

Scope

This book is an *introductory* text. As such, it covers several topics, such as algorithm complexity, hash tables, and graphs, in rather less depth than would be possible in a more advanced text. Moreover, it omits altogether a variety of interesting algorithms, such as compression, encryption, geometric, linear algebra, and scheduling algorithms.

Under Further Reading there is a list of books that do cover these more advanced topics, together with seminal papers on topics covered by this book. There are also bibliographic notes that suggest further reading on the topics covered by each chapter and by the book as a whole.

Java Collections framework

The Java 2 software development kit (SDK) provides a library of classes supporting the most common ADTs: lists, sets, and maps. These are known collectively as the *Java collection classes*, and they fit together in what is called the *Java Collections framework*. Programmers should use these collection classes rather than designing and implementing their own. However, the framework is complex and incomplete, and selecting the correct collection class for a given problem is not always easy.

This book aims to help programmers in the following ways:

- To reduce complexity, we introduce our own contract for each ADT. Our contract captures the fundamental properties of the ADT, and is not biased towards a particular implementation.
- To maintain compatibility, our ADT contract is always a simplified version of the contract of the corresponding Java collection class (if any). The applications presented in this book can all be trivially modified to use the corresponding collection classes. (The only necessary changes are to import the `java.util` package and to modify certain constructor invocations.)
- To guide the selection of the appropriate ADT for a given problem, we present one or more short examples, and a case study, illustrating how to use each ADT.
- To guide the selection of the most appropriate implementation of an ADT, we analyze the efficiency of several alternative implementations, and compare them with the corresponding Java collection class (if any).
- In addition to the list, set, and map ADTs supported by the Java Collections framework, we also cover other important ADTs: stacks, queues, priority queues, trees, and graphs. Our contracts for these ADTs follow the same style as the Java Collections framework, so readers should be able to adapt easily.

Companion Web site

This book is complemented by a Web site, whose URL is

www.wiley.co.uk/wattbrown