

Contents

Preface	ix
1 Introduction	1
1.1 Computation	3
1.2 The Limits of Computation	5
1.3 Algorithms	9
1.4 A Laboratory for Computational Experiments	12
1.5 Summary	14
2 The Python Workbench	17
<i>Introducing Python and an environment for interactive experiments</i>	
2.1 Interacting with Python	18
2.2 Numbers, Functions, and the Math Module	24
2.3 Variables	27
2.4 Defining New Functions: <code>def</code> Statements	31
2.5 Conditional Execution: <code>if</code> Statements	36
2.6 Strings	39
2.7 Objects	45
2.8 Summary	47
3 The Sieve of Eratosthenes	55
<i>An algorithm for finding prime numbers</i>	
3.1 The Algorithm	56
3.2 Lists	58
3.3 Iterating Over Lists: <code>for</code> Statements	62
3.4 List Indexes	65
3.5 The Animated Sieve	69
3.6 A Helper Function	74
3.7 The <code>sieve</code> Function	76
3.8 ♦ Running a Program from the Command Line	81
3.9 Summary	84

4	A Journey of a Thousand Miles	91
	<i>Iteration as a strategy for solving computational problems</i>	
4.1	The Linear Search Algorithm	92
4.2	Implementing Linear Search	96
4.3	The Insertion Sort Algorithm	99
4.4	A Helper Function for Insertion Sort	101
4.5	Implementing Insertion Sort	105
4.6	Scalability	107
4.7	Summary	112
5	Divide and Conquer	117
	<i>A new strategy: Breaking large problems into smaller subproblems</i>	
5.1	The Binary Search Algorithm	120
5.2	Implementing Binary Search	122
5.3	Binary Search Experiments	126
5.4	The Merge Sort Algorithm	129
5.5	Implementing Merge Sort	131
5.6	Merge Sort Experiments	136
5.7	◆ Recursive Functions	138
5.8	Summary	141
6	Spam, Spam, Spam, Mail, and Spam	147
	<i>A machine learning approach to filtering junk mail</i>	
6.1	A Closer Look at Strings	148
6.2	Reading Text from a File	154
6.3	Counting Words in a Text File	159
6.4	Dictionaries	161
6.5	Word Frequency	166
6.6	Spamicity	172
6.7	An Algorithm for Identifying Junk Mail	176
6.8	Summary	183
7	Now for Something Completely Different	191
	<i>An algorithm for generating random numbers</i>	
7.1	Pseudorandom Numbers	192
7.2	Numbers on Demand	199
7.3	Modules and Encapsulation	202
7.4	Games with Random Numbers	208
7.5	Random Shuffles	212
7.6	Tests of Randomness	218
7.7	Defining New Objects: class Statements	224
7.8	◆ Additional Features for the Card Class	234
	7.8.1 ◆ Class Variables	234
	7.8.2 ◆ Exceptions	236
	7.8.3 ◆ Optional Arguments	239
7.9	Summary	242

8 Bit by Bit	249
<i>Binary codes and algorithms for text compression and error detection</i>	
8.1 Binary Codes	251
8.2 Codes for Characters	256
8.3 Error Detection	262
8.4 Parity Bits and the Exclusive OR	265
8.5 Huffman Trees	269
8.6 Huffman Codes	276
8.7 Summary	282
9 The War of the Words	289
<i>Computer architecture and machine level programming</i>	
9.1 Hello, MARS	292
9.2 The Temperature on MARS	296
9.3 Corewar	302
9.4 Self-Referential Code	306
9.5 ♦ Clones	309
9.6 Summary	312
10 I'd Like to Have an Argument, Please	315
<i>A program that understands English (or does it?)</i>	
10.1 Overview of ELIZA	317
10.2 Sentence Patterns	321
10.3 Building Responses from Parts of Sentences	325
10.4 Substitutions	329
10.5 An Algorithm for Having a Conversation	333
10.6 ♦ Writing Scripts for ELIZA	338
10.7 ELIZA and the Turing Test	339
10.8 Summary	342
11 The Music of the Spheres	349
<i>Computer simulation and the N-body problem</i>	
11.1 Running Around in Circles	353
11.2 The Force of Gravity	358
11.3 Force Vectors	365
11.4 N-Body Simulation of the Solar System	369
11.5 Summary	373
12 The Traveling Salesman	379
<i>A genetic algorithm for a computationally demanding problem</i>	
12.1 Maps and Tours	382
12.2 Exhaustive Search	385
12.3 Random Search	390

12.4 Point Mutations	393
12.5 The Genetic Algorithm	396
12.6 Crossovers	403
12.7 ♦ TSP Helper Functions	406
12.8 Summary	410

Index	417
--------------	------------