

# Contents

<i>Preface</i>	vii
<i>About the Authors</i>	xi
<b>Chapter 1. Algorithms</b>	<b>1</b>
1.1 Some Sorting Algorithms	1
1.1.1 Bubble sort	2
1.1.2 Insertion sort	4
1.1.3 Selection sort	5
1.1.4 Quick sort	7
1.2 Algorithm Performance/Algorithm Complexity	13
1.3 Heap Sort	18
Exercise 1.1	22
<b>Chapter 2. Growth of Functions</b>	<b>23</b>
2.1 Introduction	23
2.2 Asymptotic Notations	24
2.2.1 $O$ -notation (big oh notation)	24
2.2.2 $\Omega$ -notation (big omega notation)	32
2.2.3 $\theta$ -notation (big theta notation)	32
2.3 Some Asymptotic Notations Using Limits	35
Exercise 2.1	38
2.4 Asymptotic Notations in Equations	39
2.5 Comparison of Growth of Some Typical Functions	41
Exercise 2.2	47
Exercise 2.3	47

2.6	Some Typical Examples . . . . .	47
	Exercise 2.4 . . . . .	50
<b>Chapter 3. Backtracking</b>		<b>53</b>
3.1	Backtracking Procedure . . . . .	53
3.2	Graph Coloring . . . . .	55
3.3	$n$ -Queen Problem . . . . .	56
3.4	$n$ -Queen Problem and Permutation Tree . . . . .	61
3.5	Solving Sudoku Problem Using Backtracking . . . . .	73
	Exercise 3.1 . . . . .	78
<b>Chapter 4. Divide and Conquer</b>		<b>81</b>
4.1	Divide and Conquer General Algorithm . . . . .	81
4.2	Max–Min Problem . . . . .	82
4.3	Round-Robin Tennis Tournament . . . . .	83
4.4	The Problem of Counterfeit Coin . . . . .	89
4.5	Tiling a Defective Chess Board with Exactly One Defective Square Using Triominoes . . . . .	94
4.6	Strassen’s Matrix Multiplication Algorithm . . . . .	96
4.7	Medians and Order Statistic . . . . .	101
	4.7.1 Worst-case running time of select (order statistic) . . . . .	102
	Exercise 4.1 . . . . .	108
<b>Chapter 5. Greedy Algorithms</b>		<b>111</b>
5.1	Some Examples . . . . .	111
5.2	Knapsack Problem . . . . .	117
5.3	Job Sequencing with Deadlines . . . . .	127
5.4	Huffman Code . . . . .	136
	Exercise 5.1 . . . . .	154
<b>Chapter 6. Dynamic Programming</b>		<b>155</b>
6.1	Matrix-Chain Multiplication . . . . .	156
6.2	Multistage Graphs . . . . .	168
6.3	A Business/Industry Oriented Problem . . . . .	171
6.4	Largest Common Subsequence . . . . .	174

6.5	Optimal Binary Search Tree . . . . .	185
	Exercise 6.1 . . . . .	200
6.6	Optimal Triangulation of a Polygon . . . . .	200
	Exercise 6.2 . . . . .	220
	Exercise 6.3 . . . . .	223
<b>Chapter 7. Elementary Graph Algorithms</b>		<b>225</b>
7.1	Representations of Graphs . . . . .	225
7.2	Queues . . . . .	227
	Exercise 7.1 . . . . .	231
7.3	Shortest Paths . . . . .	231
7.4	Breadth-First Search . . . . .	232
	Exercise 7.2 . . . . .	234
	7.4.1 Breadth-first tree . . . . .	235
	Exercise 7.3 . . . . .	240
7.5	Depth-First Search . . . . .	241
7.6	Directed Acyclic Graphs . . . . .	253
	7.6.1 Topological sort . . . . .	254
7.7	Strongly Connected Components . . . . .	256
	Exercise 7.4 . . . . .	268
<b>Chapter 8. Minimal Spanning Tree</b>		<b>269</b>
8.1	Prim's Algorithm . . . . .	270
8.2	Kruskal's Algorithm . . . . .	273
8.3	Some Examples . . . . .	276
	Exercise 8.1 . . . . .	292
<b>Chapter 9. Single-Source Shortest Paths</b>		<b>293</b>
9.1	Preliminaries . . . . .	294
9.2	Dijkstra's Algorithm . . . . .	297
9.3	Bellman–Ford Algorithm . . . . .	305
	Exercise 9.1 . . . . .	312
9.4	Predecessor Subgraph . . . . .	312
9.5	Difference Constraints and Shortest Paths . . . . .	313
	Exercise 9.2 . . . . .	320

<b>Chapter 10. All Pairs Shortest Paths</b>	<b>321</b>
10.1 A Recursive Algorithm for Shortest Path Lengths . . . . .	321
10.2 The Floyd–Warshall Algorithm . . . . .	333
10.2.1 Construction of a shortest path . . . . .	337
10.3 Transitive Closure . . . . .	343
Exercise 10.1 . . . . .	349
10.4 Johnson’s Algorithm . . . . .	349
Exercise 10.2 . . . . .	357
<b>Chapter 11. Flow Networks</b>	<b>359</b>
11.1 Ford and Fulkerson Theorem . . . . .	359
11.2 Labeling Procedure for Maximal Flow and Flow Pattern . . . . .	369
11.3 Maximum Bipartite Matching . . . . .	380
Exercise 11.1 . . . . .	384
Exercise 11.2 . . . . .	386
<b>Chapter 12. Polynomial Multiplication, FFT and DFT</b>	<b>387</b>
12.1 Evaluation of a Polynomial . . . . .	387
12.2 Discrete Fourier Transforms and Polynomial Multiplication . . . . .	389
12.2.1 Representation of polynomials . . . . .	389
12.3 Primitive Roots of Unity . . . . .	391
12.4 Discrete Fourier Transforms . . . . .	392
Exercise 12.1 . . . . .	397
<b>Chapter 13. String Matching</b>	<b>399</b>
13.1 Preliminaries . . . . .	399
13.2 The Naïve String-Matching Algorithm . . . . .	402
13.3 The Rabin–Karp Algorithm . . . . .	405
Exercise 13.1 . . . . .	411
13.4 Pattern-Matching with Finite Automata . . . . .	413
Exercise 13.2 . . . . .	420
13.5 The Knuth–Morris–Pratt Algorithm . . . . .	421
13.5.1 Some auxiliary functions . . . . .	421
13.5.2 The KMP procedure . . . . .	430

<b>Chapter 14. Sorting Networks</b>	<b>439</b>
14.1 Comparison Networks . . . . .	439
14.2 Batcher's Odd-Even Mergesort . . . . .	446
Exercise 14.1 . . . . .	454
14.3 The Zero-One Principle . . . . .	455
14.4 A Bitonic Sorting Network . . . . .	459
Exercise 14.2 . . . . .	466
Exercise 14.3 . . . . .	466
14.5 Another Merging Network . . . . .	467
14.6 A Sorting Network . . . . .	472
Exercise 14.4 . . . . .	476
<b>Chapter 15. NP-Complete Problems</b>	<b>479</b>
15.1 Classes P, NP and NP-Complete . . . . .	480
15.2 Optimization Problems as Decision Problems . . . . .	482
15.3 Reducibility or Reductions . . . . .	484
Exercise 15.1 . . . . .	498
15.4 Some NP-Complete Problems . . . . .	499
Exercise 15.2 . . . . .	505
<i>Bibliography</i>	507
<i>Index</i>	511

### 1.1. Some Sorting Algorithms

We begin our study of algorithms by considering some sorting algorithms. Here we are given a sequence of non-negative integers  $a_1, a_2, \dots, a_n$ , not necessarily all distinct, the aim is to obtain a permutation of the  $a$ 's to get the numbers in a non-decreasing (or non-increasing) order i.e., to obtain a sequence  $a_{i_1}, a_{i_2}, \dots, a_{i_n}$  with  $\{a_{i_1}, a_{i_2}, \dots, a_{i_n}\} = \{a_1, a_2, \dots, a_n\}$  and  $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_n}$ . We restrict ourselves to obtaining a non-decreasing sequence.