19.1 Introduction

Searching data involves determining whether a value (referred to as the **search key**) is present in the data and, if so, finding its location. Two popular search algorithms are the simple linear search and the faster but more complex binary search. **Sorting** places data in ascending or descending order, based on one or more **sort keys**. A list of names could be sorted alphabetically, bank accounts could be sorted by account number, employee payroll records could be sorted by social security number, and so on. This chapter introduces two simple sorting algorithms, the selection sort and the insertion sort, along with the more efficient but more complex merge sort. Figure 19.1 summarizes the searching and sorting algorithms discussed in the examples and exercises of this book.

Software Engineering Observation 19.1

In apps that require searching and sorting, use the predefined capabilities of the Java Collections API (Chapter 16). The techniques in this chapter are provided to introduce students to the concepts behind searching and sorting algorithms—upper-level computer-science courses typically discuss such

algorithms in detail.

Chapter	Algorithm	Location
Searching Algorithms:		
16	`binarySearch` method of class `Collections`	Fig. 16.12
19	Linear search Binary search Recursive linear search Recursive binary search	Section 19.2 Section 19.4 Exercise 19.8 Exercise 19.9
21	Linear search of a `List` Binary tree search	Exercise 21.21 Exercise 21.23
16	`sort` method of class `Collections` `SortedSet` collection	Figs. 16.6–16.9 Fig. 16.16
19	Selection sort Insertion sort Recursive merge sort Bubble sort Bucket sort Recursive quicksort	Section 19.6 Section 19.7 Section 19.8 Exercises 19.5 and 19.6 Exercise 19.7 Exercise 19.10
21	Binary tree sort	Section 21.7

Fig. 19.1

Searching and sorting algorithms covered in this text.

A Note About Counter-Controlled `for` Loops in the Examples

8

Throughout this chapter, we use many counter-controlled `for` loops to demonstrate the mechanics of searching and sorting with various algorithms. Many of these loops now can be implemented using Java SE 8's streams capabilities (Chapter 17).