Texts in Computer Science

More information about this series at http://www.springer.com/series/3191

Gabriel Valiente

Algorithms on Trees and Graphs

With Python Code

Second Edition

Springer

Gabriel Valiente
Department of Computer Science
Technical University of Catalonia
Barcelona, Spain

ISSN 1868-0941 ISSN 1868-095X (electronic)
Texts in Computer Science
ISBN 978-3-030-81884-5 ISBN 978-3-030-81885-2 (eBook)
https://doi.org/10.1007/978-3-030-81885-2

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

To my child Aleksandr,
six years old,
who is eager to grow
and read it all

Preface to the Second Edition

The first edition of this book has been extensively used for graduate teaching and research all over the world in the last two decades. We have listed hundreds of citing publications in Appendix C, including books, scientific articles in journals and conference proceedings, M.Sc. and Ph.D. theses, and even United States patents.

In this new edition, we have substituted detailed pseudocode for both the literate programming description and the implementation of the algorithms using the LEDA library of efficient data structures and algorithms. Although the pseudocode is detailed enough to allow for a straightforward implementation of the algorithms in any modern programming language, we have added a proof-of-concept implementation in Python of all the algorithms in Appendix A. This is, therefore, a thoroughly revised and extended edition.

Regarding new material, we have added an adjacency map representation of trees and graphs, and both maximum cardinality and maximum weight bipartite matching as an additional application of graph traversal techniques. Further, we have revised the end-of-chapter problems and exercises and have included solutions to all the problems in Appendix B.

It has been a pleasure for the author to work out editorial matters together with Sriram Srinivas and, especially, Wayne Wheeler of Springer Nature, whose standing support and encouragement have made this new edition possible.

Last, but not least, any minor errors found so far have been corrected in this second edition, the bibliographic notes and references have been updated, and the index has been substantially enhanced. Even though the author and the publisher have taken much care in the preparation of this book, they make no representation, express or implied, with regard to the accuracy of the information contained herein and cannot accept any legal responsibility or liability for incidental or consequential damages arising out of the use of the information, algorithms, or program code contained in this book.

Barcelona, Spain Gabriel Valiente
June 2021

Preface to the First Edition

Graph algorithms, a long-established subject in mathematics and computer science curricula, are also of much interest to disciplines such as computational molecular biology and computational chemistry. This book goes beyond the *classical* graph problems of shortest paths, spanning trees, flows in networks, and matchings in bipartite graphs, and addresses further algorithmic problems of practical application on trees and graphs. Much of the material presented on the book is only available in the specialized research literature.

The book is structured around the fundamental problem of isomorphism. Tree isomorphism is covered in much detail, together with the related problems of subtree isomorphism, maximum common subtree isomorphism, and tree comparison. Graph isomorphism is also covered in much detail, together with the related problems of subgraph isomorphism, maximal common subgraph isomorphism, and graph edit distance. A building block for solving some of these isomorphism problems are algorithms for finding maximal and maximum cliques.

Most intractable graph problems of practical application are not even approximable to within a constant bound, and several of the isomorphism problems addressed in this book are no exception. The book can thus be seen as a companion to recent texts on approximation algorithms [1, 16], but also as a complement to previous texts on combinatorial and graph algorithms [2–15, 17].

The book is conceived on the ground of first, introducing simple algorithms for these problems in order to develop some intuition before moving on to more complicated algorithms from the research literature and second, stimulating graduate research on tree and graph algorithms by providing together with the underlying theory, a solid basis for experimentation and further development.

Algorithms are presented on an intuitive basis, followed by a detailed exposition in a literate programming style. Correctness proofs are also given, together with a worst-case analysis of the algorithms. Further, full C++ implementation of all the algorithms using the LEDA library of efficient data structures and algorithms is given along the book. These implementations include result checking of implementation correctness using correctness certificates.

The choice of LEDA, which is becoming a de-facto standard for graduate courses on graph algorithms throughout the world is not casual, because it allows the student, lecturer, researcher, and practitioner to complement algorithmic graph theory with actual implementation and experimentation, building upon a thorough library of efficient implementations of modern data structures and fundamental algorithms.

An interactive demonstration including animations of all the algorithms using LEDA is given in an appendix. The interactive demonstration also includes visual checkers of implementation correctness.

The book is divided into four parts. Part I has an introductory nature and consists of two chapters. Chapter 1 includes a review of basic graph-theoretical notions and results used along the book, a brief primer of literate programming, and an exposition of the implementation correctness approach by result checking using correctness certificates. Chapter 2 is devoted exclusively to the fundamental algorithmic techniques used in the book: backtracking, branch-and-bound, divide-and-conquer, and dynamic programming. These techniques are illustrated by means of a running example: algorithms for the tree edit distance problem.

Part II also consists of two chapters. Chapter 3 addresses the most common methods for traversing general, rooted trees: depth-first prefix leftmost (preorder), depth-first prefix rightmost, depth-first postfix leftmost (postorder), depth-first postfix rightmost, breadth-first leftmost (top-down), breadth-first rightmost, and bottom-up traversal. Tree drawing is also discussed as an application of tree traversal methods. Chapter 4 addresses several isomorphism problems on ordered and unordered trees: tree isomorphism, subtree isomorphism, and maximum common subtree isomorphism. Computational molecular biology is also discussed as an application of the different isomorphism problems on trees.

Part III consists of three chapters. Chapter 5 addresses the most common methods for traversing graphs: depth-first and breadth-first traversal, which respectively generalize depth-first prefix leftmost (preorder) and breadth-first leftmost (top-down) tree traversal. Leftmost depth-first traversal of an undirected graph, a particular case of depth-first traversal, is also discussed. Isomorphism of ordered graphs is also discussed as an application of graph traversal methods. Chapter 6 addresses the related problems of finding cliques, independent sets, and vertex covers in trees and graphs. Multiple alignment of protein sequences in computational molecular biology is also discussed as an application of clique algorithms. Chapter 7 addresses several isomorphism problems on graphs: graph isomorphism, graph automorphism, subgraph isomorphism, and maximal common subgraph isomorphism. Chemical structure search is also discussed as an application of the different graph isomorphism problems.

Part IV consists of two appendices, followed by bibliographic references and an index. Appendix A gives an overview of LEDA, including a simple C++ representation of trees as LEDA graphs, and a C++ implementation of radix sort using LEDA. The interactive demonstration of graph algorithms presented along the book is put together in Appendix B. Finally, Appendix C contains a complete index to all program modules described in the book.

This book is suitable for use in upper undergraduate and graduate level courses on algorithmic graph theory. This book can also be used as a supplementary text in basic undergraduate and graduate level courses on algorithms and data structures, and in computational molecular biology and computational chemistry courses as well. Some basic knowledge of discrete mathematics, data structures, algorithms, and programming at the undergraduate level is assumed.

This book is based on lectures taught at the Technical University of Catalonia, Barcelona between 1996 and 2002, and the University of Latvia, Riga between 2000 and 2002. Numerous colleagues at the Technical University of Catalonia have influenced the approach to data structures and algorithms on trees and graphs expressed in this book. In particular, the author would like to thank José L. Balcázar, Rafel Casas, Jordi Cortadella, Josep Daz, Conrado Martnez, Xavier Messeguer, Roberto Nieuwenhuis, Fernando Orejas, Jordi Petit, Salvador Roura, and Maria Serna, to name just a few. It has been a pleasure to share teaching and research experiences with them over the last several years.

The author would also like to thank Ricardo Baeza-Yates, Francesc Rosselló, and Steven Skiena, for their standing support and encouragement, and Hans-Jörg Kreowski, for supporting basic and applied research on graph algorithms within the field of graph transformation. It has been a pleasure for the author to work out editorial matters together with Alfred Hofmann, Ingeborg Mayer, and Peter Straßer of Springer-Verlag. Special thanks are debt to the Technical University of Catalonia for funding the sabbatical year during which this book was written, and to the Institute of Mathematics and Computer Science, University of Latvia, in particular to Jānis Bārzdiņš and Rūsiņš Freivalds, for hosting the sabbatical visit.

Barcelona, Spain Gabriel Valiente
July 2002

References

1. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kahn, A. MarchettiSpaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer-Verlag, Berlin Heidelberg, 1999.
2. G. Chartrand and O. R. Oellermann. *Applied and Algorithmic Graph Theory*. McGraw-Hill, New York, 1993.
3. N. Christofides. *Graph Theory: An Algorithmic Approach*. Academic Press, New York, 1975.
4. S. Even. *GraphAlgorithms*. Computer Science Press, Rockville MD,1979.
5. A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, Cambridge, 1985.
6. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
7. D. L. Kreher and D. R. Stinson. *Combinatorial Algorithms: Generation, Enumeration, and Search*. CRC Press, Boca Raton FL, 1999.
8. J. van Leeuwen. Graph algorithms. In *Handbook of Theoretical Computer Science*, volume A, chapter 10, pages 525-631. Elsevier, Amsterdam, 1990.
9. J. A. McHugh. *Algorithmic Graph Theory*. Prentice Hall, Englewood Cliffs NJ, 1990.
10. K. Mehlhorn. *Graph Algorithms and NP-Completeness, volume 2 of Data Structures and Algorithms*. Springer-Verlag, Berlin Heidelberg, 1984.
11. K. Mehlhorn and S. Naher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, 1999.

12. C. H. Papadimitriou and K. Steiglitz. *CombinatorialOptimization: Algorithms and Complexity*. Dover, Mineola, New York, 1998.
13. E. M. Reingold, 1. Nievergelt, and N. J. Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice Hall, Englewood Cliffs NJ, 1977.
14. S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, Berlin Heidelberg, 1998.
15. R. E. Tarjan. *Space-efficient Implementations of Graph Search Methods*. ACM Transactions on Mathematical Software. 1983 9(3), 326–339
16. V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin Heidelberg, 2001.
17. H. S. Wilf. *Combinatorial Algorithms: An Update*. SIAM, Philadelphia PA, 1989.

Contents