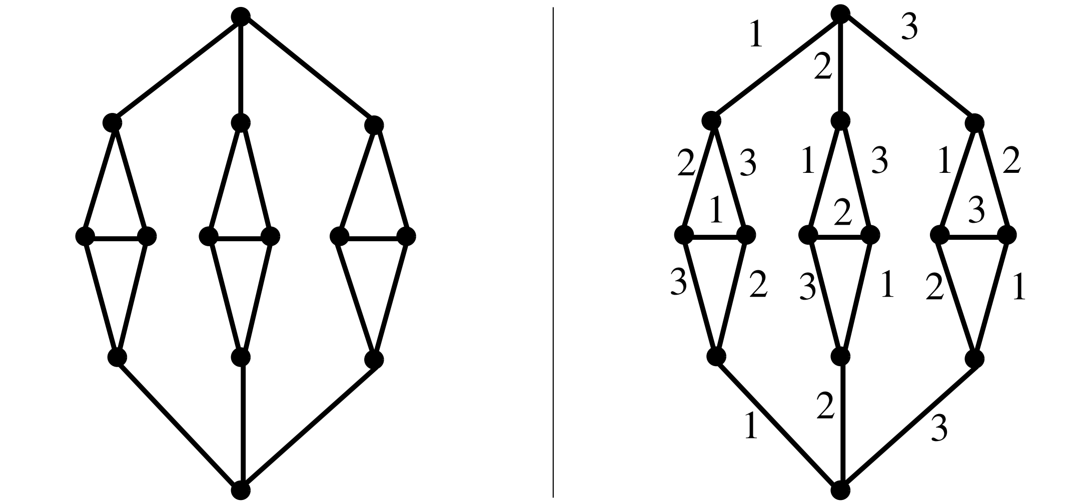

INPUT OUTPUT

16.8 Edge Coloring

Input description: A graph $G = (V, E)$.

Problem description: What is the smallest set of colors needed to color the edges of G such that no two same-color edges share a common vertex?

Discussion: The edge coloring of graphs arises in scheduling applications, typically associated with minimizing the number of noninterfering rounds needed to complete a given set of tasks. For example, consider a situation where we must schedule a given set of two-person interviews, where each interview takes one hour. All meetings could be scheduled to occur at distinct times to avoid conflicts, but it is less wasteful to schedule nonconflicting events simultaneously. We construct a graph whose vertices are people and whose edges represent the pairs of people who need to meet. An edge coloring of this graph defines the schedule. The color classes represent the different time periods in the schedule, with all meetings of the same color happening simultaneously.

The National Football League solves such an edge-coloring problem each season to make up its schedule. Each team's opponents are determined by the records of the previous season. Assigning the opponents to weeks of the season is an edge-coloring problem, complicated by extra constraints of spacing out rematches and making sure that there is a good game every Monday night.

The minimum number of colors needed to edge color a graph is called its *edge-chromatic number* by some and its *chromatic index* by others. Note that an even-length cycle can be edge-colored with 2 colors, while odd-length cycles have an edge-chromatic number of 3.

Edge coloring has a better (if less famous) theorem associated with it than vertex coloring. Vizing's theorem states that any graph with a maximum vertex degree of Δ can be edge colored using at most $\Delta + 1$ colors. To put this in perspective, note that *any* edge coloring must have at least Δ colors, since all the edges incident on any vertex must be distinct colors.

The proof of Vizing's theorem is constructive, meaning it can be turned into an $O(nm\Delta)$ algorithm to find an edge-coloring with $\Delta + 1$ colors. Since deciding whether we can get away using one less color than this is NP-complete, it hardly seems worth the effort to worry about it. An implementation of Vizing's theorem is described below.

Any edge-coloring problem on G can be converted to the problem of finding a vertex coloring on the *line graph* $L(G)$, which has a vertex of $L(G)$ for each edge of G and an edge of $L(G)$ if and only if the two edges of G share a common vertex. Line graphs can be constructed in time linear to their size, and any vertex-coloring code can be employed to color them. That said, it is disappointing to go the vertex coloring route. Vizing's theorem is our reward for the extra thought needed to see that we have an edge-coloring problem.

Implementations: Yan Dong produced an implementation of Vizing's theorem in C++ as a course project for my algorithms course while a student at Stony Brook. It can be found on the algorithm repository site at *http://www.cs.sunysb.edu/~algorith*.

GOBLIN (*http://www.math.uni-augsburg.de/~fremuth/goblin.html*) implements a branch-and-bound algorithm for edge coloring.

See Section 16.7 (page 544) for a larger collection of vertex-coloring codes and heuristics, which can be applied to the line graph of your target graph. Combinatorica [PS03] provides Mathematica implementations of edge coloring in this fashion, via the line graph transformation and vertex coloring routines. See Section 19.1.9 (page 661) for more information on Combinatorica.

Notes: Graph-theoretic results on edge coloring are surveyed in [FW77, GT94]. Vizing [Viz64] and Gupta [Gup66] independently proved that any graph can be edge colored using at most $\Delta + 1$ colors. Misra and Gries give a simple constructive proof of this result [MG92]. Despite these tight bounds, it is NP-complete to compute the edge-chromatic number [Hol81]. Bipartite graphs can be edge-colored in polynomial time [Sch98].

Whitney, in introducing line graphs [Whi32], showed that with the exception of K_3 and $K_{1,3}$, any two connected graphs with isomorphic line graphs are isomorphic. It is an interesting exercise to show that the line graph of an Eulerian graph is both Eulerian and Hamiltonian, while the line graph of a Hamiltonian graph is always Hamiltonian.

Related Problems: Vertex coloring (see page 544), scheduling (see page 468).