教育部高等教育司推荐
国外优秀信息科学与技术系列教学用书

算法分析
—— 有效的学习方法

（影印版）

ANALYSIS OF ALGORITHMS
An Active Learning Approach

■ Jeffrey J. McConnell

Z

高等教育出版社
Higher Education Press

教 育 部 高 等 教 育 司 推 荐
国外优秀信息科学与技术系列教学用书

算 法 分 析

——有效的学习方法

（影印版）

ANALYSIS OF ALGORITHMS

An Active Learning Approach

Jeffrey J. McConnell

高 等 教 育 出 版 社

图字：01-2003-0688 号

Analysis of Algorithms: An Active Learning Approach, First Edition
Jeffrey J. McConnell

ORIGINAL ENGLISH LANGUAGE EDITION PUBLISHED BY
Jones and Bartlett Publishers, Inc.
40 Tall Pine Drive
Sudbury, MA 01776

图书在版编目（CIP）数据

算法分析：有效的学习方法 /（美）麦康奈尔（McConnell, J. J.）著. —影印版. —北京：高等教育出版社，2003.3
ISBN 7-04-012659-1

Ⅰ. 算...　Ⅱ. 麦...　Ⅲ. 算法分析 - 高等学校 - 教材 - 英文　Ⅳ. TP311

中国版本图书馆 CIP 数据核字（2003）第 012578 号

出版发行	高等教育出版社	**购书热线**	010 - 64054588
社　　址	北京市东城区沙滩后街 55 号	**免费咨询**	800 - 810 - 0598
邮政编码	100009	**网　　址**	http://www.hep.edu.cn
传　　真	010 - 64014048		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京奥隆印刷厂		
开　　本	787×1092　1/16	**版　　次**	2003 年 3 月第 1 版
印　　张	19.75	**印　　次**	2003 年 3 月第 1 次印刷
字　　数	400 000	**定　　价**	28.00 元

前　　言

20 世纪末，以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用，带动了世界范围信息产业的蓬勃发展，为许多国家带来了丰厚的回报。

进入 21 世纪，尤其随着我国加入 WTO，信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展，但与发达国家相比，甚至与印度、爱尔兰等国家相比，还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力，最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材，在有条件的学校推动开展英语授课或双语教学，是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此，教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求，一是要高水平，二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下，经过比较短的时间，第一批引进的 20 多种教材已经陆续出版。这套教材出版后受到了广泛的好评，其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品，代表了目前世界信息科学技术教育的一流水平，而且价格也是最优惠的，与国内同类自编教材相当。

这项教材引进工作是在教育部高等教育司和高教社的共同组织下，由国内信息科学技术领域的专家、教授广泛参与，在对大量国外教材进行多次遴选的基础上，参考了国内和国外著名大学相关专业的课程设置进行系统引进的。其中，John Wiley 公司出版的贝尔实验室信息科学研究中心副总裁 Silberschatz 教授的经典著作《操作系统概念》，是我们经过反复谈判，做了很多努力才得以引进的。William Stallings 先生曾编写了在美国深受欢迎的信息科学技术系列教材，其中有多种教材获得过美国教材和学术著作者协会颁发的计算机科学与工程教材奖，这批引进教材中就有他的两本著作。留美中国学者 Jiawei Han 先生的《数据挖掘》是该领域中具有里程碑意义的著作。由达特茅斯学院 Thomas Cormen 和麻省理工学院、哥伦比亚大学的几

位学者共同编著的经典著作《算法导论》，在经历了 11 年的锤炼之后于 2001 年出版了第二版。目前任教于美国 Massachusetts 大学的 James Kurose 教授，曾在美国三所高校先后 10 次获得杰出教师或杰出教学奖，由他主编的《计算机网络》出版后，以其体系新颖、内容先进而倍受欢迎。在努力降低引进教材售价方面，高等教育出版社做了大量和细致的工作。这套引进的教材体现了权威性、系统性、先进性和经济性等特点。

教育部也希望国内和国外的出版商积极参与此项工作，共同促进中国信息技术教育和信息产业的发展。我们在与外商的谈判工作中，不仅要坚定不移地引进国外最优秀的教材，而且还要千方百计地将版权转让费降下来，要让引进教材的价格与国内自编教材相当，让广大教师和学生负担得起。中国的教育市场巨大，外国出版公司和国内出版社要通过扩大发行数量取得效益。

在引进教材的同时，我们还应做好消化吸收，注意学习国外先进的教学思想和教学方法，提高自编教材的水平，使我们的教学和教材在内容体系上，在理论与实践的结合上，在培养学生的动手能力上能有较大的突破和创新。

目前，教育部正在全国 35 所高校推动示范性软件学院的建设和实施，这也是加快培养信息科学技术人才的重要举措之一。示范性软件学院要立足于培养具有国际竞争力的实用性软件人才，与国外知名高校或著名企业合作办学，以国内外著名 IT 企业为实践教学基地，聘请国内外知名教授和软件专家授课，还要率先使用引进教材开展教学。

我们希望通过这些举措，能在较短的时间，为我国培养一大批高质量的信息技术人才，提高我国软件人才的国际竞争力，促进我国信息产业的快速发展，加快推动国家信息化进程，进而带动整个国民经济的跨越式发展。

教育部高等教育司

二〇〇二年三月

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

CHAPTER 1

Analysis Basics

PREREQUISITES

Before beginning this chapter, you should be able to

- Read and create algorithms
- Read and create recursive algorithms
- Identify comparison and arithmetic operations
- Use basic algebra

GOALS

At the end of this chapter you should be able to

- Describe how to analyze an algorithm
- Explain how to choose the operations that are counted and why others are not
- Explain how to do a best-case, worst-case, and average-case analysis
- Work with logarithms, probabilities, and summations
- Describe $\theta(f)$, $\Omega(f)$, $O(f)$, growth rate, and algorithm order
- Use a decision tree to determine a lower bound on complexity
- Convert a simple recurrence relation into its closed form

STUDY SUGGESTIONS

As you are working through the chapter, you should rework the examples to make sure you understand them. Additionally, you should try to answer any questions before reading on. A hint or the answer to the question is in the sentences following it.