

Computational Thinking

Paolo Ferragina • Fabrizio Luccio

Computational Thinking

First Algorithms, Then Code

 Springer

Paolo Ferragina
Dipartimento di Informatica
University of Pisa
Pisa, Italy

Fabrizio Luccio
Dipartimento di Informatica
University of Pisa
Pisa, Italy

ISBN 978-3-319-97939-7 ISBN 978-3-319-97940-3 (eBook)
<https://doi.org/10.1007/978-3-319-97940-3>

Library of Congress Control Number: 2018952631

© Springer Nature Switzerland AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The main purpose of this text is to introduce readers to algorithms and their coding for solving problems in different fields, some of which are of major importance in today's world. The presentation is accessible to anyone with elementary mathematical knowledge and is made as simple as possible, without giving up the scientific rigor that is indispensable for transforming general ideas into executable algorithms. In the course of the discussion, however, arithmetic notions that are particularly relevant for the construction of algorithms are referred to as needed.

To make it as usable as possible, the text is divided into substantially independent chapters that can be read separately, leaving the reader the freedom to choose the topics he or she prefers. In a few cases more challenging side arguments have been reported which can be omitted without affecting the understanding of the main text.

In addition, readers interested in implementing the algorithms described in the text on a computer may, even without any knowledge of “programming”, examine their operation and perhaps try to build their own programs through a very accessible website created for this purpose.¹ It contains the coding of these algorithms in Python and provides everything a reader would need to execute them and, perhaps, construct new examples.

Pisa,
February 2018

*Paolo Ferragina
Fabrizio Luccio*

¹ <https://github.com/ComputationalThinking-Springer/FirstAlgorithmsThenCode>

Contents

1	A Brief Historical Outline	1
2	A Problem with Which to Begin	5
3	Algorithms and Coding	11
3.1	The Pseudocode	12
3.2	A More Useful Program	16
3.3	Interacting with the Machine	18
3.4	Binary Search	19
4	The Tournament	25
4.1	A Lower Bound to the Problem of Maximum	28
4.2	The Knockout Stage	29
4.3	The Problem of the Runner-up	32
5	A Financial Problem	35
5.1	A “Cubic” Algorithm	38
5.2	A “Quadratic” Algorithm	39
5.3	A “Linear” Algorithm	41
5.4	Algorithm Efficiency Versus Hardware Efficiency	44
5.5	Some Interesting Variants of the Problem	45
6	Secret Messages	49
6.1	Modularity	51
6.2	Exponential Growth	53
6.3	Leon Battista Alberti’s Disk	56
6.4	Today’s Ciphers	59
6.5	Dispositions of Characters	60
6.6	The Construction of Keys	62

7	Putting Things in Order	67
	7.1 The Sorting Problem	69
	7.2 A More Efficient Algorithm	75
	7.3 A Lower Bound	80
	7.4 Sorting Integers	83
	7.5 Sorting Big Data	86
	7.6 Multi-way Merge (Sort)	90
8	“Easy” and “Difficult” Problems	95
	8.1 Walking Through a City	96
	8.2 $P=NP?$	107
9	Search Engines	111
	9.1 From Words to Hyperlinks	112
	9.2 Fourth Generation	115
	9.3 The Structure	117
	9.4 How to Solve a Query	119
	9.5 Still an Improvement	125
	9.6 A Concluding Remark	127
10	Data Compression	129
	10.1 Statistical Compressors	132
	10.2 Dictionary-Based Compressors	137
	10.3 Sort-Based Compressors	141
	10.4 Concluding Remarks	147
11	Recursion	149
	11.1 A Very Instructive Game	151
	11.2 Comparison Between Recursion and Iteration	155
	11.3 Other Examples	158
	11.4 A Famous Randomized Algorithm	163
	11.5 To Conclude, Let Us Be a Little Cautious	168
	Index	173