
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 | 14 |
| 1.3.3 Composite | 15 |
| 1.3.4 Strategy | 16 |
| 1.4 Problems, Algorithms, and Programs | 17 |
| 1.5 Further Reading | 19 |
| 1.6 Exercises | 21 |
| 2 Mathematical Preliminaries | 25 |
| 2.1 Sets and Relations | 25 |
| 2.2 Miscellaneous Notation | 29 |
| 2.3 Logarithms | 31 |
| 2.4 Summations and Recurrences | 33 |
| | iii |

| | | |
|-----------|--|-----------|
| 2.5 | Recursion | 36 |
| 2.6 | Mathematical Proof Techniques | 39 |
| 2.6.1 | Direct Proof | 40 |
| 2.6.2 | Proof by Contradiction | 40 |
| 2.6.3 | Proof by Mathematical Induction | 41 |
| 2.7 | Estimating | 47 |
| 2.8 | Further Reading | 49 |
| 2.9 | Exercises | 50 |
| 3 | Algorithm Analysis | 57 |
| 3.1 | Introduction | 57 |
| 3.2 | Best, Worst, and Average Cases | 63 |
| 3.3 | A Faster Computer, or a Faster Algorithm? | 65 |
| 3.4 | Asymptotic Analysis | 67 |
| 3.4.1 | Upper Bounds | 68 |
| 3.4.2 | Lower Bounds | 70 |
| 3.4.3 | Θ Notation | 71 |
| 3.4.4 | Simplifying Rules | 72 |
| 3.4.5 | Classifying Functions | 73 |
| 3.5 | Calculating the Running Time for a Program | 74 |
| 3.6 | Analyzing Problems | 79 |
| 3.7 | Common Misunderstandings | 81 |
| 3.8 | Multiple Parameters | 83 |
| 3.9 | Space Bounds | 84 |
| 3.10 | Speeding Up Your Programs | 86 |
| 3.11 | Empirical Analysis | 89 |
| 3.12 | Further Reading | 90 |
| 3.13 | Exercises | 91 |
| 3.14 | Projects | 95 |
| II | Fundamental Data Structures | 97 |
| 4 | Lists, Stacks, and Queues | 99 |

| | |
|--|------------|
| Contents | v |
| 4.1 Lists | 100 |
| 4.1.1 Array-Based List Implementation | 103 |
| 4.1.2 Linked Lists | 106 |
| 4.1.3 Comparison of List Implementations | 117 |
| 4.1.4 Element Implementations | 119 |
| 4.1.5 Doubly Linked Lists | 120 |
| 4.2 Stacks | 125 |
| 4.2.1 Array-Based Stacks | 125 |
| 4.2.2 Linked Stacks | 128 |
| 4.2.3 Comparison of Array-Based and Linked Stacks | 129 |
| 4.2.4 Implementing Recursion | 129 |
| 4.3 Queues | 133 |
| 4.3.1 Array-Based Queues | 134 |
| 4.3.2 Linked Queues | 137 |
| 4.3.3 Comparison of Array-Based and Linked Queues | 140 |
| 4.4 Dictionaries and Comparators | 140 |
| 4.5 Further Reading | 147 |
| 4.6 Exercises | 147 |
| 4.7 Projects | 150 |
| 5 Binary Trees | 153 |
| 5.1 Definitions and Properties | 153 |
| 5.1.1 The Full Binary Tree Theorem | 156 |
| 5.1.2 A Binary Tree Node ADT | 157 |
| 5.2 Binary Tree Traversals | 158 |
| 5.3 Binary Tree Node Implementations | 162 |
| 5.3.1 Pointer-Based Node Implementations | 163 |
| 5.3.2 Space Requirements | 169 |
| 5.3.3 Array Implementation for Complete Binary Trees | 170 |
| 5.4 Binary Search Trees | 171 |
| 5.5 Heaps and Priority Queues | 180 |
| 5.6 Huffman Coding Trees | 187 |
| 5.6.1 Building Huffman Coding Trees | 189 |

| | | |
|------------|---|------------|
| 5.6.2 | Assigning and Using Huffman Codes | 195 |
| 5.7 | Further Reading | 198 |
| 5.8 | Exercises | 198 |
| 5.9 | Projects | 202 |
| 6 | Non-Binary Trees | 205 |
| 6.1 | General Tree Definitions and Terminology | 205 |
| 6.1.1 | An ADT for General Tree Nodes | 206 |
| 6.1.2 | General Tree Traversals | 207 |
| 6.2 | The Parent Pointer Implementation | 208 |
| 6.3 | General Tree Implementations | 216 |
| 6.3.1 | List of Children | 217 |
| 6.3.2 | The Left-Child/Right-Sibling Implementation | 218 |
| 6.3.3 | Dynamic Node Implementations | 218 |
| 6.3.4 | Dynamic “Left-Child/Right-Sibling” Implementation | 220 |
| 6.4 | <i>K</i> -ary Trees | 221 |
| 6.5 | Sequential Tree Implementations | 223 |
| 6.6 | Further Reading | 226 |
| 6.7 | Exercises | 226 |
| 6.8 | Projects | 230 |
| III | Sorting and Searching | 233 |
| 7 | Internal Sorting | 235 |
| 7.1 | Sorting Terminology and Notation | 236 |
| 7.2 | Three $\Theta(n^2)$ Sorting Algorithms | 237 |
| 7.2.1 | Insertion Sort | 238 |
| 7.2.2 | Bubble Sort | 240 |
| 7.2.3 | Selection Sort | 241 |
| 7.2.4 | The Cost of Exchange Sorting | 243 |
| 7.3 | Shellsort | 244 |
| 7.4 | Mergesort | 246 |
| 7.5 | Quicksort | 249 |

| | |
|---|------------|
| Contents | vii |
| 7.6 Heapsort | 256 |
| 7.7 Binsort and Radix Sort | 259 |
| 7.8 An Empirical Comparison of Sorting Algorithms | 265 |
| 7.9 Lower Bounds for Sorting | 267 |
| 7.10 Further Reading | 271 |
| 7.11 Exercises | 272 |
| 7.12 Projects | 275 |
| 8 File Processing and External Sorting | 279 |
| 8.1 Primary versus Secondary Storage | 280 |
| 8.2 Disk Drives | 282 |
| 8.2.1 Disk Drive Architecture | 283 |
| 8.2.2 Disk Access Costs | 286 |
| 8.3 Buffers and Buffer Pools | 289 |
| 8.4 The Programmer's View of Files | 297 |
| 8.5 External Sorting | 298 |
| 8.5.1 Simple Approaches to External Sorting | 301 |
| 8.5.2 Replacement Selection | 304 |
| 8.5.3 Multiway Merging | 307 |
| 8.6 Further Reading | 310 |
| 8.7 Exercises | 311 |
| 8.8 Projects | 315 |
| 9 Searching | 317 |
| 9.1 Searching Unsorted and Sorted Arrays | 318 |
| 9.2 Self-Organizing Lists | 324 |
| 9.3 Bit Vectors for Representing Sets | 329 |
| 9.4 Hashing | 330 |
| 9.4.1 Hash Functions | 331 |
| 9.4.2 Open Hashing | 336 |
| 9.4.3 Closed Hashing | 337 |
| 9.4.4 Analysis of Closed Hashing | 346 |
| 9.4.5 Deletion | 350 |

| | | |
|-----------|---------------------------------|------------|
| 9.5 | Further Reading | 351 |
| 9.6 | Exercises | 352 |
| 9.7 | Projects | 355 |
| 10 | Indexing | 357 |
| 10.1 | Linear Indexing | 359 |
| 10.2 | ISAM | 361 |
| 10.3 | Tree-based Indexing | 364 |
| 10.4 | 2-3 Trees | 366 |
| 10.5 | B-Trees | 372 |
| 10.5.1 | B ⁺ -Trees | 375 |
| 10.5.2 | B-Tree Analysis | 381 |
| 10.6 | Further Reading | 382 |
| 10.7 | Exercises | 382 |
| 10.8 | Projects | 385 |
| IV | Advanced Data Structures | 387 |
| 11 | Graphs | 389 |
| 11.1 | Terminology and Representations | 390 |
| 11.2 | Graph Implementations | 394 |
| 11.3 | Graph Traversals | 397 |
| 11.3.1 | Depth-First Search | 400 |
| 11.3.2 | Breadth-First Search | 401 |
| 11.3.3 | Topological Sort | 405 |
| 11.4 | Shortest-Paths Problems | 407 |
| 11.4.1 | Single-Source Shortest Paths | 407 |
| 11.5 | Minimum-Cost Spanning Trees | 411 |
| 11.5.1 | Prim's Algorithm | 412 |
| 11.5.2 | Kruskal's Algorithm | 415 |
| 11.6 | Further Reading | 416 |
| 11.7 | Exercises | 416 |
| 11.8 | Projects | 420 |

| | |
|--|------------|
| 12 Lists and Arrays Revisited | 423 |
| 12.1 Multilists | 423 |
| 12.2 Matrix Representations | 427 |
| 12.3 Memory Management | 430 |
| 12.3.1 Dynamic Storage Allocation | 431 |
| 12.3.2 Failure Policies and Garbage Collection | 438 |
| 12.4 Further Reading | 443 |
| 12.5 Exercises | 444 |
| 12.6 Projects | 445 |
| 13 Advanced Tree Structures | 447 |
| 13.1 Tries | 447 |
| 13.2 Balanced Trees | 452 |
| 13.2.1 The AVL Tree | 453 |
| 13.2.2 The Splay Tree | 455 |
| 13.3 Spatial Data Structures | 459 |
| 13.3.1 The K-D Tree | 461 |
| 13.3.2 The PR quadtree | 466 |
| 13.3.3 Other Point Data Structures | 471 |
| 13.3.4 Other Spatial Data Structures | 471 |
| 13.4 Further Reading | 473 |
| 13.5 Exercises | 473 |
| 13.6 Projects | 475 |
| V Theory of Algorithms | 479 |
| 14 Analysis Techniques | 481 |
| 14.1 Summation Techniques | 482 |
| 14.2 Recurrence Relations | 487 |
| 14.2.1 Estimating Upper and Lower Bounds | 487 |
| 14.2.2 Expanding Recurrences | 491 |
| 14.2.3 Divide and Conquer Recurrences | 492 |
| 14.2.4 Average-Case Analysis of Quicksort | 495 |

| | |
|--|------------|
| 14.3 Amortized Analysis | 496 |
| 14.4 Further Reading | 499 |
| 14.5 Exercises | 500 |
| 14.6 Projects | 504 |
| 15 Lower Bounds | 505 |
| 15.1 Introduction to Lower Bounds Proofs | 506 |
| 15.2 Lower Bounds on Searching Lists | 508 |
| 15.2.1 Searching in Unsorted Lists | 508 |
| 15.2.2 Searching in Sorted Lists | 510 |
| 15.3 Finding the Maximum Value | 511 |
| 15.4 Adversarial Lower Bounds Proofs | 513 |
| 15.5 State Space Lower Bounds Proofs | 516 |
| 15.6 Finding the i th Best Element | 519 |
| 15.7 Optimal Sorting | 522 |
| 15.8 Further Reading | 524 |
| 15.9 Exercises | 525 |
| 15.10 Projects | 527 |
| 16 Patterns of Algorithms | 529 |
| 16.1 Greedy Algorithms | 529 |
| 16.2 Dynamic Programming | 530 |
| 16.2.1 Knapsack Problem | 531 |
| 16.2.2 All-Pairs Shortest Paths | 532 |
| 16.3 Randomized Algorithms | 534 |
| 16.3.1 Skip Lists | 536 |
| 16.4 Numerical Algorithms | 541 |
| 16.4.1 Exponentiation | 542 |
| 16.4.2 Largest Common Factor | 543 |
| 16.4.3 Matrix Multiplication | 543 |
| 16.4.4 Random Numbers | 546 |
| 16.4.5 Fast Fourier Transform | 546 |
| 16.5 Further Reading | 551 |

| | |
|--|------------|
| Contents | xi |
| 16.6 Exercises | 551 |
| 16.7 Projects | 552 |
| 17 Limits to Computation | 553 |
| 17.1 Reductions | 554 |
| 17.2 Hard Problems | 559 |
| 17.2.1 The Theory of \mathcal{NP} -Completeness | 560 |
| 17.2.2 \mathcal{NP} -Completeness Proofs | 565 |
| 17.2.3 Coping with \mathcal{NP} -Complete Problems | 569 |
| 17.3 Impossible Problems | 573 |
| 17.3.1 Uncountability | 574 |
| 17.3.2 The Halting Problem Is Unsolvable | 577 |
| 17.4 Further Reading | 581 |
| 17.5 Exercises | 581 |
| 17.6 Projects | 584 |
| Bibliography | 585 |
| Index | 591 |