Letting $k = |J(T)|$, this last quantity becomes

$$c([S,T]) = |I(S)|k = \sum_{i\in I(S)} k \ge \sum_{i\in I(S)} \min\{p_i, k\}.$$

Also $\sum_{i\in I(T)} p_i \ge \sum_{i\in I(T)} \min\{p_i, k\}$, and $\sum_{j\in J(T)} q_j \le \sum_{j=1}^{k} q_j$. Combining these inequalities, the condition $\sum_{i=1}^{m} \min\{p_i, k\} \ge \sum_{j=1}^{k} q_j$ implies $c([S,T]) \ge \partial(Y \cap T) - \sigma(X \cap T)$. Since this holds for each partition S, T, the network has a feasible flow, which yields the desired bipartite graph. ■

We can extend the maximum flow problem by imposing a nonnegative lower bound on the permitted flow in each edge. The capacity constraint remains as an upper bound, so we require $l(e) \le f(e) \le u(e)$ for the flow $f(e)$. We still impose conservation constraints on the internal nodes. If we have a feasible flow, then an easy modification of the Ford–Fulkerson labeling algorithm allows us to find a maximum (or minimum) feasible flow (Exercise 4). The difficulty is finding an initial feasible flow. First we present an application.

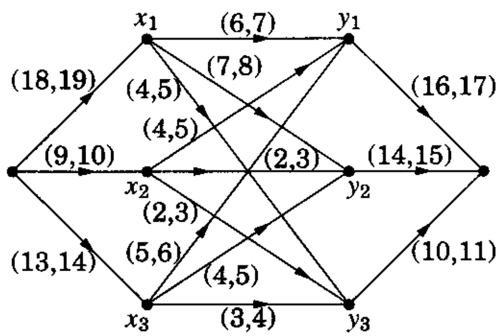
4.3.19. Application. *Matrix rounding* (Bacharach [1966]). We may want to round the entries of a data matrix up or down to integers. We also want to present integers for the row sums and column sums. The sum of each rounded row or column should be a rounding of the original sum. The resulting integer matrix, if it exists, is a **consistent rounding**.

We can represent the consistent rounding problem as a feasible flow problem. Establish vertices $x_1, \dots, x_n$ for the rows and vertices $y_1, \dots, y_n$ for the columns of the matrix. Add a source s and a sink t. Add edges sx_i, x_iy_j, y_jt for all values of i and j. If the matrix has entries $a_{i,j}$ with row-sums $r_1, \dots, r_n$ and column-sums $s_1, \dots, s_n$, set

$$\begin{array}{lll} l(sx_i) = \lfloor r_i \rfloor & l(x_iy_j) = \lfloor a_{i,j} \rfloor & l(y_jt) = \lfloor c_j \rfloor \\ u(sx_i) = \lceil r_i \rceil & u(x_iy_j) = \lceil a_{i,j} \rceil & u(y_jt) = \lceil c_j \rceil \end{array}$$

We test for a feasible flow by transforming again to an ordinary maximum flow problem. With these two transformations, we can use network flow to test for the existence of a consistent rounding. ■

$$\begin{pmatrix} 6.3 & 7.6 & 4.6 \\ 4.7 & 2.3 & 2.8 \\ 5.5 & 4.5 & 3.5 \end{pmatrix}$$



4.3.20. Solution. *Circulations and flows with lower bounds.* In a maximum flow problem with upper and lower bounds on edge capacities, the zero flow is not feasible, so the Ford–Fulkerson labeling algorithm has no place to start. We must first obtain a feasible flow, after which an easy modification of the labeling algorithm applies (Exercise 4).

The first step is to add an edge of infinite capacity from the sink to the source. The resulting network has a feasible flow with conservation at *every* node (called a **circulation**) if and only if the original network has a feasible flow. In a circulation problem, there is no source or sink.

Next, we convert a feasible circulation problem C into a maximum flow problem N by introducing supplies or demands at the nodes and adding a source and sink to satisfy the supplies and demands. Given the flow constraints $l(e) \le f(e) \le u(e)$, let $c(e) = u(e) - l(e)$ for each edge e. For each vertex v, let

$$\begin{aligned} l^-(v) &= \textstyle\sum_{e\in[V(C)-v,v]} l(e), \\ l^+(v) &= \textstyle\sum_{e\in[v,V(C)-v]} l(e), \\ b(v) &= l^-(v) - l^+(v). \end{aligned}$$

Since each $l(uv)$ contributes to $l^+(u)$ and $l^-(v)$, we have $\sum b(v) = 0$. A feasible circulation f must satisfy the flow constraints at each edge and satisfy $f^+(v) - f^-(v) = 0$ at each node. Letting $f'(e) = f(e) - l(e)$, we find that f is a feasible circulation in C if and only if f' satisfies $0 \le f'(e) \le c(e)$ on each edge and $f'^+(v) - f'^-(v) = b(v)$ at each vertex.

This transforms the feasible circulation problem into a flow problem with supplies and demands. If $b(v) \ge 0$, then v supplies flow $|b(v)|$ to the network; otherwise v demands $|b(v)|$. To restore conservation constraints, we add a source s with an edge of capacity $b(v)$ to each v with $b(v) \ge 0$, and we add a sink t with an edge of capacity $-b(v)$ from each v with $b(v) < 0$. This completes the construction of N.

Let α be the total capacity on the edges leaving s; since $\sum b(v) = 0$, the edges entering t also have total capacity α. Now C has a feasible circulation f if and only if N has a flow of value α (saturating all edges out of s or into t). ∎

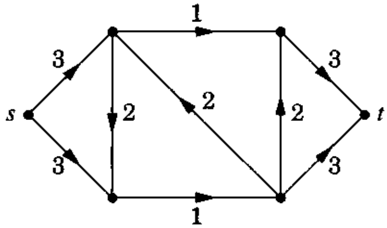
4.3.21. Corollary. A network D with conservation constraints at every node has a feasible circulation if and only if $\sum_{e\in[S,\overline{S}]} l(e) \le \sum_{e\in[\overline{S},S]} u(e)$ for every $S \subseteq V(D)$.

Proof: We can stop before the last step in the discussion of Solution 4.3.20 and interpret our problem with supplies and demands in the model of Theorem 4.3.17. Since $\sum b(v) = 0$, the only way to satisfy all the demands is to use up all the supply. Hence there is a circulation if and only if the supply/demand problem with supplies $\sigma(v) = b(v)$ for $\{v \in V(D) : b(v) \ge 0\}$ and demands $\partial(v) = -b(v)$ for $\{v \in V(D) : b(v) < 0\}$ has a solution.

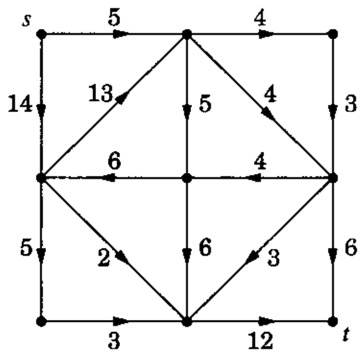
Theorem 4.3.17 characterizes when this problem has a solution. Translated back into the lower and upper bounds on flow in the original problem (Exercise 22), the criterion of Theorem 4.3.17 becomes $\sum_{e\in[S,\overline{S}]} l(e) \le \sum_{e\in[\overline{S},S]} u(e)$ for every $S \subseteq V(D)$. ∎

EXERCISES

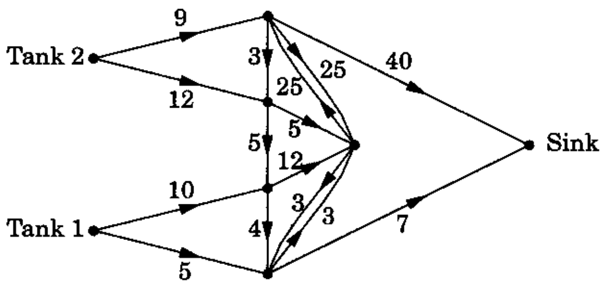
4.3.1. (−) In the network below, list all integer-valued feasible flows and select a flow of maximum value (this illustrates the advantage of duality over exhaustive search). Prove that this flow is a maximum flow by exhibiting a cut with the same value. Determine the number of source/sink cuts. (Comment: There is a nonzero flow with value 0.)



4.3.2. (−) In the network below, find a maximum flow from s to t. Prove that your answer is optimal by using the dual problem, and explain why this proves optimality.



4.3.3. (−) A kitchen sink draws water from two tanks according to the network of pipes with capacities per unit time shown below. Find the maximum flow. Prove that your answer is optimal by using the dual problem, and explain why this proves optimality.



4.3.4. Let N be a network with edge capacity and node conservation constraints plus lower bound constraints $l(e)$ on the flow in edges, meaning that $f(e) \geq l(e)$ is required. If an initial feasible flow is given, how can the Ford–Fulkerson labeling algorithm be modified to search for a maximum feasible flow in this network?

4.3.5. (!) Use network flows to prove Menger's Theorem for internally-disjoint paths in digraphs: $\kappa(x, y) = \lambda(x, y)$ when xy is not an edge. (Hint: Use the first transformation suggested in Remark 4.3.15.)

4.3.6. (!) Use network flows to prove Menger's Theorem for edge-disjoint paths in graphs: $\kappa'(x, y) = \lambda'(x, y)$. (Hint: Use the second transformation suggested in Remark 4.3.15.)

4.3.7. (!) Use network flows to prove Menger's Theorem for nonadjacent vertices in graphs: $\kappa(x, y) = \lambda(x, y)$. (Hint: Use both transformations suggested in Remark 4.3.15.)

4.3.8. Let G be a directed graph with $x, y \in V(G)$. Suppose that capacities are specified *not* on the edges of G, but rather on the *vertices* (other than x, y); for each vertex there is a fixed limit on the total flow through it. There is no restriction on flows in edges. Show how to use ordinary network flow theory to determine the maximum value of a feasible flow from x to y in the vertex-capacitated graph G.

4.3.9. Use network flows to prove that a graph G is connected if and only if for every partition of $V(G)$ into two nonempty sets S, T, there is an edge with one endpoint in S and one endpoint in T. (Comment: Chapter 1 contains an easy direct proof of the conclusion, so this is an example of "using a sledgehammer to squash a bug".)

4.3.10. (!) Use network flows to prove the König–Egerváry Theorem ($\alpha'(G) = \beta(G)$ if G is bipartite).

4.3.11. Show that the Augmenting Path Algorithm for bipartite graphs (Algorithm 3.2.1) is a special case of the Ford–Fulkerson Labeling Algorithm.

4.3.12. Let $[S, \overline{S}]$ and $[T, \overline{T}]$ be source/sink cuts in a network N.

a) Prove that $\text{cap}(S \cup T, \overline{S \cup T}) + \text{cap}(S \cap T, \overline{S \cap T}) \le \text{cap}([S, \overline{S}]) + \text{cap}(T, \overline{T})$. (Hint: Draw a picture and consider contributions from various types of edges.)

b) Suppose that $[S, \overline{S}]$ and $[T, \overline{T}]$ are minimum cuts. Conclude from part (a) that $[S \cup T, \overline{S \cup T}]$ and $[S \cap T, \overline{S \cap T}]$ are also minimum cuts. Conclude also that no edge between $S - T$ and $T - S$ has positive capacity.

4.3.13. (!) Several companies send representatives to a conference; the ith company sends m_i representatives. The organizers of the conference conduct simultaneous networking groups; the jth group can accommodate up to n_j participants. The organizers want to schedule all the participants into groups, but the participants from the same company must be in different groups. The groups need not all be filled.

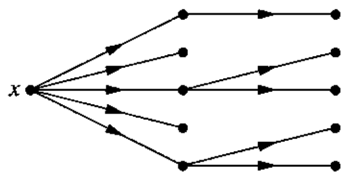
a) Show how to use network flows to test whether the constraints can be satisfied.

b) Let p be the number of companies, and let q be the number of groups, indexed so that $m_1 \ge \cdots \ge m_p$ and $n_1 \le \cdots \le n_q$. Prove that there exists an assignment of participants to groups that satisfies all the constraints if and only if, for all $0 \le k \le p$ and $0 \le l \le q$, it holds that $k(q - l) + \sum_{j=1}^{l} n_j \ge \sum_{i=1}^{k} m_i$.

4.3.14. In a large university with k academic departments, we must appoint an important committee. One professor will be chosen from each department. Some professors have joint appointments in two or more departments, but each must be the designated representative of at most one department. We must use equally many assistant professors, associate professors, and full professors among the chosen representatives (assume that k is divisible by 3). How can the committee be found? (Hint: Build a network in which units of flow correspond to professors chosen for the committee and capacities enforce the various constraints. Explain how to use the network to test whether such a committee exists and find it if it does.) (Hall [1956])

4.3.15. Let G be a weighted graph. Let the *value* of a spanning tree be the minimum weight of its edges. Let the *cap* from a edge cut $[S, \overline{S}]$ be the maximum weight of its edges. Prove that the maximum value of a spanning tree of G equals the minimum cap of an edge cut in G. (Ahuja–Magnanti–Orlin [1993, p538])

4.3.16. (+) Let x be a vertex of maximum outdegree in a tournament T. Prove that T has a spanning directed tree rooted at x such that every vertex has distance at most 2 from x and every vertex other than x has outdegree at most 2. (Hint: Create a network to model the desired paths to the non-successors of x, and show that every cut has enough capacity. Comment: This strengthens Proposition 1.4.30 about kings in tournaments; no vertex need be an intermediate vertex for more than two others.) (Lu [1996])



4.3.17. ($*-$) Use the Gale–Ryser Theorem (Theorem 4.3.18) to determine whether there is a simple bipartite graph in which the vertices in one partite set have degrees $(5, 4, 4, 2, 1)$ and the vertices in the other partite set also have degrees $(5, 4, 4, 2, 1)$.

4.3.18. ($*-$) Given list $r = (r_1, \dots, r_n)$ and $s = (s_1, \dots, s_n)$, obtain necessary and sufficient conditions for the existence of a digraph D with vertices $v_1, \dots, v_n$ such that each ordered pair occurs at most once as an edge and $d^+(v_i) = r_i$ and $d^-(v_i) = s_i$ for all i.

4.3.19. ($*-$) Find a consistent rounding of the data in the matrix below. Is it unique? (Every entry must be 0 or 1.)

$$\begin{pmatrix} .55 & .6 & .6 \\ .55 & .65 & .7 \\ .6 & .65 & .7 \end{pmatrix}$$

4.3.20. ($*$) Prove that every two-by-two matrix can be consistently rounded.

4.3.21. ($*$) Suppose that every entry in an n-by-n matrix is strictly between $1/n$ and $1/(n-1)$. Describe all consistent roundings.

4.3.22. ($*$) Complete the details of proving Corollary 4.3.21, proving the necessary and sufficient condition for a circulation in a network with lower and upper bounds.

4.3.23. ($*!$) A $(k+l)$-regular graph G is (k, l)-**orientable** if it can be oriented so that each indegree is k or l.

a) Prove that G is (k, l)-orientable if and only if there is a partition X, Y of $V(G)$ such that for every $S \subseteq V(G)$,

$$(k-l)(|X \cap S| - |Y \cap S|) \le \left|[S, \overline{S}]\right|.$$

(Hint: Use Theorem 4.3.17.)

b) Conclude that if G is (k, l)-orientable and $k > l$, then G is also $(k-1, l+1)$-orientable. (Bondy–Murty [1976, p210–211])

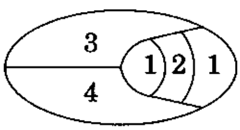
Chapter 5

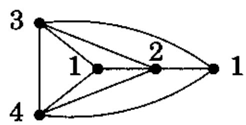
Coloring of Graphs

5.1. Vertex Coloring and Upper Bounds

The committee-scheduling example (Example 1.1.11) used graph coloring to model avoidance of conflicts. Similarly, in a university we want to assign time slots for final examinations so that two courses with a common student have different slots. The number of slots needed is the chromatic number of the graph in which two courses are adjacent if they have a common student.

Coloring the regions of a map with different colors on regions with common boundaries is another example; we return to it in Chapter 6. The map on the left below has five regions, and four colors suffice. The graph on the right models the "common boundary" relation and the corresponding coloring. Labeling of vertices is our context for coloring problems.





DEFINITIONS AND EXAMPLES

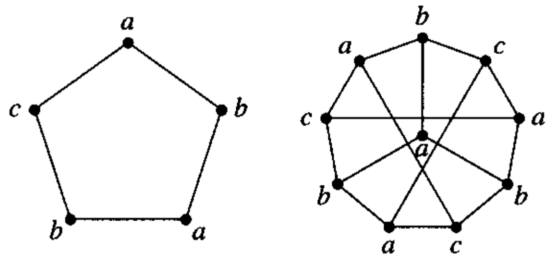
Graph coloring takes its name from the map-coloring application. We assign labels to vertices. When the numerical value of the labels is unimportant, we call them "colors" to indicate that they may be elements of any set.

5.1.1. Definition. A **k-coloring** of a graph G is a labeling $f\colon V(G) \to S$, where $|S| = k$ (often we use $S = [k]$). The labels are **colors**; the vertices of one color form a **color class**. A k-coloring is **proper** if adjacent vertices have different labels. A graph is **k-colorable** if it has a proper k-coloring. The **chromatic number** $\chi(G)$ is the least k such that G is k-colorable.

5.1.2. Remark. In a proper coloring, each color class is an independent set, so G is k-colorable if and only if $V(G)$ is the union of k independent sets. Thus "k-colorable" and "k-partite" have the same meaning. (The usage of the two terms is slightly different. Often "k-partite" is a structural hypothesis, while "k-colorable" is the result of an optimization problem.)

Graphs with loops are uncolorable; we cannot make the color of a vertex different from itself. Therefore, **in this chapter all graphs are loopless.** Also, multiple edges are irrelevant; extra copies don't affect colorings. Thus we usually think in terms of simple graphs when discussing colorings, and we will name edges by their endpoints. Most of the statements made without restriction to simple graphs remain valid when multiple edges are allowed. ■

5.1.3. Example. Since a graph is 2-colorable if and only if it is bipartite, C_5 and the Petersen graph have chromatic number at least 3. Since they are 3-colorable, as shown below, they have chromatic number exactly 3. ■



5.1.4. Definition. A graph G is **k-chromatic** if $\chi(G) = k$. A proper k-coloring of a k-chromatic graph is an **optimal coloring**. If $\chi(H) < \chi(G) = k$ for every proper subgraph H of G, then G is **color-critical** or **k-critical**.

5.1.5. Example. *k-critical graphs for small k.* Properly coloring a graph needs at least two colors if and only if the graph has an edge. Thus K_2 is the only 2-critical graph (similarly, K_1 is the only 1-critical graph). Since 2-colorable is the same as bipartite, the characterization of bipartite graphs implies that the 3-critical graphs are the odd cycles.

We can test 2-colorability of a graph G by computing distances from a vertex x (in each component). Let $X = \{u \in V(G)\colon d(u,x) \text{ is even}\}$, and let $Y = \{u \in V(G)\colon d(u,x) \text{ is odd}\}$. The graph G is bipartite if and only if X, Y is a bipartition, meaning that $G[X]$ and $G[Y]$ are independent sets.

No good characterization of 4-critical graphs or test for 3-colorability is known. Appendix B discusses the computational ramifications. ■

5.1.6. Definition. The **clique number** of a graph G, written $\omega(G)$, is the maximum size of a set of pairwise adjacent vertices (clique) in G.

We have used $\alpha(G)$ for the independence number of G; the usage of $\omega(G)$ is analogous. The letters α and ω are the first and last in the Greek alphabet.

This is consistent with viewing independent sets and cliques as the beginning and end of the "evolution" of a graph (see Section 8.5).

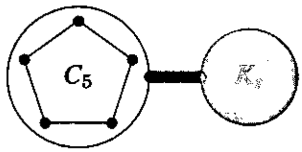
5.1.7. Proposition. For every graph G, $\chi(G) \ge \omega(G)$ and $\chi(G) \ge \frac{n(G)}{\alpha(G)}$.

Proof: The first bound holds because vertices of a clique require distinct colors. The second bound holds because each color class is an independent set and thus has at most $\alpha(G)$ vertices. ■

Both bounds in Proposition 5.1.7 are tight when G is a complete graph.

5.1.8. Example. *$\chi(G)$ may exceed $\omega(G)$.* For $r \ge 2$, let $G = C_{2r+1} \vee K_s$ (the join of C_{2r+1} and K_s—see Definition 3.3.6). Since C_{2r+1} has no triangle, $\omega(G) = s+2$.

Properly coloring the induced cycle requires at least three colors. The s-clique needs s colors. Since every vertex of the induced cycle is adjacent to every vertex of the clique, these s colors must differ from the first three, and $\chi(G) \ge s+3$. We conclude that $\chi(G) > \omega(G)$. ■

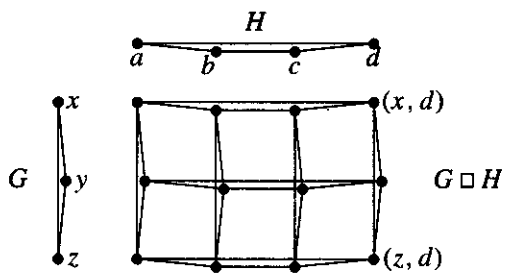


Exercises 23–30 discuss the chromatic number for special families of graphs. We can also ask how it behaves under graph operations. For the disjoint union, $\chi(G+H) = \max\{\chi(G), \chi(H)\}$. For the join, $\chi(G \vee H) = \chi(G) + \chi(H)$. Next we introduce another combining operation.

5.1.9. Definition. The **cartesian product** of G and H, written $G \,\square\, H$, is the graph with vertex set $V(G) \times V(H)$ specified by putting (u, v) adjacent to (u', v') if and only if (1) $u = u'$ and $vv' \in E(H)$, or (2) $v = v'$ and $uu' \in E(G)$.

5.1.10. Example. The cartesian product operation is symmetric; $G \square H \cong H \square G$. Below we show $C_3 \,\square\, C_4$. The hypercube is another familiar example: $Q_k = Q_{k-1} \,\square\, K_2$ when $k \ge 1$. The m-by-n **grid** is the cartesian product $P_m \,\square\, P_n$.

In general, $G \,\square\, H$ decomposes into copies of H for each vertex of G and copies of G for each vertex of H (Exercise 10). We use $\square$ instead of $\times$ to avoid confusion with other product operations, reserving $\times$ for the cartesian product of vertex sets. The symbol $\square$, due to Rödl, evokes the identity $K_2 \,\square\, K_2 = C_4$. ■

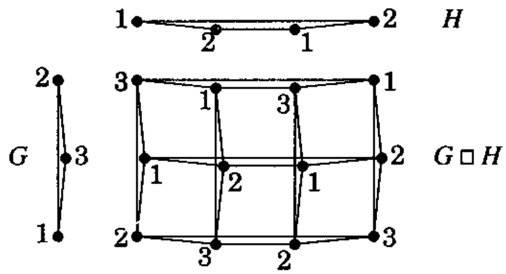


5.1.11. Proposition. (Vizing [1963], Aberth [1964]) $\chi(G \square H) = \max\{\chi(G), \chi(H)\}$.

Proof: The cartesian product $G \square H$ contains copies of G and H as subgraphs, so $\chi(G \square H) \ge \max\{\chi(G), \chi(H)\}$.

Let $k = \max\{\chi(G), \chi(H)\}$. To prove the upper bound, we produce a proper k-coloring of $G \square H$ using optimal colorings of G and H. Let g be a proper $\chi(G)$-coloring of G, and let h be a proper $\chi(H)$-coloring of H. Define a coloring f of $G \square H$ by letting $f(u, v)$ be the congruence class of $g(u) + h(v)$ modulo k. Thus f assigns colors to $V(G \square H)$ from a set of size k.

We claim that f properly colors $G \square H$. If (u, v) and (u', v') are adjacent in $G \square H$, then $g(u) + h(v)$ and $g(u') + h(v')$ agree in one summand and differ by between 1 and k in the other. Since the difference of the two sums is between 1 and k, they lie in different congruence classes modulo k. ■



The cartesian product allows us to compute chromatic numbers by computing independence numbers, because a graph G is m-colorable if and only if the cartesian product $G \square K_m$ has an independent set of size $n(G)$ (Exercise 31).

UPPER BOUNDS

Most upper bounds on the chromatic number come from algorithms that produce colorings. For example, assigning distinct colors to the vertices yields $\chi(G) \le n(G)$. This bound is best possible, since $\chi(K_n) = n$, but it holds with equality only for complete graphs. We can improve a "best-possible" bound by obtaining another bound that is always at least as good. For example, $\chi(G) \le n(G)$ uses nothing about the structure of G; we can do better by coloring the vertices in some order and always using the "least available" color.

5.1.12. Algorithm. (Greedy coloring)
The **greedy coloring** relative to a vertex ordering $v_1, \dots, v_n$ of $V(G)$ is obtained by coloring vertices in the order $v_1, \dots, v_n$, assigning to v_i the smallest-indexed color not already used on its lower-indexed neighbors. ■

5.1.13. Proposition. $\chi(G) \le \Delta(G) + 1$.

Proof: In a vertex ordering, each vertex has at most $\Delta(G)$ earlier neighbors, so the greedy coloring cannot be forced to use more than $\Delta(G) + 1$ colors. This proves constructively that $\chi(G) \le \Delta(G) + 1$. ■

The bound $\Delta(G) + 1$ is the worst upper bound that greedy coloring could produce (although optimal for cliques and odd cycles). Choosing the vertex ordering carefully yields improvements. We can avoid the trouble caused by vertices of high degree by putting them at the beginning, where they won't have many earlier neighbors (see Exercise 36 for a better ordering).

5.1.14. Proposition. (Welsh–Powell [1967]) If a graph G has degree sequence $d_1 \geq \cdots \geq d_n$, then $\chi(G) \leq 1 + \max_i \min\{d_i, i-1\}$.

Proof: We apply greedy coloring to the vertices in nonincreasing order of degree. When we color the ith vertex v_i, it has at most $\min\{d_i, i-1\}$ earlier neighbors, so at most this many colors appear on its earlier neighbors. Hence the color we assign to v_i is at most $1 + \min\{d_i, i-1\}$. This holds for each vertex, so we maximize over i to obtain the upper bound on the maximum color used. ■

The bound in Proposition 5.1.14 is always at most $1 + \Delta(G)$, so this is always at least as good as Proposition 5.1.13. It gives the optimal upper bound in Example 5.1.8, while $1 + \Delta(G)$ does not.

In Proposition 5.1.14, we use greedy coloring with a well-chosen ordering. In fact, every graph G has some vertex ordering for which the greedy algorithm uses only $\chi(G)$ colors (Exercise 33). Usually it is hard to find such an ordering.

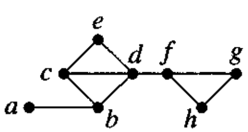
Our next example introduces a class of graphs where such an ordering is easy to find. The ordering produces a coloring that achieves equality in the bound $\chi(G) \geq \omega(G)$.

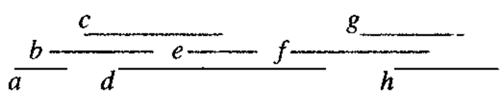
5.1.15. Example. *Register allocation and interval graphs.* A computer program stores the values of its variables in memory. For arithmetic computations, the values must be entered in easily accessed locations called *registers*. Registers are expensive, so we want to use them efficiently. If two variables are never used simultaneously, then we can allocate them to the same register. For each variable, we compute the first and last time when it is used. A variable is *active* during the interval between these times.

We define a graph whose vertices are the variables. Two vertices are adjacent if they are active at a common time. The number of registers needed is the chromatic number of this graph. The time when a variable is active is an interval, so we we obtain a special type of representation for the graph.

An **interval representation** of a graph is a family of intervals assigned to the vertices so that vertices are adjacent if and only if the corresponding intervals intersect. A graph having such a representation is an **interval graph**.

For the vertex ordering a, b, c, d, e, f, g, h of the interval graph below, greedy coloring assigns $1, 2, 1, 3, 2, 1, 2, 3$, respectively, which is optimal. Greedy colorings relative to orderings starting $a, d, \ldots$ use four colors. ■





5.1.16. Proposition. If G is an interval graph, then $\chi(G) = \omega(G)$.

Proof: Order the vertices according to the left endpoints of the intervals in an interval representation. Apply greedy coloring, and suppose that x receives k, the maximum color assigned. Since x does not receive a smaller color, the left endpoint a of its interval belongs also to intervals that already have colors 1 through $k-1$. These intervals all share the point a, so we have a k-clique consisting of x and neighbors of x with colors 1 through $k-1$. Hence $\omega(G) \geq k \geq \chi(G)$. Since $\chi(G) \geq \omega(G)$ always, this coloring is optimal. ■

5.1.17.* Remark. The greedy coloring algorithm runs rapidly. It is "on-line" in the sense that it produces a proper coloring even if it sees only one new vertex at each step and must color it with no option to change earlier colors. For a random vertex ordering in a random graph (see Section 8.5), greedy coloring almost always uses only about twice as many colors as the minimum, although with a bad ordering it may use many colors on a tree (Exercise 34). ■

We began with greedy coloring to underscore the constructive aspect of upper bounds on chromatic number. Other bounds follow from the properties of k-critical graphs but don't produce proper colorings: every k-chromatic graph *has* a k-critical subgraph, but we have no good algorithm for *finding* one. We derive the next bound using critical subgraphs; it can also be proved using greedy coloring (Exercise 36).

5.1.18. Lemma. If H is a k-critical graph, then $\delta(H) \geq k-1$.

Proof: Let x be a vertex of H. Because H is k-critical, $H - x$ is $k-1$-colorable. If $d_H(x) < k-1$, then the $k-1$ colors used on $H - x$ do not all appear on $N(x)$. We can assign x a color not used on $N(x)$ to obtain a proper $k-1$-coloring of H. This contradicts our hypothesis that $\chi(H) = k$. We conclude that $d_H(x) \geq k-1$ (for each $x \in V(H)$). ■

5.1.19. Theorem. (Szekeres–Wilf [1968]) If G is a graph, then $\chi(G) \leq 1 + \max_{H \subseteq G} \delta(H)$.

Proof: Let $k = \chi(G)$, and let H' be a k-critical subgraph of G. Lemma 5.1.18 yields $\chi(G) - 1 = \chi(H') - 1 \leq \delta(H') \leq \max_{H \subseteq G} \delta(H)$. ■

The next bound involves orientations (see also Exercises 43–45).

5.1.20. Example. If G is bipartite, then the orientation of G that directs every edge from one partite set to the other has no path (in the directed sense) of length more than 1. The next theorem thus implies that $\chi(G) \leq 2$.

Every orientation of an odd cycle must somewhere have two consecutive edges in the same direction. Thus each orientation has a path of length at least two, and the theorem confirms that an odd cycle is 3-chromatic. ■

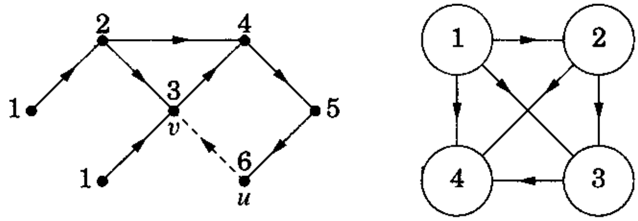
5.1.21. Theorem. Gallai–Roy–Vitaver Theorem (Gallai [1968], Roy [1967], Vitaver [1962]) If D is an orientation of G with longest path length $l(D)$, then $\chi(G) \leq 1 + l(D)$. Furthermore, equality holds for some orientation of G.

Proof: Let D be an orientation of G. Let D' be a maximal subdigraph of D that contains no cycle (in the example below, uv is the only edge of D not in D'). Note that D' includes all vertices of G. Color $V(G)$ by letting $f(v)$ be 1 plus the length of the longest path in D' that ends at v.

Let P be a path in D', and let u be the first vertex of P. Every path in D' ending at u has no other vertex on P, since D' is acyclic. Therefore, each path ending at u (including the longest such path) can be lengthened along P. This implies that f strictly increases along each path in D'.

The coloring f uses colors 1 through $1+l(D')$ on $V(D')$ (which is also $V(G)$). We claim that f is a proper coloring of G. For each $uv \in E(D)$, there is a path in D' between its endpoints (since uv is an edge of D' or its addition to D' creates a cycle). This implies that $f(u) \neq f(v)$, since f increases along paths of D'.

To prove the second statement, we construct an orientation D^* such that $l(D^*) \leq \chi(G) - 1$. Let f be an optimal coloring of G. For each edge uv in G, orient it from u to v in D^* if and only if $f(u) < f(v)$. Since f is a proper coloring, this defines an orientation. Since the labels used by f increase along each path in D^*, and there are only $\chi(G)$ labels in f, we have $l(D^*) \leq \chi(G) - 1$. ■



D' and D $\qquad\qquad$ D^*

BROOKS' THEOREM

The bound $\chi(G) \leq 1 + \Delta(G)$ holds with equality for complete graphs and odd cycles. By choosing the vertex ordering more carefully, we can show that these are essentially the only such graphs. This implies, for example, that the Petersen graph is 3-colorable, without finding an explicit coloring. To avoid unimportant complications, we phrase the statement only for connected graphs. It extends to all graphs because the chromatic number of a graph is the maximum chromatic number of its components. Many proofs are known; we present a modification of the proof by Lovasz [1975].

5.1.22. Theorem. (Brooks [1941]) If G is a connected graph other than a complete graph or an odd cycle, then $\chi(G) \leq \Delta(G)$.

Proof: Let G be a connected graph, and let $k = \Delta(G)$. We may assume that $k \geq 3$, since G is a complete graph when $k \leq 1$, and G is an odd cycle or is bipartite when $k = 2$, in which case the bound holds.

Our aim is to order the vertices so that each has at most $k-1$ lower-indexed neighbors; greedy coloring for such an ordering yields the bound.

When G is not k-regular, we can choose a vertex of degree less than k as v_n. Since G is connected, we can grow a spanning tree of G from v_n, assigning indices in decreasing order as we reach vertices. Each vertex other than v_n in the resulting ordering $v_1, \ldots, v_n$ has a higher-indexed neighbor along the path to v_n in the tree. Hence each vertex has at most $k-1$ lower-indexed neighbors, and the greedy coloring uses at most k colors.

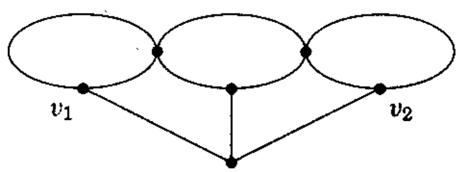
In the remaining case, G is k-regular. Suppose first that G has a cut-vertex x, and let G' be a subgraph consisting of a component of $G-x$ together with its edges to x. The degree of x in G' is less than k, so the method above provides a proper k-coloring of G'. By permuting the names of colors in the subgraphs resulting in this way from components of $G-x$, we can make the colorings agree on x to complete a proper k-coloring of G.

We may thus assume that G is 2-connected. In every vertex ordering, the last vertex has k earlier neighbors. The greedy coloring idea may still work if we arrange that two neighbors of v_n get the same color.

In particular, suppose that some vertex v_n has neighbors v_1, v_2 such that $v_1 \not\leftrightarrow v_2$ and $G - \{v_1, v_2\}$ is connected. In this case, we index the vertices of a spanning tree of $G - \{v_1, v_2\}$ using $3, \ldots, n$ such that labels increase along paths to the root v_n. As before, each vertex before v_n has at most $k-1$ lower indexed neighbors. The greedy coloring also uses at most $k-1$ colors on neighbors of v_n, since v_1 and v_2 receive the same color.

Hence it suffices to show that every 2-connected k-regular graph with $k \geq 3$ has such a triple v_1, v_2, v_n. Choose a vertex x. If $\kappa(G-x) \geq 2$, let v_1 be x and let v_2 be a vertex with distance 2 from x. Such a vertex v_2 exists because G is regular and is not a complete graph; let v_n be a common neighbor of v_1 and v_2.

If $\kappa(G-x) = 1$, let $v_n = x$. Since G has no cut-vertex, x has a neighbor in every leaf block of $G-x$. Neighbors v_1, v_2 of x in two such blocks are nonadjacent. Also, $G - \{x, v_1, v_2\}$ is connected, since blocks have no cut-vertices. Since $k \geq 3$, vertex x has another neighbor, and $G - \{v_1, v_2\}$ is connected. ■



5.1.23.* Remark. The bound $\chi(G) \leq \Delta(G)$ can be improved when G has no large clique (Exercise 50). Brooks' Theorem implies that the complete graphs and odd cycles are the only $k-1$-regular k-critical graphs (Exercise 47). Gallai

[1963b] strengthened this by proving that in the subgraph of a k-critical graph induced by the vertices of degree $k-1$, every block is a clique or an odd cycle.

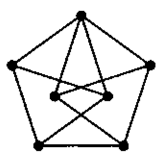
Brooks' Theorem states that $\chi(G) \le \Delta(G)$ whenever $3 \le \omega(G) \le \Delta(G)$. Borodin and Kostochka [1977] conjectured that $\omega(G) < \Delta(G)$ implies $\chi(G) < \Delta(G)$ if $\Delta(G) \ge 9$ (examples show that the condition $\Delta(G) \ge 9$ is needed). Reed [1999] proved that this is true when $\Delta(G) \ge 10^{14}$.

Reed [1998] also conjectured that the chromatic number is bounded by the average of the trivial upper and lower bounds; that is, $\chi(G) \le \lceil \frac{\Delta(G)+1+\omega(G)}{2} \rceil$. ■

Because the idea of partitioning to satisfy constraints is so fundamental, there are many, many variations and generalizations of graph coloring. In Chapter 7 we consider coloring the edges of a graph. Sticking to vertices, we could allow color classes to induce subgraphs other than independent sets ("generalized coloring"—Exercises 49–53). We could restrict the colors allowed to be used on each vertex ("list coloring"—Section 8.4). We could ask questions involving numerical values of the colors (Exercise 54). We have only touched the tip of the iceberg on coloring problems.

EXERCISES

5.1.1. (−) Compute the clique number, the independence number, and the chromatic number of the graph below. Does either bound in Proposition 5.1.7 prove optimality for some proper coloring? Is the graph color-critical?



5.1.2. (−) Prove that the chromatic number of a graph equals the maximum of the chromatic numbers of its components.

5.1.3. (−) Let $G_1, \dots, G_k$ be the blocks of a graph G. Prove that $\chi(G) = \max_i \chi(G_i)$.

5.1.4. (−) Exhibit a graph G with a vertex v so that $\chi(G-v) < \chi(G)$ and $\chi(\overline{G}-v) < \chi(\overline{G})$.

5.1.5. (−) Given graphs G and H, prove that $\chi(G+H) = \max\{\chi(G), \chi(H)\}$ and that $\chi(G \vee H) = \chi(G) + \chi(H)$.

5.1.6. (−) Suppose that $\chi(G) = \omega(G) + 1$, as in Example 5.1.8. Let $H_1 = G$ and $H_k = H_{k-1} \vee G$ for $k > 1$. Prove that $\chi(H)_k = \omega(H)_k + k$.

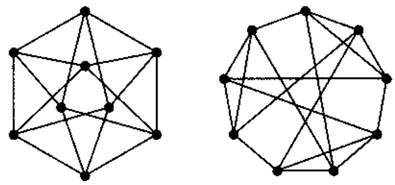
5.1.7. (−) Construct a graph G that is neither a clique nor an odd cycle but has a vertex ordering relative to which greedy coloring uses $\Delta(G) + 1$ colors.

5.1.8. (−) Prove that $\max_{H \subseteq G} \delta(H) \le \Delta(G)$ to explain why Theorem 5.1.19 is better than Proposition 5.1.13. Determine all graphs G such that $\max_{H \subseteq G} \delta(H) = \Delta(G)$.

5.1.9. (−) Draw the graph $K_{1,3} \square P_3$ and exhibit an optimal coloring of it. Draw $C_5 \square C_5$ and find a proper 3-coloring of it with color classes of sizes 9, 8, 8.

5.1.10. (−) Prove that $G \square H$ decomposes into $n(G)$ copies of H and $n(H)$ copies of G.

5.1.11. (−) Prove that each graph below is isomorphic to $C_3 \square C_3$.



5.1.12. (−) Prove or disprove: Every k-chromatic graph G has a proper k-coloring in which some color class has $\alpha(G)$ vertices.

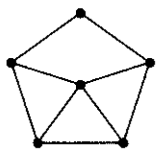
5.1.13. (−) Prove or disprove: If $G = F \cup H$, then $\chi(G) \le \chi(F) + \chi(H)$.

5.1.14. (−) Prove or disprove: For every graph G, $\chi(G) \le n(G) - \alpha(G) + 1$.

5.1.15. (−) Prove or disprove: If G is a connected graph, then $\chi(G) \le 1 + a(G)$, where $a(G)$ is the average of the vertex degrees in G.

5.1.16. (−) Use Theorem 5.1.21 to prove that every tournament has a spanning path. (Rédei [1934])

5.1.17. (−) Use Lemma 5.1.18 to prove that $\chi(G) \le 4$ for the graph G below.



5.1.18. (−) Determine the number of colors needed to label $V(K_n)$ such that each color class induces a subgraph with maximum degree at most k.

5.1.19. (−) Find the error in the false argument below for Brooks' Theorem (Theorem 5.1.22).

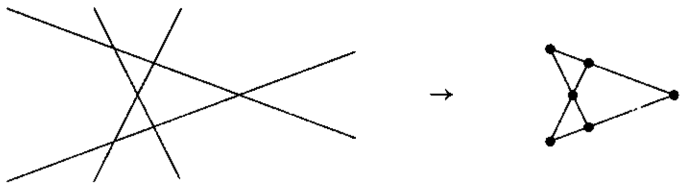
"We use induction on $n(G)$; the statement holds when $n(G) = 1$. For the induction step, suppose that G is not a complete graph or an odd cycle. Since $\kappa(G) \le \delta(G)$, the graph G has a separating set S of size at most $\Delta(G)$. Let $G_1, \ldots, G_m$ be the components of $G - S$, and let $H_i = G[V(G_i) \cup S]$. By the induction hypothesis, each H_i is $\Delta(G)$-colorable. Permute the names of the colors used on these subgraphs to agree on S. This yields a proper $\Delta(G)$-coloring of G."

• • • • •

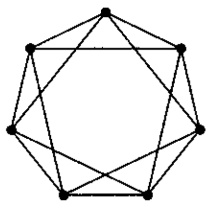
5.1.20. (!) Let G be a graph whose odd cycles are pairwise intersecting, meaning that every two odd cycles in G have a common vertex. Prove that $\chi(G) \le 5$.

5.1.21. Suppose that every edge of a graph G appears in at most one cycle. Prove that every block of G is an edge, a cycle, or an isolated vertex. Use this to prove that $\chi(G) \le 3$.

5.1.22. (!) Given a set of lines in the plane with no three meeting at a point, form a graph G whose vertices are the intersections of the lines, with two vertices adjacent if they appear consecutively on one of the lines. Prove that $\chi(G) \le 3$. (Hint: This can be solved by using the Szekeres–Wilf Theorem or by using greedy colori: ; with an appropriate vertex ordering. Comment: The conclusion may fail when three lines are allowed to share a point.) (H. Sachs)



5.1.23. (!) Place n points on a circle, where $n \geq k(k+1)$. Let $G_{n,k}$ be the $2k$-regular graph obtained by joining each point to the k nearest points in each direction on the circle. For example, $G_{n,1} = C_n$, and $G_{7,2}$ appears below. Prove that $\chi(G_{n,k}) = k+1$ if $k+1$ divides n and $\chi(G_{n,k}) = k+2$ if $k+1$ does not divide n. Prove that the lower bound on n cannot be weakened, by proving that $\chi(G_{k(k+1)-1,k}) > k+2$ if $k \geq 2$.



5.1.24. (+) Let G be any 20-regular graph with 360 vertices formed in the following way. The vertices are evenly-spaced around a circle. Vertices separated by 1 or 2 degrees are nonadjacent. Vertices separated by 3, 4, 5 or 6 degrees are adjacent. No information is given about other adjacencies (except that G is 20-regular). Prove that $\chi(G) \leq 19$. (Hint: Color successive vertices in order around the circle.) (Pritikin)

5.1.25. (+) Let G be the **unit-distance graph** in the plane; $V(G) = \mathbb{R}^2$, and two points are adjacent if their Euclidean distance is 1 (this is an infinite graph). Prove that $4 \leq \chi(G) \leq 7$. (Hint: For the upper bound, present an explicit coloring by regions, paying attention to the boundaries.) (Hadwiger [1945, 1961], Moser–Moser [1961])

5.1.26. Given finite sets $S_1, \dots, S_m$, let $U = S_1 \times \cdots \times S_m$. Define a graph G with vertex set U by putting $u \leftrightarrow v$ if and only if u and v differ in every coordinate. Determine $\chi(G)$.

5.1.27. Let H be the complement of the graph in Exercise 5.1.26. Determine $\chi(H)$.

5.1.28. Consider a traffic signal controlled by two switches, each of which can be set in n positions. For each setting of the switches, the traffic signal shows one of its n possible colors. Whenever the setting of *both* switches changes, the color changes. Prove that the color shown is determined by the position of one of the switches. Interpret this in terms of the chromatic number of some graph. (Greenwell–Lovász [1974])

5.1.29. For the graph G below, compute $\chi(G)$ and find a $\chi(G)$-critical subgraph.