© 2011 Taylor and Francis Group, LLC

Contents

1 Introduction **1**
1.1 Computation . . . 3
1.2 The Limits of Computation . . . 6
1.3 Algorithms . . . 10
1.4 A Laboratory for Computational Experiments . . . 12

2 The Ruby Workbench **19**
Introducing Ruby and the RubyLabs environment for computational experiments
2.1 Interactive Ruby . . . 20
2.2 Numbers . . . 24
2.3 Variables . . . 27
2.4 Methods . . . 32
2.5 RubyLabs . . . 37
2.6 Summary . . . 39

3 The Sieve of Eratosthenes **45**
An algorithm for finding prime numbers
3.1 The Sieve Algorithm . . . 46
3.2 The mod Operator . . . 48
3.3 Containers . . . 50
3.4 Iterators . . . 55
3.5 Boolean Values and the `delete_if` Method . . . 57
3.6 Exploring the Algorithm . . . 60
3.7 The `sieve` Method . . . 62
3.8 A Better Sieve . . . 65
3.9 Experiments with the Sieve . . . 67
3.10 Summary . . . 71

4 A Journey of a Thousand Miles **75**
Iteration as a strategy for solving computational problems
4.1 Searching and Sorting . . . 76
4.2 The Linear Search Algorithm . . . 80
4.3 The Insertion Sort Algorithm . . . 88
4.4 Scalability . . . 91
4.5 ♦ Best Case, Worst Case . . . 96
4.6 Summary . . . 98

5 Divide and Conquer **103**
A new strategy: breaking large problems into smaller subproblems
5.1 Binary Search . . . 106
5.2 Binary Search Experiments . . . 111
5.3 Merge Sort . . . 116
5.4 Merge Sort Experiments . . . 120
5.5 ♦ Recursive Methods . . . 123
5.6 Summary . . . 126

6 When Words Collide **131**
Organizing data for more efficient problem solving
6.1 Word Lists . . . 133
6.2 Hash Tables . . . 136
6.3 The mod Function Again . . . 140
6.4 Collisions . . . 147
6.5 Hash Table Experiments . . . 152
6.6 Summary . . . 156

7 Bit by Bit **161**
Binary codes and algorithms for text compression and error detection
7.1 Binary Codes . . . 163
7.2 Codes for Characters . . . 169
7.3 Parity Bits . . . 173
7.4 Huffman Trees . . . 178
7.5 Huffman Codes . . . 185
7.6 Summary . . . 192

8 The War of the Words **197**
An introduction to computer architecture and assembly language programming
8.1 Hello, MARS . . . 200
8.2 The Temperature on MARS . . . 204
8.3 Corewar . . . 209
8.4 Self-Referential Code . . . 212
8.5 ♦ Clones . . . 214
8.6 Summary . . . 216

9 Now for Something Completely Different **221**
An algorithm for generating random numbers
9.1 Pseudorandom Numbers . . . 222
9.2 Numbers on Demand . . . 229
9.3 Games with Random Numbers . . . 230
9.4 Random Shuffles . . . 234
9.5 Tests of Randomness . . . 239
9.6 Summary . . . 244

© 2011 Taylor and Francis Group, LLC

10 Ask Dr. Ruby **247**
A program that understands English (or does it?)
10.1 Overview of ELIZA . . . 249
10.2 Sentence Patterns . . . 252
10.3 Building Responses from Parts of Sentences . . . 257
10.4 Substitutions . . . 262
10.5 An Algorithm for Having a Conversation . . . 265
10.6 ♦ Writing Scripts for ELIZA . . . 270
10.7 ELIZA and the Turing Test . . . 273
10.8 Summary . . . 275

11 The Music of the Spheres **279**
Computer simulation and the N-body problem
11.1 Running Around in Circles . . . 283
11.2 The Force of Gravity . . . 288
11.3 Force Vectors . . . 295
11.4 *N*-Body Simulation of the Solar System . . . 300
11.5 Summary . . . 304

12 The Traveling Salesman **309**
A genetic algorithm for a computationally demanding problem
12.1 Maps and Tours . . . 312
12.2 Exhaustive Search . . . 315
12.3 Random Search . . . 320
12.4 Point Mutations . . . 324
12.5 The Genetic Algorithm . . . 326
12.6 Crossovers . . . 333
12.7 Summary . . . 336

A Answers to Selected Exercises **341**

B Ruby Reference **347**

© 2011 Taylor and Francis Group, LLC