The Power of Algorithms

Giorgio Ausiello • Rossella Petreschi
Editors

The Power of Algorithms

Inspiration and Examples in Everyday Life

Springer

Editors
Giorgio Ausiello
Dip. di Informatica e Sistemistica
Università di Roma "La Sapienza"
Rome, Italy

Rossella Petreschi
Dipartimento di Informatica
Università di Roma "La Sapienza"
Rome, Italy

First published in Italian in 2010 by Mondadori Education S.p.A., Milano as "L'informatica invisibile: Come gli algoritmi regolano la nostra vita . . . e tutto il resto"

ISBN 978-3-642-39651-9 ISBN 978-3-642-39652-6 (eBook)
DOI 10.1007/978-3-642-39652-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013952981

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The meaning of the word *algorithm* as found in any English dictionary is rather similar to the meaning of words such as *method* or *procedure*, that is, “a finite set of rules specifying a sequence of operations to solve a particular problem”. Simple algorithms we are all familiar with are those used to perform the four arithmetical operations, or the binary search which, more or less unconsciously, we use to find a name in a telephone directory.

Strangely, however, the very mention of the word algorithm provokes a sense of fear in many people, possibly due to its mathematical connotations. Indeed, the word’s etymological origin is the name of the Persian mathematician, al-Khwarizmi, who worked in Baghdad at the beginning of the ninth century, and its contemporary meaning is derived from the fact that he introduced Indian methods of calculation based on positional representation of numbers into the Christian countries of the West.

And so it may be that a deep-seated unease with mathematics causes many to lose sight of the central role algorithms play in computer science and of the fact that myriad activities of their lives are today governed by algorithms. Booking a plane ticket, effecting a secure transaction at the cash machine of a bank, searching for information on the Web, and zipping or unzipping files containing music or images are just a few examples of the way algorithms have come to pervade all aspects of everyday life. Algorithms are even inserted into national legislation, such as the rules defining the construction of a citizen’s fiscal code, national insurance number, etc., or the increasingly widely used digital signature for authenticating documents.

A highly important consideration to emphasize, however, is that not only do algorithms have a huge number of applications, but they also act as powerful “magnifying lenses” enabling a penetrating comprehension of problems.

Examining, analyzing, and manipulating a problem to the point of being able to design an algorithm leading to its solution is a mental exercise that can be of fundamental help in understanding a wide range of subjects, irrespective of the fields of knowledge to which they belong (natural sciences, linguistics, music, etc.).

In any case, it was the advent of computers and computer science that led to the word ‘algorithm’ becoming known to a wide range of people, so much so that even in 1977 Donald Knuth (one of the founding fathers of computer science) wrote:

> Until ten years ago the word algorithm was unknown to the vast majority of educated people and, to tell the truth, there was little need for it anyway. The furiously rapid development of computer science, whose primary focus is the study of algorithms, has changed this situation: today the word algorithm is indispensable.

Formalizing a problem as an algorithm thus leads to a better grasp of the argument to be dealt with, compared to tackling it using traditional reasoning. Indeed, a person who knows how to handle algorithms acquires a capacity for introspection that she/he will find useful not only in writing good programs for a computer, but also in achieving improved understanding of many other kinds of problem in other fields. Knuth, again, in his book “The Art of Computer Programming”, asserts that:

> If it is true that one doesn’t truly understand a problem in depth until one has to teach it to someone else, it is even truer that nothing is understood more completely than something one has to teach to a machine, that is, than something which has to be expressed by way of an algorithm.

Unfortunately, the precision demanded by the algorithmic approach (the algorithm has to be independent of the data to which it is applied and the rules it employs have to be elementary, that is, very simple and unambiguous), although useful as a means of mental development, limits the types of problem for which it can be adopted. To convince oneself of this just think of the fact that no algorithm exists for teaching “how to live a happy life”. Alternatively, as a more rigorous demonstration of these limitations, we cite one of the most important findings of twentieth century logic, whereby Alan Turing (in the wake of Gödel’s incompleteness proof) showed that no algorithm exists that would be capable of deciding whether or not a logical formula asserting a property of arithmetic is a theorem (see Chaps. 1 and 3).

For every algorithm two fundamental components can be identified: the determination of the appropriate algorithmic design technique (based on the structure of the problem) and the clear understanding of the mathematical nucleus of the problem. These two components interact closely with each other, thus it is not so much that algorithmic ideas just find solutions to well-stated problems, as that they function as a language that enables a particular problem to be expressed in the first place. It is for this reason that David Harel, in his 1987 book “Algorithmics: The Spirit of Computing” was able, without fear of contradiction, to define the algorithm as “the soul of computer science”.

The earliest algorithms can be traced back as far as 2000 BCE; Mesopotamian clay tablets and Egyptian papyrus have been found bearing the first examples of procedures for calculation defined in fairly rigorous ways. Over the successive millennia thereafter humans made ever-increasing use of algorithms to solve problems arising in widely diverse fields: from measurements of land areas to astronomy, from trade to finance, and from the design of civil engineering projects to the study of physical phenomena. All of these significantly contributed, in the eighteenth and nineteenth centuries, to the first products of the industrial revolution.

Notwithstanding this, it was not until the twentieth century that the formal definition of the concept of algorithm began to be tackled. This was done primarily by mathematical logicians, such as Alonzo Church and the already-cited Alan Turing, in a series of theoretical investigations which turned out to be the indispensable groundwork for subsequent development of the first programmable electronic computer and the first computer programming languages. As mentioned earlier, it was with the advent of computers and computer science that algorithms really began to play a central role, initially only in military and scientific fields, and then ever increasingly in the fields of commerce and management. Today we can say that algorithms are an indispensable part of our everyday lives—and it seems they are destined to become even more pervasive in the future.

Nevertheless, despite this massive influence of algorithms on the world around us, the majority of users remain totally ignorant of their role and importance in securing the performance of the computer applications with which they are most familiar, or, at best, consider them technical matters of little concern to them. Instead quite the opposite is the case: in reality it is the power, the precision, the reliability and the speed of execution which these same users have been demanding with ever-increasing pressure that have transformed the design and construction of algorithms from a highly skilled "niche craft" into a full-fledged science in its own right.

This book is aimed at all those who, perhaps without realizing it, exploit the results of this new science, and it seeks to give them the opportunity to see what otherwise would remain hidden. There are ten chapters, of which nine are divided into two parts. Part I (Chaps. 1–3) introduces the reader to the properties and techniques upon which the design of an efficient algorithm is based and shows how the intrinsic complexity of a problem is tackled. Part II (Chaps. 4–9) presents six different applications (one for each chapter) which we encounter daily in our work or leisure routines. For each of these applications the conceptual and scientific bases upon which the algorithm used is grounded are revealed and it is shown how these bases are decisive as regards the validity of the applications dealt with. The book concludes with a different format, that of the dialogue. Chapter 10 illustrates how randomness can be exploited in order to solve complex problems, and its dialogue format has been deliberately chosen to show how discussions of such issues are part of the daily life of those who work in this field.

As an aid to readers whose educational background may not include particularly advanced mathematics there are clear indications in the text as to which sections containing more demanding mathematics may be skipped without fear of losing the thread of the main argument. Moreover, in almost every chapter, boxes covering specific mathematical or technical concepts have been inserted, and those readers wishing to get a general sense of the topic can avoid tackling these, at least on a first reading.

In fact, an overriding aim of the authors is to make the role of algorithms in today's world readily comprehensible to as wide a sector of the public as possible. To this end a simple, intuitive approach that keeps technical concepts to a minimum has been used throughout. This should ensure ideas are accessible to the intellectually curious reader whose general education is of a good level, but does not necessarily include mathematical and/or computer scientific training.

At the same time, the variety of subjects dealt with should make the book interesting to those who are familiar with computer technologies and applications, but who wish to deepen their knowledge of the ideas and techniques that underlie the creation and development of efficient algorithms. It is for these reasons that the book, while having a logical progression from the first page to the last, has been written in such a way that each chapter can be read separately from the others.

Roma, Italy
July 2013

Giorgio Ausiello
Rossella Petreschi

Contents

Part I Finding One's Way in a World of Algorithms

Part II The Difficult Simplicity of Daily Life

List of Contributors

Giorgio Ausiello Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Roma, Italy

Vincenzo Bonifaci Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", Consiglio Nazionale delle Ricerche, Roma, Italy

Camil Demetrescu Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Roma, Italy

Paolo Ferragina Dipartimento di Informatica, Università di Pisa, Pisa, Italy

Raffaele Giancarlo Dipartimento di Matematica ed Informatica, Università di Palermo, Palermo, Italy

Giuseppe F. Italiano Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma "Tor Vergata", Roma, Italy

Stefano Leonardi Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Roma, Italy

Alberto Marchetti-Spaccamela Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Roma, Italy

Alessandro Panconesi Dipartimento di Informatica, Sapienza Università di Roma, Roma, Italy

Rossella Petreschi Dipartimento di Informatica, Sapienza Università di Roma, Roma, Italy

Fabrizio Rossi Dipartimento di Informatica, Università dell'Aquila, Coppito (AQ), Italy

Antonio Sassano Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Roma, Italy

Riccardo Silvestri Dipartimento di Informatica, Sapienza Università di Roma, Roma, Italy

Stefano Smriglio Dipartimento di Informatica, Università dell'Aquila, Coppito (AQ), Italy

Rossano Venturini Dipartimento di Informatica, Università di Pisa, Pisa, Italy