

A Java Library of Graph Algorithms and Optimization. By *Hang T. Lau*. Chapman & Hall/CRC, 2007. \$99.95. 368 pp., hardcover. ISBN 1-58488-718-4.

As the title suggests, this text describes a collection of routines in the Java programming language solving basic problems in graph theory, such as locating the connected components of a graph, and in graph optimization, such as finding a maximum network flow. Each of the fifty-five routines is preceded by an introduction about a page in length. This is split between a description of the problem to be solved, with some references to the literature, and a description of the parameters passed to the routine. This is followed by the verbatim listing of the code. The one-page conclusion of each routine's description is a small example, typically a graph on 8 or 10 vertices, and the program listing for a small driver program and its output.

So roughly 250 pages are given over to program listings. The use of the word "library" in the title would suggest some sort of uniformity to the routines and an interdependence, utilizing common data structures and exploiting the object-oriented features of Java for efficiency in use and presentation. This is not the case. There is not even a graph object. Typical data structures are arrays of integers, so graphs are implemented as paired one-dimensional arrays of endpoints of edges (`nodei[]`, `nodej[]`) or as a two-dimensional array that is the adjacency matrix. Worse, the code takes no advantage of Java's strengths and is written in a style reminiscent of C, Pascal or FORTRAN. Arguments to routines are often long lists of arrays, there are no objects in sight anywhere and the output requires the user to interpret the contents of the arrays containing the results. Besides questions of style, the particular choice of algorithms also raises questions. The description of depth-first search ignores the possibility of a recursive approach. The long section on graph isomorphism makes no mention of McKay's algorithm (implemented as the `nauty` package and freely available for non-military use), and instead uses an algorithm from a FORTRAN90 library.

Of the roughly 100 pages that is not code, there is little explanation of the choice of algorithms or data structures for the problem at hand. A similar effort is Skiena's *Implementing Discrete Mathematics* which describes the `Combinatorica` package for Mathematica. The reader

who expects the work under review to match Skiena's careful attempts to teach and explain will be very disappointed. The book's Introduction says, "The library of programs is intended to be used for educational and experimental purposes." While the use of such a library could be beneficial in an educational setting (or for rapid prototyping) the book itself makes very little effort to teach or inform.

Included inside the back cover is a copyrighted compact disc. There is no reference to this disc in the book, no description of its contents, no guidance on allowed uses. The closest thing to a mention of how the disc may be used is the standard boilerplate on the copyright page which says, "No part of this book may be utilized...in any information storage or retrieval system...without written permission from the publishers." Harming a copyright violation, an examination of the contents of the disc reveals it contains the class files of the routines. These are the intermediate, machine-readable files created by a Java compiler, and not the original source files as printed. So modifications to these files (perhaps for "experimental purposes") are not made easily. For a project of this nature, it would be more useful to release the source code in electronic form with a license explaining clearly what the purchaser would be allowed to do with the code, and with any programs that might incorporate it. A search uncovered no network locations that might also be hosting the source code. Using Skiena's work as an example again, consider that his *Algorithm Design Manual* is available online in a hypertext-linked edition and is supported by the comprehensive Stonybrook Algorithm Repository website.

There is little to recommend in this project. The text and the code are not instructive. Practical use of the class files on the compact disc for experimental purposes is limited at best, prohibited as described in the front matter. Whether or not the algorithms are carefully chosen and implemented, whether or not the style of coding is appropriate or instructive, delivery as a printed book with an electronic supplement containing only machine-readable implementations seems ill-advised.

ROBERT A. BEEZER
UNIVERSITY OF PUGET SOUND

An edited version of this review was published in *SIAM Review* 49, No. 3 (2007) as part of the Book Review section.