

Python Algorithms

Mastering Basic Algorithms in the
Python Language



Magnus Lie Hetland

Apress®

Python Algorithms: Mastering Basic Algorithms in the Python Language

Copyright © 2010 by Magnus Lie Hetland

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-3237-7

ISBN-13 (electronic): 978-1-4302-3238-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning

Lead Editor: Frank Pohlmann

Development Editor: Douglas Pundick

Technical Reviewer: Alex Martelli

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell,

Jonathan Gennick, Jonathan Hassell, Michelle Lowman, Matthew Moodie,

Duncan Parkes, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Coordinating Editor: Adam Heath

Compositor: Mary Sudul

Indexer: Brenda Miller

Artist: April Milne

Cover Designer: Anna Ishchenko

Photo Credit: Kai T. Dragland

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/info/bulksales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at www.apress.com

*For my students.
May your quest for knowledge be richly rewarded.*

Contents at a Glance

Contents.....	vi
About the Author	xiii
About the Technical Reviewer	xiv
Acknowledgments	xv
Preface	xvi
■ Chapter 1: Introduction.....	1
■ Chapter 2: The Basics	9
■ Chapter 3: Counting 101	45
■ Chapter 4: Induction and Recursion ... and Reduction.....	71
■ Chapter 5: Traversal: The Skeleton Key of Algorithmics	101
■ Chapter 6: Divide, Combine, and Conquer.....	125
■ Chapter 7: Greed Is Good? Prove It!.....	151
■ Chapter 8: Tangled Dependencies and Memoization	175
■ Chapter 9: From A to B with Edsger and Friends	199
■ Chapter 10: Matchings, Cuts, and Flows	221
■ Chapter 11: Hard Problems and (Limited) Sloppiness	241
■ Appendix A: Pedal to the Metal: Accelerating Python	271
■ Appendix B: List of Problems and Algorithms	275
■ Appendix C: Graph Terminology.....	285
■ Appendix D: Hints for Exercises.....	291
■ Index	307

Contents

Contents at a Glance.....	v
About the Author	xiii
About the Technical Reviewer	xiv
Acknowledgments	xv
Preface	xvi
■ Chapter 1: Introduction.....	1
What's All This, Then?	2
Why Are You Here?	3
Some Prerequisites	4
What's in This Book	5
Summary	6
If You're Curious	6
Exercises	7
References.....	7
■ Chapter 2: The Basics	9
Some Core Ideas in Computing	9
Asymptotic Notation	10
It's Greek to Me!	12
Rules of the Road	14
Taking the Asymptotics for a Spin.....	16
Three Important Cases	19
Empirical Evaluation of Algorithms.....	20

Implementing Graphs and Trees.....	23
Adjacency Lists and the Like.....	25
Adjacency Matrices	29
Implementing Trees.....	32
A Multitude of Representations	35
Beware of Black Boxes.....	36
Hidden Squares	37
The Trouble with Floats	38
Summary	40
If You're Curious	41
Exercises	42
References.....	43
■ Chapter 3: Counting 101	45
The Skinny on Sums	45
More Greek.....	46
Working with Sums	46
A Tale of Two Tournaments.....	47
Shaking Hands.....	47
The Hare and the Tortoise	49
Subsets, Permutations, and Combinations.....	53
Recursion and Recurrences.....	56
Doing It by Hand	57
A Few Important Examples.....	58
Guessing and Checking	62
The Master Theorem: A Cookie-Cutter Solution	64
So What Was All <i>That</i> About?	67
Summary	68
If You're Curious	69
Exercises	69

References.....	70
■ Chapter 4: Induction and Recursion ... and Reduction.....	71
Oh, That's Easy!	72
One, Two, Many	74
Mirror, Mirror	76
Designing with Induction (and Recursion)	81
Finding a Maximum Permutation	81
The Celebrity Problem	85
Topological Sorting.....	87
Stronger Assumptions	91
Invariants and Correctness.....	92
Relaxation and Gradual Improvement.....	93
Reduction + Contraposition = Hardness Proof	94
Problem Solving Advice	95
Summary	96
If You're Curious	97
Exercises	97
References.....	99
■ Chapter 5: Traversal: The Skeleton Key of Algorithmics	101
A Walk in the Park	107
No Cycles Allowed	108
How to Stop Walking in Circles	109
Go Deep!	110
Depth-First Timestamps and Topological Sorting (Again).....	112
Infinite Mazes and Shortest (Unweighted) Paths.....	114
Strongly Connected Components	118
Summary	121

If You're Curious . . .	122
Exercises	122
References	123
■ Chapter 6: Divide, Combine, and Conquer	125
Tree-Shaped Problems: All About the Balance	125
The Canonical D&C Algorithm	128
Searching by Halves	129
Traversing Search Trees . . . with Pruning	130
Selection	133
Sorting by Halves	135
How Fast Can We Sort?	137
Three More Examples	138
Closest Pair	138
Convex Hull	140
Greatest Slice	142
Tree Balance . . . and Balancing	143
Summary	148
If You're Curious . . .	149
Exercises	149
References	150
■ Chapter 7: Greed Is Good? Prove It!	151
Staying Safe, Step by Step	151
The Knapsack Problem	155
Fractional Knapsack	155
Integer Knapsack	156
Huffman's Algorithm	156
The Algorithm	158
The First Greedy Choice	159

Going the Rest of the Way	160
Optimal Merging	160
Minimum spanning trees	161
The Shortest Edge	162
What About the Rest?	163
Kruskal's Algorithm	164
Prim's Algorithm	166
Greed Works. But When?	168
Keeping Up with the Best	168
No Worse Than Perfect	169
Staying Safe	170
Summary	172
If You're Curious	172
Exercises	173
References.....	174
 ■ Chapter 8: Tangled Dependencies and Memoization	 175
Don't Repeat Yourself	176
Shortest Paths in Directed Acyclic Graphs	182
Longest Increasing Subsequence.....	184
Sequence Comparison.....	187
The Knapsack Strikes Back	190
Binary Sequence Partitioning	193
Summary	196
If You're Curious	196
Exercises	197
References.....	198

■ Chapter 9: From A to B with Edsger and Friends	199
Propagating Knowledge.....	200
Relaxing like Crazy	201
Finding the Hidden DAG.....	204
All Against All.....	206
Far-Fetched Subproblems	208
Meeting in the Middle.....	211
Knowing Where You're Going	213
Summary	217
If You're Curious	218
Exercises	218
References.....	219
■ Chapter 10: Matchings, Cuts, and Flows	221
Bipartite Matching	222
Disjoint Paths.....	225
Maximum Flow	227
Minimum Cut	231
Cheapest Flow and the Assignment Problem	232
Some Applications.....	234
Summary	237
If You're Curious	237
Exercises	238
References.....	239
■ Chapter 11: Hard Problems and (Limited) Sloppiness	241
Reduction Redux.....	241
Not in Kansas Anymore?.....	244
Meanwhile, Back in Kansas	246

But Where Do You Start? And Where Do You Go from There?	249
A Ménagerie of Monsters.....	254
Return of the Knapsack	254
Cliques and Colorings	256
Paths and Circuits	258
When the Going Gets Tough, the Smart Get Sloppy.....	261
Desperately Seeking Solutions	263
And the Moral of the Story Is	265
Summary	267
If You're Curious	267
Exercises	267
References.....	269
■ Appendix A: Pedal to the Metal: Accelerating Python	271
■ Appendix B: List of Problems and Algorithms	275
■ Appendix C: Graph Terminology.....	285
■ Appendix D: Hints for Exercises.....	291
Index.....	307

About the Author



■ **Magnus Lie Hetland** is an experienced Python programmer, having used the language since the late 90s. He is also an associate professor of algorithms at the Norwegian University of Science and Technology and has taught algorithms for the better part of a decade. Hetland is the author of *Beginning Python* (originally *Practical Python*).

About the Technical Reviewer



■ **Alex Martelli** was born and grew up in Italy and holds a Laurea in Ingegneria Elettronica from the Università di Bologna. He wrote *Python in a Nutshell* and coedited the *Python Cookbook*. He's a member of the PSF and won the 2002 Activators' Choice Award and the 2006 Frank Willison Award for contributions to the Python community. He currently lives in California and works as senior staff engineer for Google. His detailed profile is at www.google.com/profiles/aleaxit; a summary bio is at http://en.wikipedia.org/wiki/Alex_Martelli.

Acknowledgments

Thanks to everyone who contributed to this book, either directly or indirectly. This certainly includes my algorithm mentors, Arne Halaas and Bjørn Olstad, as well as the entire crew at Apress and my brilliant tech editor, Alex. Thanks to Nils Grimsmo, Jon Marius Venstad, Ole Edsberg, Rolv Seehuus, and Jorg Rødsjø for useful input; to my parents, Kjersti Lie and Tor M. Hetland, and my sister, Anne Lie-Hetland, for their interest and support; and to my uncle Axel, for checking my French. Finally, a big thank-you to the Python Software Foundation for their permission to reproduce parts of the Python standard library and to Randall Munroe for letting me include some of his wonderful XKCD comics.

Preface

This book is a marriage of three of my passions: algorithms, Python programming, and explaining things. To me, all three of these are about aesthetics—finding just the right way of doing something, looking until you uncover a hint of elegance, and then polishing that until it shines. Or at least until it is a bit shinier. Of course, when there's a lot of material to cover, you may not get to polish things quite as much as you want. Luckily, though, most of the contents in this book is prepolished, because I'm writing about really beautiful algorithms and proofs, as well as one of the cutest programming languages out there. As for the third part, I've tried hard to find explanations that will make things seem as obvious as possible. Even so, I'm sure I have failed in many ways, and if you have suggestions for improving the book, I'd be happy to hear from you. Who knows, maybe some of your ideas could make it into a future edition? For now, though, I hope you have fun with what's here and that you take any newfound insight and run with it. If you can, use it to make the world a more awesome place, in whatever way seems right.