Contents

Preface xi

1. **Why Data Structures Matter** 1
   - Data Structures 2
   - The Array: The Foundational Data Structure 3
   - Measuring Speed 4
   - Reading 5
   - Searching 8
   - Insertion 11
   - Deletion 13
   - Sets: How a Single Rule Can Affect Efficiency 15
   - Wrapping Up 18
   - Exercises 19

2. **Why Algorithms Matter** 21
   - Ordered Arrays 22
   - Searching an Ordered Array 24
   - Binary Search 26
   - Binary Search vs. Linear Search 31
   - Wrapping Up 34
   - Exercises 34

3. **O Yes! Big O Notation** 35
   - Big O: How Many Steps Relative to N Elements? 36
   - The Soul of Big O 37
   - An Algorithm of the Third Kind 40
   - Logarithms 41
   - O(log N) Explained 42
   - Practical Examples 43
   - Wrapping Up 45
   - Exercises 45

4. **Speeding Up Your Code with Big O** 47
   - Bubble Sort 47

Bubble Sort in Action 48
The Efficiency of Bubble Sort 53
A Quadratic Problem 56
A Linear Solution 58
Wrapping Up 60
Exercises 60

5. Optimizing Code with and Without Big O 63
Selection Sort 63
Selection Sort in Action 64
The Efficiency of Selection Sort 70
Ignoring Constants 71
Big O Categories 72
Wrapping Up 76
Exercises 76

6. Optimizing for Optimistic Scenarios 79
Insertion Sort 79
Insertion Sort in Action 80
The Efficiency of Insertion Sort 86
The Average Case 88
A Practical Example 91
Wrapping Up 93
Exercises 93

7. Big O in Everyday Code 95
Mean Average of Even Numbers 95
Word Builder 97
Array Sample 99
Average Celsius Reading 100
Clothing Labels 101
Count the Ones 102
Palindrome Checker 102
Get All the Products 103
Password Cracker 107
Wrapping Up 109
Exercises 109

8. Blazing Fast Lookup with Hash Tables 113
Hash Tables 113
Hashing with Hash Functions 114

Building a Thesaurus for Fun and Profit, but Mainly Profit 115
Hash Table Lookups 117
Dealing with Collisions 119
Making an Efficient Hash Table 122
Hash Tables for Organization 124
Hash Tables for Speed 125
Wrapping Up 130
Exercises 131

9. Crafting Elegant Code with Stacks and Queues 133
Stacks 133
Abstract Data Types 136
Stacks in Action 137
The Importance of Constrained Data Structures 143
Queues 144
Queues in Action 146
Wrapping Up 147
Exercises 148

10. Recursively Recurse with Recursion 149
Recurse Instead of Loop 149
The Base Case 151
Reading Recursive Code 151
Recursion in the Eyes of the Computer 154
Filesystem Traversal 156
Wrapping Up 159
Exercises 159

11. Learning to Write in Recursive 161
Recursive Category: Repeatedly Execute 161
Recursive Category: Calculations 166
Top-Down Recursion: A New Way of Thinking 168
The Staircase Problem 173
Anagram Generation 177
Wrapping Up 181
Exercises 181

12. Dynamic Programming 183
Unnecessary Recursive Calls 183
The Little Fix for Big O 187
The Efficiency of Recursion 188

Overlapping Subproblems 189
Dynamic Programming through Memoization 191
Dynamic Programming through Going Bottom-Up 194
Wrapping Up 196
Exercises 197

13. Recursive Algorithms for Speed 199
Partitioning 199
Quicksort 205
The Efficiency of Quicksort 211
Quicksort in the Worst-Case Scenario 216
Quickselect 218
Sorting as a Key to Other Algorithms 222
Wrapping Up 223
Exercises 224

14. Node-Based Data Structures 225
Linked Lists 225
Implementing a Linked List 227
Reading 229
Searching 231
Insertion 232
Deletion 236
Efficiency of Linked List Operations 238
Linked Lists in Action 239
Doubly Linked Lists 240
Queues as Doubly Linked Lists 242
Wrapping Up 244
Exercises 244

15. Speeding Up All the Things with Binary Search Trees . . 247
Trees 248
Binary Search Trees 250
Searching 251
Insertion 256
Deletion 260
Binary Search Trees in Action 271
Binary Search Tree Traversal 272
Wrapping Up 276
Exercises 276

16. Keeping Your Priorities Straight with Heaps 279
Priority Queues 279
Heaps 281
Heap Properties 284
Heap Insertion 285
Looking for the Last Node 287
Heap Deletion 288
Heaps vs. Ordered Arrays 292
The Problem of the Last Node...Again 293
Arrays as Heaps 295
Heaps as Priority Queues 302
Wrapping Up 302
Exercises 303

17. It Doesn't Hurt to Trie 305
Tries 306
Storing Words 307
Trie Search 311
The Efficiency of Trie Search 315
Trie Insertion 316
Building Autocomplete 320
Completing Autocomplete 326
Tries with Values: A Better Autocomplete 327
Wrapping Up 328
Exercises 329

18. Connecting Everything with Graphs 331
Graphs 332
Directed Graphs 334
Object-Oriented Graph Implementation 334
Graph Search 337
Depth-First Search 339
Breadth-First Search 348
The Efficiency of Graph Search 361
Weighted Graphs 364
Dijkstra's Algorithm 367
Wrapping Up 384
Exercises 384

19. Dealing with Space Constraints 387
Big O of Space Complexity 387

Trade-Offs Between Time and Space 390
The Hidden Cost of Recursion 393
Wrapping Up 395
Exercises 395

20. Techniques for Code Optimization 397
Prerequisite: Determine Your Current Big O 397
Start Here: The Best-Imaginable Big O 397
Magical Lookups 399
Recognizing Patterns 406
Greedy Algorithms 414
Change the Data Structure 427
Wrapping Up 433
Parting Thoughts 433
Exercises 434

A1. Exercise Solutions 439
Chapter 1 439
Chapter 2 440
Chapter 3 440
Chapter 4 441
Chapter 5 442
Chapter 6 442
Chapter 7 443
Chapter 8 444
Chapter 9 446
Chapter 10 447
Chapter 11 448
Chapter 12 450
Chapter 13 451
Chapter 14 453
Chapter 15 456
Chapter 16 458
Chapter 17 459
Chapter 18 461
Chapter 19 464
Chapter 20 465

Index 475