Analysis of Algorithms: An Active Learning Approach

Jeffrey J. McConnell
Canisius College

JONES AND BARTLETT PUBLISHERS
Sudbury, Massachusetts
BOSTON TORONTO LONDON SINGAPORE

PREFACE

The two major goals of this book are to raise awareness of the impact that algorithms can have on the efficiency of a program and to develop the skills necessary to analyze any algorithms that are used in programs. In looking at many commercial products today, it appears that some software designers are unconcerned about space and time efficiency. If a program takes too much space, they expect that the user will buy more memory. If a program takes too long, they expect that the user will buy a faster computer.

There are limits, however, on how fast computers can ever become because there are limits on how fast electrons can travel down "wires," how fast light can travel along fiber optic cables, and how fast the circuits that do the calculations can switch. There are other limits on computation that go beyond the speed of the computer and are directly related to the complexity of the problems being solved. There are some problems for which the fastest algorithm known will not complete execution in our lifetime. Since these are important problems, algorithms are needed that provide approximate answers.

In the early 1980s, computer architecture severely limited the amount of speed and space on a computer. Some computers of that time frequently limited programs and their data to 64K of memory, where today's personal computers regularly come equipped with more than 1,000 times that amount. Though today's software is much more complex than that in the 1980s and today's computers are much more capable, these changes do not mean we can ignore efficiency in our program design. Some project specifications will include time and space limitations on the final software that may force programmers to look for places to save memory and increase speed. The compact size of personal digital assistants (PDAs) also limits the size and speed of software.

Pedagogy

What I hear, I forget.
What I see, I remember.
What I do, I understand.
—*Confucius*

The material in this book is presented with the expectation that it can be read independently or used as part of a course that incorporates an active and cooperative learning methodology. To accomplish this, the chapters are clear and complete so as to be easy to understand and to encourage readers to prepare by reading before group meetings. All chapters include study suggestions. Many include additional data sets that the reader can use to hand-execute the algorithms for increased understanding of them. The results of the algorithms applied to this additional data are presented in Appendix C. Each section has a number of exercises that include simple tracing of the algorithm to more complex proof-based exercises. The reader should be able to work the exercises in each chapter. They can, in connection with a course, be assigned as homework or can be used as in-class assignments for students to work individually or in small groups. An instructor's manual that provides background on how to teach this material using active and cooperative learning as well as giving exercise solutions is available. Chapters 2, 3, 5, 6, and 9 include programming exercises. These programming projects encourage readers to implement and test the algorithms from the chapter, and then compare actual algorithm results with the theoretical analysis in the book.

Active learning is based on the premise that people learn better and retain information longer when they are participants in the learning process. To achieve that, students must be given the opportunity to do more that just listen to the professor during class. This can best be accomplished in an analysis of algorithms course by the professor giving a short introductory lecture on the material, and then having students work problems while the instructor circulates around the room answering questions that this application of the material raises.

Cooperative work gives students an opportunity to answer simple questions that others in their group have and allows the professor to deal with bigger questions that have stumped an entire group. In this way, students have a greater opportunity to ask questions and have their concerns addressed in a timely manner. It is important that the professor observe group work to make sure that group-wide misconceptions are not reinforced. An additional way for the professor to identify and correct misunderstandings is to have groups regularly submit exercise answers for comments or grading.

To support student preparation and learning, each chapter includes the prerequisites needed, and the goals or skills that students should have on completion, as well as suggestions for studying the material.

Algorithms

Since the analysis of algorithms is independent of the computer or programming language used, algorithms are given in pseudo-code. These algorithms are readily understandable by anyone who knows the concepts of conditional statements (for example, IF and CASE/SWITCH), loops (for example, FOR and WHILE), and recursion.

Course Use

One way that this material could be covered in a one-semester course is by using the following approximate schedule:

Chapter 1	2 weeks
Chapter 2	1 week
Chapter 3	2 weeks
Chapter 4	1 week
Chapter 5	1 week
Chapter 6	2 weeks
Chapter 7	2 weeks
Chapter 8	1 week
Chapter 9	2 weeks

Chapters 2, 4, and 5 are not likely to need a full week, which will provide time for an introduction to the course, an explanation of the active and cooperative learning pedagogy, and hour examinations. Depending on the background of the students, Chapter 1 may be covered more quickly as well.

Acknowledgements

I would like to acknowledge all those who helped in the development of this book. First, I would like to thank the students in my "Automata and Algorithms" course (Spring 1997, Spring 1998, Spring 1999, Spring 2000, and Fall 2000) for all of their comments on earlier versions of this material.

The reviews that Jones and Bartlett Publishers obtained about the manuscript were most helpful and produced some good additions and clarifications. I would like to thank Douglas Campbell (Brigham Young University), Nancy Kinnersley (University of Kansas), and Kirk Pruhs (University of Pittsburgh) for their reviews.

At Jones and Bartlett, I would like to thank my editors Amy Rose and Michael Stranz, and production assistant Tara McCormick for their support of this book. I am especially grateful to Amy for remembering a brief conversation about this project from a few years ago. Her memory and efforts are appreciated very much. I would also like to thank Nancy Young for her copyediting and Brooke Albright for her proofreading. Any errors that remain are solely the author's responsibility.

Lastly, I am grateful to Fred Dansereau for his support and suggestions during the many stages of this book, and to Barney for the wonderful diversions that only a dog can provide.

CONTENTS