

Data Structures and Algorithm Analysis

Edition 3.2 (C++ Version)

Clifford A. Shaffer

Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

March 28, 2013

Update 3.2.0.10

For a list of changes, see

<http://people.cs.vt.edu/~shaffer/Book/errata.html>

Copyright © 2009-2012 by Clifford A. Shaffer.

This document is made freely available in PDF form for educational and other non-commercial use. You may make copies of this file and redistribute in electronic form without charge. You may extract portions of this document provided that the front page, including the title, author, and this notice are included. Any commercial use of this document requires the written consent of the author. The author can be reached at

shaffer@cs.vt.edu.

If you wish to have a printed version of this document, print copies are published by Dover Publications

(see <http://store.doverpublications.com/048648582x.html>).

Further information about this text is available at

<http://people.cs.vt.edu/~shaffer/Book/>.

Contents

Preface	xiii
I Preliminaries	1
1 Data Structures and Algorithms	3
1.1 A Philosophy of Data Structures	4
1.1.1 The Need for Data Structures	4
1.1.2 Costs and Benefits	6
1.2 Abstract Data Types and Data Structures	8
1.3 Design Patterns	12
1.3.1 Flyweight	13
1.3.2 Visitor	13
1.3.3 Composite	14
1.3.4 Strategy	15
1.4 Problems, Algorithms, and Programs	16
1.5 Further Reading	18
1.6 Exercises	20
2 Mathematical Preliminaries	25
2.1 Sets and Relations	25
2.2 Miscellaneous Notation	29
2.3 Logarithms	31
2.4 Summations and Recurrences	32
2.5 Recursion	36
2.6 Mathematical Proof Techniques	38

2.6.1	Direct Proof	39
2.6.2	Proof by Contradiction	39
2.6.3	Proof by Mathematical Induction	40
2.7	Estimation	46
2.8	Further Reading	47
2.9	Exercises	48
3	Algorithm Analysis	55
3.1	Introduction	55
3.2	Best, Worst, and Average Cases	61
3.3	A Faster Computer, or a Faster Algorithm?	62
3.4	Asymptotic Analysis	65
3.4.1	Upper Bounds	65
3.4.2	Lower Bounds	67
3.4.3	Θ Notation	68
3.4.4	Simplifying Rules	69
3.4.5	Classifying Functions	70
3.5	Calculating the Running Time for a Program	71
3.6	Analyzing Problems	76
3.7	Common Misunderstandings	77
3.8	Multiple Parameters	79
3.9	Space Bounds	80
3.10	Speeding Up Your Programs	82
3.11	Empirical Analysis	85
3.12	Further Reading	86
3.13	Exercises	86
3.14	Projects	90
II	Fundamental Data Structures	93
4	Lists, Stacks, and Queues	95
4.1	Lists	96
4.1.1	Array-Based List Implementation	100
4.1.2	Linked Lists	103
4.1.3	Comparison of List Implementations	112

Contents	v
4.1.4 Element Implementations	114
4.1.5 Doubly Linked Lists	115
4.2 Stacks	120
4.2.1 Array-Based Stacks	121
4.2.2 Linked Stacks	123
4.2.3 Comparison of Array-Based and Linked Stacks	123
4.2.4 Implementing Recursion	125
4.3 Queues	127
4.3.1 Array-Based Queues	128
4.3.2 Linked Queues	133
4.3.3 Comparison of Array-Based and Linked Queues	133
4.4 Dictionaries	133
4.5 Further Reading	145
4.6 Exercises	145
4.7 Projects	148
5 Binary Trees	151
5.1 Definitions and Properties	151
5.1.1 The Full Binary Tree Theorem	153
5.1.2 A Binary Tree Node ADT	155
5.2 Binary Tree Traversals	155
5.3 Binary Tree Node Implementations	160
5.3.1 Pointer-Based Node Implementations	160
5.3.2 Space Requirements	166
5.3.3 Array Implementation for Complete Binary Trees	168
5.4 Binary Search Trees	168
5.5 Heaps and Priority Queues	178
5.6 Huffman Coding Trees	185
5.6.1 Building Huffman Coding Trees	186
5.6.2 Assigning and Using Huffman Codes	192
5.6.3 Search in Huffman Trees	195
5.7 Further Reading	196
5.8 Exercises	196
5.9 Projects	200
6 Non-Binary Trees	203

6.1	General Tree Definitions and Terminology	203
6.1.1	An ADT for General Tree Nodes	204
6.1.2	General Tree Traversals	205
6.2	The Parent Pointer Implementation	207
6.3	General Tree Implementations	213
6.3.1	List of Children	214
6.3.2	The Left-Child/Right-Sibling Implementation	215
6.3.3	Dynamic Node Implementations	215
6.3.4	Dynamic “Left-Child/Right-Sibling” Implementation	218
6.4	<i>K</i> -ary Trees	218
6.5	Sequential Tree Implementations	219
6.6	Further Reading	223
6.7	Exercises	223
6.8	Projects	226
III	Sorting and Searching	229
7	Internal Sorting	231
7.1	Sorting Terminology and Notation	232
7.2	Three $\Theta(n^2)$ Sorting Algorithms	233
7.2.1	Insertion Sort	233
7.2.2	Bubble Sort	235
7.2.3	Selection Sort	237
7.2.4	The Cost of Exchange Sorting	238
7.3	Shellsort	239
7.4	Mergesort	241
7.5	Quicksort	244
7.6	Heapsort	251
7.7	Binsort and Radix Sort	252
7.8	An Empirical Comparison of Sorting Algorithms	259
7.9	Lower Bounds for Sorting	261
7.10	Further Reading	265
7.11	Exercises	265
7.12	Projects	269

[Click here to download full PDF material](#)