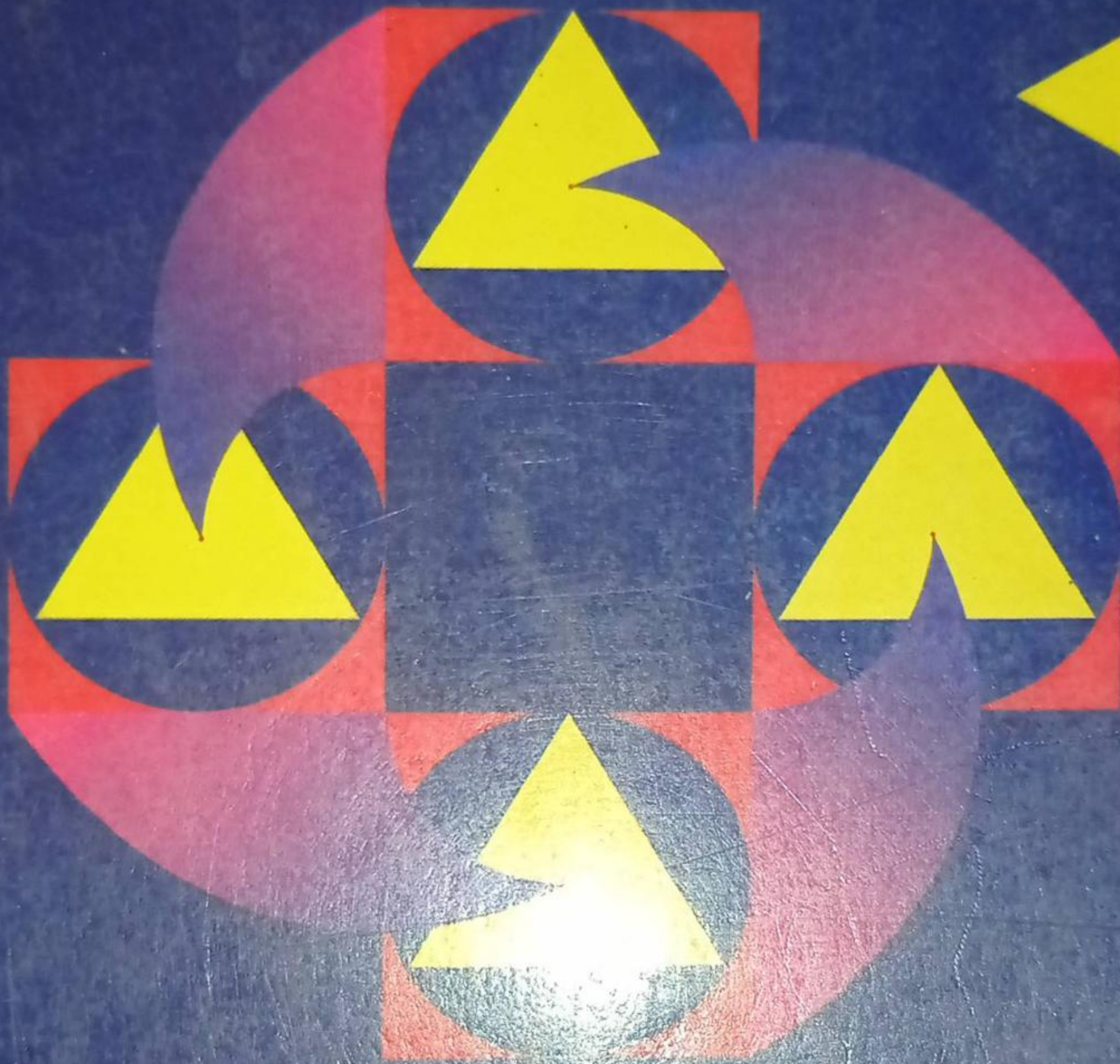


Education
Asia

COMPUTER ALGORITHMS

Introduction to Design & Analysis

*Third
Edition*



*Sara Baase
Allen Van Gelder*

Contents

Preface

vii

1 Analyzing Algorithms and Problems: Principles and Examples

1

- 1.1 Introduction 2
- 1.2 Java as an Algorithm Language 3
- 1.3 Mathematical Background 11
- 1.4 Analyzing Algorithms and Problems 30
- 1.5 Classifying Functions by Their Asymptotic Growth Rates 43
- 1.6 Searching an Ordered Array 53
 - Exercises 61
 - Notes and References 67

2 Data Abstraction and Basic Data Structures

69

- 2.1 Introduction 70
- 2.2 ADT Specification and Design Techniques 71
- 2.3 Elementary ADTs—Lists and Trees 73
- 2.4 Stacks and Queues 86
- 2.5 ADTs for Dynamic Sets 89
 - Exercises 95
 - Notes and References 100

3 Recursion and Induction

101

- 3.1 Introduction 102
- 3.2 Recursive Procedures 102
- 3.3 What Is a Proof? 108
- 3.4 Induction Proofs 111
- 3.5 Proving Correctness of Procedures 118

| | | |
|-----|----------------------|-----|
| 3.6 | Recurrence Equations | 130 |
| 3.7 | Recursion Trees | 134 |
| * | Exercises | 141 |
| | Notes and References | 146 |

4 Sorting 149

| | | |
|------|--|-----|
| 4.1 | Introduction | 150 |
| 4.2 | Insertion Sort | 151 |
| 4.3 | Divide and Conquer | 157 |
| 4.4 | Quicksort | 159 |
| 4.5 | Merging Sorted Sequences | 171 |
| 4.6 | Mergesort | 174 |
| 4.7 | Lower Bounds for Sorting by Comparison of Keys | 178 |
| 4.8 | Heapsort | 182 |
| 4.9 | Comparison of Four Sorting Algorithms | 197 |
| 4.10 | Shellsort | 197 |
| 4.11 | Radix Sorting | 201 |
| | Exercises | 206 |
| | Programs | 221 |
| | Notes and References | 221 |

5 Selection and Adversary Arguments 223

| | | |
|-------|--------------------------------------|-----|
| 5.1 | Introduction | 224 |
| 5.2 | Finding max and min | 226 |
| 5.3 | Finding the Second-Largest Key | 229 |
| 5.4 | The Selection Problem | 233 |
| * 5.5 | A Lower Bound for Finding the Median | 238 |
| 5.6 | Designing Against an Adversary | 240 |
| | Exercises | 242 |
| | Notes and References | 246 |

6 Dynamic Sets and Searching 249

| | | |
|-------|---|-----|
| 6.1 | Introduction | 250 |
| 6.2 | Array Doubling | 250 |
| 6.3 | Amortized Time Analysis | 251 |
| 6.4 | Red-Black Trees | 253 |
| 6.5 | Hashing | 275 |
| 6.6 | Dynamic Equivalence Relations and Union-Find Programs | 283 |
| * 6.7 | Priority Queues with a Decrease Key Operation | 295 |
| | Exercises | 302 |

Programs 309
 Notes and References 309

7 Graphs and Graph Traversals 313

7.1 Introduction 314
 7.2 Definitions and Representations 314
 7.3 Traversing Graphs 328
 7.4 Depth-First Search on Directed Graphs 336
 7.5 Strongly Connected Components of a Directed Graph 357
 7.6 Depth-First Search on Undirected Graphs 364
 7.7 Biconnected Components of an Undirected Graph 366
 Exercises 375
 Programs 384
 Notes and References 385

8 Graph Optimization Problems and Greedy Algorithms 387

8.1 Introduction 388
 8.2 Prim's Minimum Spanning Tree Algorithm 388
 8.3 Single-Source Shortest Paths 403
 8.4 Kruskal's Minimum Spanning Tree Algorithm 412
 Exercises 416
 Programs 421
 Notes and References 422

9 Transitive Closure, All-Pairs Shortest Paths 425

9.1 Introduction 426
 9.2 The Transitive Closure of a Binary Relation 426
 9.3 Warshall's Algorithm for Transitive Closure 430
 9.4 All-Pairs Shortest Paths in Graphs 433
 9.5 Computing Transitive Closure by Matrix Operations 436
 9.6 Multiplying Bit Matrices—Kronrod's Algorithm 439
 * Exercises 446
 Programs 449
 Notes and References 449

10 Dynamic Programming 451

10.1 Introduction 452
 10.2 Subproblem Graphs and Their Traversal 453
 10.3 Multiplying a Sequence of Matrices 457

Programs 309
 Notes and References 309

7 Graphs and Graph Traversals 313

7.1 Introduction 314
 7.2 Definitions and Representations 314
 7.3 Traversing Graphs 328
 7.4 Depth-First Search on Directed Graphs 336
 7.5 Strongly Connected Components of a Directed Graph 357
 7.6 Depth-First Search on Undirected Graphs 364
 7.7 Biconnected Components of an Undirected Graph 366
 Exercises 375
 Programs 384
 Notes and References 385

8 Graph Optimization Problems and Greedy Algorithms 387

8.1 Introduction 388
 8.2 Prim's Minimum Spanning Tree Algorithm 388
 8.3 Single-Source Shortest Paths 403
 8.4 Kruskal's Minimum Spanning Tree Algorithm 412
 Exercises 416
 Programs 421
 Notes and References 422

9 Transitive Closure, All-Pairs Shortest Paths 425

9.1 Introduction 426
 9.2 The Transitive Closure of a Binary Relation 426
 9.3 Warshall's Algorithm for Transitive Closure 430
 9.4 All-Pairs Shortest Paths in Graphs 433
 9.5 Computing Transitive Closure by Matrix Operations 436
 9.6 Multiplying Bit Matrices—Kronrod's Algorithm 439
 * Exercises 446
 Programs 449
 Notes and References 449

10 Dynamic Programming 451

10.1 Introduction 452
 10.2 Subproblem Graphs and Their Traversal 453
 10.3 Multiplying a Sequence of Matrices 457

Programs 309
 Notes and References 309

7 Graphs and Graph Traversals 313

7.1 Introduction 314
 7.2 Definitions and Representations 314
 7.3 Traversing Graphs 328
 7.4 Depth-First Search on Directed Graphs 336
 7.5 Strongly Connected Components of a Directed Graph 357
 7.6 Depth-First Search on Undirected Graphs 364
 7.7 Biconnected Components of an Undirected Graph 366
 Exercises 375
 Programs 384
 Notes and References 385

8 Graph Optimization Problems and Greedy Algorithms 387

8.1 Introduction 388
 8.2 Prim's Minimum Spanning Tree Algorithm 388
 8.3 Single-Source Shortest Paths 403
 8.4 Kruskal's Minimum Spanning Tree Algorithm 412
 Exercises 416
 Programs 421
 Notes and References 422

9 Transitive Closure, All-Pairs Shortest Paths 425

9.1 Introduction 426
 9.2 The Transitive Closure of a Binary Relation 426
 9.3 Warshall's Algorithm for Transitive Closure 430
 9.4 All-Pairs Shortest Paths in Graphs 433
 9.5 Computing Transitive Closure by Matrix Operations 436
 9.6 Multiplying Bit Matrices—Kronrod's Algorithm 439
 * Exercises 446
 Programs 449
 Notes and References 449

10 Dynamic Programming 451

10.1 Introduction 452
 10.2 Subproblem Graphs and Their Traversal 453
 10.3 Multiplying a Sequence of Matrices 457

| | | |
|------|--|-----|
| 10.4 | Constructing Optimal Binary Search Trees | 466 |
| 10.5 | Separating Sequences of Words into Lines | 471 |
| 10.6 | Developing a Dynamic Programming Algorithm | 474 |
| | Exercises | 475 |
| | Programs | 481 |
| | Notes and References | 482 |

11 String Matching 483

| | | |
|------|----------------------------------|-----|
| 11.1 | Introduction | 484 |
| 11.2 | A Straightforward Solution | 485 |
| 11.3 | The Knuth-Morris-Pratt Algorithm | 487 |
| 11.4 | The Boyer-Moore Algorithm | 495 |
| 11.5 | Approximate String Matching | 504 |
| | Exercises | 508 |
| | Programs | 512 |
| | Notes and References | 512 |

12 Polynomials and Matrices 515

| | | |
|--------|--|-----|
| 12.1 | Introduction | 516 |
| 12.2 | Evaluating Polynomial Functions | 516 |
| 12.3 | Vector and Matrix Multiplication | 522 |
| * 12.4 | The Fast Fourier Transform and Convolution | 528 |
| | Exercises | 542 |
| | Programs | 546 |
| | Notes and References | 546 |

13 NP-Complete Problems 547

| | | |
|------|--------------------------------------|-----|
| 13.1 | Introduction | 548 |
| 13.2 | \mathcal{P} and \mathcal{NP} | 548 |
| 13.3 | NP-Complete Problems | 559 |
| 13.4 | Approximation Algorithms | 570 |
| 13.5 | Bin Packing | 572 |
| 13.6 | The Knapsack and Subset Sum Problems | 577 |
| 13.7 | Graph Coloring | 581 |
| 13.8 | The Traveling Salesperson Problem | 589 |
| 13.9 | Computing with DNA | 592 |
| | Exercises | 600 |
| | Notes and References | 608 |

| | | |
|-----------|--|------------|
| 14 | Parallel Algorithms | 611 |
| 14.1 | Introduction | 612 |
| 14.2 | Parallelism, the PRAM, and Other Models | 612 |
| 14.3 | Some Simple PRAM Algorithms | 616 |
| 14.4 | Handling Write Conflicts | 622 |
| 14.5 | Merging and Sorting | 624 |
| 14.6 | Finding Connected Components | 628 |
| 14.7 | A Lower Bound for Adding n Integers | 641 |
| | Exercises | 643 |
| | Notes and References | 647 |
| A | Java Examples and Techniques | 649 |
| A.1 | Introduction | 650 |
| A.2 | A Java Main Program | 651 |
| A.3 | A Simple Input Library | 656 |
| A.4 | Documenting Java Classes | 658 |
| A.5 | Generic Order and the “Comparable” Interface | 659 |
| A.6 | Subclasses Extend the Capability of Their Superclass | 663 |
| A.7 | Copy via the “Cloneable” Interface | 667 |
| | Bibliography | 669 |
| | Index | 679 |