# CONTENTS

Preface  
Introduction .................................... 1

## PART ONE. ITERATIVE ALGORITHMS AND LOOP INVARIANTS

1 **Iterative Algorithms: Measures of Progress and Loop Invariants** . . . . . 5
   1.1 A Paradigm Shift: A Sequence of Actions vs. a Sequence of Arguments 5
   1.2 The Steps to Develop an Iterative Algorithm 8
   1.3 More about the Steps 12
   1.4 Different Types of Iterative Algorithms 21
   1.5 Typical Errors 26
   1.6 Exercises 27

2 **Examples Using More-of-the-Input Loop Invariants** . . . . . . . . . . . 29
   2.1 Coloring the Plane 29
   2.2 Deterministic Finite Automaton 31
   2.3 More of the Input vs. More of the Output 39

3 **Abstract Data Types** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
   3.1 Specifications and Hints at Implementations 44
   3.2 Link List Implementation 51
   3.3 Merging with a Queue 56
   3.4 Parsing with a Stack 57

4 **Narrowing the Search Space: Binary Search** . . . . . . . . . . . . . . . 60
   4.1 Binary Search Trees 60
   4.2 Magic Sevens 62
   4.3 VLSI Chip Testing 65
   4.4 Exercises 69

5 **Iterative Sorting Algorithms** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
   5.1 Bucket Sort by Hand 71
Contents

5.2 Counting Sort (a Stable Sort) ................. 72
5.3 Radix Sort .................................. 75
5.4 Radix Counting Sort ......................... 76

6 Euclid’s GCD Algorithm ........................ 79

7 The Loop Invariant for Lower Bounds ............ 85

PART TWO. RECURSION

8 Abstractions, Techniques, and Theory ............ 97
  8.1 Thinking about Recursion ....................... 97
  8.2 Looking Forward vs. Backward ................ 99
  8.3 With a Little Help from Your Friends .......... 100
  8.4 The Towers of Hanoi ........................ 102
  8.5 Checklist for Recursive Algorithms ............ 104
  8.6 The Stack Frame ............................ 110
  8.7 Proving Correctness with Strong Induction .... 112

9 Some Simple Examples of Recursive Algorithms .... 114
  9.1 Sorting and Selecting Algorithms ............... 114
  9.2 Operations on Integers ........................ 122
  9.3 Ackermann’s Function ........................ 127
  9.4 Exercises ................................... 128

10 Recursion on Trees ................................ 130
  10.1 Tree Traversals .............................. 133
  10.2 Simple Examples ............................. 135
  10.3 Generalizing the Problem Solved ............... 138
  10.4 Heap Sort and Priority Queues ............... 141
  10.5 Representing Expressions with Trees .......... 149

11 Recursive Images ................................ 153
  11.1 Drawing a Recursive Image from a Fixed Recursive and a Base
       Case Image .................................. 153
  11.2 Randomly Generating a Maze .................... 156

12 Parsing with Context-Free Grammars .............. 159

PART THREE. OPTIMIZATION PROBLEMS

13 Definition of Optimization Problems .............. 171

14 Graph Search Algorithms ........................ 173
  14.1 A Generic Search Algorithm ................... 174
  14.2 Breadth-First Search for Shortest Paths ....... 179
  14.3 Dijkstra’s Shortest-Weighted-Path Algorithm ... 183
  14.4 Depth-First Search ........................... 188
  14.5 Recursive Depth-First Search ................. 192
  14.6 Linear Ordering of a Partial Order ............ 194
  14.7 Exercise ................................... 196
### Contents

#### 15 Network Flows and Linear Programming

15.1 A Hill-Climbing Algorithm with a Small Local Maximum
15.2 The Primal–Dual Hill-Climbing Method
15.3 The Steepest-Ascent Hill-Climbing Algorithm
15.4 Linear Programming
15.5 Exercises

#### 16 Greedy Algorithms

16.1 Abstractions, Techniques, and Theory
16.2 Examples of Greedy Algorithms
   - 16.2.1 Example: The Job/Event Scheduling Problem
   - 16.2.2 Example: The Interval Cover Problem
   - 16.2.3 Example: The Minimum-Spanning-Tree Problem
16.3 Exercises

#### 17 Recursive Backtracking

17.1 Recursive Backtracking Algorithms
17.2 The Steps in Developing a Recursive Backtracking
17.3 Pruning Branches
17.4 Satisfiability
17.5 Exercises

#### 18 Dynamic Programming Algorithms

18.1 Start by Developing a Recursive Backtracking
18.2 The Steps in Developing a Dynamic Programming Algorithm
18.3 Subtle Points
   - 18.3.1 The Question for the Little Bird
   - 18.3.2 Subinstances and Subsolutions
   - 18.3.3 The Set of Subinstances
   - 18.3.4 Decreasing Time and Space
   - 18.3.5 Counting the Number of Solutions
   - 18.3.6 The New Code
18.4 All Pairs Using Matrix Multiplication
18.5 Parsing with Context-Free Grammars
18.6 Designing Dynamic Programming Algorithms via Reductions

© Cambridge University Press
www.cambridge.org
Contents

20 Reductions and NP-Completeness .......................................................... 324
  20.1 Satisfiability Is at Least as Hard as Any Optimization Problem .......... 326
  20.2 Steps to Prove NP-Completeness ...................................................... 330
  20.3 Example: 3-Coloring Is NP-Complete ............................................. 338
  20.4 An Algorithm for Bipartite Matching Using the Network Flow Algorithm 342

21 Randomized Algorithms ............................................................................. 346
  21.1 Using Randomness to Hide the Worst Cases ..................................... 347
  21.2 Solutions of Optimization Problems with a Random Structure ........... 350

PART FOUR. APPENDIX

22 Existential and Universal Quantifiers ....................................................... 357

23 Time Complexity ....................................................................................... 366
  23.1 The Time (and Space) Complexity of an Algorithm .......................... 366
  23.2 The Time Complexity of a Computational Problem ......................... 371

24 Logarithms and Exponentials .................................................................... 374

25 Asymptotic Growth ................................................................................... 377
  25.1 Steps to Classify a Function ............................................................... 379
  25.2 More about Asymptotic Notation ....................................................... 384

26 Adding-Made-Easy Approximations ......................................................... 388
  26.1 The Technique .................................................................................... 389
  26.2 Some Proofs for the Adding-Made-Easy Technique ......................... 393

27 Recurrence Relations ................................................................................. 398
  27.1 The Technique .................................................................................... 399
  27.2 Some Proofs ....................................................................................... 401

28 A Formal Proof of Correctness ................................................................... 408

PART FIVE. EXERCISE SOLUTIONS ................................................................. 411

Conclusion ..................................................................................................... 437

Index ............................................................................................................... 439