Algorithms

Copyright ©2006 S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani

July 18, 2006

Contents

Preface **9**

0 Prologue **11**
0.1 Books and algorithms . . . 11
0.2 Enter Fibonacci . . . 12
0.3 Big-O notation . . . 15
Exercises . . . 18

1 Algorithms with numbers **21**
1.1 Basic arithmetic . . . 21
1.2 Modular arithmetic . . . 25
1.3 Primality testing . . . 33
1.4 Cryptography . . . 39
1.5 Universal hashing . . . 43
Exercises . . . 48

Randomized algorithms: a virtual chapter **39**

2 Divide-and-conquer algorithms **55**
2.1 Multiplication . . . 55
2.2 Recurrence relations . . . 58
2.3 Mergesort . . . 60
2.4 Medians . . . 64
2.5 Matrix multiplication . . . 66
2.6 The fast Fourier transform . . . 68
Exercises . . . 83

3 Decompositions of graphs **91**
3.1 Why graphs? . . . 91
3.2 Depth-first search in undirected graphs . . . 93
3.3 Depth-first search in directed graphs . . . 98
3.4 Strongly connected components . . . 101
Exercises . . . 106

4 Paths in graphs **115**
4.1 Distances . . . 115
4.2 Breadth-first search . . . 116
4.3 Lengths on edges . . . 118
4.4 Dijkstra's algorithm . . . 119
4.5 Priority queue implementations . . . 126
4.6 Shortest paths in the presence of negative edges . . . 128
4.7 Shortest paths in dags . . . 130
Exercises . . . 132

5 Greedy algorithms **139**
5.1 Minimum spanning trees . . . 139
5.2 Huffman encoding . . . 153
5.3 Horn formulas . . . 157
5.4 Set cover . . . 158
Exercises . . . 161

6 Dynamic programming **169**
6.1 Shortest paths in dags, revisited . . . 169
6.2 Longest increasing subsequences . . . 170
6.3 Edit distance . . . 174
6.4 Knapsack . . . 181
6.5 Chain matrix multiplication . . . 184
6.6 Shortest paths . . . 186
6.7 Independent sets in trees . . . 189
Exercises . . . 191

7 Linear programming and reductions **201**
7.1 An introduction to linear programming . . . 201
7.2 Flows in networks . . . 211
7.3 Bipartite matching . . . 219
7.4 Duality . . . 220
7.5 Zero-sum games . . . 224
7.6 The simplex algorithm . . . 227
7.7 Postscript: circuit evaluation . . . 236
Exercises . . . 239

8 NP-complete problems **247**
8.1 Search problems . . . 247
8.2 NP-complete problems . . . 257
8.3 The reductions . . . 262
Exercises . . . 278

9 Coping with NP-completeness **283**
9.1 Intelligent exhaustive search . . . 284
9.2 Approximation algorithms . . . 290
9.3 Local search heuristics . . . 297
Exercises . . . 306

10 Quantum algorithms **311**
10.1 Qubits, superposition, and measurement . . . 311
10.2 The plan . . . 315
10.3 The quantum Fourier transform . . . 316
10.4 Periodicity . . . 318
10.5 Quantum circuits . . . 321
10.6 Factoring as periodicity . . . 324
10.7 The quantum algorithm for factoring . . . 326
Exercises . . . 329

Historical notes and further reading **331**

Index **333**

List of boxes

Bases and logs . . . 21
Two's complement . . . 27
Is your social security number a prime? . . . 33
Hey, that was group theory! . . . 36
Carmichael numbers . . . 37
Randomized algorithms: a virtual chapter . . . 39
An application of number theory? . . . 40

Binary search . . . 60
An $n \log n$ lower bound for sorting . . . 62
The Unix `sort` command . . . 66
Why multiply polynomials? . . . 68
The slow spread of a fast algorithm . . . 82

How big is your graph? . . . 93
Crawling fast . . . 105

Which heap is best? . . . 125

Trees . . . 140
A randomized algorithm for minimum cut . . . 150
Entropy . . . 155

Recursion? No, thanks. . . . 173
Programming? . . . 173
Common subproblems . . . 177
Of mice and men . . . 179
Memoization . . . 183
On time and memory . . . 189

A magic trick called duality . . . 205
Reductions . . . 209
Matrix-vector notation . . . 211
Visualizing duality . . . 222
Gaussian elimination . . . 234

Linear programming in polynomial time 236

The story of Sissa and Moore 247
Why **P** and **NP**? 258
The two ways to use reductions 259
Unsolvable problems 276

Entanglement 314
The Fourier transform of a periodic vector 320
Setting up a periodic superposition 325
Quantum physics meets computation 327