + d_n = \binom{n}{2}, \qquad\qquad d_1^2 + \quad + d_n^2 = \frac{n}{2}\binom{n}{2}. \qquad (*)$$

141. [*26*] The conditions of exercise 140(b) can hold only if $n \bmod 16 \in \{0, 1, 8\}$. Write a program to find all of the proportional graphs that have $n = 8$ vertices.

142. [*M30*] (S. Janson and J. Kratochvíl, 1991.) Prove that no graph G on 4 or more vertices can be "extraproportional," in the sense that its subgraph counts $\#(H{:}G)$ agree with the expected values in exercise 139 for each of the eleven nonisomorphic graphs H of order 4. *Hint:* $(n-3)\#(K_3{:}G) = 4\#(K_4{:}G) + 2\#(K_{1,1,2}{:}G) + \#(K_1 {\oplus} K_3{:}G)$.

▶ **143.** [*M25*] Let A be any matrix with $m > 1$ distinct rows, and $n \ge m$ columns. Prove that at least one column of A can be deleted, without making any two rows equal.

▶ **144.** [*21*] Let X be an $m \times n$ matrix whose entries x_{ij} are either 0, 1, or $*$. A "completion" of X is a matrix X^* in which every $*$ has been replaced by either 0 or 1. Show that the problem of finding a completion with fewest distinct rows is equivalent to the problem of finding the chromatic number of a graph.

▶ **145.** [*25*] (R. S. Boyer and J. S. Moore, 1980.) Suppose the array $a_1 \ldots a_n$ contains a *majority element*, namely a value that occurs more than $n/2$ times. Design an algorithm that finds it after making fewer than n comparisons. *Hint:* If $n \ge 3$ and $a_{n-1} \ne a_n$, the majority element of $a_1 \ldots a_n$ is also the majority element of $a_1 \ldots a_{n-2}$.