

Table of Contents

Part I: Programming environments for motion, graphics, and geometry	7
1. Reducing a task to given primitives: programming motion	9
A robot car, its capabilities, and the task to be performed	9
Wall-following algorithm described informally	10
Algorithm specified in a high-level language	11
Algorithm programmed in the robot's language	12
The robot's program optimized	12
2. Graphics primitives and environments	14
Turtle graphics: a basic environment	14
QuickDraw: a graphics toolbox	16
A graphics frame program	19
3. Algorithm animation	24
Computer-driven visualization: characteristics and techniques	24
A gallery of algorithm snapshots	27
Part II: Programming concepts: beyond notation	33
4. Algorithms and programs as literature: substance and form	34
Programming in the large versus programming in the small	34
Documentation versus literature: is it meant to be read?	35
Pascal and its dialects: lingua franca of computer science	40
5. Divide-and-conquer and recursion	45
An algorithmic principle	45
Divide-and-conquer expressed as a diagram: merge sort	46
Recursively defined trees	47
Recursive tree traversal	49
Recursion versus iteration: the Tower of Hanoi	50
The flag of Alfanumerica: an algorithmic novel on iteration and recursion	52
6. Syntax	53
Syntax and semantics	53
Grammars and their representation: syntax diagrams and EBNF	54
An overly simple syntax for simple expressions	57
Parenthesis-free notation for arithmetic expressions	59
7. Syntax analysis	62
The role of syntax analysis	62
Syntax analysis of parenthesis-free expressions by counting	63
Analysis by recursive descent	64
Turning syntax diagrams into a parser	65
Part III: Objects, algorithms, programs	67
8. Truth values, the data type 'set', and bit acrobatics	69
Bits and boolean functions	69
Swapping and crossovers: the versatile exclusive-or	70
The bit sum or "population count"	71
9. Ordered sets	78
Sequential search	78
Binary search	79
In-place permutation	82
10. Strings	87
Recognizing a pattern consisting of a single string	87
Recognizing a set of strings: a finite-state-machine interpreter	88
11. Matrices and graphs: transitive closure	93

Paths in a graph	93
Boolean matrix multiplication	94
Warshall's algorithm	95
Minimum spanning tree in a graph	97
12. Integers	100
Operations on integers	100
The Euclidean algorithm	102
The prime number sieve of Eratosthenes	103
Large integers	104
Modular number systems: the poor man's large integers	105
Random numbers	107
13. Reals	110
Floating-point numbers	110
Some dangers	112
Horner's method	113
Bisection	114
Newton's method for computing the square root	115
14. Straight lines and circles	119
Intersection	119
Clipping	122
Drawing digitized lines	123
The riddle of the braiding straight lines	126
Digitized circles	131
Part IV: Complexity of problems and algorithms	134
15. Computability and complexity	135
Models of computation: the ultimate RISC	135
Almost nothing is computable	138
The halting problem is undecidable	139
Computable, yet unknown	140
Multiplication of complex numbers	142
Complexity of matrix multiplication	142
16. The mathematics of algorithm analysis	146
Growth rates and orders of magnitude	146
Asymptotics	147
Summation formulas	148
Recurrence relations	150
Asymptotic performance of divide-and-conquer algorithms	153
Permutations	154
Trees	155
17. Sorting and its complexity	158
What is sorting? How difficult is it?	158
Types of sorting algorithms	160
Simple sorting algorithms that work in time $\Theta(n^2)$	163
A lower bound $\Omega(n \cdot \log n)$	165
Quicksort	166
Analysis for three cases: best, "typical", and worst	169
Is it possible to sort in linear time?	174
Sorting networks	174
Part V: Data structures	179
18. What is a data structure?	180
Data structures old and new	180

<u>The range of data structures studied</u>	181
<u>Performance criteria and measures</u>	182
19. <u>Abstract data types</u>	184
<u>Concepts: What and why?</u>	184
<u>Stack</u>	185
<u>First-in-first-out queue</u>	189
<u>Priority queue</u>	190
<u>Dictionary</u>	191
20. <u>Implicit data structures</u>	196
<u>What is an implicit data structure?</u>	196
<u>Array storage</u>	197
<u>Implementation of the fixed-length fifo queue as a circular buffer</u>	202
<u>Implementation of the fixed-length priority queue as a heap</u>	205
<u>Heapsort</u>	209
21. <u>List structures</u>	211
<u>Lists, memory management, pointer variables</u>	211
<u>The fifo queue implemented as a one-way list</u>	214
<u>Tree traversal</u>	214
<u>Binary search trees</u>	223
<u>Height-balanced trees</u>	228
22. <u>Address computation</u>	239
<u>Concepts and terminology</u>	239
<u>The special case of small key domains</u>	240
<u>The special case of perfect hashing: table contents known a priori</u>	241
<u>Conventional hash tables: collision resolution</u>	242
<u>Choice of hash function: randomization</u>	246
<u>Performance analysis</u>	248
<u>Extendible hashing</u>	249
<u>A virtual radix tree: order-preserving extendible hashing</u>	251
23. <u>Metric data structures</u>	254
<u>Organizing the embedding space versus organizing its contents</u>	254
<u>Radix trees, tries</u>	255
<u>Quadtrees and octtrees</u>	255
<u>Spatial data structures: objectives and constraints</u>	257
<u>The grid file</u>	259
<u>Simple geometric objects and their parameter spaces</u>	263
<u>Region queries of arbitrary shape</u>	264
<u>Evaluating region queries with a grid file</u>	267
<u>Interaction between query processing and data access</u>	267
<u>Part VI: Interaction between algorithms and data structures: case studies in geometric computation</u>	271
24. <u>Sample problems and algorithms</u>	272
<u>Geometry and geometric computation</u>	272
<u>Convex hull: a multitude of algorithms</u>	273
<u>The uses of convexity: basic operations on polygons</u>	277
<u>Visibility in the plane: a simple algorithm whose analysis is not</u>	279
25. <u>Plane-sweep: a general-purpose algorithm for two-dimensional problems illustrated using line segment intersection</u>	286
<u>The line segment intersection test</u>	286
<u>The skeleton: Turning a space dimension into a time dimension</u>	288
<u>Data structures</u>	288

This book is licensed under a [Creative Commons Attribution 3.0 License](#)

Updating the y-table and detecting an intersection	289
Sweeping across intersections	290
Degenerate configurations, numerical errors, robustness	291
26. The closest pair	293
The problem	293
Plane-sweep applied to the closest pair problem	294
Implementation	295
Analysis	297
Sweeping in three or more dimensions	298