## grokking algorithms

An *illustrated* guide for programmers and other curious people

Aditya Y. Bhargava



For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

> Special Sales Department Manning Publications Co. 20 Baldwin Road, PO Box 761 Shelter Island, NY 11964 Email: orders@manning.com

©2016 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

⊗ Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.



Manning Publications Co. 20 Baldwin Road Shelter Island, NY 11964 Development editor: Jennifer Stout Technical development editor: Damien White Project manager: Tiffany Taylor Copyeditor: Tiffany Taylor Technical proofreader: Jean-François Morin Typesetter: Leslie Haimes Cover and interior design: Leslie Haimes Illustrations by the author

ISBN: 9781617292231 Printed in the United States of America 1 2 3 4 5 6 7 8 9 10 – EBM – 21 20 19 18 17 16 For my parents, Sangeeta and Yogesh



## contents

preface	xiii
acknowledgments	xiv
about this book	xv
<b>1</b> Introduction to algorithms	1
Introduction	1
What you'll learn about performance	2
What you'll learn about solving problems	2
Binary search	3
A better way to search	5
Running time	10
Big O notation	10
Algorithm running times grow at different rates	11
Visualizing different Big O run times	13
Big O establishes a worst-case run time	15
Some common Big O run times	15
The traveling salesperson	17
Recap	19
2 Selection sort	21
How memory works	22
Arrays and linked lists	24
Linked lists	25
Arrays	26
Terminology	27
Inserting into the middle of a list	29
Deletions	30

Selection sort	32
Recap	36
<b>3</b> Recursion	37
Recursion	38
Base case and recursive case	40
The stack	42
The call stack	43
The call stack with recursion	45
Recap	50
4 Quicksort	51
Divide & conquer	52
Quicksort	60
Big O notation revisited	66
Merge sort vs. quicksort	67
Average case vs. worst case	68
Recap	72
<b>5</b> Hash tables	73
Hash functions	76
Use cases	79
Using hash tables for lookups	79
Preventing duplicate entries	81
Using hash tables as a cache	83
Recap	86
Collisions	86
Collisions Performance	86 88
Collisions Performance Load factor	86 88 90
<b>Collisions</b> <b>Performance</b> Load factor A good hash function	86 88 90 92
Collisions Performance Load factor A good hash function Recap	86 88 90 92 93
Collisions Performance Load factor A good hash function Recap 6 Breadth-first search	86 88 90 92 93 95
Collisions Performance Load factor A good hash function Recap 6 Breadth-first search Introduction to graphs	86 88 90 92 93 95 95
Collisions Performance Load factor A good hash function Recap 6 Breadth-first search Introduction to graphs What is a graph?	86 88 90 92 93 95 95 96 98
Collisions Performance Load factor A good hash function Recap 6 Breadth-first search Introduction to graphs What is a graph? Breadth-first search	86 88 90 92 93 95 95 96 98 99

Queues	103
Implementing the graph	105
Implementing the algorithm	107
Running time	111
Recap	114
<b>7</b> Dijkstra's algorithm	115
Working with Dijkstra's algorithm	116
Terminology	120
Trading for a piano	122
Negative-weight edges	128
Implementation	131
Recap	140
8 Greedy algorithms	141
The classroom scheduling problem	142
The knapsack problem	144
The set-covering problem	146
Approximation algorithms	147
NP-complete problems	152
Traveling salesperson, step by step	153
How do you tell if a problem is NP-complete?	158
Recap	160
<b>9</b> Dynamic programming	161
The knapsack problem	161
The simple solution	162
Dynamic programming	163
Knapsack problem FAQ	171
What happens if you add an item?	171
What happens if you change the order of the rows?	174
Can you fill in the grid column-wise instead	
of row-wise?	174
What happens if you add a smaller item?	174
Can you steal fractions of an item?	175
Optimizing your travel itinerary	175
Handling items that depend on each other	177

Is it possible that the solution will require	
more than two sub-knapsacks?	177
Is it possible that the best solution doesn't fill	
the knapsack completely?	178
Longest common substring	178
Making the grid	179
Filling in the grid	180
The solution	182
Longest common subsequence	183
Longest common subsequence—solution	184
Recap	186
<b>10</b> K-nearest neighbors	187
Classifying oranges vs. grapefruit	187
Building a recommendations system	189
Feature extraction	191
Regression	195
Picking good features	198
Introduction to machine learning	199
OCR	199
Building a spam filter	200
Predicting the stock market	201
Recap	201
<b>11</b> Where to go next	203
Trees	203
Inverted indexes	206
The Fourier transform	207
Parallel algorithms	208
MapReduce	209
Why are distributed algorithms useful?	209
The map function	209
The reduce function	210
Bloom filters and HyperLogLog	211
Bloom filters	212

HyperLogLog	213
The SHA algorithms	213
Comparing files	214
Checking passwords	215
Locality-sensitive hashing	216
Diffie-Hellman key exchange	217
Linear programming	218
Epilogue	219
answers to exercises	221
index	235