Kurt Mehlhorn and Peter Sanders

Algorithms and Data Structures

The Basic Toolbox

August 21, 2008

Springer

To all algorithmicists

Preface

Algorithms are at the heart of every nontrivial computer application. Therefore every computer scientist and every professional programmer should know about the basic algorithmic toolbox: structures that allow efficient organization and retrieval of data, frequently used algorithms, and generic techniques for modeling, understanding, and solving algorithmic problems.

This book is a concise introduction to this basic toolbox, intended for students and professionals familiar with programming and basic mathematical language. We have used the book in undergraduate courses on algorithmics. In our graduate-level courses, we make most of the book a prerequisite, and concentrate on the starred sections and the more advanced material. We believe that, even for undergraduates, a concise yet clear and simple presentation makes material more accessible, as long as it includes examples, pictures, informal explanations, exercises, and some linkage to the real world.

Most chapters have the same basic structure. We begin by discussing a problem as it occurs in a real-life situation. We illustrate the most important applications and then introduce simple solutions *as informally as possible and as formally as necessary* to really understand the issues at hand. When we move to more advanced and optional issues, this approach gradually leads to a more mathematical treatment, including theorems and proofs. This way, the book should work for readers with a wide range of mathematical expertise. There are also advanced sections (marked with a *) where we *recommend* that readers should skip them on first reading. Exercises provide additional examples, alternative approaches and opportunities to think about the problems. It is highly recommended to take a look at the exercises even if there is no time to solve them during the first reading. In order to be able to concentrate on ideas rather than programming details, we use pictures, words, and high-level pseudocode to explain our algorithms. A section "implementation notes" links these abstract ideas to clean, efficient implementations in real programming languages such as C++ and Java. Each chapter ends with a section on further findings that provides a glimpse at the state of the art, generalizations, and advanced solutions.

Algorithmics is a modern and active area of computer science, even at the level of the basic toolbox. We have made sure that we present algorithms in a modern

way, including explicitly formulated invariants. We also discuss recent trends, such as algorithm engineering, memory hierarchies, algorithm libraries, and certifying algorithms.

We have chosen to organize most of the material by problem domain and not by solution technique. The final chapter on optimization techniques is an exception. We find that presentation by problem domain allows a more concise presentation. However, it is also important that readers and students obtain a good grasp of the available techniques. Therefore, we have structured the final chapter by techniques, and an extensive index provides cross-references between different applications of the same technique. Bold page numbers in the Index indicate the pages where concepts are defined.

Karlsruhe, Saarbrücken, *Kurt Mehlhorn*
February, 2008 *Peter Sanders*

Contents