世界优秀教材中国版

理科类系列教材

改编版

Discrete Mathematics (Fourth Edition)

离散数学（第4版）

Dossey
Otto
Spence
Vanden Eynden
原著

俞正光 陆玫 改编

高等教育出版社
Higher Education Press

世界优秀教材中国版

理科类系列教材

改编版

Discrete Mathematics

(Fourth Edition)

离散数学（第4版）

John A.Dossey
Albert D.Otto
Lawrence E.Spence
Charles Vanden Eynden 原著
Illinios State University

俞正光 陆玫 改编
清华大学

高等教育出版社
Higher Education Press

图字:01－2005－2299号

图书在版编目（CIP）数据

离散数学＝Discrete Mathematics：第4版：改编版／（美）多塞等著；俞正光，陆玫改编．—北京：高等教育出版社，2005.7
（理科类系列教材）
ISBH 7-04-016632-1

Ⅰ.离… Ⅱ.①多…②俞…③陆… Ⅲ.离散数学－高等学校－教材－英文 Ⅳ.0158

中国版本图书馆 CIP 数据核字（2005）第067628号

出版发行	高等教育出版社	**购书热线**	010－58581118
社　　址	北京市西城区德外大街4号	**免费咨询**	800－810－0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010－58581000		http://www.hep.com.cn
经　　销	北京蓝色畅想图书发行有限公司	**网上订购**	http://www.landraco.com
印　　刷	北京外文印刷厂		http://www.landraco.com.cn
开　　本	787×1092　1/16	**版　　次**	2005年7月第1版
印　　张	30.75	**印　　次**	2005年7月第1次印刷
字　　数	550 000	**定　　价**	31.90元

物料号　16632－00

DISCRETE MATHEMATICS TIMELINE

c. 300 B.C. | 0 | 1200 | 1500 | 1600 | 1700

■ **c. 300 B.C.**
Euclid published the Euclidean Algorithm in his *Elements*.

■ **1202**
Fibonacci introduced the Fibonacci numbers in his *Liber Abaci*.

■ **1575**
Francesco Maurolico wrote his *Arithmeticorum*, which employed the principle of mathematical induction in proofs.

■ **1654**
Blaise Pascal's famous triangle appeared in *Arithmetical Triangle*.

■ **1654**
Blaise Pascal and Pierre Fermat corresponded on the fundamentals of probability theory.

■ **1666**
Gottfried Leibniz published his ideas on combinatorics in *Diertatio de Arte Combinatoria*.

■ **1713**
James Bernoulli's *Ars Conjectandi* introduced several ideas related to permutations and combinations.

■ **1718**
Abraham De Moivre wrote about the principle of inclusion-exclusion in his *Doctrine of Chances*.

■ **1720**
Abraham De Moivre began work on generating functions.

■ **1735**
Leonhard Euler introduced functional notation.

■ **1736**
Leonhard Euler published a paper on the Königsberg Bridge problem.

■ **1750**
Leonhard Euler proposed his formula for the number of edges, vertices, and faces of polyhedra.

c. 300 B.C. | 0 | 1200 | 1500 | 1600 | 1700

1800 1850 1900 1950 2000

■ **1801**
Carl Friedrich Gauss published *Disquisitiones Arithmeticae*, which outlined congruence modulo *m* and other number theory topics.

■ **1844**
Gabriel Lamé provided an analysis of the complexity of the Euclidean algorithm.

■ **1847**
Gustav Kirchhoff examined trees in the study of electrical circuits.

■ **1852**
Augustus De Morgan wrote William Rowen Hamilton outlining the 4-color problem.

■ **1854**
George Boole published *An Investigation of the Laws of Thought*, which formalized the algebra of sets and logic.

■ **1855**
Thomas P. Kirkman published a paper containing the traveling salesperson problem.

■ **1856**
William Rowen Hamilton proposed the traveling salesperson problem.

■ **1857**
Arthur Cayley introduced the name "tree" and enumerated the number of rooted trees with *n* edges.

■ **1872**
George Boole published *A Treatise on the Calculus of Finite Differences*.

■ **1877**
James Joseph Sylvester introduced the word "graph" in a paper.

■ **1881**
John Venn introduced the usage of Venn diagrams for reasoning.

■ **1922**
Oswald Veblen proved that every connected graph contains a spanning tree.

■ **1931**
Dénes König published his paper on matchings in graphs.

■ **1935**
Philip Hall published necessary and sufficient conditions for the existence of a system of distinct representatives.

■ **1936**
Dénes König wrote the first book on graph theory, *Theorie der Endlichen und Unendlichen Graphen*.

■ **1938**
Claude Shannon devised the algebra of switching circuits and showed its connections to logic.

■ **1951**
George Dantzig published his simplex algorithm for solving linear programming problems.

■ **1953**
Maurice Karnaugh introduced the use of Karnaugh maps for the simplification of Boolean circuits.

■ **1954**
G.H. Mealy developed a model for a finite state machine with output.

■ **1956**
Joseph B. Kruskal, Jr., published his algorithm for minimum spanning tree length.

■ **1956**
Lester R. Ford, Jr., and Delbert R. Fulkerson published their work on maximal flows in a network.

■ **1957**
Robert Prim developed Prim's algorithm.

■ **1958**
PERT algorithm was developed and applied in the construction of the *Nautilus* submarine.

■ **1959**
Edsger W. Dijkstra published a paper containing Dijkstra's algorithm.

■ **1976**
Kenneth Appel and Wolfgang Haken released their proof of the 4-color theorem.

■ **1979**
L.G. Khachian published his ellipsoidal algorithm for solving linear programming problems in polynomial time.

■ **1984**
Narendra Karmarkar devised his interior algorithm for solving classes of linear programming problems.

■ **1991**
Donald Miller and Joseph Pekny published an algorithm for solving a class of traveling salesperson problems.

1800 1850 1900 1950 2000

出版者的话

为适应当前我国高校各类创新人才培养的需要，大力推进教育部倡导的双语教学，配合教育部实施的“高等学校教学质量与教学改革工程”和“精品课程”建设的需要，我社开始有计划、大规模地开展了海外优秀理科系列教材的影印及改编工作。海外优秀教材在立体化配套、多种教学资源的整合以及为课程提供整体教学解决方案等方面对我们有不少可资借鉴之处。但一个不容忽视的问题是，外版教材与我国现行的教学内容、教学体系、教学模式和习惯等存在着巨大的差异。譬如，重点课程的原版教材通常很厚，内容很多，容量是国内自编教材的好几倍，国外的情况是，老师未必会都讲，剩下大量的内容留给学生自学；而国内的情况则不尽相同。受国内教学学时所限，完全照搬是不合时宜的。教材的国际化必须与本民族的文化教育传统相融合，在原有的基础上吸收国外优秀教材的长处，这使得我们需要对外文原版教材进行适当的改编。改编不是简单地使内容增删，而是结合国内教学特点，引进国外先进的教学思想，在教学内容和方式上更中国化，使之更符合国内的课程设置及教学环境。

在引进改编海外优秀教材的过程中，我们坚持了两条原则：1. 精选版本，打造精品系列；2. 慎选改编者，保证品质。

首先，我们和 Pearson Education，John Wiley & Sons，McGraw-Hill 以及 Thomson Learning 等国外出版公司进行了广泛接触，经推荐并在国内专家的协助下，提交引进版权总数达200余种，学科专业领域涉及数学、物理、化学化工、地理、环境等。收到样书后，我们聘请了国内高校一线教师、专家学者参与这些原版教材的评介工作，从中遴选出了一批优秀教材进行改编，并组织出版。这批教材普遍具有以下特点：(1) 基本上是近几年出版的，在国际上被广泛使用，在同类教材中具有相当的权威性；(2) 高版次，历经多年教学实践检验，内容翔实准确，反映时代要求；(3) 各种教学资源配套整

齐，为师生提供了极大的便利；(4) 插图精美、丰富，图文并茂，与正文相辅相成；(5) 语言简练、流畅，可读性强，比较适合非英语国家的学生阅读。

其次，慎选改编者。原版教材确定后，随之碰到的问题是寻找合适的改编者。要改编一本教材，必须要从头到尾吃透它，有这样的精力自编一本教材都绰绰有余了。我们与国内众多高等院校的众多专家学者进行了广泛的接触和细致的协商，几经酝酿，最终确定下来改编者。大多数改编者都是有国外留学背景的中青年学者，他们既有相当高的学术水平，又热爱教学，活跃在教学第一线。他们能够承担此任，不单是因为他们了解引进版教材的知识结构、表达方式和写作方法，更重要的是他们有精力、有热情，愿意付出，有的甚至付出了比写一本新教材更多的劳动。我们向他们表示最真诚的谢意。

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，这套引进改编的教材体现了一定的权威性、系统性、先进性和经济性等特点。

这套教材出版后，我们将结合各高校的双语教学计划，开展大规模的宣传、培训工作，及时地将本套丛书推荐给高校使用。在使用过程中，我们衷心希望广大高校教师和同学提出宝贵的意见和建议。如有好的教材值得引进，也请与高等教育出版社高等理科分社联系。

联系电话：010-58581384(数学)；010-58581374(物理)；010-58581380(化学化工)。E-mail: xuke@hep.com.cn。

高等教育出版社

2005 年 5 月

Preface

Today an increasing proportion of the applications of mathematics involves discrete rather than continuous models. The main reason for this trend is the integration of the computer into more and more of modern society. This book is intended for a one-semester introductory course in discrete mathematics.

Prerequisites Even though a course taught from this book requires few formal mathematical prerequisites, students are assumed to have the mathematical maturity ordinarily obtained by taking at least two years of high school mathematics, including problem-solving and algorithmic skills, and the ability to think abstractly.

Approach This book has a strong algorithmic emphasis that serves to unify the material. Algorithms are presented in English so that knowledge of a particular programming language is not required.

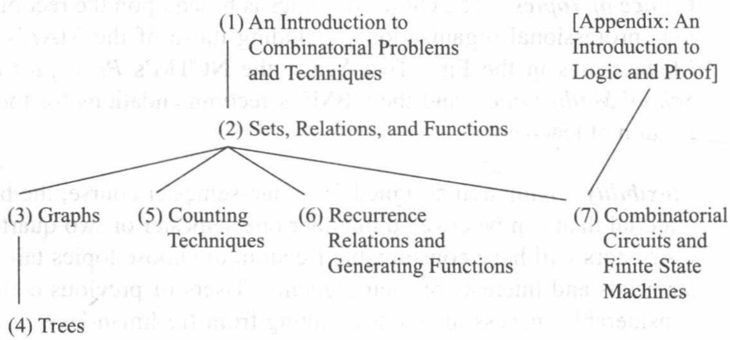
Choice of Topics The choice of topics is based upon the recommendations of various professional organizations, including those of the MAA's Panel on Discrete Mathematics in the First Two Years, the NCTM's *Principles and Standards for School Mathematics*, and the CBMS's recommendations for the mathematical education of teachers.

Flexibility Although designed for a one-semester course, the book contains more material than can be covered in either one semester or two quarters. Consequently, instructors will have considerable freedom to choose topics tailored to the particular needs and interests of their students. Users of previous editions have reported considerable success in courses ranging from freshman-level courses for computer science students to upper-level courses for mathematics majors. The present edition continues to allow instructors the flexibility to devise a course that is appropriate for a variety of different types of students.

Changes in the Fourth Edition At the suggestion of several users of the third edition, additional historical comments have been added; these are included at the end of each chapter. In addition, Chapters 3 and 4 have been rewritten so as to give the breadth-first search algorithm a more prominent role. (It now appears in Section 3.3 and is used in Sections 3.4 and 4.2.) Many examples in Chapters 3 and 4 have also been rewritten to be more useful to instructors who do not wish to discuss the details of the formal presentations of the algorithms. These examples now precede the algorithms and better reveal how the algorithms work without requiring discussion of the formal algorithms themselves. The previously separate sections on spanning trees and minimal and maximal spanning trees have been combined into a new Section 4.2, and the introductory material on matrices has been removed from Chapter 3. Another new appendix (Appendix B) describes the looping and branching structures used in the book's algorithms. Additional changes to the exposition have also been made throughout the book to improve the clarity of the writing.

Exercises The exercise sets in this book have been designed for flexibility. Many straightforward computational and algorithmic exercises are included after each section. These exercises give students hands-on practice with the concepts and algorithms of discrete mathematics and are especially important for students whose mathematical backgrounds are weak. Other exercises extend the material in the text or introduce new concepts not treated there. Exercise numbers in color indicate the more challenging problems. An instructor should choose those exercises appropriate to his or her course and students. Answers to odd-numbered computational exercises appear at the end of the book. At the end of each chapter, a set of **Supplementary Exercises** is provided. These reprise the most important concepts and techniques of the chapter and also explore new ideas not covered elsewhere.

Chapter Independence The sequence of chapters allows considerable flexibility in teaching a course from this book. The following diagram shows the logical dependence of the chapters. Although this book assumes only the familiarity with logic and proof ordinarily gained in high-school geometry, an appendix (Appendix A) is provided for those who prefer a more formal treatment. If this appendix is covered, it may be taught at any time as an independent unit or in combination with Chapter 7.



Chapters 1 and 2 are introductory in nature. Chapter 1, which should be covered fairly quickly, gives a sampling of the sort of discrete problems the course treats. Some questions are raised that will not be answered until later in the book. Section 1.4 contains a discussion of complexity that some instructors may want to omit or delay until students have had more experience with algorithms. An instructor may wish to cover only the illustrative algorithms in this section that are most relevant to his or her students.

Chapter 2 reviews various basic topics, including sets, relations, functions, and mathematical induction. It can be taught more or less rapidly depending on the mathematical backgrounds of the students and the level of the course. It should be possible for students with good mathematics backgrounds to be able to read much of Chapter 2 on their own. The remaining chapters are, as the diagram shows, independent except that Chapter 4 depend on Chapter 3.

Possible Courses A course emphasizing graph theory and its applications would cover most of Chapters 3–4, while a course with less graph theory would concentrate on Chapters 5–7.

Courses of various levels of sophistication can be taught from this book. For example, the topic of computational complexity is of great importance, and so attention is given to the complexity of many algorithms in this text. Yet it is a difficult topic, and the detail with which it is treated should correspond to the intended level of the course and the preparation of students.

Computer Projects Each chapter ends with a set of computer projects related to its content, algorithmic and otherwise. These are purposely stated in general terms, so as to be appropriate to students using various computing systems and languages.

Supplements A *Student's Solution Manual*, available for purchase by students, provides detailed, worked-out solutions to the odd-numbered exercises (ISBN 0-201-75483-5). An *Instructor's Answer Manual*, containing answers to all even-numbered computational exercises, is also available (ISBN 0-201-75482-7).

Acknowledgements We would like to thank the following mathematics professionals whose reviews guided the preparation of this text: Dorothee Blum, Millersville University; Richard Brualdi, University of Wisconsin, Madison; John L. Bryant, Florida State University; Richard Crittenden, Portland State University; Klaus Fischer, George Mason University; Dennis Grantham, East Texas State University; William R. Hare, Clemson University; Christopher Hee, Eastern Michigan University; Frederick Hoffman, Florida Atlantic University; Julian L. Hook, Florida International University; Carmelita Keyes, Broome Community College; Richard K. Molnar, Macalester College; Catherine Murphy, Purdue University, Calumet; Charles Nelson, University of Florida; Fred Schuurmann, Miami University; Karen Sharp, Charles S. Mott Community College; and Donovan H. Van Osdol, University of New Hampshire. The second edition was improved by the helpful comments of our colleagues Saad El-Zanati, Michael Plantholt, and Shailesh Tipnis, and other users of the text throughout the country, as well as by Elaine Bohanon, Bemidji State University; George Dimitroff, Evergreen State College; Richard Enstad, University of Wisconsin-Whitewater; Donald Goldsmith, Western Michigan University; Thomas R. Graviss, Kentucky Educational Network; Gary Klatt, University of Wisconsin-Whitewater; Mark Michael, Kings College; Peter Morris, Shepherd College; Dix H. Pettey, University of Missouri; Matt Pickard, University of Puget Sound; Terry Walters, University of Tennessee; Porter Webster, University of Southern Mississippi; Richard Weimer, Frostburg State University; Thomas Weininger, University of Wisconsin-Eau Claire; and Mark Woodard, Furman University. The third edition was further improved by the comments of users of the text throughout the country, as well as by the following list of colleague reviewers: Veena Chadha, University of Wisconsin-Eau Claire; Gary Chartrand, West Michigan University; Tilak de Alwis, Southeastern Louisiana University; Ron Dalla, Eastern Washington University; George Dimitroff, Evergreen State College; Gayla S. Domke, Georgia State University; Jerome Eisenfeld, University of Texas at Arlington; Kathleen Elder, Frostburg State University; Klaus Fischer, George Mason University; Donald A. Goral, Northern Virginia Community College; Natasa Jonoska, University of South Florida; Thomas Kiley, George Mason University; Theresa D. Magnus, St. Mary's College; Chris Mawata, University of Tennessee; Robert C. Mers, North Carolina A & T University; Catherine M. Murphy, Purdue University, Calumet; Anne Quinn, Edinboro University of Pennsylvania; Steen Pedersen, Wright State University; Dix H. Pettey, University of Missouri, Columbia; James L. Richards, Bemidji State University; A. Allan Riveland, Washburn University; Mohan Shrikhande, Central Michigan University; Allan Schwenk, Western Michigan University; and our colleague, Roger Eggleton.

We wish to offer special thanks to Michael Plantholt and Dean Sanders of Illinois State University, who independently checked all the algorithms in the third edition for both correctness and readability. Their suggestions led to substantial revisions and improvements of the algorithms.

Changes made for this fourth edition were guided by the comments of the following reviewers: Mark Ferris, Midwestern State University; Johanne Hattingh, Georgia State University; Colleen Hoover, St. Mary's College; Jason Miller, Truman State University; and Richard Rockwell, Pacific Union College.

We also appreciate the excellent editorial work of Cindy Cody and Kevin Bradley during the production process.

John A. Dossey
Albert D. Otto
Lawrence E. Spence
Charles Vanden Eynden

To the Student

This book is concerned with the *discrete*, that is, finite processes and sets of elements that can be listed. This contrasts with calculus, which has to do with infinite processes and intervals of real numbers.

Although discrete mathematics has been around for a long time, it has enjoyed a recent rapid expansion, paralleling the growth in the importance of computers. A digital computer is a complicated, but essentially finite, machine. At any given time it can be described by a large, but finite, sequence of 0s and 1s, corresponding to the internal states of its electronic components. Thus discrete mathematics is essential in understanding computers and how they can be applied.

An important part of discrete mathematics has to do with *algorithms*, which are explicit instructions for performing certain computations. You first learned algorithms in elementary school, for arithmetic is full of them. For example, there is the *long division algorithm*, which might cause an elementary school student to write down something like the following tableau.

$$\begin{array}{r} 32 \\ 13\overline{)425} \\ \underline{39} \\ 35 \\ \underline{26} \\ 9 \end{array}$$

Internally, the student is applying certain memorized procedures: *There are three 13s in 42, 3 times 13 is 39, 42 minus 39 is 3, bring down the 5*, etc. These procedures comprise the algorithm.

Another example of an algorithm is a computer program. Suppose a small business wants to identify all customers who owe it more than \$100 and have been delinquent in payments for at least 3 months. Even though the company's computer files contain this information, it constitutes only a small portion of their data. Thus a program must be written to sift out exactly what the company wants to know. This program consists of a precise set of instructions to the computer, covering all possibilities, that causes it to isolate the desired list of customers.

Our two examples of algorithms are similar in that the entity executing the algorithm does not have to understand why it works. Students in elementary school generally do not know why the long division algorithm gives the correct answer, only what the proper steps are. Of course, a computer doesn't understand anything; it just follows orders (and if its orders are incorrect, so that the program is wrong, the computer will dutifully produce the wrong answer).

If you are taking a course using this book, however, you are no longer in elementary school and you are a human being, not a computer. Thus you will be expected to know not only *how* our algorithms work, but *why*.

We will investigate some algorithms you probably have never seen before. For example, suppose you are planning to drive from Miami, Florida, to Seattle, Washington. Even if you stick to the interstate highways, there are hundreds of ways to go. Which way is the shortest? You might get out a map and, after playing around, find a route you *thought* was shortest, but could you be sure?

There is an algorithm that you could apply to this problem that would give you the correct answer. Better yet, you could program the algorithm into a computer, and let it find the shortest route. That algorithm is explained in this book.

We will be interested not only in the how and why of algorithms, but also in the *how long*. Computer time can be expensive, so before we give a computer a job to do we may want an estimate of how long it will take. Sometimes the surprising answer is that the computation will take so long as to make a computer solution impractical, even if we use the largest and fastest existing machines. It is a popular but incorrect idea that computers can do any computation. No computer can take the data from the world's weather stations and use it to predict future weather accurately more than a few days in advance. The fact that no one knows how to do certain computations efficiently can actually be useful. For example, if n is the product of two primes of about 150 decimal digits, then to factor n takes hundreds of years (even using the best methods and computers known), and this is the basis of an important system of cryptography.

You have probably already heard a number of times that mathematics is not a spectator sport, and that the only way to learn mathematics is by doing it. There is an important reason we are repeating this advice here. IT'S TRUE! Moreover, it's the best thing we know to tell you. You can't learn to play the guitar or shoot free throws just by watching someone else do these things, and you can't learn discrete mathematics just by reading this book or attending lectures. The mind must be in gear and active. When reading a mathematics book, you should always have paper and pencil handy to work out examples and the details of computations. When attending a mathematics lecture, it is best if you have read the material already. Then you can concentrate on seeing if your understanding of the content agrees with that of the professor, and you can ask questions about any difficult points.

Of course, one of the best ways to be active in learning mathematics is by doing exercises. There are many of these in this book. Some are purely computational, others test understanding of concepts, and some require constructing proofs. Answers to odd-numbered computational exercises are in the back of the book, but *don't look before you have determined your own answer.* If your work consistently gives the same answer as in the back of the book, then you can have confidence that you are on the right track.

Some exercises are harder than others. The more time you spend on such exercises, the more you learn. There is a common notion (reinforced by some courses) that if you can't figure out how to do a problem in five minutes you should go on to the next problem. This attitude becomes less and less relevant the more skillful you become. Very few accomplishments of any importance can be done in five minutes.

Many students do not realize the importance of learning the technical language of what they are studying. It is traditional in mathematics to assign special meaning to short, common words such as *set, function, relation, graph, tree, network.* These words have precise definitions that you must learn. Otherwise, how can you understand what you read in this book, or what your professor is saying? These technical words are necessary for efficient communication. How would you like to explain a baseball game to someone if you were not allowed to use the particular language of that sport? Every time you wanted to say that a pitch was a *ball*, you would have to say that it was a pitch that was not in the strike zone and that the batter didn't swing at it. For that matter, *strike zone* is a technical term that would also need an explanation in each instance. Communication on such a basis would be almost impossible.

Finally, proper terminology is necessary to share information in a useful way with others. Mathematics is a human endeavor, and human cooperation depends on communication. In the real world, it is seldom sufficient simply to figure something out. You must be able to explain it to other people, and to convince them that your solution is correct.

We hope your study of discrete mathematics is successful, and that you get from it techniques and attitudes that you will find useful in many contexts.

目 录