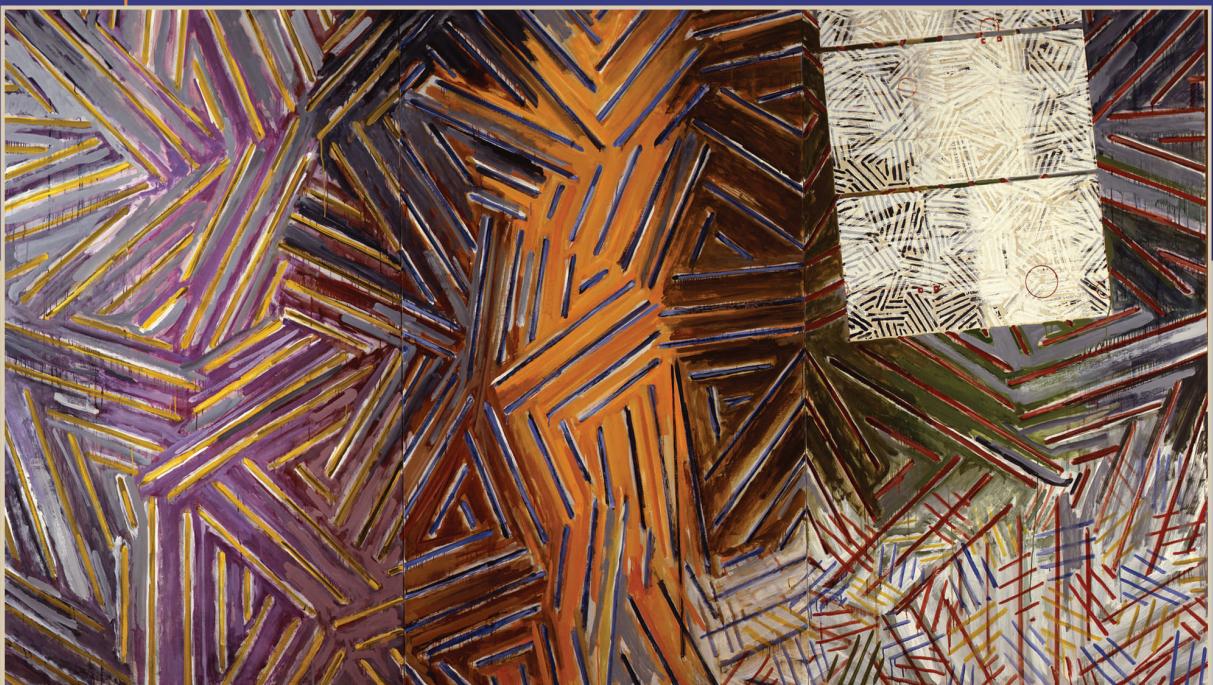


Kenneth H. Rosen



**Discrete
Mathematics
and Its
Applications**

SEVENTH EDITION

Discrete Mathematics and Its Applications

Seventh Edition

Kenneth H. Rosen

*Monmouth University
(and formerly AT&T Laboratories)*



The McGraw-Hill Companies



DISCRETE MATHEMATICS AND ITS APPLICATIONS, SEVENTH EDITION

Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2012 by The McGraw-Hill Companies, Inc. All rights reserved. Previous editions © 2007, 2003, and 1999. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of The McGraw-Hill Companies, Inc., including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 0 DOW/DOW 1 0 9 8 7 6 5 4 3 2 1

ISBN 978-0-07-338309-5

MHID 0-07-338309-0

Vice President & Editor-in-Chief: *Marty Lange*

Editorial Director: *Michael Lange*

Global Publisher: *Raghothaman Srinivasan*

Executive Editor: *Bill Stenquist*

Development Editors: *Lorraine K. Buczak/Rose Kernan*

Senior Marketing Manager: *Curt Reynolds*

Project Manager: *Robin A. Reed*

Buyer: *Sandy Ludovissy*

Design Coordinator: *Brenda A. Rolwes*

Cover painting: Jasper Johns, *Between the Clock and the Bed*, 1981. Oil on Canvas (72 × 126 1/4 inches)

Collection of the artist. Photograph by Glenn Stiegelman. Cover Art © Jasper Johns/Licensed by VAGA, New York, NY

Cover Designer: *Studio Montage, St. Louis, Missouri*

Lead Photo Research Coordinator: *Carrie K. Burger*

Media Project Manager: *Tammy Juran*

Production Services/Compositor: *RPK Editorial Services/PreTeX, Inc.*

Typeface: *10.5/12 Times Roman*

Printer: *R.R. Donnelley*

All credits appearing on this page or at the end of the book are considered to be an extension of the copyright page.

Library of Congress Cataloging-in-Publication Data

Rosen, Kenneth H.

Discrete mathematics and its applications / Kenneth H. Rosen. — 7th ed.

p. cm.

Includes index.

ISBN 0-07-338309-0

1. Mathematics. 2. Computer science—Mathematics. I. Title.

QA39.3.R67 2012

511—dc22

2011011060

Contents

About the Author vi

Preface vii

The Companion Website xvi

To the Student xvii

1	The Foundations: Logic and Proofs	1
1.1	Propositional Logic	1
1.2	Applications of Propositional Logic.....	16
1.3	Propositional Equivalences	25
1.4	Predicates and Quantifiers	36
1.5	Nested Quantifiers	57
1.6	Rules of Inference	69
1.7	Introduction to Proofs	80
1.8	Proof Methods and Strategy.....	92
	<i>End-of-Chapter Material</i>	109
2	Basic Structures: Sets, Functions, Sequences, Sums, and Matrices .	115
2.1	Sets	115
2.2	Set Operations.....	127
2.3	Functions	138
2.4	Sequences and Summations.....	156
2.5	Cardinality of Sets	170
2.6	Matrices	177
	<i>End-of-Chapter Material</i>	185
3	Algorithms	191
3.1	Algorithms	191
3.2	The Growth of Functions.....	204
3.3	Complexity of Algorithms	218
	<i>End-of-Chapter Material</i>	232
4	Number Theory and Cryptography.....	237
4.1	Divisibility and Modular Arithmetic	237
4.2	Integer Representations and Algorithms	245
4.3	Primes and Greatest Common Divisors	257
4.4	Solving Congruences.....	274
4.5	Applications of Congruences.....	287
4.6	Cryptography	294
	<i>End-of-Chapter Material</i>	306

5 Induction and Recursion	311
5.1 Mathematical Induction	311
5.2 Strong Induction and Well-Ordering	333
5.3 Recursive Definitions and Structural Induction.....	344
5.4 Recursive Algorithms	360
5.5 Program Correctness	372
<i>End-of-Chapter Material</i>	377
6 Counting	385
6.1 The Basics of Counting.....	385
6.2 The Pigeonhole Principle	399
6.3 Permutations and Combinations	407
6.4 Binomial Coefficients and Identities	415
6.5 Generalized Permutations and Combinations	423
6.6 Generating Permutations and Combinations	434
<i>End-of-Chapter Material</i>	439
7 Discrete Probability	445
7.1 An Introduction to Discrete Probability	445
7.2 Probability Theory	452
7.3 Bayes' Theorem	468
7.4 Expected Value and Variance	477
<i>End-of-Chapter Material</i>	494
8 Advanced Counting Techniques	501
8.1 Applications of Recurrence Relations	501
8.2 Solving Linear Recurrence Relations	514
8.3 Divide-and-Conquer Algorithms and Recurrence Relations.....	527
8.4 Generating Functions	537
8.5 Inclusion–Exclusion	552
8.6 Applications of Inclusion–Exclusion	558
<i>End-of-Chapter Material</i>	565
9 Relations	573
9.1 Relations and Their Properties	573
9.2 n -ary Relations and Their Applications	583
9.3 Representing Relations	591
9.4 Closures of Relations	597
9.5 Equivalence Relations	607
9.6 Partial Orderings	618
<i>End-of-Chapter Material</i>	633

10 Graphs	641
10.1 Graphs and Graph Models	641
10.2 Graph Terminology and Special Types of Graphs	651
10.3 Representing Graphs and Graph Isomorphism	668
10.4 Connectivity	678
10.5 Euler and Hamilton Paths	693
10.6 Shortest-Path Problems	707
10.7 Planar Graphs	718
10.8 Graph Coloring	727
<i>End-of-Chapter Material</i>	735
11 Trees	745
11.1 Introduction to Trees	745
11.2 Applications of Trees	757
11.3 Tree Traversal	772
11.4 Spanning Trees	785
11.5 Minimum Spanning Trees	797
<i>End-of-Chapter Material</i>	803
12 Boolean Algebra	811
12.1 Boolean Functions	811
12.2 Representing Boolean Functions	819
12.3 Logic Gates	822
12.4 Minimization of Circuits	828
<i>End-of-Chapter Material</i>	843
13 Modeling Computation	847
13.1 Languages and Grammars	847
13.2 Finite-State Machines with Output	858
13.3 Finite-State Machines with No Output	865
13.4 Language Recognition	878
13.5 Turing Machines	888
<i>End-of-Chapter Material</i>	899
Appendices	A-1
1 Axioms for the Real Numbers and the Positive Integers	1
2 Exponential and Logarithmic Functions	7
3 Pseudocode	11
Suggested Readings B-1	
Answers to Odd-Numbered Exercises S-1	
Photo Credits C-1	
Index of Biographies I-1	
Index I-2	

About the Author

Kenneth H. Rosen has had a long career as a Distinguished Member of the Technical Staff at AT&T Laboratories in Monmouth County, New Jersey. He currently holds the position of Visiting Research Professor at Monmouth University, where he teaches graduate courses in computer science.

Dr. Rosen received his B.S. in Mathematics from the University of Michigan, Ann Arbor (1972), and his Ph.D. in Mathematics from M.I.T. (1976), where he wrote his thesis in the area of number theory under the direction of Harold Stark. Before joining Bell Laboratories in 1982, he held positions at the University of Colorado, Boulder; The Ohio State University, Columbus; and the University of Maine, Orono, where he was an associate professor of mathematics. While working at AT&T Labs, he taught at Monmouth University, teaching courses in discrete mathematics, coding theory, and data security. He currently teaches courses in algorithm design and in computer security and cryptography.

Dr. Rosen has published numerous articles in professional journals in number theory and in mathematical modeling. He is the author of the widely used *Elementary Number Theory and Its Applications*, published by Pearson, currently in its sixth edition, which has been translated into Chinese. He is also the author of *Discrete Mathematics and Its Applications*, published by McGraw-Hill, currently in its seventh edition. *Discrete Mathematics and Its Applications* has sold more than 350,000 copies in North America during its lifetime, and hundreds of thousands of copies throughout the rest of the world. This book has also been translated into Spanish, French, Greek, Chinese, Vietnamese, and Korean. He is also co-author of *UNIX: The Complete Reference*; *UNIX System V Release 4: An Introduction*; and *Best UNIX Tips Ever*, all published by Osborne McGraw-Hill. These books have sold more than 150,000 copies, with translations into Chinese, German, Spanish, and Italian. Dr. Rosen is also the editor of the *Handbook of Discrete and Combinatorial Mathematics*, published by CRC Press, and he is the advisory editor of the CRC series of books in discrete mathematics, consisting of more than 55 volumes on different aspects of discrete mathematics, most of which are introduced in this book. Dr. Rosen serves as an Associate Editor for the journal *Discrete Mathematics*, where he works with submitted papers in several areas of discrete mathematics, including graph theory, enumeration, and number theory. He is also interested in integrating mathematical software into the educational and professional environments, and worked on several projects with Waterloo Maple Inc.'s MapleTM software in both these areas. Dr. Rosen has also worked with several publishing companies on their homework delivery platforms.

At Bell Laboratories and AT&T Laboratories, Dr. Rosen worked on a wide range of projects, including operations research studies, product line planning for computers and data communications equipment, and technology assessment. He helped plan AT&T's products and services in the area of multimedia, including video communications, speech recognition, speech synthesis, and image networking. He evaluated new technology for use by AT&T and did standards work in the area of image networking. He also invented many new services, and holds more than 55 patents. One of his more interesting projects involved helping evaluate technology for the AT&T attraction that was part of EPCOT Center.

Preface

In writing this book, I was guided by my long-standing experience and interest in teaching discrete mathematics. For the student, my purpose was to present material in a precise, readable manner, with the concepts and techniques of discrete mathematics clearly presented and demonstrated. My goal was to show the relevance and practicality of discrete mathematics to students, who are often skeptical. I wanted to give students studying computer science all of the mathematical foundations they need for their future studies. I wanted to give mathematics students an understanding of important mathematical concepts together with a sense of why these concepts are important for applications. And most importantly, I wanted to accomplish these goals without watering down the material.

For the instructor, my purpose was to design a flexible, comprehensive teaching tool using proven pedagogical techniques in mathematics. I wanted to provide instructors with a package of materials that they could use to teach discrete mathematics effectively and efficiently in the most appropriate manner for their particular set of students. I hope that I have achieved these goals.

I have been extremely gratified by the tremendous success of this text. The many improvements in the seventh edition have been made possible by the feedback and suggestions of a large number of instructors and students at many of the more than 600 North American schools, and at many universities in parts of the world, where this book has been successfully used.

This text is designed for a one- or two-term introductory discrete mathematics course taken by students in a wide variety of majors, including mathematics, computer science, and engineering. College algebra is the only explicit prerequisite, although a certain degree of mathematical maturity is needed to study discrete mathematics in a meaningful way. This book has been designed to meet the needs of almost all types of introductory discrete mathematics courses. It is highly flexible and extremely comprehensive. The book is designed not only to be a successful textbook, but also to serve as valuable resource students can consult throughout their studies and professional life.

Goals of a Discrete Mathematics Course

A discrete mathematics course has more than one purpose. Students should learn a particular set of mathematical facts and how to apply them; more importantly, such a course should teach students how to think logically and mathematically. To achieve these goals, this text stresses mathematical reasoning and the different ways problems are solved. Five important themes are interwoven in this text: mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and applications and modeling. A successful discrete mathematics course should carefully blend and balance all five themes.

1. *Mathematical Reasoning:* Students must understand mathematical reasoning in order to read, comprehend, and construct mathematical arguments. This text starts with a discussion of mathematical logic, which serves as the foundation for the subsequent discussions of methods of proof. Both the science and the art of constructing proofs are addressed. The technique of mathematical induction is stressed through many different types of examples of such proofs and a careful explanation of why mathematical induction is a valid proof technique.

2. *Combinatorial Analysis:* An important problem-solving skill is the ability to count or enumerate objects. The discussion of enumeration in this book begins with the basic techniques of counting. The stress is on performing combinatorial analysis to solve counting problems and analyze algorithms, not on applying formulae.
3. *Discrete Structures:* A course in discrete mathematics should teach students how to work with discrete structures, which are the abstract mathematical structures used to represent discrete objects and relationships between these objects. These discrete structures include sets, permutations, relations, graphs, trees, and finite-state machines.
4. *Algorithmic Thinking:* Certain classes of problems are solved by the specification of an algorithm. After an algorithm has been described, a computer program can be constructed implementing it. The mathematical portions of this activity, which include the specification of the algorithm, the verification that it works properly, and the analysis of the computer memory and time required to perform it, are all covered in this text. Algorithms are described using both English and an easily understood form of pseudocode.
5. *Applications and Modeling:* Discrete mathematics has applications to almost every conceivable area of study. There are many applications to computer science and data networking in this text, as well as applications to such diverse areas as chemistry, biology, linguistics, geography, business, and the Internet. These applications are natural and important uses of discrete mathematics and are not contrived. Modeling with discrete mathematics is an extremely important problem-solving skill, which students have the opportunity to develop by constructing their own models in some of the exercises.

Changes in the Seventh Edition

Although the sixth edition has been an extremely effective text, many instructors, including longtime users, have requested changes designed to make this book more effective. I have devoted a significant amount of time and energy to satisfy their requests and I have worked hard to find my own ways to make the book more effective and more compelling to students.

The seventh edition is a major revision, with changes based on input from more than 40 formal reviewers, feedback from students and instructors, and author insights. The result is a new edition that offers an improved organization of topics making the book a more effective teaching tool. Substantial enhancements to the material devoted to logic, algorithms, number theory, and graph theory make this book more flexible and comprehensive. Numerous changes in the seventh edition have been designed to help students more easily learn the material. Additional explanations and examples have been added to clarify material where students often have difficulty. New exercises, both routine and challenging, have been added. Highly relevant applications, including many related to the Internet, to computer science, and to mathematical biology, have been added. The companion website has benefited from extensive development activity and now provides tools students can use to master key concepts and explore the world of discrete mathematics, and many new tools under development will be released in the year following publication of this book.

I hope that instructors will closely examine this new edition to discover how it might meet their needs. Although it is impractical to list all the changes in this edition, a brief list that highlights some key changes, listed by the benefits they provide, may be useful.

More Flexible Organization

- Applications of propositional logic are found in a new dedicated section, which briefly introduces logic circuits.
- Recurrence relations are now covered in Chapter 2.
- Expanded coverage of countability is now found in a dedicated section in Chapter 2.

- Separate chapters now provide expanded coverage of algorithms (Chapter 3) and number theory and cryptography (Chapter 4).
- More second and third level heads have been used to break sections into smaller coherent parts.

Tools for Easier Learning

- Difficult discussions and proofs have been marked with the famous Bourbaki “dangerous bend” symbol in the margin.
- New marginal notes make connections, add interesting notes, and provide advice to students.
- More details and added explanations, in both proofs and exposition, make it easier for students to read the book.
- Many new exercises, both routine and challenging, have been added, while many existing exercises have been improved.

Enhanced Coverage of Logic, Sets, and Proof

- The satisfiability problem is addressed in greater depth, with Sudoku modeled in terms of satisfiability.
- Hilbert’s Grand Hotel is used to help explain uncountability.
- Proofs throughout the book have been made more accessible by adding steps and reasons behind these steps.
- A template for proofs by mathematical induction has been added.
- The step that applies the inductive hypothesis in mathematical induction proof is now explicitly noted.

Algorithms

- The pseudocode used in the book has been updated.
- Explicit coverage of algorithmic paradigms, including brute force, greedy algorithms, and dynamic programming, is now provided.
- Useful rules for big- O estimates of logarithms, powers, and exponential functions have been added.

Number Theory and Cryptography

- Expanded coverage allows instructors to include just a little or a lot of number theory in their courses.
- The relationship between the **mod** function and congruences has been explained more fully.
- The sieve of Eratosthenes is now introduced earlier in the book.
- Linear congruences and modular inverses are now covered in more detail.
- Applications of number theory, including check digits and hash functions, are covered in great depth.
- A new section on cryptography integrates previous coverage, and the notion of a cryptosystem has been introduced.
- Cryptographic protocols, including digital signatures and key sharing, are now covered.

Graph Theory

- A structured introduction to graph theory applications has been added.
- More coverage has been devoted to the notion of social networks.
- Applications to the biological sciences and motivating applications for graph isomorphism and planarity have been added.
- Matchings in bipartite graphs are now covered, including Hall's theorem and its proof.
- Coverage of vertex connectivity, edge connectivity, and n -connectedness has been added, providing more insight into the connectedness of graphs.

Enrichment Material

- Many biographies have been expanded and updated, and new biographies of Bellman, Bézout Bienyamé, Cardano, Catalan, Cocks, Cook, Dirac, Hall, Hilbert, Ore, and Tao have been added.
- Historical information has been added throughout the text.
- Numerous updates for latest discoveries have been made.

Expanded Media

- Extensive effort has been devoted to producing valuable web resources for this book.
- Extra examples in key parts of the text have been provided on companion website.
- Interactive algorithms have been developed, with tools for using them to explore topics and for classroom use.
- A new online ancillary, *The Virtual Discrete Mathematics Tutor*, available in fall 2012, will help students overcome problems learning discrete mathematics.
- A new homework delivery system, available in fall 2012, will provide automated homework for both numerical and conceptual exercises.
- Student assessment modules are available for key concepts.
- Powerpoint transparencies for instructor use have been developed.
- A supplement *Exploring Discrete Mathematics* has been developed, providing extensive support for using MapleTM or MathematicaTM in conjunction with the book.
- An extensive collection of external web links is provided.

Features of the Book

ACCESSIBILITY This text has proved to be easily read and understood by beginning students. There are no mathematical prerequisites beyond college algebra for almost all the content of the text. Students needing extra help will find tools on the companion website for bringing their mathematical maturity up to the level of the text. The few places in the book where calculus is referred to are explicitly noted. Most students should easily understand the pseudocode used in the text to express algorithms, regardless of whether they have formally studied programming languages. There is no formal computer science prerequisite.

Each chapter begins at an easily understood and accessible level. Once basic mathematical concepts have been carefully developed, more difficult material and applications to other areas of study are presented.

FLEXIBILITY This text has been carefully designed for flexible use. The dependence of chapters on previous material has been minimized. Each chapter is divided into sections of approximately the same length, and each section is divided into subsections that form natural blocks of material for teaching. Instructors can easily pace their lectures using these blocks.

WRITING STYLE The writing style in this book is direct and pragmatic. Precise mathematical language is used without excessive formalism and abstraction. Care has been taken to balance the mix of notation and words in mathematical statements.

MATHEMATICAL RIGOR AND PRECISION All definitions and theorems in this text are stated extremely carefully so that students will appreciate the precision of language and rigor needed in mathematics. Proofs are motivated and developed slowly; their steps are all carefully justified. The axioms used in proofs and the basic properties that follow from them are explicitly described in an appendix, giving students a clear idea of what they can assume in a proof. Recursive definitions are explained and used extensively.

WORKED EXAMPLES Over 800 examples are used to illustrate concepts, relate different topics, and introduce applications. In most examples, a question is first posed, then its solution is presented with the appropriate amount of detail.

APPLICATIONS The applications included in this text demonstrate the utility of discrete mathematics in the solution of real-world problems. This text includes applications to a wide variety of areas, including computer science, data networking, psychology, chemistry, engineering, linguistics, biology, business, and the Internet.

ALGORITHMS Results in discrete mathematics are often expressed in terms of algorithms; hence, key algorithms are introduced in each chapter of the book. These algorithms are expressed in words and in an easily understood form of structured pseudocode, which is described and specified in Appendix 3. The computational complexity of the algorithms in the text is also analyzed at an elementary level.

HISTORICAL INFORMATION The background of many topics is succinctly described in the text. Brief biographies of 83 mathematicians and computer scientists are included as footnotes. These biographies include information about the lives, careers, and accomplishments of these important contributors to discrete mathematics and images, when available, are displayed.

In addition, numerous historical footnotes are included that supplement the historical information in the main body of the text. Efforts have been made to keep the book up-to-date by reflecting the latest discoveries.

KEY TERMS AND RESULTS A list of key terms and results follows each chapter. The key terms include only the most important that students should learn, and not every term defined in the chapter.

EXERCISES There are over 4000 exercises in the text, with many different types of questions posed. There is an ample supply of straightforward exercises that develop basic skills, a large number of intermediate exercises, and many challenging exercises. Exercises are stated clearly and unambiguously, and all are carefully graded for level of difficulty. Exercise sets contain special discussions that develop new concepts not covered in the text, enabling students to discover new ideas through their own work.

Exercises that are somewhat more difficult than average are marked with a single star *; those that are much more challenging are marked with two stars **. Exercises whose solutions require calculus are explicitly noted. Exercises that develop results used in the text are clearly identified with the right pointing hand symbol . Answers or outlined solutions to all odd-

numbered exercises are provided at the back of the text. The solutions include proofs in which most of the steps are clearly spelled out.

REVIEW QUESTIONS A set of review questions is provided at the end of each chapter. These questions are designed to help students focus their study on the most important concepts and techniques of that chapter. To answer these questions students need to write long answers, rather than just perform calculations or give short replies.

SUPPLEMENTARY EXERCISE SETS Each chapter is followed by a rich and varied set of supplementary exercises. These exercises are generally more difficult than those in the exercise sets following the sections. The supplementary exercises reinforce the concepts of the chapter and integrate different topics more effectively.

COMPUTER PROJECTS Each chapter is followed by a set of computer projects. The approximately 150 computer projects tie together what students may have learned in computing and in discrete mathematics. Computer projects that are more difficult than average, from both a mathematical and a programming point of view, are marked with a star, and those that are extremely challenging are marked with two stars.

COMPUTATIONS AND EXPLORATIONS A set of computations and explorations is included at the conclusion of each chapter. These exercises (approximately 120 in total) are designed to be completed using existing software tools, such as programs that students or instructors have written or mathematical computation packages such as MapleTM or MathematicaTM. Many of these exercises give students the opportunity to uncover new facts and ideas through computation. (Some of these exercises are discussed in the *Exploring Discrete Mathematics* companion workbooks available online.)

WRITING PROJECTS Each chapter is followed by a set of writing projects. To do these projects students need to consult the mathematical literature. Some of these projects are historical in nature and may involve looking up original sources. Others are designed to serve as gateways to new topics and ideas. All are designed to expose students to ideas not covered in depth in the text. These projects tie mathematical concepts together with the writing process and help expose students to possible areas for future study. (Suggested references for these projects can be found online or in the printed *Student's Solutions Guide*.)

APPENDICES There are three appendixes to the text. The first introduces axioms for real numbers and the positive integers, and illustrates how facts are proved directly from these axioms. The second covers exponential and logarithmic functions, reviewing some basic material used heavily in the course. The third specifies the pseudocode used to describe algorithms in this text.

SUGGESTED READINGS A list of suggested readings for the overall book and for each chapter is provided after the appendices. These suggested readings include books at or below the level of this text, more difficult books, expository articles, and articles in which discoveries in discrete mathematics were originally published. Some of these publications are classics, published many years ago, while others have been published in the last few years.

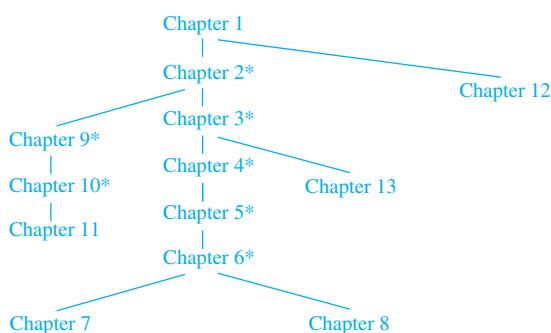
How to Use This Book

This text has been carefully written and constructed to support discrete mathematics courses at several levels and with differing foci. The following table identifies the core and optional sections. An introductory one-term course in discrete mathematics at the sophomore level can be based on the core sections of the text, with other sections covered at the discretion of the

instructor. A two-term introductory course can include all the optional mathematics sections in addition to the core sections. A course with a strong computer science emphasis can be taught by covering some or all of the optional computer science sections. Instructors can find sample syllabi for a wide range of discrete mathematics courses and teaching suggestions for using each section of the text can be found in the *Instructor's Resource Guide* available on the website for this book.

<i>Chapter</i>	<i>Core</i>	<i>Optional CS</i>	<i>Optional Math</i>
1	1.1–1.8 (as needed)		
2	2.1–2.4, 2.6 (as needed)		
3		3.1–3.3 (as needed)	2.5
4	4.1–4.4 (as needed)	4.5, 4.6	
5	5.1–5.3	5.4, 5.5	
6	6.1–6.3	6.6	6.4, 6.5
7	7.1	7.4	7.2, 7.3
8	8.1, 8.5	8.3	8.2, 8.4, 8.6
9	9.1, 9.3, 9.5	9.2	9.4, 9.6
10	10.1–10.5		10.6–10.8
11	11.1	11.2, 11.3	11.4, 11.5
12		12.1–12.4	
13		13.1–13.5	

Instructors using this book can adjust the level of difficulty of their course by choosing either to cover or to omit the more challenging examples at the end of sections, as well as the more challenging exercises. The chapter dependency chart shown here displays the strong dependencies. A star indicates that only relevant sections of the chapter are needed for study of a later chapter. Weak dependencies have been ignored. More details can be found in the Instructor Resource Guide.



Ancillaries

STUDENT'S SOLUTIONS GUIDE This student manual, available separately, contains full solutions to all odd-numbered problems in the exercise sets. These solutions explain why a particular method is used and why it works. For some exercises, one or two other possible approaches are described to show that a problem can be solved in several different ways. Suggested references for the writing projects found at the end of each chapter are also included in this volume. Also included are a guide to writing proofs and an extensive description of common

mistakes students make in discrete mathematics, plus sample tests and a sample crib sheet for each chapter designed to help students prepare for exams.

(ISBN-10: 0-07-735350-1) (ISBN-13: 978-0-07-735350-6)

INSTRUCTOR'S RESOURCE GUIDE This manual, available on the website and in printed form by request for instructors, contains full solutions to even-numbered exercises in the text. Suggestions on how to teach the material in each chapter of the book are provided, including the points to stress in each section and how to put the material into perspective. It also offers sample tests for each chapter and a test bank containing over 1500 exam questions to choose from. Answers to all sample tests and test bank questions are included. Finally, several sample syllabi are presented for courses with differing emphases and student ability levels.

(ISBN-10: 0-07-735349-8) (ISBN-13: 978-0-07-735349-0)

Acknowledgments

I would like to thank the many instructors and students at a variety of schools who have used this book and provided me with their valuable feedback and helpful suggestions. Their input has made this a much better book than it would have been otherwise. I especially want to thank Jerrold Grossman, Jean-Claude Evard, and Georgia Mederer for their technical reviews of the seventh edition and their “eagle eyes,” which have helped ensure the accuracy of this book. I also appreciate the help provided by all those who have submitted comments via the website.

I thank the reviewers of this seventh and the six previous editions. These reviewers have provided much helpful criticism and encouragement to me. I hope this edition lives up to their high expectations.

Reviewers for the Seventh Edition

Philip Barry <i>University of Minnesota, Minneapolis</i>	T.J. Duda <i>Columbus State Community College</i>	Jerry Ianni <i>LaGuardia Community College</i>
Miklos Bona <i>University of Florida</i>	Bruce Elenbogen <i>University of Michigan, Dearborn</i>	Ravi Janardan <i>University of Minnesota, Minneapolis</i>
Kirby Brown <i>Queens College</i>	Norma Elias <i>Purdue University, Calumet-Hammond</i>	Norliza Katuk <i>University of Utara Malaysia</i>
John Carter <i>University of Toronto</i>	Herbert Enderton <i>University of California, Los Angeles</i>	William Klostermeyer <i>University of North Florida</i>
Narendra Chaudhari <i>Nanyang Technological University</i>	Anthony Evans <i>Wright State University</i>	Przemo Kranz <i>University of Mississippi</i>
Allan Cochran <i>University of Arkansas</i>	Kim Factor <i>Marquette University</i>	Jaromy Kuhl <i>University of West Florida</i>
Daniel Cunningham <i>Buffalo State College</i>	Margaret Fleck <i>University of Illinois, Champaign</i>	Loredana Lanzani <i>University of Arkansas, Fayetteville</i>
George Davis <i>Georgia State University</i>	Peter Gillespie <i>Fayetteville State University</i>	Steven Leonhardi <i>Winona State University</i>
Andrzej Derdzinski <i>The Ohio State University</i>	Johannes Hattingh <i>Georgia State University</i>	Xu Liutong <i>Beijing University of Posts and Telecommunications</i>
Ronald Dotzel <i>University of Missouri-St. Louis</i>	Ken Holladay <i>University of New Orleans</i>	Vladimir Logvinenko <i>De Anza Community College</i>

Darrell Minor
Columbus State Community College

Keith Olson
Utah Valley University

Yongyuth Permpoontanalarp
King Mongkut's University of Technology, Thonburi

Galin Piatniskaia
University of Missouri, St. Louis

Stefan Robila
Montclair State University

Chris Rodger
Auburn University

Sukhit Singh
Texas State University, San Marcos

David Snyder
Texas State University, San Marcos

Wasin So
San Jose State University

Bogdan Suceava
California State University, Fullerton

Christopher Swanson
Ashland University

Bon Sy
Queens College

Matthew Walsh
Indiana-Purdue University, Fort Wayne

Gideon Weinstein
Western Governors University

David Wilczynski
University of Southern California

I would like to thank Bill Stenquist, Executive Editor, for his advocacy, enthusiasm, and support. His assistance with this edition has been essential. I would also like to thank the original editor, Wayne Yuhasz, whose insights and skills helped ensure the book's success, as well as all the many other previous editors of this book.

I want to express my appreciation to the staff of RPK Editorial Services for their valuable work on this edition, including Rose Kerman, who served as both the developmental editor and the production editor, and the other members of the RPK team, Fred Dahl, Martha McMaster, Erin Wagner, Harlan James, and Shelly Gerger-Knecht. I thank Paul Mailhot of PreTeX, Inc., the compositor, for the tremendous amount of work he devoted to producing this edition, and for his intimate knowledge of LaTeX. Thanks also to Danny Meldung of Photo Affairs, Inc., who was resourceful obtaining images for the new biographical footnotes.

The accuracy and quality of this new edition owe much to Jerry Grossman and Jean-Claude Evard, who checked the entire manuscript for technical accuracy and Georgia Mederer, who checked the accuracy of the answers at the end of the book and the solutions in the *Student's Solutions Guide* and *Instructor's Resource Guide*. As usual, I cannot thank Jerry Grossman enough for all his work authoring these two essential ancillaries.

I would also express my appreciation to the Science, Engineering, and Mathematics (SEM) Division of McGraw-Hill Higher Education for their valuable support for this new edition and the associated media content. In particular, thanks go to Kurt Strand: President, SEM, McGraw-Hill Higher Education, Marty Lange: Editor-in-Chief, SEM, Michael Lange: Editorial Director, Raghathan Srinivasan: Global Publisher, Bill Stenquist: Executive Editor, Curt Reynolds: Executive Marketing Manager, Robin A. Reed: Project Manager, Sandy Ludovissey: Buyer, Lorraine Buczek: In-house Developmental Editor, Brenda Rowles: Design Coordinator, Carrie K. Burger: Lead Photo Research Coordinator, and Tammy Juran: Media Project Manager.

Kenneth H. Rosen

The Companion Website

The extensive companion website accompanying this text has been substantially enhanced for the seventh edition. This website is accessible at www.mhhe.com/rosen. The homepage shows the *Information Center*, and contains login links for the site's *Student Site* and *Instructor Site*. Key features of each area are described below:

THE INFORMATION CENTER

The Information Center contains basic information about the book including the expanded table of contents (including subsection heads), the preface, descriptions of the ancillaries, and a sample chapter. It also provides a link that can be used to submit errata reports and other feedback about the book.

STUDENT SITE

The Student site contains a wealth of resources available for student use, including the following, tied into the text wherever the special icons displayed below are found in the text:

- Extra Examples**  You can find a large number of additional examples on the site, covering all chapters of the book. These examples are concentrated in areas where students often ask for additional material. Although most of these examples amplify the basic concepts, more-challenging examples can also be found here.
- Demo**  These applets enable you to interactively explore how important algorithms work, and are tied directly to material in the text with linkages to examples and exercises. Additional resources are provided on how to use and apply these applets.
- Assessment**  These interactive guides help you assess your understanding of 14 key concepts, providing a question bank where each question includes a brief tutorial followed by a multiple-choice question. If you select an incorrect answer, advice is provided to help you understand your error. Using these Self Assessments, you should be able to diagnose your problems and find appropriate help.
- Links**  This guide provides annotated links to hundreds of external websites containing relevant material such as historical and biographical information, puzzles and problems, discussions, applets, programs, and more. These links are keyed to the text by page number.

Additional resources in the Student site include:

- Exploring Discrete Mathematics** This ancillary provides help for using a computer algebra system to do a wide range of computations in discrete mathematics. Each chapter provides a description of relevant functions in the computer algebra system and how they are used, programs to carry out computations in discrete mathematics, examples, and exercises that can be worked using this computer algebra system. Two versions, *Exploring Discrete Mathematics with Maple*TM and *Exploring Discrete Mathematics with Mathematica*TM will be available.
- Applications of Discrete Mathematics** This ancillary contains 24 chapters—each with its own set of exercises—presenting a wide variety of interesting and important applications

covering three general areas in discrete mathematics: discrete structures, combinatorics, and graph theory. These applications are ideal for supplementing the text or for independent study.

- **A Guide to Proof-Writing** This guide provides additional help for writing proofs, a skill that many students find difficult to master. By reading this guide at the beginning of the course and periodically thereafter when proof writing is required, you will be rewarded as your proof-writing ability grows. (Also available in the *Student's Solutions Guide*.)
- **Common Mistakes in Discrete Mathematics** This guide includes a detailed list of common misconceptions that students of discrete mathematics often have and the kinds of errors they tend to make. You are encouraged to review this list from time to time to help avoid these common traps. (Also available in the *Student's Solutions Guide*.)
- **Advice on Writing Projects** This guide offers helpful hints and suggestions for the Writing Projects in the text, including an extensive bibliography of helpful books and articles for research; discussion of various resources available in print and online; tips on doing library research; and suggestions on how to write well. (Also available in the *Student's Solutions Guide*.)
- **The Virtual Discrete Mathematics Tutor** This extensive ancillary provides students with valuable assistance as they make the transition from lower-level courses to discrete mathematics. The errors students have made when studying discrete mathematics using this text has been analyzed to design this resource. Students will be able to get many of their questions answered and can overcome many obstacles via this ancillaries. The *Virtual Discrete Mathematics Tutor* is expected to be available in the fall of 2012.

INSTRUCTOR SITE

This part of the website provides access to all of the resources on the Student Site, as well as these resources for instructors:

- **Suggested Syllabi** Detailed course outlines are shown, offering suggestions for courses with different emphases and different student backgrounds and ability levels.
- **Teaching Suggestions** This guide contains detailed teaching suggestions for instructors, including chapter overviews for the entire text, detailed remarks on each section, and comments on the exercise sets.
- **Printable Tests** Printable tests are offered in TeX and Word format for every chapter, and can be customized by instructors.
- **PowerPoints Lecture Slides and PowerPoint Figures and Tables** An extensive collection of PowerPoint slides for all chapters of the text are provided for instructor use. In addition, images of all figures and tables from the text are provided as PowerPoint slides.
- **Homework Delivery System** An extensive homework delivery system, under development for availability in fall 2012, will provide questions tied directly to the text, so that students will be able to do assignments on-line. Moreover, they will be able to use this system in a tutorial mode. This system will be able to automatically grade assignments, and deliver free-form student input to instructors for their own analysis. Course management capabilities will be provided that will allow instructors to create assignments, automatically assign and grade homework, quiz, and test questions from a bank of questions tied directly to the text, create and edit their own questions, manage course announcements and due dates, and track student progress.

To the Student

What is discrete mathematics? Discrete mathematics is the part of mathematics devoted to the study of discrete objects. (Here *discrete* means consisting of distinct or unconnected elements.) The kinds of problems solved using discrete mathematics include:

- How many ways are there to choose a valid password on a computer system?
- What is the probability of winning a lottery?
- Is there a link between two computers in a network?
- How can I identify spam e-mail messages?
- How can I encrypt a message so that no unintended recipient can read it?
- What is the shortest path between two cities using a transportation system?
- How can a list of integers be sorted so that the integers are in increasing order?
- How many steps are required to do such a sorting?
- How can it be proved that a sorting algorithm correctly sorts a list?
- How can a circuit that adds two integers be designed?
- How many valid Internet addresses are there?

You will learn the discrete structures and techniques needed to solve problems such as these.

More generally, discrete mathematics is used whenever objects are counted, when relationships between finite (or countable) sets are studied, and when processes involving a finite number of steps are analyzed. A key reason for the growth in the importance of discrete mathematics is that information is stored and manipulated by computing machines in a discrete fashion.

WHY STUDY DISCRETE MATHEMATICS? There are several important reasons for studying discrete mathematics. First, through this course you can develop your mathematical maturity: that is, your ability to understand and create mathematical arguments. You will not get very far in your studies in the mathematical sciences without these skills.

Second, discrete mathematics is the gateway to more advanced courses in all parts of the mathematical sciences. Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems. Students find these courses much more difficult when they have not had the appropriate mathematical foundations from discrete math. One student has sent me an e-mail message saying that she used the contents of this book in every computer science course she took!

Math courses based on the material studied in discrete mathematics include logic, set theory, number theory, linear algebra, abstract algebra, combinatorics, graph theory, and probability theory (the discrete part of the subject).

Also, discrete mathematics contains the necessary mathematical background for solving problems in operations research (including many discrete optimization techniques), chemistry, engineering, biology, and so on. In the text, we will study applications to some of these areas.

Many students find their introductory discrete mathematics course to be significantly more challenging than courses they have previously taken. One reason for this is that one of the primary goals of this course is to teach mathematical reasoning and problem solving, rather than a discrete set of skills. The exercises in this book are designed to reflect this goal. Although there are plenty of exercises in this text similar to those addressed in the examples, a large

percentage of the exercises require original thought. This is intentional. The material discussed in the text provides the tools needed to solve these exercises, but your job is to successfully apply these tools using your own creativity. One of the primary goals of this course is to learn how to attack problems that may be somewhat different from any you may have previously seen. Unfortunately, learning how to solve only particular types of exercises is not sufficient for success in developing the problem-solving skills needed in subsequent courses and professional work. This text addresses many different topics, but discrete mathematics is an extremely diverse and large area of study. One of my goals as an author is to help you develop the skills needed to master the additional material you will need in your own future pursuits.

THE EXERCISES I would like to offer some advice about how you can best learn discrete mathematics (and other subjects in the mathematical and computing sciences). You will learn the most by actively working exercises. I suggest that you solve as many as you possibly can. After working the exercises your instructor has assigned, I encourage you to solve additional exercises such as those in the exercise sets following each section of the text and in the supplementary exercises at the end of each chapter. (Note the key explaining the markings preceding exercises.)

Key to the Exercises

no marking	A routine exercise
*	A difficult exercise
**	An extremely challenging exercise
	An exercise containing a result used in the book (Table 1 on the following page shows where these exercises are used.)
(Requires calculus)	An exercise whose solution requires the use of limits or concepts from differential or integral calculus

The best approach is to try exercises yourself before you consult the answer section at the end of this book. Note that the odd-numbered exercise answers provided in the text are answers only and not full solutions; in particular, the reasoning required to obtain answers is omitted in these answers. The *Student's Solutions Guide*, available separately, provides complete, worked solutions to all odd-numbered exercises in this text. When you hit an impasse trying to solve an odd-numbered exercise, I suggest you consult the *Student's Solutions Guide* and look for some guidance as to how to solve the problem. The more work you do yourself rather than passively reading or copying solutions, the more you will learn. The answers and solutions to the even-numbered exercises are intentionally not available from the publisher; ask your instructor if you have trouble with these.

WEB RESOURCES You are *strongly* encouraged to take advantage of additional resources available on the Web, especially those on the companion website for this book found at www.mhhe.com/rosen. You will find many Extra Examples designed to clarify key concepts; Self Assessments for gauging how well you understand core topics; Interactive Demonstration Applets exploring key algorithms and other concepts; a Web Resources Guide containing an extensive selection of links to external sites relevant to the world of discrete mathematics; extra explanations and practice to help you master core concepts; added instruction on writing proofs and on avoiding common mistakes in discrete mathematics; in-depth discussions of important applications; and guidance on utilizing Maple™ software to explore the computational aspects of discrete mathematics. Places in the text where these additional online resources are available are identified in the margins by special icons. You will also find (after fall 2012) the *Virtual Discrete Mathematics Tutor*, an on-line resource that provides extra support to help you make the transition from lower level courses to discrete mathematics. This tutorial should help answer many of your questions and correct errors that you may make, based on errors other students using this book, have made. For more details on these and other online resources, see the description of the companion website immediately preceding this “To the Student” message.

TABLE 1 Hand-Icon Exercises and Where They Are Used

<i>Section</i>	<i>Exercise</i>	<i>Section Where Used</i>	<i>Pages Where Used</i>
1.1	40	1.3	31
1.1	41	1.3	31
1.3	9	1.6	71
1.3	10	1.6	70, 71
1.3	15	1.6	71
1.3	30	1.6	71, 74
1.3	42	12.2	820
1.7	16	1.7	86
2.3	72	2.3	144
2.3	79	2.5	170
2.5	15	2.5	174
2.5	16	2.5	173
3.1	43	3.1	197
3.2	72	11.2	761
4.2	36	4.2	270
4.3	37	4.1	239
4.4	2	4.6	301
4.4	44	7.2	464
6.4	17	7.2	466
6.4	21	7.4	480
7.2	15	7.2	466
9.1	26	9.4	598
10.4	59	11.1	747
11.1	15	11.1	750
11.1	30	11.1	755
11.1	48	11.2	762
12.1	12	12.3	825
A.2	4	8.3	531

THE VALUE OF THIS BOOK My intention is to make your substantial investment in this text an excellent value. The book, the associated ancillaries, and companion website have taken many years of effort to develop and refine. I am confident that most of you will find that the text and associated materials will help you master discrete mathematics, just as so many previous students have. Even though it is likely that you will not cover some chapters in your current course, you should find it helpful—as many other students have—to read the relevant sections of the book as you take additional courses. Most of you will return to this book as a useful tool throughout your future studies, especially for those of you who continue in computer science, mathematics, and engineering. I have designed this book to be a gateway for future studies and explorations, and to be comprehensive reference, and I wish you luck as you begin your journey.

Kenneth H. Rosen

1

The Foundations: Logic and Proofs

- [1.1 Propositional Logic](#)
- [1.2 Applications of Propositional Logic](#)
- [1.3 Propositional Equivalences](#)
- [1.4 Predicates and Quantifiers](#)
- [1.5 Nested Quantifiers](#)
- [1.6 Rules of Inference](#)
- [1.7 Introduction to Proofs](#)
- [1.8 Proof Methods and Strategy](#)

The rules of logic specify the meaning of mathematical statements. For instance, these rules help us understand and reason with statements such as “There exists an integer that is not the sum of two squares” and “For every positive integer n , the sum of the positive integers not exceeding n is $n(n + 1)/2$.” Logic is the basis of all mathematical reasoning, and of all automated reasoning. It has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages, and to other areas of computer science, as well as to many other fields of study.

To understand mathematics, we must understand what makes up a correct mathematical argument, that is, a proof. Once we prove a mathematical statement is true, we call it a theorem. A collection of theorems on a topic organize what we know about this topic. To learn a mathematical topic, a person needs to actively construct mathematical arguments on this topic, and not just read exposition. Moreover, knowing the proof of a theorem often makes it possible to modify the result to fit new situations.

Everyone knows that proofs are important throughout mathematics, but many people find it surprising how important proofs are in computer science. In fact, proofs are used to verify that computer programs produce the correct output for all possible input values, to show that algorithms always produce the correct result, to establish the security of a system, and to create artificial intelligence. Furthermore, automated reasoning systems have been created to allow computers to construct their own proofs.

In this chapter, we will explain what makes up a correct mathematical argument and introduce tools to construct these arguments. We will develop an arsenal of different proof methods that will enable us to prove many different types of results. After introducing many different methods of proof, we will introduce several strategies for constructing proofs. We will introduce the notion of a conjecture and explain the process of developing mathematics by studying conjectures.

1.1 Propositional Logic

Introduction

The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments. Because a major goal of this book is to teach the reader how to understand and how to construct correct mathematical arguments, we begin our study of discrete mathematics with an introduction to logic.

Besides the importance of logic in understanding mathematical reasoning, logic has numerous applications to computer science. These rules are used in the design of computer circuits, the construction of computer programs, the verification of the correctness of programs, and in many other ways. Furthermore, software systems have been developed for constructing some, but not all, types of proofs automatically. We will discuss these applications of logic in this and later chapters.

Propositions

Our discussion begins with an introduction to the basic building blocks of logic—propositions. A **proposition** is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

EXAMPLE 1 All the following declarative sentences are propositions.



1. Washington, D.C., is the capital of the United States of America.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

Propositions 1 and 3 are true, whereas 2 and 4 are false.

Some sentences that are not propositions are given in Example 2.

EXAMPLE 2 Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. $x + 1 = 2$.
4. $x + y = z$.

Sentences 1 and 2 are not propositions because they are not declarative sentences. Sentences 3 and 4 are not propositions because they are neither true nor false. Note that each of sentences 3 and 4 can be turned into a proposition if we assign values to the variables. We will also discuss other ways to turn sentences such as these into propositions in Section 1.4.

We use letters to denote **propositional variables** (or **statement variables**), that is, variables that represent propositions, just as letters are used to denote numerical variables. The



ARISTOTLE (384 B.C.E.–322 B.C.E.) Aristotle was born in Stagirus (Stagira) in northern Greece. His father was the personal physician of the King of Macedonia. Because his father died when Aristotle was young, Aristotle could not follow the custom of following his father's profession. Aristotle became an orphan at a young age when his mother also died. His guardian who raised him taught him poetry, rhetoric, and Greek. At the age of 17, his guardian sent him to Athens to further his education. Aristotle joined Plato's Academy, where for 20 years he attended Plato's lectures, later presenting his own lectures on rhetoric. When Plato died in 347 B.C.E., Aristotle was not chosen to succeed him because his views differed too much from those of Plato. Instead, Aristotle joined the court of King Hermeas where he remained for three years, and married the niece of the King. When the Persians defeated Hermeas, Aristotle moved to Mytilene and, at the invitation of King Philip of Macedonia, he tutored Alexander, Philip's son, who later became Alexander the Great. Aristotle tutored Alexander for five years and after the death of King Philip, he returned to Athens and set up his own school, called the Lyceum.

Aristotle's followers were called the peripatetics, which means "to walk about," because Aristotle often walked around as he discussed philosophical questions. Aristotle taught at the Lyceum for 13 years where he lectured to his advanced students in the morning and gave popular lectures to a broad audience in the evening. When Alexander the Great died in 323 B.C.E., a backlash against anything related to Alexander led to trumped-up charges of impiety against Aristotle. Aristotle fled to Chalcis to avoid prosecution. He only lived one year in Chalcis, dying of a stomach ailment in 322 B.C.E.

Aristotle wrote three types of works: those written for a popular audience, compilations of scientific facts, and systematic treatises. The systematic treatises included works on logic, philosophy, psychology, physics, and natural history. Aristotle's writings were preserved by a student and were hidden in a vault where a wealthy book collector discovered them about 200 years later. They were taken to Rome, where they were studied by scholars and issued in new editions, preserving them for posterity.

conventional letters used for propositional variables are p, q, r, s, \dots . The **truth value** of a proposition is true, denoted by T, if it is a true proposition, and the truth value of a proposition is false, denoted by F, if it is a false proposition.

The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**. It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.



We now turn our attention to methods for producing new propositions from those that we already have. These methods were discussed by the English mathematician George Boole in 1854 in his book *The Laws of Thought*. Many mathematical statements are constructed by combining one or more propositions. New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

DEFINITION 1

Let p be a proposition. The *negation of p* , denoted by $\neg p$ (also denoted by \overline{p}), is the statement

“It is not the case that p .”

The proposition $\neg p$ is read “not p .” The truth value of the negation of p , $\neg p$, is the opposite of the truth value of p .

EXAMPLE 3

Find the negation of the proposition



“Michael’s PC runs Linux”

and express this in simple English.

Solution: The negation is

“It is not the case that Michael’s PC runs Linux.”

This negation can be more simply expressed as

“Michael’s PC does not run Linux.”

EXAMPLE 4

Find the negation of the proposition

“Vandana’s smartphone has at least 32GB of memory”

and express this in simple English.

Solution: The negation is

“It is not the case that Vandana’s smartphone has at least 32GB of memory.”

This negation can also be expressed as

“Vandana’s smartphone does not have at least 32GB of memory”

or even more simply as

“Vandana’s smartphone has less than 32GB of memory.”

TABLE 1 The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

Table 1 displays the **truth table** for the negation of a proposition p . This table has a row for each of the two possible truth values of a proposition p . Each row shows the truth value of $\neg p$ corresponding to the truth value of p for this row.

The negation of a proposition can also be considered the result of the operation of the **negation operator** on a proposition. The negation operator constructs a new proposition from a single existing proposition. We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called **connectives**.

DEFINITION 2

Let p and q be propositions. The *conjunction* of p and q , denoted by $p \wedge q$, is the proposition “ p and q .” The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Table 2 displays the truth table of $p \wedge q$. This table has a row for each of the four possible combinations of truth values of p and q . The four rows correspond to the pairs of truth values TT, TF, FT, and FF, where the first truth value in the pair is the truth value of p and the second truth value is the truth value of q .

Note that in logic the word “but” sometimes is used instead of “and” in a conjunction. For example, the statement “The sun is shining, but it is raining” is another way of saying “The sun is shining and it is raining.” (In natural language, there is a subtle difference in meaning between “and” and “but”; we will not be concerned with this nuance here.)

EXAMPLE 5

Find the conjunction of the propositions p and q where p is the proposition “Rebecca’s PC has more than 16 GB free hard disk space” and q is the proposition “The processor in Rebecca’s PC runs faster than 1 GHz.”

Solution: The conjunction of these propositions, $p \wedge q$, is the proposition “Rebecca’s PC has more than 16 GB free hard disk space, and the processor in Rebecca’s PC runs faster than 1 GHz.” This conjunction can be expressed more simply as “Rebecca’s PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz.” For this conjunction to be true, both conditions given must be true. It is false, when one or both of these conditions are false. \blacktriangleleft

DEFINITION 3

Let p and q be propositions. The *disjunction* of p and q , denoted by $p \vee q$, is the proposition “ p or q .” The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

Table 3 displays the truth table for $p \vee q$.

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

The use of the connective *or* in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, as an **inclusive or**. A disjunction is true when at least one of the two propositions is true. For instance, the inclusive or is being used in the statement

“Students who have taken calculus or computer science can take this class.”

Here, we mean that students who have taken both calculus and computer science can take the class, as well as the students who have taken only one of the two subjects. On the other hand, we are using the **exclusive or** when we say

“Students who have taken calculus or computer science, but not both, can enroll in this class.”

Here, we mean that students who have taken both calculus and a computer science course cannot take the class. Only those who have taken exactly one of the two courses can take the class.

Similarly, when a menu at a restaurant states, “Soup or salad comes with an entrée,” the restaurant almost always means that customers can have either soup or salad, but not both. Hence, this is an exclusive, rather than an inclusive, or.

EXAMPLE 6 What is the disjunction of the propositions p and q where p and q are the same propositions as in Example 5?

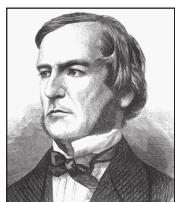


Solution: The disjunction of p and q , $p \vee q$, is the proposition

“Rebecca’s PC has at least 16 GB free hard disk space, or the processor in Rebecca’s PC runs faster than 1 GHz.”

This proposition is true when Rebecca’s PC has at least 16 GB free hard disk space, when the PC’s processor runs faster than 1 GHz, and when both conditions are true. It is false when both of these conditions are false, that is, when Rebecca’s PC has less than 16 GB free hard disk space and the processor in her PC runs at 1 GHz or slower.

As was previously remarked, the use of the connective *or* in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, in an inclusive way. Thus, a disjunction is true when at least one of the two propositions in it is true. Sometimes, we use *or* in an exclusive sense. When the exclusive or is used to connect the propositions p and q , the proposition “ p or q (but not both)” is obtained. This proposition is true when p is true and q is false, and when p is false and q is true. It is false when both p and q are false and when both are true.



GEORGE BOOLE (1815–1864) George Boole, the son of a cobbler, was born in Lincoln, England, in November 1815. Because of his family’s difficult financial situation, Boole struggled to educate himself while supporting his family. Nevertheless, he became one of the most important mathematicians of the 1800s. Although he considered a career as a clergyman, he decided instead to go into teaching, and soon afterward opened a school of his own. In his preparation for teaching mathematics, Boole—unsatisfied with textbooks of his day—decided to read the works of the great mathematicians. While reading papers of the great French mathematician Lagrange, Boole made discoveries in the calculus of variations, the branch of analysis dealing with finding curves and surfaces by optimizing certain parameters.

In 1848 Boole published *The Mathematical Analysis of Logic*, the first of his contributions to symbolic logic. In 1849 he was appointed professor of mathematics at Queen’s College in Cork, Ireland. In 1854 he published *The Laws of Thought*, his most famous work. In this book, Boole introduced what is now called *Boolean algebra* in his honor. Boole wrote textbooks on differential equations and on difference equations that were used in Great Britain until the end of the nineteenth century. Boole married in 1855; his wife was the niece of the professor of Greek at Queen’s College. In 1864 Boole died from pneumonia, which he contracted as a result of keeping a lecture engagement even though he was soaking wet from a rainstorm.

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

TABLE 5 The Truth Table for the Conditional Statement $p \rightarrow q$.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

DEFINITION 4

Let p and q be propositions. The *exclusive or* of p and q , denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

The truth table for the exclusive or of two propositions is displayed in Table 4.

Conditional Statements

We will discuss several other important ways in which propositions can be combined.

DEFINITION 5

Let p and q be propositions. The *conditional statement* $p \rightarrow q$ is the proposition “if p , then q .” The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).



The statement $p \rightarrow q$ is called a conditional statement because $p \rightarrow q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

The truth table for the conditional statement $p \rightarrow q$ is shown in Table 5. Note that the statement $p \rightarrow q$ is true when both p and q are true and when p is false (no matter what truth value q has).

Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express $p \rightarrow q$. You will encounter most if not all of the following ways to express this conditional statement:

“if p , then q ”	“ p implies q ”
“if p , q ”	“ p only if q ”
“ p is sufficient for q ”	“a sufficient condition for q is p ”
“ q if p ”	“ q whenever p ”
“ q when p ”	“ q is necessary for p ”
“a necessary condition for p is q ”	“ q follows from p ”
“ q unless $\neg p$ ”	

A useful way to understand the truth value of a conditional statement is to think of an obligation or a contract. For example, the pledge many politicians make when running for office is

“If I am elected, then I will lower taxes.”

If the politician is elected, voters would expect this politician to lower taxes. Furthermore, if the politician is not elected, then voters will not have any expectation that this person will lower taxes, although the person may have sufficient influence to cause those in power to lower taxes. It is only when the politician is elected but does not lower taxes that voters can say that the politician has broken the campaign pledge. This last scenario corresponds to the case when p is true but q is false in $p \rightarrow q$.

Similarly, consider a statement that a professor might make:

“If you get 100% on the final, then you will get an A.”

If you manage to get a 100% on the final, then you would expect to receive an A. If you do not get 100% you may or may not receive an A depending on other factors. However, if you do get 100%, but the professor does not give you an A, you will feel cheated.

Of the various ways to express the conditional statement $p \rightarrow q$, the two that seem to cause the most confusion are “ p only if q ” and “ q unless $\neg p$.” Consequently, we will provide some guidance for clearing up this confusion.

To remember that “ p only if q ” expresses the same thing as “if p , then q ,” note that “ p only if q ” says that p cannot be true when q is not true. That is, the statement is false if p is true, but q is false. When p is false, q may be either true or false, because the statement says nothing about the truth value of q . Be careful not to use “ q only if p ” to express $p \rightarrow q$ because this is incorrect. To see this, note that the true values of “ q only if p ” and $p \rightarrow q$ are different when p and q have different truth values.

To remember that “ q unless $\neg p$ ” expresses the same conditional statement as “if p , then q ,” note that “ q unless $\neg p$ ” means that if $\neg p$ is false, then q must be true. That is, the statement “ q unless $\neg p$ ” is false when p is true but q is false, but it is true otherwise. Consequently, “ q unless $\neg p$ ” and $p \rightarrow q$ always have the same truth value.

We illustrate the translation between conditional statements and English statements in Example 7.

EXAMPLE 7

Let p be the statement “Maria learns discrete mathematics” and q the statement “Maria will find a good job.” Express the statement $p \rightarrow q$ as a statement in English.



Solution: From the definition of conditional statements, we see that when p is the statement “Maria learns discrete mathematics” and q is the statement “Maria will find a good job,” $p \rightarrow q$ represents the statement

“If Maria learns discrete mathematics, then she will find a good job.”

There are many other ways to express this conditional statement in English. Among the most natural of these are:

“Maria will find a good job when she learns discrete mathematics.”

“For Maria to get a good job, it is sufficient for her to learn discrete mathematics.”

and

“Maria will find a good job unless she does not learn discrete mathematics.”

Note that the way we have defined conditional statements is more general than the meaning attached to such statements in the English language. For instance, the conditional statement in Example 7 and the statement

“If it is sunny, then we will go to the beach.”

are statements used in normal language where there is a relationship between the hypothesis and the conclusion. Further, the first of these statements is true unless Maria learns discrete mathematics, but she does not get a good job, and the second is true unless it is indeed sunny, but we do not go to the beach. On the other hand, the statement

“If Juan has a smartphone, then $2 + 3 = 5$ ”

is true from the definition of a conditional statement, because its conclusion is true. (The truth value of the hypothesis does not matter then.) The conditional statement

“If Juan has a smartphone, then $2 + 3 = 6$ ”

is true if Juan does not have a smartphone, even though $2 + 3 = 6$ is false. We would not use these last two conditional statements in natural language (except perhaps in sarcasm), because there is no relationship between the hypothesis and the conclusion in either statement. In mathematical reasoning, we consider conditional statements of a more general sort than we use in English. The mathematical concept of a conditional statement is independent of a cause-and-effect relationship between hypothesis and conclusion. Our definition of a conditional statement specifies its truth values; it is not based on English usage. Propositional language is an artificial language; we only parallel English usage to make it easy to use and remember.

The if-then construction used in many programming languages is different from that used in logic. Most programming languages contain statements such as **if** p **then** S , where p is a proposition and S is a program segment (one or more statements to be executed). When execution of a program encounters such a statement, S is executed if p is true, but S is not executed if p is false, as illustrated in Example 8.

EXAMPLE 8 What is the value of the variable x after the statement

if $2 + 2 = 4$ **then** $x := x + 1$

if $x = 0$ before this statement is encountered? (The symbol $:=$ stands for assignment. The statement $x := x + 1$ means the assignment of the value of $x + 1$ to x .)

Solution: Because $2 + 2 = 4$ is true, the assignment statement $x := x + 1$ is executed. Hence, x has the value $0 + 1 = 1$ after this statement is encountered. 

CONVERSE, CONTRAPOSITIVE, AND INVERSE We can form some new conditional statements starting with a conditional statement $p \rightarrow q$. In particular, there are three related conditional statements that occur so often that they have special names. The proposition $q \rightarrow p$ is called the **converse** of $p \rightarrow q$. The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$. The proposition $\neg p \rightarrow \neg q$ is called the **inverse** of $p \rightarrow q$. We will see that of these three conditional statements formed from $p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.

We first show that the contrapositive, $\neg q \rightarrow \neg p$, of a conditional statement $p \rightarrow q$ always has the same truth value as $p \rightarrow q$. To see this, note that the contrapositive is false only when $\neg p$ is false and $\neg q$ is true, that is, only when p is true and q is false. We now show that neither the converse, $q \rightarrow p$, nor the inverse, $\neg p \rightarrow \neg q$, has the same truth value as $p \rightarrow q$ for all possible truth values of p and q . Note that when p is true and q is false, the original conditional statement is false, but the converse and the inverse are both true.

When two compound propositions always have the same truth value we call them **equivalent**, so that a conditional statement and its contrapositive are equivalent. The converse and the inverse of a conditional statement are also equivalent, as the reader can verify, but neither is equivalent to the original conditional statement. (We will study equivalent propositions in Section 1.3.) Take note that one of the most common logical errors is to assume that the converse or the inverse of a conditional statement is equivalent to this conditional statement.

We illustrate the use of conditional statements in Example 9.

Remember that the contrapositive, but neither the converse or inverse, of a conditional statement is equivalent to it.

EXAMPLE 9 What are the contrapositive, the converse, and the inverse of the conditional statement
“The home team wins whenever it is raining?”



Solution: Because “ q whenever p ” is one of the ways to express the conditional statement $p \rightarrow q$, the original statement can be rewritten as

“If it is raining, then the home team wins.”

Consequently, the contrapositive of this conditional statement is

“If the home team does not win, then it is not raining.”

The converse is

“If the home team wins, then it is raining.”

The inverse is

“If it is not raining, then the home team does not win.”

Only the contrapositive is equivalent to the original statement.

BICONDITIONALS We now introduce another way to combine propositions that expresses that two propositions have the same truth value.

DEFINITION 6

Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition “ p if and only if q .” The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

The truth table for $p \leftrightarrow q$ is shown in Table 6. Note that the statement $p \leftrightarrow q$ is true when both the conditional statements $p \rightarrow q$ and $q \rightarrow p$ are true and is false otherwise. That is why we use the words “if and only if” to express this logical connective and why it is symbolically written by combining the symbols \rightarrow and \leftarrow . There are some other common ways to express $p \leftrightarrow q$:

- “ p is necessary and sufficient for q ”
- “if p then q , and conversely”
- “ p iff q .”

The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation “iff” for “if and only if.” Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$.

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

EXAMPLE 10 Let p be the statement “You can take the flight,” and let q be the statement “You buy a ticket.” Then $p \leftrightarrow q$ is the statement

“You can take the flight if and only if you buy a ticket.”



This statement is true if p and q are either both true or both false, that is, if you buy a ticket and can take the flight or if you do not buy a ticket and you cannot take the flight. It is false when p and q have opposite truth values, that is, when you do not buy a ticket, but you can take the flight (such as when you get a free trip) and when you buy a ticket but you cannot take the flight (such as when the airline bumps you).

IMPLICIT USE OF BICONDITIONALS You should be aware that biconditionals are not always explicit in natural language. In particular, the “if and only if” construction used in biconditionals is rarely used in common language. Instead, biconditionals are often expressed using an “if, then” or an “only if” construction. The other part of the “if and only if” is implicit. That is, the converse is implied, but not stated. For example, consider the statement in English “If you finish your meal, then you can have dessert.” What is really meant is “You can have dessert if and only if you finish your meal.” This last statement is logically equivalent to the two statements “If you finish your meal, then you can have dessert” and “You can have dessert only if you finish your meal.” Because of this imprecision in natural language, we need to make an assumption whether a conditional statement in natural language implicitly includes its converse. Because precision is essential in mathematics and in logic, we will always distinguish between the conditional statement $p \rightarrow q$ and the biconditional statement $p \leftrightarrow q$.

Truth Tables of Compound Propositions



We have now introduced four important logical connectives—conjunctions, disjunctions, conditional statements, and biconditional statements—as well as negations. We can use these connectives to build up complicated compound propositions involving any number of propositional variables. We can use truth tables to determine the truth values of these compound propositions, as Example 11 illustrates. We use a separate column to find the truth value of each compound expression that occurs in the compound proposition as it is built up. The truth values of the compound proposition for each combination of truth values of the propositional variables in it is found in the final column of the table.

EXAMPLE 11 Construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

Solution: Because this truth table involves two propositional variables p and q , there are four rows in this truth table, one for each of the pairs of truth values TT, TF, FT, and FF. The first two columns are used for the truth values of p and q , respectively. In the third column we find the truth value of $\neg q$, needed to find the truth value of $p \vee \neg q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \rightarrow (p \wedge q)$ is found in the last column. The resulting truth table is shown in Table 7.

TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Precedence of Logical Operators

TABLE 8
Precedence of
Logical Operators.

Operator	Precedence
\neg	1
\wedge	2
\vee	3
\rightarrow	4
\leftrightarrow	5

We can construct compound propositions using the negation operator and the logical operators defined so far. We will generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied. For instance, $(p \vee q) \wedge (\neg r)$ is the conjunction of $p \vee q$ and $\neg r$. However, to reduce the number of parentheses, we specify that the negation operator is applied before all other logical operators. This means that $\neg p \wedge q$ is the conjunction of $\neg p$ and q , namely, $(\neg p) \wedge q$, not the negation of the conjunction of p and q , namely $\neg(p \wedge q)$.

Another general rule of precedence is that the conjunction operator takes precedence over the disjunction operator, so that $p \wedge q \vee r$ means $(p \wedge q) \vee r$ rather than $p \wedge (q \vee r)$. Because this rule may be difficult to remember, we will continue to use parentheses so that the order of the disjunction and conjunction operators is clear.

Finally, it is an accepted rule that the conditional and biconditional operators \rightarrow and \leftrightarrow have lower precedence than the conjunction and disjunction operators, \wedge and \vee . Consequently, $p \vee q \rightarrow r$ is the same as $(p \vee q) \rightarrow r$. We will use parentheses when the order of the conditional operator and biconditional operator is at issue, although the conditional operator has precedence over the biconditional operator. Table 8 displays the precedence levels of the logical operators, \neg , \wedge , \vee , \rightarrow , and \leftrightarrow .

Logic and Bit Operations

Computers represent information using bits. A **bit** is a symbol with two possible values, namely, 0 (zero) and 1 (one). This meaning of the word bit comes from *binary digit*, because zeros and ones are the digits used in binary representations of numbers. The well-known statistician John Tukey introduced this terminology in 1946. A bit can be used to represent a truth value, because there are two truth values, namely, *true* and *false*. As is customarily done, we will use a 1 bit to represent true and a 0 bit to represent false. That is, 1 represents T (true), 0 represents F (false). A variable is called a **Boolean variable** if its value is either true or false. Consequently, a Boolean variable can be represented using a bit.

Computer **bit operations** correspond to the logical connectives. By replacing true by a one and false by a zero in the truth tables for the operators \wedge , \vee , and \oplus , the tables shown in Table 9 for the corresponding bit operations are obtained. We will also use the notation *OR*, *AND*, and *XOR* for the operators \vee , \wedge , and \oplus , as is done in various programming languages.

Truth Value	Bit
T	1
F	0



JOHN WILDER TUKEY (1915–2000) Tukey, born in New Bedford, Massachusetts, was an only child. His parents, both teachers, decided home schooling would best develop his potential. His formal education began at Brown University, where he studied mathematics and chemistry. He received a master's degree in chemistry from Brown and continued his studies at Princeton University, changing his field of study from chemistry to mathematics. He received his Ph.D. from Princeton in 1939 for work in topology, when he was appointed an instructor in mathematics at Princeton. With the start of World War II, he joined the Fire Control Research Office, where he began working in statistics. Tukey found statistical research to his liking and impressed several leading statisticians with his skills. In 1945, at the conclusion of the war, Tukey returned to the mathematics department at Princeton as a professor of statistics, and he also took a position at AT&T Bell Laboratories. Tukey founded

the Statistics Department at Princeton in 1966 and was its first chairman. Tukey made significant contributions to many areas of statistics, including the analysis of variance, the estimation of spectra of time series, inferences about the values of a set of parameters from a single experiment, and the philosophy of statistics. However, he is best known for his invention, with J. W. Cooley, of the fast Fourier transform. In addition to his contributions to statistics, Tukey was noted as a skilled wordsmith; he is credited with coining the terms *bit* and *software*.

Tukey contributed his insight and expertise by serving on the President's Science Advisory Committee. He chaired several important committees dealing with the environment, education, and chemicals and health. He also served on committees working on nuclear disarmament. Tukey received many awards, including the National Medal of Science.

HISTORICAL NOTE There were several other suggested words for a binary digit, including *binit* and *bigit*, that never were widely accepted. The adoption of the word *bit* may be due to its meaning as a common English word. For an account of Tukey's coining of the word *bit*, see the April 1984 issue of *Annals of the History of Computing*.

TABLE 9 Table for the Bit Operators *OR*, *AND*, and *XOR*.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Information is often represented using bit strings, which are lists of zeros and ones. When this is done, operations on the bit strings can be used to manipulate this information.

DEFINITION 7

A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

EXAMPLE 12 101010011 is a bit string of length nine.

We can extend bit operations to bit strings. We define the **bitwise OR**, **bitwise AND**, and **bitwise XOR** of two strings of the same length to be the strings that have as their bits the *OR*, *AND*, and *XOR* of the corresponding bits in the two strings, respectively. We use the symbols \vee , \wedge , and \oplus to represent the bitwise *OR*, bitwise *AND*, and bitwise *XOR* operations, respectively. We illustrate bitwise operations on bit strings with Example 13.

EXAMPLE 13 Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit strings 0110110110 and 1100011101. (Here, and throughout this book, bit strings will be split into blocks of four bits to make them easier to read.)

Solution: The bitwise *OR*, bitwise *AND*, and bitwise *XOR* of these strings are obtained by taking the *OR*, *AND*, and *XOR* of the corresponding bits, respectively. This gives us

$$\begin{array}{r} 01\ 1011\ 0110 \\ 11\ 0001\ 1101 \\ \hline 11\ 1011\ 1111 \end{array} \quad \begin{array}{l} \text{bitwise } OR \\ \text{bitwise } AND \\ \text{bitwise } XOR \end{array}$$

Exercises

- Which of these sentences are propositions? What are the truth values of those that are propositions?
 - Boston is the capital of Massachusetts.
 - Miami is the capital of Florida.
 - $2 + 3 = 5$.
 - $5 + 7 = 10$.
 - $x + 2 = 11$.
 - Answer this question.
- Which of these are propositions? What are the truth values of those that are propositions?
 - Do not pass go.
 - What time is it?
 - There are no black flies in Maine.
4. Which of these sentences are propositions? What are the truth values of those that are propositions?
 - $4 + x = 5$.
 - The moon is made of green cheese.
 - $2^n \geq 100$.
- What is the negation of each of these propositions?
 - Mei has an MP3 player.
 - There is no pollution in New Jersey.
 - $2 + 1 = 3$.
 - The summer in Maine is hot and sunny.
- What is the negation of each of these propositions?
 - Jennifer and Teja are friends.
 - There are 13 items in a baker's dozen.
 - Abby sent more than 100 text messages every day.
 - 121 is a perfect square.

5. What is the negation of each of these propositions?
- Steve has more than 100 GB free disk space on his laptop.
 - Zach blocks e-mails and texts from Jennifer.
 - $7 \cdot 11 \cdot 13 = 999$.
 - Diane rode her bicycle 100 miles on Sunday.
6. Suppose that Smartphone A has 256 MB RAM and 32 GB ROM, and the resolution of its camera is 8 MP; Smartphone B has 288 MB RAM and 64 GB ROM, and the resolution of its camera is 4 MP; and Smartphone C has 128 MB RAM and 32 GB ROM, and the resolution of its camera is 5 MP. Determine the truth value of each of these propositions.
- Smartphone B has the most RAM of these three smartphones.
 - Smartphone C has more ROM or a higher resolution camera than Smartphone B.
 - Smartphone B has more RAM, more ROM, and a higher resolution camera than Smartphone A.
 - If Smartphone B has more RAM and more ROM than Smartphone C, then it also has a higher resolution camera.
 - Smartphone A has more RAM than Smartphone B if and only if Smartphone B has more RAM than Smartphone A.
7. Suppose that during the most recent fiscal year, the annual revenue of Acme Computer was 138 billion dollars and its net profit was 8 billion dollars, the annual revenue of Nadir Software was 87 billion dollars and its net profit was 5 billion dollars, and the annual revenue of Quixote Media was 111 billion dollars and its net profit was 13 billion dollars. Determine the truth value of each of these propositions for the most recent fiscal year.
- Quixote Media had the largest annual revenue.
 - Nadir Software had the lowest net profit and Acme Computer had the largest annual revenue.
 - Acme Computer had the largest net profit or Quixote Media had the largest net profit.
 - If Quixote Media had the smallest net profit, then Acme Computer had the largest annual revenue.
 - Nadir Software had the smallest net profit if and only if Acme Computer had the largest annual revenue.
8. Let p and q be the propositions
- p : I bought a lottery ticket this week.
 q : I won the million dollar jackpot.
- Express each of these propositions as an English sentence.
- $\neg p$
 - $p \vee q$
 - $p \rightarrow q$
 - $p \wedge q$
 - $p \leftrightarrow q$
 - $\neg p \rightarrow \neg q$
 - $\neg p \wedge \neg q$
 - $\neg p \vee (p \wedge q)$
9. Let p and q be the propositions “Swimming at the New Jersey shore is allowed” and “Sharks have been spotted near the shore,” respectively. Express each of these compound propositions as an English sentence.
- $\neg q$
 - $p \wedge q$
 - $\neg p \vee q$
 - $p \rightarrow \neg q$
 - $\neg q \rightarrow p$
 - $\neg p \rightarrow \neg q$
 - $p \leftrightarrow \neg q$
 - $\neg p \wedge (p \vee \neg q)$

10. Let p and q be the propositions “The election is decided” and “The votes have been counted,” respectively. Express each of these compound propositions as an English sentence.

- $\neg p$
- $p \vee q$
- $\neg p \wedge q$
- $q \rightarrow p$
- $\neg q \rightarrow \neg p$
- $\neg p \rightarrow \neg q$
- $p \leftrightarrow q$
- $\neg q \vee (\neg p \wedge q)$

11. Let p and q be the propositions

p : It is below freezing.
 q : It is snowing.

Write these propositions using p and q and logical connectives (including negations).

- It is below freezing and snowing.
- It is below freezing but not snowing.
- It is not below freezing and it is not snowing.
- It is either snowing or below freezing (or both).
- If it is below freezing, it is also snowing.
- Either it is below freezing or it is snowing, but it is not snowing if it is below freezing.
- That it is below freezing is necessary and sufficient for it to be snowing.

12. Let p , q , and r be the propositions

p : You have the flu.
 q : You miss the final examination.
 r : You pass the course.

Express each of these propositions as an English sentence.

- $p \rightarrow q$
- $\neg q \leftrightarrow r$
- $q \rightarrow \neg r$
- $p \vee q \vee r$
- $(p \rightarrow \neg r) \vee (q \rightarrow \neg r)$
- $(p \wedge q) \vee (\neg q \wedge r)$

13. Let p and q be the propositions

p : You drive over 65 miles per hour.
 q : You get a speeding ticket.

Write these propositions using p and q and logical connectives (including negations).

- You do not drive over 65 miles per hour.
- You drive over 65 miles per hour, but you do not get a speeding ticket.
- You will get a speeding ticket if you drive over 65 miles per hour.
- If you do not drive over 65 miles per hour, then you will not get a speeding ticket.
- Driving over 65 miles per hour is sufficient for getting a speeding ticket.
- You get a speeding ticket, but you do not drive over 65 miles per hour.
- Whenever you get a speeding ticket, you are driving over 65 miles per hour.

14. Let p , q , and r be the propositions

p : You get an A on the final exam.
 q : You do every exercise in this book.
 r : You get an A in this class.

Write these propositions using p , q , and r and logical connectives (including negations).

- a) You get an A in this class, but you do not do every exercise in this book.
- b) You get an A on the final, you do every exercise in this book, and you get an A in this class.
- c) To get an A in this class, it is necessary for you to get an A on the final.
- d) You get an A on the final, but you don't do every exercise in this book; nevertheless, you get an A in this class.
- e) Getting an A on the final and doing every exercise in this book is sufficient for getting an A in this class.
- f) You will get an A in this class if and only if you either do every exercise in this book or you get an A on the final.
- 15.** Let p , q , and r be the propositions
- p : Grizzly bears have been seen in the area.
 - q : Hiking is safe on the trail.
 - r : Berries are ripe along the trail.
- Write these propositions using p , q , and r and logical connectives (including negations).
- a) Berries are ripe along the trail, but grizzly bears have not been seen in the area.
- b) Grizzly bears have not been seen in the area and hiking on the trail is safe, but berries are ripe along the trail.
- c) If berries are ripe along the trail, hiking is safe if and only if grizzly bears have not been seen in the area.
- d) It is not safe to hike on the trail, but grizzly bears have not been seen in the area and the berries along the trail are ripe.
- e) For hiking on the trail to be safe, it is necessary but not sufficient that berries not be ripe along the trail and for grizzly bears not to have been seen in the area.
- f) Hiking is not safe on the trail whenever grizzly bears have been seen in the area and berries are ripe along the trail.
- 16.** Determine whether these biconditionals are true or false.
- a) $2 + 2 = 4$ if and only if $1 + 1 = 2$.
- b) $1 + 1 = 2$ if and only if $2 + 3 = 4$.
- c) $1 + 1 = 3$ if and only if monkeys can fly.
- d) $0 > 1$ if and only if $2 > 1$.
- 17.** Determine whether each of these conditional statements is true or false.
- a) If $1 + 1 = 2$, then $2 + 2 = 5$.
- b) If $1 + 1 = 3$, then $2 + 2 = 4$.
- c) If $1 + 1 = 3$, then $2 + 2 = 5$.
- d) If monkeys can fly, then $1 + 1 = 3$.
- 18.** Determine whether each of these conditional statements is true or false.
- a) If $1 + 1 = 3$, then unicorns exist.
- b) If $1 + 1 = 3$, then dogs can fly.
- c) If $1 + 1 = 2$, then dogs can fly.
- d) If $2 + 2 = 4$, then $1 + 2 = 3$.
- 19.** For each of these sentences, determine whether an inclusive or, or an exclusive or, is intended. Explain your answer.
- a) Coffee or tea comes with dinner.
- b) A password must have at least three digits or be at least eight characters long.
- c) The prerequisite for the course is a course in number theory or a course in cryptography.
- d) You can pay using U.S. dollars or euros.
- 20.** For each of these sentences, determine whether an inclusive or, or an exclusive or, is intended. Explain your answer.
- a) Experience with C++ or Java is required.
- b) Lunch includes soup or salad.
- c) To enter the country you need a passport or a voter registration card.
- d) Publish or perish.
- 21.** For each of these sentences, state what the sentence means if the logical connective or is an inclusive or (that is, a disjunction) versus an exclusive or. Which of these meanings of or do you think is intended?
- a) To take discrete mathematics, you must have taken calculus or a course in computer science.
- b) When you buy a new car from Acme Motor Company, you get \$2000 back in cash or a 2% car loan.
- c) Dinner for two includes two items from column A or three items from column B.
- d) School is closed if more than 2 feet of snow falls or if the wind chill is below -100 .
- 22.** Write each of these statements in the form "if p , then q " in English. [Hint: Refer to the list of common ways to express conditional statements provided in this section.]
- a) It is necessary to wash the boss's car to get promoted.
- b) Winds from the south imply a spring thaw.
- c) A sufficient condition for the warranty to be good is that you bought the computer less than a year ago.
- d) Willy gets caught whenever he cheats.
- e) You can access the website only if you pay a subscription fee.
- f) Getting elected follows from knowing the right people.
- g) Carol gets seasick whenever she is on a boat.
- 23.** Write each of these statements in the form "if p , then q " in English. [Hint: Refer to the list of common ways to express conditional statements.]
- a) It snows whenever the wind blows from the northeast.
- b) The apple trees will bloom if it stays warm for a week.
- c) That the Pistons win the championship implies that they beat the Lakers.
- d) It is necessary to walk 8 miles to get to the top of Long's Peak.
- e) To get tenure as a professor, it is sufficient to be world-famous.
- f) If you drive more than 400 miles, you will need to buy gasoline.
- g) Your guarantee is good only if you bought your CD player less than 90 days ago.
- h) Jan will go swimming unless the water is too cold.

- 24.** Write each of these statements in the form “if p , then q ” in English. [Hint: Refer to the list of common ways to express conditional statements provided in this section.]
- I will remember to send you the address only if you send me an e-mail message.
 - To be a citizen of this country, it is sufficient that you were born in the United States.
 - If you keep your textbook, it will be a useful reference in your future courses.
 - The Red Wings will win the Stanley Cup if their goalie plays well.
 - That you get the job implies that you had the best credentials.
 - The beach erodes whenever there is a storm.
 - It is necessary to have a valid password to log on to the server.
 - You will reach the summit unless you begin your climb too late.
- 25.** Write each of these propositions in the form “ p if and only if q ” in English.
- If it is hot outside you buy an ice cream cone, and if you buy an ice cream cone it is hot outside.
 - For you to win the contest it is necessary and sufficient that you have the only winning ticket.
 - You get promoted only if you have connections, and you have connections only if you get promoted.
 - If you watch television your mind will decay, and conversely.
 - The trains run late on exactly those days when I take it.
- 26.** Write each of these propositions in the form “ p if and only if q ” in English.
- For you to get an A in this course, it is necessary and sufficient that you learn how to solve discrete mathematics problems.
 - If you read the newspaper every day, you will be informed, and conversely.
 - It rains if it is a weekend day, and it is a weekend day if it rains.
 - You can see the wizard only if the wizard is not in, and the wizard is not in only if you can see him.
- 27.** State the converse, contrapositive, and inverse of each of these conditional statements.
- If it snows today, I will ski tomorrow.
 - I come to class whenever there is going to be a quiz.
 - A positive integer is a prime only if it has no divisors other than 1 and itself.
- 28.** State the converse, contrapositive, and inverse of each of these conditional statements.
- If it snows tonight, then I will stay at home.
 - I go to the beach whenever it is a sunny summer day.
 - When I stay up late, it is necessary that I sleep until noon.
- 29.** How many rows appear in a truth table for each of these compound propositions?
- $p \rightarrow \neg p$
 - $(p \vee \neg r) \wedge (q \vee \neg s)$
 - $q \vee p \vee \neg s \vee \neg r \vee \neg t \vee u$
 - $(p \wedge r \wedge t) \leftrightarrow (q \wedge t)$
- 30.** How many rows appear in a truth table for each of these compound propositions?
- $(q \rightarrow \neg p) \vee (\neg p \rightarrow \neg q)$
 - $(p \vee \neg t) \wedge (p \vee \neg s)$
 - $(p \rightarrow r) \vee (\neg s \rightarrow \neg t) \vee (\neg u \rightarrow v)$
 - $(p \wedge r \wedge s) \vee (q \wedge t) \vee (r \wedge \neg t)$
- 31.** Construct a truth table for each of these compound propositions.
- $p \wedge \neg p$
 - $p \vee \neg p$
 - $(p \vee \neg q) \rightarrow q$
 - $(p \vee q) \rightarrow (p \wedge q)$
 - $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$
 - $(p \rightarrow q) \rightarrow (q \rightarrow p)$
- 32.** Construct a truth table for each of these compound propositions.
- $p \rightarrow \neg p$
 - $p \leftrightarrow \neg p$
 - $p \oplus (p \vee q)$
 - $(p \wedge q) \rightarrow (p \vee q)$
 - $(q \rightarrow \neg p) \leftrightarrow (p \leftrightarrow q)$
 - $(p \leftrightarrow q) \oplus (p \leftrightarrow \neg q)$
- 33.** Construct a truth table for each of these compound propositions.
- $(p \vee q) \rightarrow (p \oplus q)$
 - $(p \oplus q) \rightarrow (p \wedge q)$
 - $(p \vee q) \oplus (p \wedge q)$
 - $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg q)$
 - $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$
 - $(p \oplus q) \rightarrow (p \oplus \neg q)$
- 34.** Construct a truth table for each of these compound propositions.
- $p \oplus p$
 - $p \oplus \neg p$
 - $p \oplus \neg q$
 - $\neg p \oplus \neg q$
 - $(p \oplus q) \vee (p \oplus \neg q)$
 - $(p \oplus q) \wedge (p \oplus \neg q)$
- 35.** Construct a truth table for each of these compound propositions.
- $p \rightarrow \neg q$
 - $\neg p \leftrightarrow q$
 - $(p \rightarrow q) \vee (\neg p \rightarrow q)$
 - $(p \rightarrow q) \wedge (\neg p \rightarrow q)$
 - $(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$
 - $(\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$
- 36.** Construct a truth table for each of these compound propositions.
- $(p \vee q) \vee r$
 - $(p \vee q) \wedge r$
 - $(p \wedge q) \vee r$
 - $(p \wedge q) \wedge r$
 - $(p \vee q) \wedge \neg r$
 - $(p \wedge q) \vee \neg r$
- 37.** Construct a truth table for each of these compound propositions.
- $p \rightarrow (\neg q \vee r)$
 - $\neg p \rightarrow (q \rightarrow r)$
 - $(p \rightarrow q) \vee (\neg p \rightarrow r)$
 - $(p \rightarrow q) \wedge (\neg p \rightarrow r)$
 - $(p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$
 - $(\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$
- 38.** Construct a truth table for $((p \rightarrow q) \rightarrow r) \rightarrow s$.
- 39.** Construct a truth table for $(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow s)$.

- 40.** Explain, without using a truth table, why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when p , q , and r have the same truth value and it is false otherwise.
- 41.** Explain, without using a truth table, why $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p , q , and r is true and at least one is false, but is false when all three variables have the same truth value.
- 42.** What is the value of x after each of these statements is encountered in a computer program, if $x = 1$ before the statement is reached?
- if** $x + 2 = 3$ **then** $x := x + 1$
 - if** $(x + 1 = 3)$ **OR** $(2x + 2 = 3)$ **then** $x := x + 1$
 - if** $(2x + 3 = 5)$ **AND** $(3x + 4 = 7)$ **then** $x := x + 1$
 - if** $(x + 1 = 2)$ **XOR** $(x + 2 = 3)$ **then** $x := x + 1$
 - if** $x < 2$ **then** $x := x + 1$
- 43.** Find the bitwise **OR**, bitwise **AND**, and bitwise **XOR** of each of these pairs of bit strings.
- 101 1110, 010 0001
 - 1111 0000, 1010 1010
 - 00 0111 0001, 10 0100 1000
 - 11 1111 1111, 00 0000 0000
- 44.** Evaluate each of these expressions.
- $1 \ 1000 \wedge (0 \ 1011 \vee 1 \ 1011)$
 - $(0 \ 1111 \wedge 1 \ 0101) \vee 0 \ 1000$
 - $(0 \ 1010 \oplus 1 \ 1011) \oplus 0 \ 1000$
 - $(1 \ 1011 \vee 0 \ 1010) \wedge (1 \ 0001 \vee 1 \ 1011)$

Fuzzy logic is used in artificial intelligence. In fuzzy logic, a proposition has a truth value that is a number between 0 and 1, inclusive. A proposition with a truth value of 0 is false and one with a truth value of 1 is true. Truth values that are between 0 and 1 indicate varying degrees of truth. For instance, the truth value 0.8 can be assigned to the statement “Fred is happy,”

because Fred is happy most of the time, and the truth value 0.4 can be assigned to the statement “John is happy,” because John is happy slightly less than half the time. Use these truth values to solve Exercises 45–47.

- 45.** The truth value of the negation of a proposition in fuzzy logic is 1 minus the truth value of the proposition. What are the truth values of the statements “Fred is not happy” and “John is not happy?”
- 46.** The truth value of the conjunction of two propositions in fuzzy logic is the minimum of the truth values of the two propositions. What are the truth values of the statements “Fred and John are happy” and “Neither Fred nor John is happy?”
- 47.** The truth value of the disjunction of two propositions in fuzzy logic is the maximum of the truth values of the two propositions. What are the truth values of the statements “Fred is happy, or John is happy” and “Fred is not happy, or John is not happy?”
- ***48.** Is the assertion “This statement is false” a proposition?
- ***49.** The n th statement in a list of 100 statements is “Exactly n of the statements in this list are false.”
- What conclusions can you draw from these statements?
 - Answer part (a) if the n th statement is “At least n of the statements in this list are false.”
 - Answer part (b) assuming that the list contains 99 statements.
- 50.** An ancient Sicilian legend says that the barber in a remote town who can be reached only by traveling a dangerous mountain road shaves those people, and only those people, who do not shave themselves. Can there be such a barber?

1.2 Applications of Propositional Logic

Introduction

Logic has many important applications to mathematics, computer science, and numerous other disciplines. Statements in mathematics and the sciences and in natural language often are imprecise or ambiguous. To make such statements precise, they can be translated into the language of logic. For example, logic is used in the specification of software and hardware, because these specifications need to be precise before development begins. Furthermore, propositional logic and its rules can be used to design computer circuits, to construct computer programs, to verify the correctness of programs, and to build expert systems. Logic can be used to analyze and solve many familiar puzzles. Software systems based on the rules of logic have been developed for constructing some, but not all, types of proofs automatically. We will discuss some of these applications of propositional logic in this section and in later chapters.

Translating English Sentences

There are many reasons to translate English sentences into expressions involving propositional variables and logical connectives. In particular, English (and every other human language) is

often ambiguous. Translating sentences into compound statements (and other types of logical expressions, which we will introduce later in this chapter) removes the ambiguity. Note that this may involve making a set of reasonable assumptions based on the intended meaning of the sentence. Moreover, once we have translated sentences from English into logical expressions we can analyze these logical expressions to determine their truth values, we can manipulate them, and we can use rules of inference (which are discussed in Section 1.6) to reason about them.

To illustrate the process of translating an English sentence into a logical expression, consider Examples 1 and 2.

EXAMPLE 1 How can this English sentence be translated into a logical expression?

“You can access the Internet from campus only if you are a computer science major or you are not a freshman.”



Solution: There are many ways to translate this sentence into a logical expression. Although it is possible to represent the sentence by a single propositional variable, such as p , this would not be useful when analyzing its meaning or reasoning with it. Instead, we will use propositional variables to represent each sentence part and determine the appropriate logical connectives between them. In particular, we let a , c , and f represent “You can access the Internet from campus,” “You are a computer science major,” and “You are a freshman,” respectively. Noting that “only if” is one way a conditional statement can be expressed, this sentence can be represented as

$$a \rightarrow (c \vee \neg f).$$



EXAMPLE 2 How can this English sentence be translated into a logical expression?

“You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.”

Solution: Let q , r , and s represent “You can ride the roller coaster,” “You are under 4 feet tall,” and “You are older than 16 years old,” respectively. Then the sentence can be translated to

$$(r \wedge \neg s) \rightarrow \neg q.$$

Of course, there are other ways to represent the original sentence as a logical expression, but the one we have used should meet our needs.



System Specifications

Translating sentences in natural language (such as English) into logical expressions is an essential part of specifying both hardware and software systems. System and software engineers take requirements in natural language and produce precise and unambiguous specifications that can be used as the basis for system development. Example 3 shows how compound propositions can be used in this process.

EXAMPLE 3 Express the specification “The automated reply cannot be sent when the file system is full” using logical connectives.



Solution: One way to translate this is to let p denote “The automated reply can be sent” and q denote “The file system is full.” Then $\neg p$ represents “It is not the case that the automated

reply can be sent,” which can also be expressed as “The automated reply cannot be sent.” Consequently, our specification can be represented by the conditional statement $q \rightarrow \neg p$. 

System specifications should be **consistent**, that is, they should not contain conflicting requirements that could be used to derive a contradiction. When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

EXAMPLE 4 Determine whether these system specifications are consistent:

- “The diagnostic message is stored in the buffer or it is retransmitted.”
- “The diagnostic message is not stored in the buffer.”
- “If the diagnostic message is stored in the buffer, then it is retransmitted.”

Solution: To determine whether these specifications are consistent, we first express them using logical expressions. Let p denote “The diagnostic message is stored in the buffer” and let q denote “The diagnostic message is retransmitted.” The specifications can then be written as $p \vee q$, $\neg p$, and $p \rightarrow q$. An assignment of truth values that makes all three specifications true must have p false to make $\neg p$ true. Because we want $p \vee q$ to be true but p must be false, q must be true. Because $p \rightarrow q$ is true when p is false and q is true, we conclude that these specifications are consistent, because they are all true when p is false and q is true. We could come to the same conclusion by use of a truth table to examine the four possible assignments of truth values to p and q . 

EXAMPLE 5 Do the system specifications in Example 4 remain consistent if the specification “The diagnostic message is not retransmitted” is added?

Solution: By the reasoning in Example 4, the three specifications from that example are true only in the case when p is false and q is true. However, this new specification is $\neg q$, which is false when q is true. Consequently, these four specifications are inconsistent. 

Boolean Searches



Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages. Because these searches employ techniques from propositional logic, they are called **Boolean searches**.

In Boolean searches, the connective *AND* is used to match records that contain both of two search terms, the connective *OR* is used to match one or both of two search terms, and the connective *NOT* (sometimes written as *AND NOT*) is used to exclude a particular search term. Careful planning of how logical connectives are used is often required when Boolean searches are used to locate information of potential interest. Example 6 illustrates how Boolean searches are carried out.

EXAMPLE 6 **Web Page Searching** Most Web search engines support Boolean searching techniques, which usually can help find Web pages about particular subjects. For instance, using Boolean searching to find Web pages about universities in New Mexico, we can look for pages matching NEW AND MEXICO AND UNIVERSITIES. The results of this search will include those pages that contain the three words NEW, MEXICO, and UNIVERSITIES. This will include all of the pages of interest, together with others such as a page about new universities in Mexico. (Note that in Google, and many other search engines, the word “AND” is not needed, although it is understood, because all search terms are included by default. These search engines also support the use of quotation marks to search for specific phrases. So, it may be more effective to search for pages matching “New Mexico” AND UNIVERSITIES.)



Next, to find pages that deal with universities in New Mexico or Arizona, we can search for pages matching (NEW AND MEXICO OR ARIZONA) AND UNIVERSITIES. (*Note:* Here the *AND* operator takes precedence over the *OR* operator. Also, in Google, the terms used for this search would be NEW MEXICO OR ARIZONA.) The results of this search will include all pages that contain the word UNIVERSITIES and either both the words NEW and MEXICO or the word ARIZONA. Again, pages besides those of interest will be listed. Finally, to find Web pages that deal with universities in Mexico (and not New Mexico), we might first look for pages matching MEXICO AND UNIVERSITIES, but because the results of this search will include pages about universities in New Mexico, as well as universities in Mexico, it might be better to search for pages matching (MEXICO AND UNIVERSITIES) NOT NEW. The results of this search include pages that contain both the words MEXICO and UNIVERSITIES but do not contain the word NEW. (In Google, and many other search engines, the word “NOT” is replaced by the symbol “-”. In Google, the terms used for this last search would be MEXICO UNIVERSITIES -NEW.)

Logic Puzzles



Puzzles that can be solved using logical reasoning are known as **logic puzzles**. Solving logic puzzles is an excellent way to practice working with the rules of logic. Also, computer programs designed to carry out logical reasoning often use well-known logic puzzles to illustrate their capabilities. Many people enjoy solving logic puzzles, published in periodicals, books, and on the Web, as a recreational activity.

We will discuss two logic puzzles here. We begin with a puzzle originally posed by Raymond Smullyan, a master of logic puzzles, who has published more than a dozen books containing challenging puzzles that involve logical reasoning. In Section 1.3 we will also discuss the extremely popular logic puzzle Sudoku.

EXAMPLE 7



In [Sm78] Smullyan posed many puzzles about an island that has two kinds of inhabitants, knights, who always tell the truth, and their opposites, knaves, who always lie. You encounter two people *A* and *B*. What are *A* and *B* if *A* says “*B* is a knight” and *B* says “The two of us are opposite types?”

Solution: Let *p* and *q* be the statements that *A* is a knight and *B* is a knight, respectively, so that $\neg p$ and $\neg q$ are the statements that *A* is a knave and *B* is a knave, respectively.

We first consider the possibility that *A* is a knight; this is the statement that *p* is true. If *A* is a knight, then he is telling the truth when he says that *B* is a knight, so that *q* is true, and *A* and *B* are the same type. However, if *B* is a knight, then *B*’s statement that *A* and *B* are of opposite types, the statement $(p \wedge \neg q) \vee (\neg p \wedge q)$, would have to be true, which it is not, because *A* and *B* are both knights. Consequently, we can conclude that *A* is not a knight, that is, that *p* is false.

If *A* is a knave, then because everything a knave says is false, *A*’s statement that *B* is a knight, that is, that *q* is true, is a lie. This means that *q* is false and *B* is also a knave. Furthermore, if *B* is a knave, then *B*’s statement that *A* and *B* are opposite types is a lie, which is consistent with both *A* and *B* being knaves. We can conclude that both *A* and *B* are knaves.

We pose more of Smullyan’s puzzles about knights and knaves in Exercises 19–23. In Exercises 24–31 we introduce related puzzles where we have three types of people, knights and knaves as in this puzzle together with spies who can lie.

Next, we pose a puzzle known as the **muddy children puzzle** for the case of two children.

EXAMPLE 8 A father tells his two children, a boy and a girl, to play in their backyard without getting dirty. However, while playing, both children get mud on their foreheads. When the children stop playing, the father says “At least one of you has a muddy forehead,” and then asks the children to answer “Yes” or “No” to the question: “Do you know whether you have a muddy forehead?” The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead? Assume that both children are honest and that the children answer each question simultaneously.

Solution: Let s be the statement that the son has a muddy forehead and let d be the statement that the daughter has a muddy forehead. When the father says that at least one of the two children has a muddy forehead, he is stating that the disjunction $s \vee d$ is true. Both children will answer “No” the first time the question is asked because each sees mud on the other child’s forehead. That is, the son knows that d is true, but does not know whether s is true, and the daughter knows that s is true, but does not know whether d is true.

After the son has answered “No” to the first question, the daughter can determine that d must be true. This follows because when the first question is asked, the son knows that $s \vee d$ is true, but cannot determine whether s is true. Using this information, the daughter can conclude that d must be true, for if d were false, the son could have reasoned that because $s \vee d$ is true, then s must be true, and he would have answered “Yes” to the first question. The son can reason in a similar way to determine that s must be true. It follows that both children answer “Yes” the second time the question is asked. 

Logic Circuits

Propositional logic can be applied to the design of computer hardware. This was first observed in 1938 by Claude Shannon in his MIT master’s thesis. In Chapter 12 we will study this topic in depth. (See that chapter for a biography of Shannon.) We give a brief introduction to this application here.

A **logic circuit** (or **digital circuit**) receives input signals p_1, p_2, \dots, p_n , each a bit [either 0 (off) or 1 (on)], and produces output signals s_1, s_2, \dots, s_n , each a bit. In this section we will restrict our attention to logic circuits with a single output signal; in general, digital circuits may have multiple outputs.

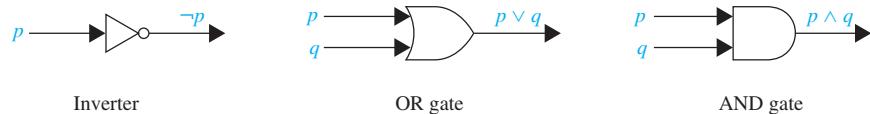
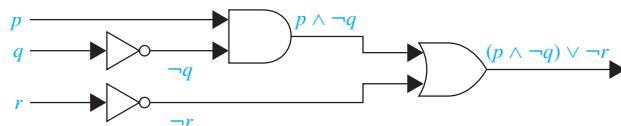
In Chapter 12 we design some useful circuits.



RAYMOND SMULLYAN (BORN 1919) Raymond Smullyan dropped out of high school. He wanted to study what he was really interested in and not standard high school material. After jumping from one university to the next, he earned an undergraduate degree in mathematics at the University of Chicago in 1955. He paid his college expenses by performing magic tricks at parties and clubs. He obtained a Ph.D. in logic in 1959 at Princeton, studying under Alonzo Church. After graduating from Princeton, he taught mathematics and logic at Dartmouth College, Princeton University, Yeshiva University, and the City University of New York. He joined the philosophy department at Indiana University in 1981 where he is now an emeritus professor.

Smullyan has written many books on recreational logic and mathematics, including *Satan, Cantor, and Infinity; What Is the Name of This Book?; The Lady or the Tiger?; Alice in Puzzland; To Mock a Mockingbird; Forever Undecided; and The Riddle of Scheherazade: Amazing Logic Puzzles, Ancient and Modern*. Because his logic puzzles are challenging, entertaining, and thought-provoking, he is considered to be a modern-day Lewis Carroll. Smullyan has also written several books about the application of deductive logic to chess, three collections of philosophical essays and aphorisms, and several advanced books on mathematical logic and set theory. He is particularly interested in self-reference and has worked on extending some of Gödel’s results that show that it is impossible to write a computer program that can solve all mathematical problems. He is also particularly interested in explaining ideas from mathematical logic to the public.

Smullyan is a talented musician and often plays piano with his wife, who is a concert-level pianist. Making telescopes is one of his hobbies. He is also interested in optics and stereo photography. He states “I’ve never had a conflict between teaching and research as some people do because when I’m teaching, I’m doing research.” Smullyan is the subject of a documentary short film entitled *This Film Needs No Title*.

**FIGURE 1** Basic logic gates.**FIGURE 2** A combinatorial circuit.

Complicated digital circuits can be constructed from three basic circuits, called **gates**, shown in Figure 1. The **inverter**, or **NOT gate**, takes an input bit p , and produces as output $\neg p$. The **OR gate** takes two input signals p and q , each a bit, and produces as output the signal $p \vee q$. Finally, the **AND gate** takes two input signals p and q , each a bit, and produces as output the signal $p \wedge q$. We use combinations of these three basic gates to build more complicated circuits, such as that shown in Figure 2.

Given a circuit built from the basic logic gates and the inputs to the circuit, we determine the output by tracing through the circuit, as Example 9 shows.

EXAMPLE 9 Determine the output for the combinatorial circuit in Figure 2.

Solution: In Figure 2 we display the output of each logic gate in the circuit. We see that the AND gate takes input of p and $\neg q$, the output of the inverter with input q , and produces $p \wedge \neg q$. Next, we note that the OR gate takes input $p \wedge \neg q$ and $\neg r$, the output of the inverter with input r , and produces the final output $(p \wedge \neg q) \vee \neg r$.

Suppose that we have a formula for the output of a digital circuit in terms of negations, disjunctions, and conjunctions. Then, we can systematically build a digital circuit with the desired output, as illustrated in Example 10.

EXAMPLE 10 Build a digital circuit that produces the output $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$ when given input bits p , q , and r .

Solution: To construct the desired circuit, we build separate circuits for $p \vee \neg r$ and for $\neg p \vee (q \vee \neg r)$ and combine them using an AND gate. To construct a circuit for $p \vee \neg r$, we use an inverter to produce $\neg r$ from the input r . Then, we use an OR gate to combine p and $\neg r$. To build a circuit for $\neg p \vee (q \vee \neg r)$, we first use an inverter to obtain $\neg p$. Then we use an OR gate with inputs q and $\neg r$ to obtain $q \vee \neg r$. Finally, we use another inverter and an OR gate to get $\neg p \vee (q \vee \neg r)$ from the inputs p and $q \vee \neg r$.

To complete the construction, we employ a final AND gate, with inputs $p \vee \neg r$ and $\neg p \vee (q \vee \neg r)$. The resulting circuit is displayed in Figure 3.

We will study logic circuits in great detail in Chapter 12 in the context of Boolean algebra, and with different notation.

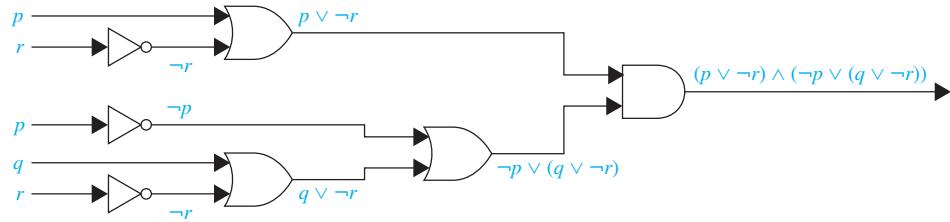


FIGURE 3 The circuit for $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$.

Exercises

In Exercises 1–6, translate the given statement into propositional logic using the propositions provided.

1. You cannot edit a protected Wikipedia entry unless you are an administrator. Express your answer in terms of e : “You can edit a protected Wikipedia entry” and a : “You are an administrator.”
2. You can see the movie only if you are over 18 years old or you have the permission of a parent. Express your answer in terms of m : “You can see the movie,” e : “You are over 18 years old,” and p : “You have the permission of a parent.”
3. You can graduate only if you have completed the requirements of your major and you do not owe money to the university and you do not have an overdue library book. Express your answer in terms of g : “You can graduate,” m : “You owe money to the university,” r : “You have completed the requirements of your major,” and b : “You have an overdue library book.”
4. To use the wireless network in the airport you must pay the daily fee unless you are a subscriber to the service. Express your answer in terms of w : “You can use the wireless network in the airport,” d : “You pay the daily fee,” and s : “You are a subscriber to the service.”
5. You are eligible to be President of the U.S.A. only if you are at least 35 years old, were born in the U.S.A., or at the time of your birth both of your parents were citizens, and you have lived at least 14 years in the country. Express your answer in terms of e : “You are eligible to be President of the U.S.A.”, a : “You are at least 35 years old,” b : “You were born in the U.S.A.”, p : “At the time of your birth, both of your parents were citizens,” and r : “You have lived at least 14 years in the U.S.A.”
6. You can upgrade your operating system only if you have a 32-bit processor running at 1 GHz or faster, at least 1 GB RAM, and 16 GB free hard disk space, or a 64-bit processor running at 2 GHz or faster, at least 2 GB RAM, and at least 32 GB free hard disk space. Express your answer in terms of u : “You can upgrade your operating system,” b_{32} : “You have a 32-bit processor,” b_{64} :

“You have a 64-bit processor,” g_1 : “Your processor runs at 1 GHz or faster,” g_2 : “Your processor runs at 2 GHz or faster,” r_1 : “Your processor has at least 1 GB RAM,” r_2 : “Your processor has at least 2 GB RAM,” h_{16} : “You have at least 16 GB free hard disk space,” and h_{32} : “You have at least 32 GB free hard disk space.”

7. Express these system specifications using the propositions p “The message is scanned for viruses” and q “The message was sent from an unknown system” together with logical connectives (including negations).
 - a) “The message is scanned for viruses whenever the message was sent from an unknown system.”
 - b) “The message was sent from an unknown system but it was not scanned for viruses.”
 - c) “It is necessary to scan the message for viruses whenever it was sent from an unknown system.”
 - d) “When a message is not sent from an unknown system it is not scanned for viruses.”
8. Express these system specifications using the propositions p “The user enters a valid password,” q “Access is granted,” and r “The user has paid the subscription fee” and logical connectives (including negations).
 - a) “The user has paid the subscription fee, but does not enter a valid password.”
 - b) “Access is granted whenever the user has paid the subscription fee and enters a valid password.”
 - c) “Access is denied if the user has not paid the subscription fee.”
 - d) “If the user has not entered a valid password but has paid the subscription fee, then access is granted.”
9. Are these system specifications consistent? “The system is in multiuser state if and only if it is operating normally. If the system is operating normally, the kernel is functioning. The kernel is not functioning or the system is in interrupt mode. If the system is not in multiuser state, then it is in interrupt mode. The system is not in interrupt mode.”

10. Are these system specifications consistent? “Whenever the system software is being upgraded, users cannot access the file system. If users can access the file system, then they can save new files. If users cannot save new files, then the system software is not being upgraded.”
11. Are these system specifications consistent? “The router can send packets to the edge system only if it supports the new address space. For the router to support the new address space it is necessary that the latest software release be installed. The router can send packets to the edge system if the latest software release is installed. The router does not support the new address space.”
12. Are these system specifications consistent? “If the file system is not locked, then new messages will be queued. If the file system is not locked, then the system is functioning normally, and conversely. If new messages are not queued, then they will be sent to the message buffer. If the file system is not locked, then new messages will be sent to the message buffer. New messages will not be sent to the message buffer.”
13. What Boolean search would you use to look for Web pages about beaches in New Jersey? What if you wanted to find Web pages about beaches on the isle of Jersey (in the English Channel)?
14. What Boolean search would you use to look for Web pages about hiking in West Virginia? What if you wanted to find Web pages about hiking in Virginia, but not in West Virginia?
- *15. Each inhabitant of a remote village always tells the truth or always lies. A villager will give only a “Yes” or a “No” response to a question a tourist asks. Suppose you are a tourist visiting this area and come to a fork in the road. One branch leads to the ruins you want to visit; the other branch leads deep into the jungle. A villager is standing at the fork in the road. What one question can you ask the villager to determine which branch to take?
16. An explorer is captured by a group of cannibals. There are two types of cannibals—those who always tell the truth and those who always lie. The cannibals will barbecue the explorer unless he can determine whether a particular cannibal always lies or always tells the truth. He is allowed to ask the cannibal exactly one question.
 - a) Explain why the question “Are you a liar?” does not work.
 - b) Find a question that the explorer can use to determine whether the cannibal always lies or always tells the truth.
17. When three professors are seated in a restaurant, the hostess asks them: “Does everyone want coffee?” The first professor says: “I do not know.” The second professor then says: “I do not know.” Finally, the third professor says: “No, not everyone wants coffee.” The hostess comes back and gives coffee to the professors who want it. How did she figure out who wanted coffee?
18. When planning a party you want to know whom to invite. Among the people you would like to invite are three touchy friends. You know that if Jasmine attends, she will

become unhappy if Samir is there, Samir will attend only if Kanti will be there, and Kanti will not attend unless Jasmine also does. Which combinations of these three friends can you invite so as not to make someone unhappy?

Exercises 19–23 relate to inhabitants of the island of knights and knaves created by Smullyan, where knights always tell the truth and knaves always lie. You encounter two people, *A* and *B*. Determine, if possible, what *A* and *B* are if they address you in the ways described. If you cannot determine what these two people are, can you draw any conclusions?

19. *A* says “At least one of us is a knave” and *B* says nothing.
20. *A* says “The two of us are both knights” and *B* says “*A* is a knave.”
21. *A* says “I am a knave or *B* is a knight” and *B* says nothing.
22. Both *A* and *B* say “I am a knight.”
23. *A* says “We are both knaves” and *B* says nothing.

Exercises 24–31 relate to inhabitants of an island on which there are three kinds of people: knights who always tell the truth, knaves who always lie, and spies (called normals by Smullyan [Sm78]) who can either lie or tell the truth. You encounter three people, *A*, *B*, and *C*. You know one of these people is a knight, one is a knave, and one is a spy. Each of the three people knows the type of person each of other two is. For each of these situations, if possible, determine whether there is a unique solution and determine who the knave, knight, and spy are. When there is no unique solution, list all possible solutions or state that there are no solutions.

24. *A* says “*C* is the knave,” *B* says, “*A* is the knight,” and *C* says “I am the spy.”
25. *A* says “I am the knight,” *B* says “I am the knave,” and *C* says “*B* is the knight.”
26. *A* says “I am the knave,” *B* says “I am the knave,” and *C* says “I am the knave.”
27. *A* says “I am the knight,” *B* says “*A* is telling the truth,” and *C* says “I am the spy.”
28. *A* says “I am the knight,” *B* says, “*A* is not the knave,” and *C* says “*B* is not the knave.”
29. *A* says “I am the knight,” *B* says “I am the knight,” and *C* says “I am the knight.”
30. *A* says “I am not the spy,” *B* says “I am not the spy,” and *C* says “*A* is the spy.”
31. *A* says “I am not the spy,” *B* says “I am not the spy,” and *C* says “I am not the spy.”

Exercises 32–38 are puzzles that can be solved by translating statements into logical expressions and reasoning from these expressions using truth tables.

32. The police have three suspects for the murder of Mr. Cooper: Mr. Smith, Mr. Jones, and Mr. Williams. Smith, Jones, and Williams each declare that they did not kill Cooper. Smith also states that Cooper was a friend of Jones and that Williams disliked him. Jones also states that he did not know Cooper and that he was out of town the day Cooper was killed. Williams also states that he

saw both Smith and Jones with Cooper the day of the killing and that either Smith or Jones must have killed him. Can you determine who the murderer was if

- a) one of the three men is guilty, the two innocent men are telling the truth, but the statements of the guilty man may or may not be true?
- b) innocent men do not lie?

33. Steve would like to determine the relative salaries of three coworkers using two facts. First, he knows that if Fred is not the highest paid of the three, then Janice is. Second, he knows that if Janice is not the lowest paid, then Maggie is paid the most. Is it possible to determine the relative salaries of Fred, Maggie, and Janice from what Steve knows? If so, who is paid the most and who the least? Explain your reasoning.

34. Five friends have access to a chat room. Is it possible to determine who is chatting if the following information is known? Either Kevin or Heather, or both, are chatting. Either Randy or Vijay, but not both, are chatting. If Abby is chatting, so is Randy. Vijay and Kevin are either both chatting or neither is. If Heather is chatting, then so are Abby and Kevin. Explain your reasoning.

35. A detective has interviewed four witnesses to a crime. From the stories of the witnesses the detective has concluded that if the butler is telling the truth then so is the cook; the cook and the gardener cannot both be telling the truth; the gardener and the handyman are not both lying; and if the handyman is telling the truth then the cook is lying. For each of the four witnesses, can the detective determine whether that person is telling the truth or lying? Explain your reasoning.

36. Four friends have been identified as suspects for an unauthorized access into a computer system. They have made statements to the investigating authorities. Alice said “Carlos did it.” John said “I did not do it.” Carlos said “Diana did it.” Diana said “Carlos lied when he said that I did it.”

- a) If the authorities also know that exactly one of the four suspects is telling the truth, who did it? Explain your reasoning.
- b) If the authorities also know that exactly one is lying, who did it? Explain your reasoning.

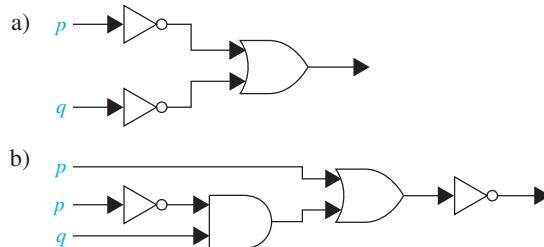
37. Suppose there are signs on the doors to two rooms. The sign on the first door reads “In this room there is a lady, and in the other one there is a tiger”; and the sign on the second door reads “In one of these rooms, there is a lady, and in one of them there is a tiger.” Suppose that you know that one of these signs is true and the other is false. Behind which door is the lady?

*38. Solve this famous logic puzzle, attributed to Albert Einstein, and known as the **zebra puzzle**. Five men with different nationalities and with different jobs live in consecutive houses on a street. These houses are painted different colors. The men have different pets and have different favorite drinks. Determine who owns a zebra and

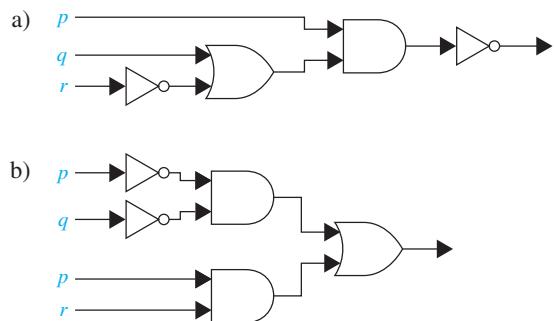
whose favorite drink is mineral water (which is one of the favorite drinks) given these clues: The Englishman lives in the red house. The Spaniard owns a dog. The Japanese man is a painter. The Italian drinks tea. The Norwegian lives in the first house on the left. The green house is immediately to the right of the white one. The photographer breeds snails. The diplomat lives in the yellow house. Milk is drunk in the middle house. The owner of the green house drinks coffee. The Norwegian’s house is next to the blue one. The violinist drinks orange juice. The fox is in a house next to that of the physician. The horse is in a house next to that of the diplomat. [Hint: Make a table where the rows represent the men and columns represent the color of their houses, their jobs, their pets, and their favorite drinks and use logical reasoning to determine the correct entries in the table.]

39. Freedonia has fifty senators. Each senator is either honest or corrupt. Suppose you know that at least one of the Freedonian senators is honest and that, given any two Freedonian senators, at least one is corrupt. Based on these facts, can you determine how many Freedonian senators are honest and how many are corrupt? If so, what is the answer?

40. Find the output of each of these combinatorial circuits.



41. Find the output of each of these combinatorial circuits.



42. Construct a combinatorial circuit using inverters, OR gates, and AND gates that produces the output $(p \wedge \neg r) \vee (\neg q \wedge r)$ from input bits p , q , and r .

43. Construct a combinatorial circuit using inverters, OR gates, and AND gates that produces the output $((\neg p \vee \neg r) \wedge \neg q) \vee (\neg p \wedge (q \vee r))$ from input bits p , q , and r .

1.3 Propositional Equivalences

Introduction

An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value. Because of this, methods that produce propositions with the same truth value as a given compound proposition are used extensively in the construction of mathematical arguments. Note that we will use the term “compound proposition” to refer to an expression formed from propositional variables using logical operators, such as $p \wedge q$.

We begin our discussion with a classification of compound propositions according to their possible truth values.

DEFINITION 1

A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a *tautology*. A compound proposition that is always false is called a *contradiction*. A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

Tautologies and contradictions are often important in mathematical reasoning. Example 1 illustrates these types of compound propositions.

EXAMPLE 1

We can construct examples of tautologies and contradictions using just one propositional variable. Consider the truth tables of $p \vee \neg p$ and $p \wedge \neg p$, shown in Table 1. Because $p \vee \neg p$ is always true, it is a tautology. Because $p \wedge \neg p$ is always false, it is a contradiction. 

Logical Equivalences



Compound propositions that have the same truth values in all possible cases are called **logically equivalent**. We can also define this notion as follows.

DEFINITION 2

The compound propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

Remark: The symbol \equiv is not a logical connective, and $p \equiv q$ is not a compound proposition but rather is the statement that $p \leftrightarrow q$ is a tautology. The symbol \Leftrightarrow is sometimes used instead of \equiv to denote logical equivalence.

One way to determine whether two compound propositions are equivalent is to use a truth table. In particular, the compound propositions p and q are equivalent if and only if the columns

TABLE 1 Examples of a Tautology and a Contradiction.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

TABLE 2 De Morgan's Laws.

$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$



giving their truth values agree. Example 2 illustrates this method to establish an extremely important and useful logical equivalence, namely, that of $\neg(p \vee q)$ with $\neg p \wedge \neg q$. This logical equivalence is one of the two **De Morgan laws**, shown in Table 2, named after the English mathematician Augustus De Morgan, of the mid-nineteenth century.

EXAMPLE 2 Show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent.

Solution: The truth tables for these compound propositions are displayed in Table 3. Because the truth values of the compound propositions $\neg(p \vee q)$ and $\neg p \wedge \neg q$ agree for all possible combinations of the truth values of p and q , it follows that $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ is a tautology and that these compound propositions are logically equivalent.

TABLE 3 Truth Tables for $\neg(p \vee q)$ and $\neg p \wedge \neg q$.

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

EXAMPLE 3 Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

Solution: We construct the truth table for these compound propositions in Table 4. Because the truth values of $\neg p \vee q$ and $p \rightarrow q$ agree, they are logically equivalent.

TABLE 4 Truth Tables for $\neg p \vee q$ and $p \rightarrow q$.

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

We will now establish a logical equivalence of two compound propositions involving three different propositional variables p , q , and r . To use a truth table to establish such a logical equivalence, we need eight rows, one for each possible combination of truth values of these three variables. We symbolically represent these combinations by listing the truth values of p , q , and r , respectively. These eight combinations of truth values are TTT, TTF, TFT, TFF, FTT, FTF, FFT, and FFF; we use this order when we display the rows of the truth table. Note that we need to double the number of rows in the truth tables we use to show that compound propositions are equivalent for each additional propositional variable, so that 16 rows are needed to establish the logical equivalence of two compound propositions involving four propositional variables, and so on. In general, 2^n rows are required if a compound proposition involves n propositional variables.

TABLE 5 A Demonstration That $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ Are Logically Equivalent.

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

EXAMPLE 4 Show that $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ are logically equivalent. This is the *distributive law* of disjunction over conjunction.

Solution: We construct the truth table for these compound propositions in Table 5. Because the truth values of $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ agree, these compound propositions are logically equivalent. 

The identities in Table 6 are a special case of Boolean algebra identities found in Table 5 of Section 12.1. See Table 1 in Section 2.2 for analogous set identities.

Table 6 contains some important equivalences. In these equivalences, T denotes the compound proposition that is always true and F denotes the compound proposition that is always

TABLE 6 Logical Equivalences.

Equivalence	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation laws

TABLE 7 Logical Equivalences Involving Conditional Statements.

$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

false. We also display some useful equivalences for compound propositions involving conditional statements and biconditional statements in Tables 7 and 8, respectively. The reader is asked to verify the equivalences in Tables 6–8 in the exercises.

The associative law for disjunction shows that the expression $p \vee q \vee r$ is well defined, in the sense that it does not matter whether we first take the disjunction of p with q and then the disjunction of $p \vee q$ with r , or if we first take the disjunction of q and r and then take the disjunction of p with $q \vee r$. Similarly, the expression $p \wedge q \wedge r$ is well defined. By extending this reasoning, it follows that $p_1 \vee p_2 \vee \cdots \vee p_n$ and $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ are well defined whenever p_1, p_2, \dots, p_n are propositions.

Furthermore, note that De Morgan's laws extend to

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

and

$$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n).$$

We will sometimes use the notation $\bigvee_{j=1}^n p_j$ for $p_1 \vee p_2 \vee \cdots \vee p_n$ and $\bigwedge_{j=1}^n p_j$ for $p_1 \wedge p_2 \wedge \cdots \wedge p_n$. Using this notation, the extended version of De Morgan's laws can be written concisely as $\neg(\bigvee_{j=1}^n p_j) \equiv \bigwedge_{j=1}^n \neg p_j$ and $\neg(\bigwedge_{j=1}^n p_j) \equiv \bigvee_{j=1}^n \neg p_j$. (Methods for proving these identities will be given in Section 5.1.)

Using De Morgan's Laws

When using De Morgan's laws, remember to change the logical connective after you negate.

The two logical equivalences known as De Morgan's laws are particularly important. They tell us how to negate conjunctions and how to negate disjunctions. In particular, the equivalence $\neg(p \vee q) \equiv \neg p \wedge \neg q$ tells us that the negation of a disjunction is formed by taking the conjunction of the negations of the component propositions. Similarly, the equivalence $\neg(p \wedge q) \equiv \neg p \vee \neg q$ tells us that the negation of a conjunction is formed by taking the disjunction of the negations of the component propositions. Example 5 illustrates the use of De Morgan's laws.

EXAMPLE 5 Use De Morgan's laws to express the negations of "Miguel has a cellphone and he has a laptop computer" and "Heather will go to the concert or Steve will go to the concert."



Solution: Let p be "Miguel has a cellphone" and q be "Miguel has a laptop computer." Then "Miguel has a cellphone and he has a laptop computer" can be represented by $p \wedge q$. By the first of De Morgan's laws, $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$. Consequently, we can express the negation of our original statement as "Miguel does not have a cellphone or he does not have a laptop computer."

Let r be "Heather will go to the concert" and s be "Steve will go to the concert." Then "Heather will go to the concert or Steve will go to the concert" can be represented by $r \vee s$. By the second of De Morgan's laws, $\neg(r \vee s)$ is equivalent to $\neg r \wedge \neg s$. Consequently, we can express the negation of our original statement as "Heather will not go to the concert and Steve will not go to the concert."

Constructing New Logical Equivalences

The logical equivalences in Table 6, as well as any others that have been established (such as those shown in Tables 7 and 8), can be used to construct additional logical equivalences. The reason for this is that a proposition in a compound proposition can be replaced by a compound proposition that is logically equivalent to it without changing the truth value of the original compound proposition. This technique is illustrated in Examples 6–8, where we also use the fact that if p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent (see Exercise 56).

EXAMPLE 6 Show that $\neg(p \rightarrow q)$ and $p \wedge \neg q$ are logically equivalent.



Solution: We could use a truth table to show that these compound propositions are equivalent (similar to what we did in Example 4). Indeed, it would not be hard to do so. However, we want to illustrate how to use logical identities that we already know to establish new logical identities, something that is of practical importance for establishing equivalences of compound propositions with a large number of variables. So, we will establish this equivalence by developing a series of



AUGUSTUS DE MORGAN (1806–1871) Augustus De Morgan was born in India, where his father was a colonel in the Indian army. De Morgan's family moved to England when he was 7 months old. He attended private schools, where in his early teens he developed a strong interest in mathematics. De Morgan studied at Trinity College, Cambridge, graduating in 1827. Although he considered medicine or law, he decided on mathematics for his career. He won a position at University College, London, in 1828, but resigned after the college dismissed a fellow professor without giving reasons. However, he resumed this position in 1836 when his successor died, remaining until 1866.

De Morgan was a noted teacher who stressed principles over techniques. His students included many famous mathematicians, including Augusta Ada, Countess of Lovelace, who was Charles Babbage's collaborator in his work on computing machines (see page 31 for biographical notes on Augusta Ada). (De Morgan cautioned the countess against studying too much mathematics, because it might interfere with her childbearing abilities!)

De Morgan was an extremely prolific writer, publishing more than 1000 articles in more than 15 periodicals. De Morgan also wrote textbooks on many subjects, including logic, probability, calculus, and algebra. In 1838 he presented what was perhaps the first clear explanation of an important proof technique known as *mathematical induction* (discussed in Section 5.1 of this text), a term he coined. In the 1840s De Morgan made fundamental contributions to the development of symbolic logic. He invented notations that helped him prove propositional equivalences, such as the laws that are named after him. In 1842 De Morgan presented what is considered to be the first precise definition of a limit and developed new tests for convergence of infinite series. De Morgan was also interested in the history of mathematics and wrote biographies of Newton and Halley.

In 1837 De Morgan married Sophia Frend, who wrote his biography in 1882. De Morgan's research, writing, and teaching left little time for his family or social life. Nevertheless, he was noted for his kindness, humor, and wide range of knowledge.

logical equivalences, using one of the equivalences in Table 6 at a time, starting with $\neg(p \rightarrow q)$ and ending with $p \wedge \neg q$. We have the following equivalences.

$$\begin{aligned}\neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) && \text{by Example 3} \\ &\equiv \neg(\neg p) \wedge \neg q && \text{by the second De Morgan law} \\ &\equiv p \wedge \neg q && \text{by the double negation law}\end{aligned}$$



EXAMPLE 7 Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent by developing a series of logical equivalences.

Solution: We will use one of the equivalences in Table 6 at a time, starting with $\neg(p \vee (\neg p \wedge q))$ and ending with $\neg p \wedge \neg q$. (Note: we could also easily establish this equivalence using a truth table.) We have the following equivalences.

$$\begin{aligned}\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{by the second De Morgan law} \\ &\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{by the first De Morgan law} \\ &\equiv \neg p \wedge (p \vee \neg q) && \text{by the double negation law} \\ &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{by the second distributive law} \\ &\equiv \mathbf{F} \vee (\neg p \wedge \neg q) && \text{because } \neg p \wedge p \equiv \mathbf{F} \\ &\equiv (\neg p \wedge \neg q) \vee \mathbf{F} && \text{by the commutative law for disjunction} \\ &\equiv \neg p \wedge \neg q && \text{by the identity law for F}\end{aligned}$$

Consequently $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent.



EXAMPLE 8 Show that $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

Solution: To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to **T**. (Note: This could also be done using a truth table.)

$$\begin{aligned}(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{by Example 3} \\ &\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{by the first De Morgan law} \\ &\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{by the associative and commutative laws for disjunction} \\ &\equiv \mathbf{T} \vee \mathbf{T} && \text{by Example 1 and the commutative law for disjunction} \\ &\equiv \mathbf{T} && \text{by the domination law}\end{aligned}$$



Propositional Satisfiability

A compound proposition is **satisfiable** if there is an assignment of truth values to its variables that makes it true. When no such assignments exists, that is, when the compound proposition is false for all assignments of truth values to its variables, the compound proposition is **unsatisfiable**.

Note that a compound proposition is unsatisfiable if and only if its negation is true for all assignments of truth values to the variables, that is, if and only if its negation is a tautology.

When we find a particular assignment of truth values that makes a compound proposition true, we have shown that it is satisfiable; such an assignment is called a **solution** of this particular

satisfiability problem. However, to show that a compound proposition is unsatisfiable, we need to show that *every* assignment of truth values to its variables makes it false. Although we can always use a truth table to determine whether a compound proposition is satisfiable, it is often more efficient not to, as Example 9 demonstrates.

EXAMPLE 9 Determine whether each of the compound propositions $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$, and $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable.

Solution: Instead of using truth table to solve this problem, we will reason about truth values. Note that $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when the three variable p , q , and r have the same truth value (see Exercise 40 of Section 1.1). Hence, it is satisfiable as there is at least one assignment of truth values for p , q , and r that makes it true. Similarly, note that $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p , q , and r is true and at least one is false (see Exercise 41 of Section 1.1). Hence, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable, as there is at least one assignment of truth values for p , q , and r that makes it true.

Finally, note that for $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ to be true, $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ and $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ must both be true. For the first to be true, the three variables must have the same truth values, and for the second to be true, at least one of three variables must be true and at least one must be false. However, these conditions are contradictory. From these observations we conclude that no assignment of truth values to p , q , and r makes $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ true. Hence, it is unsatisfiable. \blacktriangleleft



AUGUSTA ADA, COUNTESS OF LOVELACE (1815–1852) Augusta Ada was the only child from the marriage of the famous poet Lord Byron and Lady Byron, Annabella Millbanke, who separated when Ada was 1 month old, because of Lord Byron's scandalous affair with his half sister. The Lord Byron had quite a reputation, being described by one of his lovers as "mad, bad, and dangerous to know." Lady Byron was noted for her intellect and had a passion for mathematics; she was called by Lord Byron "The Princess of Parallelograms." Augusta was raised by her mother, who encouraged her intellectual talents especially in music and mathematics, to counter what Lady Byron considered dangerous poetic tendencies. At this time, women were not allowed to attend universities and could not join learned societies. Nevertheless, Augusta pursued her mathematical studies independently and with mathematicians, including William Frend. She was also encouraged by another female mathematician, Mary Somerville, and in 1834 at a dinner party hosted by Mary Somerville, she learned about Charles Babbage's ideas for a calculating machine, called the Analytic Engine. In 1838 Augusta Ada married Lord King, later elevated to Earl of Lovelace. Together they had three children.

Augusta Ada continued her mathematical studies after her marriage. Charles Babbage had continued work on his Analytic Engine and lectured on this in Europe. In 1842 Babbage asked Augusta Ada to translate an article in French describing Babbage's invention. When Babbage saw her translation, he suggested she add her own notes, and the resulting work was three times the length of the original. The most complete accounts of the Analytic Engine are found in Augusta Ada's notes. In her notes, she compared the working of the Analytic Engine to that of the Jacquard loom, with Babbage's punch cards analogous to the cards used to create patterns on the loom. Furthermore, she recognized the promise of the machine as a general purpose computer much better than Babbage did. She stated that the "engine is the material expression of any indefinite function of any degree of generality and complexity." Her notes on the Analytic Engine anticipate many future developments, including computer-generated music. Augusta Ada published her writings under her initials A.A.L. concealing her identity as a woman as did many women at a time when women were not considered to be the intellectual equals of men. After 1845 she and Babbage worked toward the development of a system to predict horse races. Unfortunately, their system did not work well, leaving Augusta Ada heavily in debt at the time of her death at an unfortunately young age from uterine cancer.

In 1953 Augusta Ada's notes on the Analytic Engine were republished more than 100 years after they were written, and after they had been long forgotten. In his work in the 1950s on the capacity of computers to think (and his famous Turing Test), Alan Turing responded to Augusta Ada's statement that "The Analytic Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform." This "dialogue" between Turing and Augusta Ada is still the subject of controversy. Because of her fundamental contributions to computing, the programming language Ada is named in honor of the Countess of Lovelace.

	2	9				4		
			5			1		
4								
			4	2				
6						7		
5								
7		3					5	
1			9					
						6		

FIGURE 1 A 9×9 Sudoku puzzle.

Applications of Satisfiability

Many problems, in diverse areas such as robotics, software testing, computer-aided design, machine vision, integrated circuit design, computer networking, and genetics, can be modeled in terms of propositional satisfiability. Although most of these applications are beyond the scope of this book, we will study one application here. In particular, we will show how to use propositional satisfiability to model Sudoku puzzles.



SUDOKU A **Sudoku puzzle** is represented by a 9×9 grid made up of nine 3×3 subgrids, known as **blocks**, as shown in Figure 1. For each puzzle, some of the 81 cells, called **givens**, are assigned one of the numbers 1, 2, ..., 9, and the other cells are blank. The puzzle is solved by assigning a number to each blank cell so that every row, every column, and every one of the nine 3×3 blocks contains each of the nine possible numbers. Note that instead of using a 9×9 grid, Sudoku puzzles can be based on $n^2 \times n^2$ grids, for any positive integer n , with the $n^2 \times n^2$ grid made up of $n^2 \times n^2$ subgrids.

The popularity of Sudoku dates back to the 1980s when it was introduced in Japan. It took 20 years for Sudoku to spread to rest of the world, but by 2005, Sudoku puzzles were a worldwide craze. The name Sudoku is short for the Japanese *suuji wa dokushin ni kagiru*, which means “the digits must remain single.” The modern game of Sudoku was apparently designed in the late 1970s by an American puzzle designer. The basic ideas of Sudoku date back even further; puzzles printed in French newspapers in the 1890s were quite similar, but not identical, to modern Sudoku.

Sudoku puzzles designed for entertainment have two additional important properties. First, they have exactly one solution. Second, they can be solved using reasoning alone, that is, without resorting to searching all possible assignments of numbers to the cells. As a Sudoku puzzle is solved, entries in blank cells are successively determined by already known values. For instance, in the grid in Figure 1, the number 4 must appear in exactly one cell in the second row. How can we determine which of the seven blank cells it must appear? First, we observe that 4 cannot appear in one of the first three cells or in one of the last three cells of this row, because it already appears in another cell in the block each of these cells is in. We can also see that 4 cannot appear in the fifth cell in this row, as it already appears in the fifth column in the fourth row. This means that 4 must appear in the sixth cell of the second row.

Many strategies based on logic and mathematics have been devised for solving Sudoku puzzles (see [Da10], for example). Here, we discuss one of the ways that have been developed for solving Sudoku puzzles with the aid of a computer, which depends on modeling the puzzle as a propositional satisfiability problem. Using the model we describe, particular Sudoku puzzles can be solved using software developed to solve satisfiability problems. Currently, Sudoku puzzles can be solved in less than 10 milliseconds this way. It should be noted that there are many other approaches for solving Sudoku puzzles via computers using other techniques.

To encode a Sudoku puzzle, let $p(i, j, n)$ denote the proposition that is true when the number n is in the cell in the i th row and j th column. There are $9 \times 9 \times 9 = 729$ such propositions, as i , j , and n all range from 1 to 9. For example, for the puzzle in Figure 1, the number 6 is given as the value in the fifth row and first column. Hence, we see that $p(5, 1, 6)$ is true, but $p(5, j, 6)$ is false for $j = 2, 3, \dots, 9$.

Given a particular Sudoku puzzle, we begin by encoding each of the given values. Then, we construct compound propositions that assert that every row contains every number, every column contains every number, every 3×3 block contains every number, and each cell contains no more than one number. It follows, as the reader should verify, that the Sudoku puzzle is solved by finding an assignment of truth values to the 729 propositions $p(i, j, n)$ with i , j , and n each ranging from 1 to 9 that makes the conjunction of all these compound propositions true. After listing these assertions, we will explain how to construct the assertion that every row contains every integer from 1 to 9. We will leave the construction of the other assertions that every column contains every number and each of the nine 3×3 blocks contains every number to the exercises.

- For each cell with a given value, we assert $p(i, j, n)$ when the cell in row i and column j has the given value n .
- We assert that every row contains every number:

$$\bigwedge_{i=1}^9 \bigwedge_{n=1}^9 \bigvee_{j=1}^9 p(i, j, n)$$

- We assert that every column contains every number:

$$\bigwedge_{j=1}^9 \bigwedge_{n=1}^9 \bigvee_{i=1}^9 p(i, j, n)$$

- We assert that each of the nine 3×3 blocks contains every number:

$$\bigwedge_{r=0}^2 \bigwedge_{s=0}^2 \bigwedge_{n=1}^9 \bigvee_{i=1}^3 \bigvee_{j=1}^3 p(3r + i, 3s + j, n)$$

- To assert that no cell contains more than one number, we take the conjunction over all values of n , n' , i , and j where each variable ranges from 1 to 9 and $n \neq n'$ of $p(i, j, n) \rightarrow \neg p(i, j, n')$.

We now explain how to construct the assertion that every row contains every number. First, to assert that row i contains the number n , we form $\bigvee_{j=1}^9 p(i, j, n)$. To assert that row i contains all n numbers, we form the conjunction of these disjunctions over all nine possible values of n , giving us $\bigwedge_{n=1}^9 \bigvee_{j=1}^9 p(i, j, n)$. Finally, to assert that every row contains every number, we take the conjunction of $\bigwedge_{n=1}^9 \bigvee_{j=1}^9 p(i, j, n)$ over all nine rows. This gives us $\bigwedge_{i=1}^9 \bigwedge_{n=1}^9 \bigvee_{j=1}^9 p(i, j, n)$. (Exercises 65 and 66 ask for explanations of the assertions that every column contains every number and that each of the nine 3×3 blocks contains every number.)

Given a particular Sudoku puzzle, to solve this puzzle we can find a solution to the satisfiability problems that asks for a set of truth values for the 729 variables $p(i, j, n)$ that makes the conjunction of all the listed assertions true.

 It is tricky setting up the two inner indices so that all nine cells in each square block are examined.

Solving Satisfiability Problems

A truth table can be used to determine whether a compound proposition is satisfiable, or equivalently, whether its negation is a tautology (see Exercise 60). This can be done by hand for a compound proposition with a small number of variables, but when the number of variables grows, this becomes impractical. For instance, there are $2^{20} = 1,048,576$ rows in the truth table for a compound proposition with 20 variables. Clearly, you need a computer to help you determine, in this way, whether a compound proposition in 20 variables is satisfiable.

When many applications are modeled, questions concerning the satisfiability of compound propositions with hundreds, thousands, or millions of variables arise. Note, for example, that when there are 1000 variables, checking every one of the 2^{1000} (a number with more than 300 decimal digits) possible combinations of truth values of the variables in a compound proposition cannot be done by a computer in even trillions of years. No procedure is known that a computer can follow to determine in a reasonable amount of time whether an arbitrary compound proposition in such a large number of variables is satisfiable. However, progress has been made developing methods for solving the satisfiability problem for the particular types of compound propositions that arise in practical applications, such as for the solution of Sudoku puzzles. Many computer programs have been developed for solving satisfiability problems which have practical use. In our discussion of the subject of algorithms in Chapter 3, we will discuss this question further. In particular, we will explain the important role the propositional satisfiability problem plays in the study of the complexity of algorithms.



Exercises

1. Use truth tables to verify these equivalences.
 - a) $p \wedge T \equiv p$
 - b) $p \vee F \equiv p$
 - c) $p \wedge F \equiv F$
 - d) $p \vee T \equiv T$
 - e) $p \vee p \equiv p$
 - f) $p \wedge p \equiv p$
2. Show that $\neg(\neg p)$ and p are logically equivalent.
3. Use truth tables to verify the commutative laws
 - a) $p \vee q \equiv q \vee p$.
 - b) $p \wedge q \equiv q \wedge p$.
4. Use truth tables to verify the associative laws
 - a) $(p \vee q) \vee r \equiv p \vee (q \vee r)$.
 - b) $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$.
5. Use a truth table to verify the distributive law

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r).$$
6. Use a truth table to verify the first De Morgan law

$$\neg(p \wedge q) \equiv \neg p \vee \neg q.$$
7. Use De Morgan's laws to find the negation of each of the following statements.
 - a) Jan is rich and happy.
 - b) Carlos will bicycle or run tomorrow.



HENRY MAURICE SHEFFER (1883–1964) Henry Maurice Sheffer, born to Jewish parents in the western Ukraine, emigrated to the United States in 1892 with his parents and six siblings. He studied at the Boston Latin School before entering Harvard, where he completed his undergraduate degree in 1905, his master's in 1907, and his Ph.D. in philosophy in 1908. After holding a postdoctoral position at Harvard, Henry traveled to Europe on a fellowship. Upon returning to the United States, he became an academic nomad, spending one year each at the University of Washington, Cornell, the University of Minnesota, the University of Missouri, and City College in New York. In 1916 he returned to Harvard as a faculty member in the philosophy department. He remained at Harvard until his retirement in 1952.

Sheffer introduced what is now known as the Sheffer stroke in 1913; it became well known only after its use in the 1925 edition of Whitehead and Russell's *Principia Mathematica*. In this same edition Russell wrote that Sheffer had invented a powerful method that could be used to simplify the *Principia*. Because of this comment, Sheffer was something of a mystery man to logicians, especially because Sheffer, who published little in his career, never published the details of this method, only describing it in mimeographed notes and in a brief published abstract.

Sheffer was a dedicated teacher of mathematical logic. He liked his classes to be small and did not like auditors. When strangers appeared in his classroom, Sheffer would order them to leave, even his colleagues or distinguished guests visiting Harvard. Sheffer was barely five feet tall; he was noted for his wit and vigor, as well as for his nervousness and irritability. Although widely liked, he was quite lonely. He is noted for a quip he spoke at his retirement: "Old professors never die, they just become emeriti." Sheffer is also credited with coining the term "Boolean algebra" (the subject of Chapter 12 of this text). Sheffer was briefly married and lived most of his later life in small rooms at a hotel packed with his logic books and vast files of slips of paper he used to jot down his ideas. Unfortunately, Sheffer suffered from severe depression during the last two decades of his life.

- c) Mei walks or takes the bus to class.
d) Ibrahim is smart and hard working.
8. Use De Morgan's laws to find the negation of each of the following statements.
- Kwame will take a job in industry or go to graduate school.
 - Yoshiko knows Java and calculus.
 - James is young and strong.
 - Rita will move to Oregon or Washington.
9. Show that each of these conditional statements is a tautology by using truth tables.
- $(p \wedge q) \rightarrow p$
 - $p \rightarrow (p \vee q)$
 - $\neg p \rightarrow (p \rightarrow q)$
 - $(p \wedge q) \rightarrow (p \rightarrow q)$
 - $\neg(p \rightarrow q) \rightarrow p$
 - $\neg(p \rightarrow q) \rightarrow \neg q$
10. Show that each of these conditional statements is a tautology by using truth tables.
- $[\neg p \wedge (p \vee q)] \rightarrow q$
 - $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$
 - $[p \wedge (p \rightarrow q)] \rightarrow q$
 - $[(p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)] \rightarrow r$
11. Show that each conditional statement in Exercise 9 is a tautology without using truth tables.
12. Show that each conditional statement in Exercise 10 is a tautology without using truth tables.
13. Use truth tables to verify the absorption laws.
- $p \vee (p \wedge q) \equiv p$
 - $p \wedge (p \vee q) \equiv p$
14. Determine whether $(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$ is a tautology.
15. Determine whether $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$ is a tautology.

Each of Exercises 16–28 asks you to show that two compound propositions are logically equivalent. To do this, either show that both sides are true, or that both sides are false, for exactly the same combinations of truth values of the propositional variables in these expressions (whichever is easier).

- Show that $p \leftrightarrow q$ and $(p \wedge q) \vee (\neg p \wedge \neg q)$ are logically equivalent.
- Show that $\neg(p \leftrightarrow q)$ and $p \leftrightarrow \neg q$ are logically equivalent.
- Show that $p \rightarrow q$ and $\neg q \rightarrow \neg p$ are logically equivalent.
- Show that $\neg p \leftrightarrow q$ and $p \leftrightarrow \neg q$ are logically equivalent.
- Show that $\neg(p \oplus q)$ and $p \leftrightarrow q$ are logically equivalent.
- Show that $\neg(p \leftrightarrow q)$ and $\neg p \leftrightarrow q$ are logically equivalent.
- Show that $(p \rightarrow q) \wedge (p \rightarrow r)$ and $p \rightarrow (q \wedge r)$ are logically equivalent.
- Show that $(p \rightarrow r) \wedge (q \rightarrow r)$ and $(p \vee q) \rightarrow r$ are logically equivalent.
- Show that $(p \rightarrow q) \vee (p \rightarrow r)$ and $p \rightarrow (q \vee r)$ are logically equivalent.
- Show that $(p \rightarrow r) \vee (q \rightarrow r)$ and $(p \wedge q) \rightarrow r$ are logically equivalent.
- Show that $\neg p \rightarrow (q \rightarrow r)$ and $q \rightarrow (p \vee r)$ are logically equivalent.
- Show that $p \leftrightarrow q$ and $(p \rightarrow q) \wedge (q \rightarrow p)$ are logically equivalent.
- Show that $p \leftrightarrow q$ and $\neg p \leftrightarrow \neg q$ are logically equivalent.

29. Show that $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$ is a tautology.
30. Show that $(p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$ is a tautology.
31. Show that $(p \rightarrow q) \rightarrow r$ and $p \rightarrow (q \rightarrow r)$ are not logically equivalent.
32. Show that $(p \wedge q) \rightarrow r$ and $(p \rightarrow r) \wedge (q \rightarrow r)$ are not logically equivalent.
33. Show that $(p \rightarrow q) \rightarrow (r \rightarrow s)$ and $(p \rightarrow r) \rightarrow (q \rightarrow s)$ are not logically equivalent.

The **dual** of a compound proposition that contains only the logical operators \vee , \wedge , and \neg is the compound proposition obtained by replacing each \vee by \wedge , each \wedge by \vee , each **T** by **F**, and each **F** by **T**. The dual of s is denoted by s^* .

34. Find the dual of each of these compound propositions.

- $p \vee \neg q$
- $p \wedge (q \vee (r \wedge T))$
- $(p \wedge \neg q) \vee (q \wedge F)$

35. Find the dual of each of these compound propositions.

- $p \wedge \neg q \wedge \neg r$
- $(p \wedge q \wedge r) \vee s$
- $(p \vee F) \wedge (q \vee T)$

36. When does $s^* = s$, where s is a compound proposition?

37. Show that $(s^*)^* = s$ when s is a compound proposition.

38. Show that the logical equivalences in Table 6, except for the double negation law, come in pairs, where each pair contains compound propositions that are duals of each other.

- **39. Why are the duals of two equivalent compound propositions also equivalent, where these compound propositions contain only the operators \wedge , \vee , and \neg ?
40. Find a compound proposition involving the propositional variables p , q , and r that is true when p and q are true and r is false, but is false otherwise. [Hint: Use a conjunction of each propositional variable or its negation.]
41. Find a compound proposition involving the propositional variables p , q , and r that is true when exactly two of p , q , and r are true and is false otherwise. [Hint: Form a disjunction of conjunctions. Include a conjunction for each combination of values for which the compound proposition is true. Each conjunction should include each of the three propositional variables or its negations.]
42. Suppose that a truth table in n propositional variables is specified. Show that a compound proposition with this truth table can be formed by taking the disjunction of conjunctions of the variables or their negations, with one conjunction included for each combination of values for which the compound proposition is true. The resulting compound proposition is said to be in **disjunctive normal form**.

A collection of logical operators is called **functionally complete** if every compound proposition is logically equivalent to a compound proposition involving only these logical operators.

43. Show that \neg , \wedge , and \vee form a functionally complete collection of logical operators. [Hint: Use the fact that every compound proposition is logically equivalent to one in disjunctive normal form, as shown in Exercise 42.]

- *44. Show that \neg and \wedge form a functionally complete collection of logical operators. [Hint: First use a De Morgan law to show that $p \vee q$ is logically equivalent to $\neg(\neg p \wedge \neg q)$.]
- *45. Show that \neg and \vee form a functionally complete collection of logical operators.

The following exercises involve the logical operators *NAND* and *NOR*. The proposition $p \text{ NAND } q$ is true when either p or q , or both, are false; and it is false when both p and q are true. The proposition $p \text{ NOR } q$ is true when both p and q are false, and it is false otherwise. The propositions $p \text{ NAND } q$ and $p \text{ NOR } q$ are denoted by $p \mid q$ and $p \downarrow q$, respectively. (The operators \mid and \downarrow are called the **Sheffer stroke** and the **Peirce arrow** after H. M. Sheffer and C. S. Peirce, respectively.)

46. Construct a truth table for the logical operator *NAND*.
47. Show that $p \mid q$ is logically equivalent to $\neg(p \wedge q)$.
48. Construct a truth table for the logical operator *NOR*.
49. Show that $p \downarrow q$ is logically equivalent to $\neg(p \vee q)$.
50. In this exercise we will show that $\{\downarrow\}$ is a functionally complete collection of logical operators.
- Show that $p \downarrow p$ is logically equivalent to $\neg p$.
 - Show that $(p \downarrow q) \downarrow (p \downarrow q)$ is logically equivalent to $p \vee q$.
 - Conclude from parts (a) and (b), and Exercise 49, that $\{\downarrow\}$ is a functionally complete collection of logical operators.
- *51. Find a compound proposition logically equivalent to $p \rightarrow q$ using only the logical operator \downarrow .
52. Show that $\{\mid\}$ is a functionally complete collection of logical operators.
53. Show that $p \mid q$ and $q \mid p$ are equivalent.
54. Show that $p \mid (q \mid r)$ and $(p \mid q) \mid r$ are not equivalent, so that the logical operator \mid is not associative.
- *55. How many different truth tables of compound propositions are there that involve the propositional variables p and q ?
56. Show that if p , q , and r are compound propositions such that p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent.
57. The following sentence is taken from the specification of a telephone system: “If the directory database is opened, then the monitor is put in a closed state, if the system is not in its initial state.” This specification is hard to understand because it involves two conditional statements. Find an equivalent, easier-to-understand specification that involves disjunctions and negations but not conditional statements.

58. How many of the disjunctions $p \vee \neg q$, $\neg p \vee q$, $q \vee r$, $q \vee \neg r$, and $\neg q \vee \neg r$ can be made simultaneously true by an assignment of truth values to p , q , and r ?
59. How many of the disjunctions $p \vee \neg q \vee s$, $\neg p \vee \neg r \vee s$, $\neg p \vee q \vee \neg s$, $q \vee r \vee \neg s$, $q \vee \neg r \vee \neg s$, $\neg p \vee \neg q \vee \neg s$, $p \vee r \vee s$, and $p \vee r \vee \neg s$ can be made simultaneously true by an assignment of truth values to p , q , r , and s ?
60. Show that the negation of an unsatisfiable compound proposition is a tautology and the negation of a compound proposition that is a tautology is unsatisfiable.
61. Determine whether each of these compound propositions is satisfiable.
- $(p \vee \neg q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q)$
 - $(p \rightarrow q) \wedge (p \rightarrow \neg q) \wedge (\neg p \rightarrow q) \wedge (\neg p \rightarrow \neg q)$
 - $(p \leftrightarrow q) \wedge (\neg p \leftrightarrow q)$
62. Determine whether each of these compound propositions is satisfiable.
- $(p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg s) \wedge (p \vee \neg r \vee \neg s) \wedge (\neg p \vee \neg q \vee \neg s) \wedge (p \vee q \vee \neg s)$
 - $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg s) \wedge (\neg p \vee \neg r \vee \neg s) \wedge (p \vee q \vee \neg r) \wedge (p \vee \neg r \vee \neg s)$
 - $(p \vee q \vee r) \wedge (p \vee \neg q \vee \neg s) \wedge (q \vee \neg r \vee s) \wedge (\neg p \vee r \vee s) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee s) \wedge (\neg p \vee \neg r \vee \neg s)$
63. Show how the solution of a given 4×4 Sudoku puzzle can be found by solving a satisfiability problem.
64. Construct a compound proposition that asserts that every cell of a 9×9 Sudoku puzzle contains at least one number.
65. Explain the steps in the construction of the compound proposition given in the text that asserts that every column of a 9×9 Sudoku puzzle contains every number.
- *66. Explain the steps in the construction of the compound proposition given in the text that asserts that each of the nine 3×3 blocks of a 9×9 Sudoku puzzle contains every number.

1.4 Predicates and Quantifiers

Introduction

Propositional logic, studied in Sections 1.1–1.3, cannot adequately express the meaning of all statements in mathematics and in natural language. For example, suppose that we know that

“Every computer connected to the university network is functioning properly.”

No rules of propositional logic allow us to conclude the truth of the statement

“MATH3 is functioning properly,”

where MATH3 is one of the computers connected to the university network. Likewise, we cannot use the rules of propositional logic to conclude from the statement

“CS2 is under attack by an intruder,”

where CS2 is a computer on the university network, to conclude the truth of

“There is a computer on the university network that is under attack by an intruder.”

In this section we will introduce a more powerful type of logic called **predicate logic**. We will see how predicate logic can be used to express the meaning of a wide range of statements in mathematics and computer science in ways that permit us to reason and explore relationships between objects. To understand predicate logic, we first need to introduce the concept of a predicate. Afterward, we will introduce the notion of quantifiers, which enable us to reason with statements that assert that a certain property holds for all objects of a certain type and with statements that assert the existence of an object with a particular property.

Predicates

Statements involving variables, such as

“ $x > 3$,” “ $x = y + 3$,” “ $x + y = z$,”

and

“computer x is under attack by an intruder,”

and

“computer x is functioning properly,”

are often found in mathematical assertions, in computer programs, and in system specifications. These statements are neither true nor false when the values of the variables are not specified. In this section, we will discuss the ways that propositions can be produced from such statements.

The statement “ x is greater than 3” has two parts. The first part, the variable x , is the subject of the statement. The second part—the **predicate**, “is greater than 3”—refers to a property that the subject of the statement can have. We can denote the statement “ x is greater than 3” by $P(x)$, where P denotes the predicate “is greater than 3” and x is the variable. The statement $P(x)$ is also said to be the value of the **propositional function** P at x . Once a value has been assigned to the variable x , the statement $P(x)$ becomes a proposition and has a truth value. Consider Examples 1 and 2.

EXAMPLE 1 Let $P(x)$ denote the statement “ $x > 3$.” What are the truth values of $P(4)$ and $P(2)$?

Solution: We obtain the statement $P(4)$ by setting $x = 4$ in the statement “ $x > 3$.” Hence, $P(4)$, which is the statement “ $4 > 3$,” is true. However, $P(2)$, which is the statement “ $2 > 3$,” is false. 

EXAMPLE 2 Let $A(x)$ denote the statement “Computer x is under attack by an intruder.” Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders. What are truth values of $A(\text{CS1})$, $A(\text{CS2})$, and $A(\text{MATH1})$?

Solution: We obtain the statement $A(\text{CS1})$ by setting $x = \text{CS1}$ in the statement “Computer x is under attack by an intruder.” Because CS1 is not on the list of computers currently under attack, we conclude that $A(\text{CS1})$ is false. Similarly, because CS2 and MATH1 are on the list of computers under attack, we know that $A(\text{CS2})$ and $A(\text{MATH1})$ are true. 

We can also have statements that involve more than one variable. For instance, consider the statement “ $x = y + 3$.” We can denote this statement by $Q(x, y)$, where x and y are variables and Q is the predicate. When values are assigned to the variables x and y , the statement $Q(x, y)$ has a truth value.

EXAMPLE 3 Let $Q(x, y)$ denote the statement “ $x = y + 3$.” What are the truth values of the propositions $Q(1, 2)$ and $Q(3, 0)$?



Solution: To obtain $Q(1, 2)$, set $x = 1$ and $y = 2$ in the statement $Q(x, y)$. Hence, $Q(1, 2)$ is the statement “ $1 = 2 + 3$,” which is false. The statement $Q(3, 0)$ is the proposition “ $3 = 0 + 3$,” which is true. 



CHARLES SANDERS PEIRCE (1839–1914) Many consider Charles Peirce, born in Cambridge, Massachusetts, to be the most original and versatile American intellect. He made important contributions to an amazing number of disciplines, including mathematics, astronomy, chemistry, geodesy, metrology, engineering, psychology, philology, the history of science, and economics. Peirce was also an inventor, a lifelong student of medicine, a book reviewer, a dramatist and an actor, a short story writer, a phenomenologist, a logician, and a metaphysician. He is noted as the preeminent system-building philosopher competent and productive in logic, mathematics, and a wide range of sciences. He was encouraged by his father, Benjamin Peirce, a professor of mathematics and natural philosophy at Harvard, to pursue a career in science. Instead, he decided to study logic and scientific methodology. Peirce attended Harvard (1855–1859) and received a Harvard master of arts degree (1862) and an advanced degree in chemistry from the Lawrence Scientific School (1863).

In 1861, Peirce became an aide in the U.S. Coast Survey, with the goal of better understanding scientific methodology. His service for the Survey exempted him from military service during the Civil War. While working for the Survey, Peirce did astronomical and geodesic work. He made fundamental contributions to the design of pendulums and to map projections, applying new mathematical developments in the theory of elliptic functions. He was the first person to use the wavelength of light as a unit of measurement. Peirce rose to the position of Assistant for the Survey, a position he held until forced to resign in 1891 when he disagreed with the direction taken by the Survey’s new administration.

While making his living from work in the physical sciences, Peirce developed a hierarchy of sciences, with mathematics at the top rung, in which the methods of one science could be adapted for use by those sciences under it in the hierarchy. During this time, he also founded the American philosophical theory of pragmatism.

The only academic position Peirce ever held was lecturer in logic at Johns Hopkins University in Baltimore (1879–1884). His mathematical work during this time included contributions to logic, set theory, abstract algebra, and the philosophy of mathematics. His work is still relevant today, with recent applications of this work on logic to artificial intelligence. Peirce believed that the study of mathematics could develop the mind’s powers of imagination, abstraction, and generalization. His diverse activities after retiring from the Survey included writing for periodicals, contributing to scholarly dictionaries, translating scientific papers, guest lecturing, and textbook writing. Unfortunately, his income from these pursuits was insufficient to protect him and his second wife from abject poverty. He was supported in his later years by a fund created by his many admirers and administered by the philosopher William James, his lifelong friend. Although Peirce wrote and published voluminously in a vast range of subjects, he left more than 100,000 pages of unpublished manuscripts. Because of the difficulty of studying his unpublished writings, scholars have only recently started to understand some of his varied contributions. A group of people is devoted to making his work available over the Internet to bring a better appreciation of Peirce’s accomplishments to the world.

EXAMPLE 4 Let $A(c, n)$ denote the statement “Computer c is connected to network n ,” where c is a variable representing a computer and n is a variable representing a network. Suppose that the computer MATH1 is connected to network CAMPUS2, but not to network CAMPUS1. What are the values of $A(\text{MATH1}, \text{CAMPUS1})$ and $A(\text{MATH1}, \text{CAMPUS2})$?

Solution: Because MATH1 is not connected to the CAMPUS1 network, we see that $A(\text{MATH1}, \text{CAMPUS1})$ is false. However, because MATH1 is connected to the CAMPUS2 network, we see that $A(\text{MATH1}, \text{CAMPUS2})$ is true. 

Similarly, we can let $R(x, y, z)$ denote the statement “ $x + y = z$.” When values are assigned to the variables x , y , and z , this statement has a truth value.

EXAMPLE 5 What are the truth values of the propositions $R(1, 2, 3)$ and $R(0, 0, 1)$?

Solution: The proposition $R(1, 2, 3)$ is obtained by setting $x = 1$, $y = 2$, and $z = 3$ in the statement $R(x, y, z)$. We see that $R(1, 2, 3)$ is the statement “ $1 + 2 = 3$,” which is true. Also note that $R(0, 0, 1)$, which is the statement “ $0 + 0 = 1$,” is false. 

In general, a statement involving the n variables x_1, x_2, \dots, x_n can be denoted by

$$P(x_1, x_2, \dots, x_n).$$

A statement of the form $P(x_1, x_2, \dots, x_n)$ is the value of the **propositional function** P at the n -tuple (x_1, x_2, \dots, x_n) , and P is also called an **n -place predicate** or a **n -ary predicate**.

Propositional functions occur in computer programs, as Example 6 demonstrates.

EXAMPLE 6 Consider the statement

if $x > 0$ **then** $x := x + 1$.

When this statement is encountered in a program, the value of the variable x at that point in the execution of the program is inserted into $P(x)$, which is “ $x > 0$.” If $P(x)$ is true for this value of x , the assignment statement $x := x + 1$ is executed, so the value of x is increased by 1. If $P(x)$ is false for this value of x , the assignment statement is not executed, so the value of x is not changed. 

PRECONDITIONS AND POSTCONDITIONS Predicates are also used to establish the correctness of computer programs, that is, to show that computer programs always produce the desired output when given valid input. (Note that unless the correctness of a computer program is established, no amount of testing can show that it produces the desired output for all input values, unless every input value is tested.) The statements that describe valid input are known as **preconditions** and the conditions that the output should satisfy when the program has run are known as **postconditions**. As Example 7 illustrates, we use predicates to describe both preconditions and postconditions. We will study this process in greater detail in Section 5.5.

EXAMPLE 7 Consider the following program, designed to interchange the values of two variables x and y .

```
temp := x
x := y
y := temp
```

Find predicates that we can use as the precondition and the postcondition to verify the correctness of this program. Then explain how to use them to verify that for all valid input the program does what is intended.

Solution: For the precondition, we need to express that x and y have particular values before we run the program. So, for this precondition we can use the predicate $P(x, y)$, where $P(x, y)$ is the statement “ $x = a$ and $y = b$,” where a and b are the values of x and y before we run the program. Because we want to verify that the program swaps the values of x and y for all input values, for the postcondition we can use $Q(x, y)$, where $Q(x, y)$ is the statement “ $x = b$ and $y = a$.”

To verify that the program always does what it is supposed to do, suppose that the precondition $P(x, y)$ holds. That is, we suppose that the statement “ $x = a$ and $y = b$ ” is true. This means that $x = a$ and $y = b$. The first step of the program, $\text{temp} := x$, assigns the value of x to the variable temp , so after this step we know that $x = a$, $\text{temp} = a$, and $y = b$. After the second step of the program, $x := y$, we know that $x = b$, $\text{temp} = a$, and $y = b$. Finally, after the third step, we know that $x = b$, $\text{temp} = a$, and $y = a$. Consequently, after this program is run, the postcondition $Q(x, y)$ holds, that is, the statement “ $x = b$ and $y = a$ ” is true. 

Quantifiers



When the variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value. However, there is another important way, called **quantification**, to create a proposition from a propositional function. Quantification expresses the extent to which a predicate is true over a range of elements. In English, the words *all*, *some*, *many*, *none*, and *few* are used in quantifications. We will focus on two types of quantification here: universal quantification, which tells us that a predicate is true for every element under consideration, and existential quantification, which tells us that there is one or more element under consideration for which the predicate is true. The area of logic that deals with predicates and quantifiers is called the **predicate calculus**.



THE UNIVERSAL QUANTIFIER Many mathematical statements assert that a property is true for all values of a variable in a particular domain, called the **domain of discourse** (or the **universe of discourse**), often just referred to as the **domain**. Such a statement is expressed using universal quantification. The universal quantification of $P(x)$ for a particular domain is the proposition that asserts that $P(x)$ is true for all values of x in this domain. Note that the domain specifies the possible values of the variable x . The meaning of the universal quantification of $P(x)$ changes when we change the domain. The domain must always be specified when a universal quantifier is used; without it, the universal quantification of a statement is not defined.

DEFINITION 1

The *universal quantification* of $P(x)$ is the statement

“ $P(x)$ for all values of x in the domain.”

The notation $\forall x P(x)$ denotes the universal quantification of $P(x)$. Here \forall is called the **universal quantifier**. We read $\forall x P(x)$ as “for all $x P(x)$ ” or “for every $x P(x)$.” An element for which $P(x)$ is false is called a **counterexample** of $\forall x P(x)$.

The meaning of the universal quantifier is summarized in the first row of Table 1. We illustrate the use of the universal quantifier in Examples 8–13.

TABLE 1 Quantifiers.

<i>Statement</i>	<i>When True?</i>	<i>When False?</i>
$\forall x P(x)$	$P(x)$ is true for every x .	There is an x for which $P(x)$ is false.
$\exists x P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x .

EXAMPLE 8 Let $P(x)$ be the statement “ $x + 1 > x$.” What is the truth value of the quantification $\forall x P(x)$, where the domain consists of all real numbers?



Solution: Because $P(x)$ is true for all real numbers x , the quantification

$$\forall x P(x)$$

is true.

Remark: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. Note that if the domain is empty, then $\forall x P(x)$ is true for any propositional function $P(x)$ because there are no elements x in the domain for which $P(x)$ is false.

Remember that the truth value of $\forall x P(x)$ depends on the domain!

Besides “for all” and “for every,” universal quantification can be expressed in many other ways, including “all of,” “for each,” “given any,” “for arbitrary,” “for each,” and “for any.”

Remark: It is best to avoid using “for any x ” because it is often ambiguous as to whether “any” means “every” or “some.” In some cases, “any” is unambiguous, such as when it is used in negatives, for example, “there is not any reason to avoid studying.”

A statement $\forall x P(x)$ is false, where $P(x)$ is a propositional function, if and only if $P(x)$ is not always true when x is in the domain. One way to show that $P(x)$ is not always true when x is in the domain is to find a counterexample to the statement $\forall x P(x)$. Note that a single counterexample is all we need to establish that $\forall x P(x)$ is false. Example 9 illustrates how counterexamples are used.

EXAMPLE 9 Let $Q(x)$ be the statement “ $x < 2$.” What is the truth value of the quantification $\forall x Q(x)$, where the domain consists of all real numbers?

Solution: $Q(x)$ is not true for every real number x , because, for instance, $Q(3)$ is false. That is, $x = 3$ is a counterexample for the statement $\forall x Q(x)$. Thus

$$\forall x Q(x)$$

is false.

EXAMPLE 10 Suppose that $P(x)$ is “ $x^2 > 0$.” To show that the statement $\forall x P(x)$ is false where the universe of discourse consists of all integers, we give a counterexample. We see that $x = 0$ is a counterexample because $x^2 = 0$ when $x = 0$, so that x^2 is not greater than 0 when $x = 0$.

Looking for counterexamples to universally quantified statements is an important activity in the study of mathematics, as we will see in subsequent sections of this book.

When all the elements in the domain can be listed—say, x_1, x_2, \dots, x_n —it follows that the universal quantification $\forall x P(x)$ is the same as the conjunction

$$P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n),$$

because this conjunction is true if and only if $P(x_1), P(x_2), \dots, P(x_n)$ are all true.

EXAMPLE 11 What is the truth value of $\forall x P(x)$, where $P(x)$ is the statement “ $x^2 < 10$ ” and the domain consists of the positive integers not exceeding 4?

Solution: The statement $\forall x P(x)$ is the same as the conjunction

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4),$$

because the domain consists of the integers 1, 2, 3, and 4. Because $P(4)$, which is the statement “ $4^2 < 10$,” is false, it follows that $\forall x P(x)$ is false. 

EXAMPLE 12 What does the statement $\forall x N(x)$ mean if $N(x)$ is “Computer x is connected to the network” and the domain consists of all computers on campus?

Solution: The statement $\forall x N(x)$ means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as “Every computer on campus is connected to the network.” 

As we have pointed out, specifying the domain is mandatory when quantifiers are used. The truth value of a quantified statement often depends on which elements are in this domain, as Example 13 shows.

EXAMPLE 13 What is the truth value of $\forall x (x^2 \geq x)$ if the domain consists of all real numbers? What is the truth value of this statement if the domain consists of all integers?

Solution: The universal quantification $\forall x (x^2 \geq x)$, where the domain consists of all real numbers, is false. For example, $(\frac{1}{2})^2 \not\geq \frac{1}{2}$. Note that $x^2 \geq x$ if and only if $x^2 - x = x(x - 1) \geq 0$. Consequently, $x^2 \geq x$ if and only if $x \leq 0$ or $x \geq 1$. It follows that $\forall x (x^2 \geq x)$ is false if the domain consists of all real numbers (because the inequality is false for all real numbers x with $0 < x < 1$). However, if the domain consists of the integers, $\forall x (x^2 \geq x)$ is true, because there are no integers x with $0 < x < 1$. 

THE EXISTENTIAL QUANTIFIER Many mathematical statements assert that there is an element with a certain property. Such statements are expressed using existential quantification. With existential quantification, we form a proposition that is true if and only if $P(x)$ is true for at least one value of x in the domain.

DEFINITION 2

The *existential quantification* of $P(x)$ is the proposition

“There exists an element x in the domain such that $P(x)$.”

We use the notation $\exists x P(x)$ for the existential quantification of $P(x)$. Here \exists is called the *existential quantifier*.

A domain must always be specified when a statement $\exists x P(x)$ is used. Furthermore, the meaning of $\exists x P(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists x P(x)$ has no meaning.

Besides the phrase “there exists,” we can also express existential quantification in many other ways, such as by using the words “for some,” “for at least one,” or “there is.” The existential quantification $\exists x P(x)$ is read as

“There is an x such that $P(x)$,”
“There is at least one x such that $P(x)$,”

or

“For some $x P(x)$.”

The meaning of the existential quantifier is summarized in the second row of Table 1. We illustrate the use of the existential quantifier in Examples 14–16.

EXAMPLE 14 Let $P(x)$ denote the statement “ $x > 3$.” What is the truth value of the quantification $\exists x P(x)$, where the domain consists of all real numbers?



Solution: Because “ $x > 3$ ” is sometimes true—for instance, when $x = 4$ —the existential quantification of $P(x)$, which is $\exists x P(x)$, is true.

Observe that the statement $\exists x P(x)$ is false if and only if there is no element x in the domain for which $P(x)$ is true. That is, $\exists x P(x)$ is false if and only if $P(x)$ is false for every element of the domain. We illustrate this observation in Example 15.

EXAMPLE 15 Let $Q(x)$ denote the statement “ $x = x + 1$.” What is the truth value of the quantification $\exists x Q(x)$, where the domain consists of all real numbers?

Solution: Because $Q(x)$ is false for every real number x , the existential quantification of $Q(x)$, which is $\exists x Q(x)$, is false.

Remember that the truth value of $\exists x P(x)$ depends on the domain!

Remark: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. If the domain is empty, then $\exists x Q(x)$ is false whenever $Q(x)$ is a propositional function because when the domain is empty, there can be no element x in the domain for which $Q(x)$ is true.

When all elements in the domain can be listed—say, x_1, x_2, \dots, x_n —the existential quantification $\exists x P(x)$ is the same as the disjunction

$$P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n),$$

because this disjunction is true if and only if at least one of $P(x_1), P(x_2), \dots, P(x_n)$ is true.

EXAMPLE 16 What is the truth value of $\exists x P(x)$, where $P(x)$ is the statement “ $x^2 > 10$ ” and the universe of discourse consists of the positive integers not exceeding 4?

Solution: Because the domain is $\{1, 2, 3, 4\}$, the proposition $\exists x P(x)$ is the same as the disjunction

$$P(1) \vee P(2) \vee P(3) \vee P(4).$$

Because $P(4)$, which is the statement “ $4^2 > 10$,” is true, it follows that $\exists x P(x)$ is true.

It is sometimes helpful to think in terms of looping and searching when determining the truth value of a quantification. Suppose that there are n objects in the domain for the variable x . To determine whether $\forall x P(x)$ is true, we can loop through all n values of x to see whether $P(x)$ is always true. If we encounter a value x for which $P(x)$ is false, then we have shown that $\forall x P(x)$ is false. Otherwise, $\forall x P(x)$ is true. To see whether $\exists x P(x)$ is true, we loop through the n values of x searching for a value for which $P(x)$ is true. If we find one, then $\exists x P(x)$ is true. If we never find such an x , then we have determined that $\exists x P(x)$ is false. (Note that this searching procedure does not apply if there are infinitely many values in the domain. However, it is still a useful way of thinking about the truth values of quantifications.)

THE UNIQUENESS QUANTIFIER We have now introduced universal and existential quantifiers. These are the most important quantifiers in mathematics and computer science. However, there is no limitation on the number of different quantifiers we can define, such as “there are exactly two,” “there are no more than three,” “there are at least 100,” and so on. Of these other quantifiers, the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$ or \exists_1 . The notation $\exists!x P(x)$ [or $\exists_1 x P(x)$] states “There exists a unique x such that $P(x)$ is true.” (Other phrases for uniqueness quantification include “there is exactly one” and “there is one and only one.”) For instance, $\exists!x(x - 1 = 0)$, where the domain is the set of real numbers, states that there is a unique real number x such that $x - 1 = 0$. This is a true statement, as $x = 1$ is the unique real number such that $x - 1 = 0$. Observe that we can use quantifiers and propositional logic to express uniqueness (see Exercise 52 in Section 1.5), so the uniqueness quantifier can be avoided. Generally, it is best to stick with existential and universal quantifiers so that rules of inference for these quantifiers can be used.

Quantifiers with Restricted Domains

An abbreviated notation is often used to restrict the domain of a quantifier. In this notation, a condition a variable must satisfy is included after the quantifier. This is illustrated in Example 17. We will also describe other forms of this notation involving set membership in Section 2.1.

EXAMPLE 17 What do the statements $\forall x < 0 (x^2 > 0)$, $\forall y \neq 0 (y^3 \neq 0)$, and $\exists z > 0 (z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

Solution: The statement $\forall x < 0 (x^2 > 0)$ states that for every real number x with $x < 0$, $x^2 > 0$. That is, it states “The square of a negative real number is positive.” This statement is the same as $\forall x(x < 0 \rightarrow x^2 > 0)$.

The statement $\forall y \neq 0 (y^3 \neq 0)$ states that for every real number y with $y \neq 0$, we have $y^3 \neq 0$. That is, it states “The cube of every nonzero real number is nonzero.” Note that this statement is equivalent to $\forall y(y \neq 0 \rightarrow y^3 \neq 0)$.

Finally, the statement $\exists z > 0 (z^2 = 2)$ states that there exists a real number z with $z > 0$ such that $z^2 = 2$. That is, it states “There is a positive square root of 2.” This statement is equivalent to $\exists z(z > 0 \wedge z^2 = 2)$. 

Note that the restriction of a universal quantification is the same as the universal quantification of a conditional statement. For instance, $\forall x < 0 (x^2 > 0)$ is another way of expressing $\forall x(x < 0 \rightarrow x^2 > 0)$. On the other hand, the restriction of an existential quantification is the same as the existential quantification of a conjunction. For instance, $\exists z > 0 (z^2 = 2)$ is another way of expressing $\exists z(z > 0 \wedge z^2 = 2)$.

Precedence of Quantifiers

The quantifiers \forall and \exists have higher precedence than all logical operators from propositional calculus. For example, $\forall x P(x) \vee Q(x)$ is the disjunction of $\forall x P(x)$ and $Q(x)$. In other words, it means $(\forall x P(x)) \vee Q(x)$ rather than $\forall x(P(x) \vee Q(x))$.

Binding Variables

When a quantifier is used on the variable x , we say that this occurrence of the variable is **bound**. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be **free**. All the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition. This can be done using a combination of universal quantifiers, existential quantifiers, and value assignments.

The part of a logical expression to which a quantifier is applied is called the **scope** of this quantifier. Consequently, a variable is free if it is outside the scope of all quantifiers in the formula that specify this variable.

EXAMPLE 18 In the statement $\exists x(x + y = 1)$, the variable x is bound by the existential quantification $\exists x$, but the variable y is free because it is not bound by a quantifier and no value is assigned to this variable. This illustrates that in the statement $\exists x(x + y = 1)$, x is bound, but y is free.

In the statement $\exists x(P(x) \wedge Q(x)) \vee \forall x R(x)$, all variables are bound. The scope of the first quantifier, $\exists x$, is the expression $P(x) \wedge Q(x)$ because $\exists x$ is applied only to $P(x) \wedge Q(x)$, and not to the rest of the statement. Similarly, the scope of the second quantifier, $\forall x$, is the expression $R(x)$. That is, the existential quantifier binds the variable x in $P(x) \wedge Q(x)$ and the universal quantifier $\forall x$ binds the variable x in $R(x)$. Observe that we could have written our statement using two different variables x and y , as $\exists x(P(x) \wedge Q(x)) \vee \forall y R(y)$, because the scopes of the two quantifiers do not overlap. The reader should be aware that in common usage, the same letter is often used to represent variables bound by different quantifiers with scopes that do not overlap. 

Logical Equivalences Involving Quantifiers

In Section 1.3 we introduced the notion of logical equivalences of compound propositions. We can extend this notion to expressions involving predicates and quantifiers.

DEFINITION 3

Statements involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value no matter which predicates are substituted into these statements and which domain of discourse is used for the variables in these propositional functions. We use the notation $S \equiv T$ to indicate that two statements S and T involving predicates and quantifiers are logically equivalent.

Example 19 illustrates how to show that two statements involving predicates and quantifiers are logically equivalent.

EXAMPLE 19 Show that $\forall x(P(x) \wedge Q(x))$ and $\forall x P(x) \wedge \forall x Q(x)$ are logically equivalent (where the same domain is used throughout). This logical equivalence shows that we can distribute a universal quantifier over a conjunction. Furthermore, we can also distribute an existential quantifier over a disjunction. However, we cannot distribute a universal quantifier over a disjunction, nor can we distribute an existential quantifier over a conjunction. (See Exercises 50 and 51.)

Solution: To show that these statements are logically equivalent, we must show that they always take the same truth value, no matter what the predicates P and Q are, and no matter which domain of discourse is used. Suppose we have particular predicates P and Q , with a common domain. We can show that $\forall x(P(x) \wedge Q(x))$ and $\forall x P(x) \wedge \forall x Q(x)$ are logically equivalent by doing two things. First, we show that if $\forall x(P(x) \wedge Q(x))$ is true, then $\forall x P(x) \wedge \forall x Q(x)$ is true. Second, we show that if $\forall x P(x) \wedge \forall x Q(x)$ is true, then $\forall x(P(x) \wedge Q(x))$ is true.

So, suppose that $\forall x(P(x) \wedge Q(x))$ is true. This means that if a is in the domain, then $P(a) \wedge Q(a)$ is true. Hence, $P(a)$ is true and $Q(a)$ is true. Because $P(a)$ is true and $Q(a)$ is true for every element in the domain, we can conclude that $\forall x P(x)$ and $\forall x Q(x)$ are both true. This means that $\forall x P(x) \wedge \forall x Q(x)$ is true.

Next, suppose that $\forall x P(x) \wedge \forall x Q(x)$ is true. It follows that $\forall x P(x)$ is true and $\forall x Q(x)$ is true. Hence, if a is in the domain, then $P(a)$ is true and $Q(a)$ is true [because $P(x)$ and $Q(x)$ are both true for all elements in the domain, there is no conflict using the same value of a here].

It follows that for all a , $P(a) \wedge Q(a)$ is true. It follows that $\forall x(P(x) \wedge Q(x))$ is true. We can now conclude that

$$\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x).$$



Negating Quantified Expressions

We will often want to consider the negation of a quantified expression. For instance, consider the negation of the statement

“Every student in your class has taken a course in calculus.”

This statement is a universal quantification, namely,

$$\forall x P(x),$$



where $P(x)$ is the statement “ x has taken a course in calculus” and the domain consists of the students in your class. The negation of this statement is “It is not the case that every student in your class has taken a course in calculus.” This is equivalent to “There is a student in your class who has not taken a course in calculus.” And this is simply the existential quantification of the negation of the original propositional function, namely,

$$\exists x \neg P(x).$$

This example illustrates the following logical equivalence:

$$\neg \forall x P(x) \equiv \exists x \neg P(x).$$

To show that $\neg \forall x P(x)$ and $\exists x \neg P(x)$ are logically equivalent no matter what the propositional function $P(x)$ is and what the domain is, first note that $\neg \forall x P(x)$ is true if and only if $\forall x P(x)$ is false. Next, note that $\forall x P(x)$ is false if and only if there is an element x in the domain for which $P(x)$ is false. This holds if and only if there is an element x in the domain for which $\neg P(x)$ is true. Finally, note that there is an element x in the domain for which $\neg P(x)$ is true if and only if $\exists x \neg P(x)$ is true. Putting these steps together, we can conclude that $\neg \forall x P(x)$ is true if and only if $\exists x \neg P(x)$ is true. It follows that $\neg \forall x P(x)$ and $\exists x \neg P(x)$ are logically equivalent.

Suppose we wish to negate an existential quantification. For instance, consider the proposition “There is a student in this class who has taken a course in calculus.” This is the existential quantification

$$\exists x Q(x),$$

where $Q(x)$ is the statement “ x has taken a course in calculus.” The negation of this statement is the proposition “It is not the case that there is a student in this class who has taken a course in calculus.” This is equivalent to “Every student in this class has not taken calculus,” which is just the universal quantification of the negation of the original propositional function, or, phrased in the language of quantifiers,

$$\forall x \neg Q(x).$$

This example illustrates the equivalence

$$\neg \exists x Q(x) \equiv \forall x \neg Q(x).$$

To show that $\neg \exists x Q(x)$ and $\forall x \neg Q(x)$ are logically equivalent no matter what $Q(x)$ is and what the domain is, first note that $\neg \exists x Q(x)$ is true if and only if $\exists x Q(x)$ is false. This is true if and

TABLE 2 De Morgan's Laws for Quantifiers.

<i>Negation</i>	<i>Equivalent Statement</i>	<i>When Is Negation True?</i>	<i>When False?</i>
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x , $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x .

only if no x exists in the domain for which $Q(x)$ is true. Next, note that no x exists in the domain for which $Q(x)$ is true if and only if $Q(x)$ is false for every x in the domain. Finally, note that $Q(x)$ is false for every x in the domain if and only if $\neg Q(x)$ is true for all x in the domain, which holds if and only if $\forall x \neg Q(x)$ is true. Putting these steps together, we see that $\neg\exists x Q(x)$ is true if and only if $\forall x \neg Q(x)$ is true. We conclude that $\neg\exists x Q(x)$ and $\forall x \neg Q(x)$ are logically equivalent.

The rules for negations for quantifiers are called **De Morgan's laws for quantifiers**. These rules are summarized in Table 2.

Remark: When the domain of a predicate $P(x)$ consists of n elements, where n is a positive integer greater than one, the rules for negating quantified statements are exactly the same as De Morgan's laws discussed in Section 1.3. This is why these rules are called De Morgan's laws for quantifiers. When the domain has n elements x_1, x_2, \dots, x_n , it follows that $\neg\forall x P(x)$ is the same as $\neg(P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n))$, which is equivalent to $\neg P(x_1) \vee \neg P(x_2) \vee \dots \vee \neg P(x_n)$ by De Morgan's laws, and this is the same as $\exists x \neg P(x)$. Similarly, $\neg\exists x P(x)$ is the same as $\neg(P(x_1) \vee P(x_2) \vee \dots \vee P(x_n))$, which by De Morgan's laws is equivalent to $\neg P(x_1) \wedge \neg P(x_2) \wedge \dots \wedge \neg P(x_n)$, and this is the same as $\forall x \neg P(x)$.

We illustrate the negation of quantified statements in Examples 20 and 21.

EXAMPLE 20 What are the negations of the statements “There is an honest politician” and “All Americans eat cheeseburgers”?

Solution: Let $H(x)$ denote “ x is honest.” Then the statement “There is an honest politician” is represented by $\exists x H(x)$, where the domain consists of all politicians. The negation of this statement is $\neg\exists x H(x)$, which is equivalent to $\forall x \neg H(x)$. This negation can be expressed as “Every politician is dishonest.” (Note: In English, the statement “All politicians are not honest” is ambiguous. In common usage, this statement often means “Not all politicians are honest.” Consequently, we do not use this statement to express this negation.)

Let $C(x)$ denote “ x eats cheeseburgers.” Then the statement “All Americans eat cheeseburgers” is represented by $\forall x C(x)$, where the domain consists of all Americans. The negation of this statement is $\neg\forall x C(x)$, which is equivalent to $\exists x \neg C(x)$. This negation can be expressed in several different ways, including “Some American does not eat cheeseburgers” and “There is an American who does not eat cheeseburgers.”

EXAMPLE 21 What are the negations of the statements $\forall x(x^2 > x)$ and $\exists x(x^2 = 2)$?

Solution: The negation of $\forall x(x^2 > x)$ is the statement $\neg\forall x(x^2 > x)$, which is equivalent to $\exists x \neg(x^2 > x)$. This can be rewritten as $\exists x(x^2 \leq x)$. The negation of $\exists x(x^2 = 2)$ is the statement $\neg\exists x(x^2 = 2)$, which is equivalent to $\forall x \neg(x^2 = 2)$. This can be rewritten as $\forall x(x^2 \neq 2)$. The truth values of these statements depend on the domain.

We use De Morgan's laws for quantifiers in Example 22.



EXAMPLE 22 Show that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Solution: By De Morgan's law for universal quantifiers, we know that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(\neg(P(x) \rightarrow Q(x)))$ are logically equivalent. By the fifth logical equivalence in Table 7 in Section 1.3, we know that $\neg(P(x) \rightarrow Q(x))$ and $P(x) \wedge \neg Q(x)$ are logically equivalent for every x . Because we can substitute one logically equivalent expression for another in a logical equivalence, it follows that $\neg\forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent. 

Translating from English into Logical Expressions

Translating sentences in English (or other natural languages) into logical expressions is a crucial task in mathematics, logic programming, artificial intelligence, software engineering, and many other disciplines. We began studying this topic in Section 1.1, where we used propositions to express sentences in logical expressions. In that discussion, we purposely avoided sentences whose translations required predicates and quantifiers. Translating from English to logical expressions becomes even more complex when quantifiers are needed. Furthermore, there can be many ways to translate a particular sentence. (As a consequence, there is no "cookbook" approach that can be followed step by step.) We will use some examples to illustrate how to translate sentences from English into logical expressions. The goal in this translation is to produce simple and useful logical expressions. In this section, we restrict ourselves to sentences that can be translated into logical expressions using a single quantifier; in the next section, we will look at more complicated sentences that require multiple quantifiers.

EXAMPLE 23 Express the statement "Every student in this class has studied calculus" using predicates and quantifiers.

Solution: First, we rewrite the statement so that we can clearly identify the appropriate quantifiers to use. Doing so, we obtain:

"For every student in this class, that student has studied calculus."



Next, we introduce a variable x so that our statement becomes

"For every student x in this class, x has studied calculus."

Continuing, we introduce $C(x)$, which is the statement " x has studied calculus." Consequently, if the domain for x consists of the students in the class, we can translate our statement as $\forall x C(x)$.

However, there are other correct approaches; different domains of discourse and other predicates can be used. The approach we select depends on the subsequent reasoning we want to carry out. For example, we may be interested in a wider group of people than only those in this class. If we change the domain to consist of all people, we will need to express our statement as

"For every person x , if person x is a student in this class then x has studied calculus."



If $S(x)$ represents the statement that person x is in this class, we see that our statement can be expressed as $\forall x(S(x) \rightarrow C(x))$. [Caution! Our statement *cannot* be expressed as $\forall x(S(x) \wedge C(x))$ because this statement says that all people are students in this class and have studied calculus!]

Finally, when we are interested in the background of people in subjects besides calculus, we may prefer to use the two-variable quantifier $Q(x, y)$ for the statement "student x has studied subject y ." Then we would replace $C(x)$ by $Q(x, \text{calculus})$ in both approaches to obtain $\forall x Q(x, \text{calculus})$ or $\forall x(S(x) \rightarrow Q(x, \text{calculus}))$. 

In Example 23 we displayed different approaches for expressing the same statement using predicates and quantifiers. However, we should always adopt the simplest approach that is adequate for use in subsequent reasoning.

EXAMPLE 24 Express the statements “Some student in this class has visited Mexico” and “Every student in this class has visited either Canada or Mexico” using predicates and quantifiers.

Solution: The statement “Some student in this class has visited Mexico” means that

“There is a student in this class with the property that the student has visited Mexico.”

We can introduce a variable x , so that our statement becomes

“There is a student x in this class having the property that x has visited Mexico.”

We introduce $M(x)$, which is the statement “ x has visited Mexico.” If the domain for x consists of the students in this class, we can translate this first statement as $\exists x M(x)$.

However, if we are interested in people other than those in this class, we look at the statement a little differently. Our statement can be expressed as

“There is a person x having the properties that x is a student in this class and x has visited Mexico.”

In this case, the domain for the variable x consists of all people. We introduce $S(x)$ to represent “ x is a student in this class.” Our solution becomes $\exists x(S(x) \wedge M(x))$ because the statement is that there is a person x who is a student in this class and who has visited Mexico. [Caution! Our statement cannot be expressed as $\exists x(S(x) \rightarrow M(x))$, which is true when there is someone not in the class because, in that case, for such a person x , $S(x) \rightarrow M(x)$ becomes either $\mathbf{F} \rightarrow \mathbf{T}$ or $\mathbf{F} \rightarrow \mathbf{F}$, both of which are true.]

Similarly, the second statement can be expressed as

“For every x in this class, x has the property that x has visited Mexico or x has visited Canada.”

(Note that we are assuming the inclusive, rather than the exclusive, or here.) We let $C(x)$ be “ x has visited Canada.” Following our earlier reasoning, we see that if the domain for x consists of the students in this class, this second statement can be expressed as $\forall x(C(x) \vee M(x))$. However, if the domain for x consists of all people, our statement can be expressed as

“For every person x , if x is a student in this class, then x has visited Mexico or x has visited Canada.”

In this case, the statement can be expressed as $\forall x(S(x) \rightarrow (C(x) \vee M(x)))$.

Instead of using $M(x)$ and $C(x)$ to represent that x has visited Mexico and x has visited Canada, respectively, we could use a two-place predicate $V(x, y)$ to represent “ x has visited country y .” In this case, $V(x, \text{Mexico})$ and $V(x, \text{Canada})$ would have the same meaning as $M(x)$ and $C(x)$ and could replace them in our answers. If we are working with many statements that involve people visiting different countries, we might prefer to use this two-variable approach. Otherwise, for simplicity, we would stick with the one-variable predicates $M(x)$ and $C(x)$. 

Using Quantifiers in System Specifications

In Section 1.2 we used propositions to represent system specifications. However, many system specifications involve predicates and quantifications. This is illustrated in Example 25.

EXAMPLE 25

Use predicates and quantifiers to express the system specifications “Every mail message larger than one megabyte will be compressed” and “If a user is active, at least one network link will be available.”



Remember the rules of precedence for quantifiers and logical connectives!

Solution: Let $S(m, y)$ be “Mail message m is larger than y megabytes,” where the variable x has the domain of all mail messages and the variable y is a positive real number, and let $C(m)$ denote “Mail message m will be compressed.” Then the specification “Every mail message larger than one megabyte will be compressed” can be represented as $\forall m(S(m, 1) \rightarrow C(m))$.

Let $A(u)$ represent “User u is active,” where the variable u has the domain of all users, let $S(n, x)$ denote “Network link n is in state x ,” where n has the domain of all network links and x has the domain of all possible states for a network link. Then the specification “If a user is active, at least one network link will be available” can be represented by $\exists u A(u) \rightarrow \exists n S(n, \text{available})$.

Examples from Lewis Carroll

Lewis Carroll (really C. L. Dodgson writing under a pseudonym), the author of *Alice in Wonderland*, is also the author of several works on symbolic logic. His books contain many examples of reasoning using quantifiers. Examples 26 and 27 come from his book *Symbolic Logic*; other examples from that book are given in the exercises at the end of this section. These examples illustrate how quantifiers are used to express various types of statements.

EXAMPLE 26

Consider these statements. The first two are called *premises* and the third is called the *conclusion*. The entire set is called an *argument*.

- “All lions are fierce.”
- “Some lions do not drink coffee.”
- “Some fierce creatures do not drink coffee.”

(In Section 1.6 we will discuss the issue of determining whether the conclusion is a valid consequence of the premises. In this example, it is.) Let $P(x)$, $Q(x)$, and $R(x)$ be the statements “ x is a lion,” “ x is fierce,” and “ x drinks coffee,” respectively. Assuming that the domain consists of all creatures, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, and $R(x)$.



CHARLES LUTWIDGE DODGSON (1832–1898) We know Charles Dodgson as Lewis Carroll—the pseudonym he used in his literary works. Dodgson, the son of a clergyman, was the third of 11 children, all of whom stuttered. He was uncomfortable in the company of adults and is said to have spoken without stuttering only to young girls, many of whom he entertained, corresponded with, and photographed (sometimes in poses that today would be considered inappropriate). Although attracted to young girls, he was extremely puritanical and religious. His friendship with the three young daughters of Dean Liddell led to his writing *Alice in Wonderland*, which brought him money and fame.

Dodgson graduated from Oxford in 1854 and obtained his master of arts degree in 1857. He was appointed lecturer in mathematics at Christ Church College, Oxford, in 1855. He was ordained in the Church of England in 1861 but never practiced his ministry. His writings published under this real name include articles and books on geometry, determinants, and the mathematics of tournaments and elections. (He also used the pseudonym Lewis Carroll for his many works on recreational logic.)

Solution: We can express these statements as:

$$\begin{aligned}\forall x(P(x) \rightarrow Q(x)). \\ \exists x(P(x) \wedge \neg R(x)). \\ \exists x(Q(x) \wedge \neg R(x)).\end{aligned}$$

Notice that the second statement cannot be written as $\exists x(P(x) \rightarrow \neg R(x))$. The reason is that $P(x) \rightarrow \neg R(x)$ is true whenever x is not a lion, so that $\exists x(P(x) \rightarrow \neg R(x))$ is true as long as there is at least one creature that is not a lion, even if every lion drinks coffee. Similarly, the third statement cannot be written as

$$\exists x(Q(x) \rightarrow \neg R(x)).$$



EXAMPLE 27 Consider these statements, of which the first three are premises and the fourth is a valid conclusion.

- “All hummingbirds are richly colored.”
- “No large birds live on honey.”
- “Birds that do not live on honey are dull in color.”
- “Hummingbirds are small.”

Let $P(x)$, $Q(x)$, $R(x)$, and $S(x)$ be the statements “ x is a hummingbird,” “ x is large,” “ x lives on honey,” and “ x is richly colored,” respectively. Assuming that the domain consists of all birds, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.

Solution: We can express the statements in the argument as

$$\begin{aligned}\forall x(P(x) \rightarrow S(x)). \\ \neg \exists x(Q(x) \wedge R(x)). \\ \forall x(\neg R(x) \rightarrow \neg S(x)). \\ \forall x(P(x) \rightarrow \neg Q(x)).\end{aligned}$$

(Note we have assumed that “small” is the same as “not large” and that “dull in color” is the same as “not richly colored.” To show that the fourth statement is a valid conclusion of the first three, we need to use rules of inference that will be discussed in Section 1.6.)



Logic Programming



An important type of programming language is designed to reason using the rules of predicate logic. Prolog (from *Programming in Logic*), developed in the 1970s by computer scientists working in the area of artificial intelligence, is an example of such a language. Prolog programs include a set of declarations consisting of two types of statements, **Prolog facts** and **Prolog rules**. Prolog facts define predicates by specifying the elements that satisfy these predicates. Prolog rules are used to define new predicates using those already defined by Prolog facts. Example 28 illustrates these notions.

EXAMPLE 28 Consider a Prolog program given facts telling it the instructor of each class and in which classes students are enrolled. The program uses these facts to answer queries concerning the professors who teach particular students. Such a program could use the predicates *instructor(p, c)* and

enrolled(s, c) to represent that professor p is the instructor of course c and that student s is enrolled in course c , respectively. For example, the Prolog facts in such a program might include:

```
instructor(chan,math273)
instructor(patel,ee222)
instructor(grossman,cs301)
enrolled(kevin,math273)
enrolled(juana,ee222)
enrolled(juana,cs301)
enrolled(kiko,math273)
enrolled(kiko,cs301)
```

(Lowercase letters have been used for entries because Prolog considers names beginning with an uppercase letter to be variables.)

A new predicate *teaches(p, s)*, representing that professor p teaches student s , can be defined using the Prolog rule

```
teaches(P,S) :- instructor(P,C), enrolled(S,C)
```

which means that *teaches(p, s)* is true if there exists a class c such that professor p is the instructor of class c and student s is enrolled in class c . (Note that a comma is used to represent a conjunction of predicates in Prolog. Similarly, a semicolon is used to represent a disjunction of predicates.)

Prolog answers queries using the facts and rules it is given. For example, using the facts and rules listed, the query

```
?enrolled(kevin,math273)
```

produces the response

```
yes
```

because the fact *enrolled(kevin, math273)* was provided as input. The query

```
?enrolled(X,math273)
```

produces the response

```
kevin
kiko
```

To produce this response, Prolog determines all possible values of X for which *enrolled(X, math273)* has been included as a Prolog fact. Similarly, to find all the professors who are instructors in classes being taken by Juana, we use the query

```
?teaches(X,juana)
```

This query returns

```
patel
grossman
```



Exercises

1. Let $P(x)$ denote the statement " $x \leq 4$." What are these truth values?
 - $P(0)$
 - $P(4)$
 - $P(6)$
2. Let $P(x)$ be the statement "the word x contains the letter a ." What are these truth values?
 - $P(\text{orange})$
 - $P(\text{lemon})$
 - $P(\text{true})$
 - $P(\text{false})$
3. Let $Q(x, y)$ denote the statement " x is the capital of y ." What are these truth values?
 - $Q(\text{Denver, Colorado})$
 - $Q(\text{Detroit, Michigan})$
 - $Q(\text{Massachusetts, Boston})$
 - $Q(\text{New York, New York})$
4. State the value of x after the statement **if** $P(x)$ **then** $x := 1$ is executed, where $P(x)$ is the statement " $x > 1$," if the value of x when this statement is reached is
 - $x = 0$.
 - $x = 1$.
 - $x = 2$.
5. Let $P(x)$ be the statement " x spends more than five hours every weekday in class," where the domain for x consists of all students. Express each of these quantifications in English.
 - $\exists x P(x)$
 - $\forall x P(x)$
 - $\exists x \neg P(x)$
 - $\forall x \neg P(x)$
6. Let $N(x)$ be the statement " x has visited North Dakota," where the domain consists of the students in your school. Express each of these quantifications in English.
 - $\exists x N(x)$
 - $\forall x N(x)$
 - $\neg \exists x N(x)$
 - $\exists x \neg N(x)$
 - $\neg \forall x N(x)$
 - $\forall x \neg N(x)$
7. Translate these statements into English, where $C(x)$ is " x is a comedian" and $F(x)$ is " x is funny" and the domain consists of all people.
 - $\forall x(C(x) \rightarrow F(x))$
 - $\forall x(C(x) \wedge F(x))$
 - $\exists x(C(x) \rightarrow F(x))$
 - $\exists x(C(x) \wedge F(x))$
8. Translate these statements into English, where $R(x)$ is " x is a rabbit" and $H(x)$ is " x hops" and the domain consists of all animals.
 - $\forall x(R(x) \rightarrow H(x))$
 - $\forall x(R(x) \wedge H(x))$
 - $\exists x(R(x) \rightarrow H(x))$
 - $\exists x(R(x) \wedge H(x))$
9. Let $P(x)$ be the statement " x can speak Russian" and let $Q(x)$ be the statement " x knows the computer language C++." Express each of these sentences in terms of $P(x)$, $Q(x)$, quantifiers, and logical connectives. The domain for quantifiers consists of all students at your school.
 - There is a student at your school who can speak Russian and who knows C++.
 - There is a student at your school who can speak Russian but who doesn't know C++.
 - Every student at your school either can speak Russian or knows C++.
 - No student at your school can speak Russian or knows C++.
10. Let $C(x)$ be the statement " x has a cat," let $D(x)$ be the statement " x has a dog," and let $F(x)$ be the statement " x has a ferret." Express each of these statements in terms of $C(x)$, $D(x)$, $F(x)$, quantifiers, and logical connectives. Let the domain consist of all students in your class.
 - A student in your class has a cat, a dog, and a ferret.
 - All students in your class have a cat, a dog, or a ferret.
 - Some student in your class has a cat and a ferret, but not a dog.
 - No student in your class has a cat, a dog, and a ferret.
 - For each of the three animals, cats, dogs, and ferrets, there is a student in your class who has this animal as a pet.
11. Let $P(x)$ be the statement " $x = x^2$." If the domain consists of the integers, what are these truth values?
 - $P(0)$
 - $P(1)$
 - $P(2)$
 - $P(-1)$
 - $\exists x P(x)$
 - $\forall x P(x)$
12. Let $Q(x)$ be the statement " $x + 1 > 2x$." If the domain consists of all integers, what are these truth values?
 - $Q(0)$
 - $Q(-1)$
 - $Q(1)$
 - $\exists x Q(x)$
 - $\forall x Q(x)$
 - $\exists x \neg Q(x)$
 - $\forall x \neg Q(x)$
13. Determine the truth value of each of these statements if the domain consists of all integers.
 - $\forall n(n + 1 > n)$
 - $\exists n(2n = 3n)$
 - $\exists n(n = -n)$
 - $\forall n(3n \leq 4n)$
14. Determine the truth value of each of these statements if the domain consists of all real numbers.
 - $\exists x(x^3 = -1)$
 - $\exists x(x^4 < x^2)$
 - $\forall x((-x)^2 = x^2)$
 - $\forall x(2x > x)$
15. Determine the truth value of each of these statements if the domain for all variables consists of all integers.
 - $\forall n(n^2 \geq 0)$
 - $\exists n(n^2 = 2)$
 - $\forall n(n^2 \geq n)$
 - $\exists n(n^2 < 0)$
16. Determine the truth value of each of these statements if the domain of each variable consists of all real numbers.
 - $\exists x(x^2 = 2)$
 - $\exists x(x^2 = -1)$
 - $\forall x(x^2 + 2 \geq 1)$
 - $\forall x(x^2 \neq x)$
17. Suppose that the domain of the propositional function $P(x)$ consists of the integers 0, 1, 2, 3, and 4. Write out each of these propositions using disjunctions, conjunctions, and negations.
 - $\exists x P(x)$
 - $\forall x P(x)$
 - $\exists x \neg P(x)$
 - $\forall x \neg P(x)$
 - $\neg \exists x P(x)$
 - $\neg \forall x P(x)$
18. Suppose that the domain of the propositional function $P(x)$ consists of the integers $-2, -1, 0, 1$, and 2 . Write out each of these propositions using disjunctions, conjunctions, and negations.
 - $\exists x P(x)$
 - $\forall x P(x)$
 - $\exists x \neg P(x)$
 - $\forall x \neg P(x)$
 - $\neg \exists x P(x)$
 - $\neg \forall x P(x)$

- 19.** Suppose that the domain of the propositional function $P(x)$ consists of the integers 1, 2, 3, 4, and 5. Express these statements without using quantifiers, instead using only negations, disjunctions, and conjunctions.

- a) $\exists x P(x)$
- b) $\forall x P(x)$
- c) $\neg \exists x P(x)$
- d) $\neg \forall x P(x)$
- e) $\forall x((x \neq 3) \rightarrow P(x)) \vee \exists x \neg P(x)$

- 20.** Suppose that the domain of the propositional function $P(x)$ consists of $-5, -3, -1, 1, 3$, and 5. Express these statements without using quantifiers, instead using only negations, disjunctions, and conjunctions.

- a) $\exists x P(x)$
- b) $\forall x P(x)$
- c) $\forall x((x \neq 1) \rightarrow P(x))$
- d) $\exists x((x \geq 0) \wedge P(x))$
- e) $\exists x(\neg P(x)) \wedge \forall x((x < 0) \rightarrow P(x))$

- 21.** For each of these statements find a domain for which the statement is true and a domain for which the statement is false.

- a) Everyone is studying discrete mathematics.
- b) Everyone is older than 21 years.
- c) Every two people have the same mother.
- d) No two different people have the same grandmother.

- 22.** For each of these statements find a domain for which the statement is true and a domain for which the statement is false.

- a) Everyone speaks Hindi.
- b) There is someone older than 21 years.
- c) Every two people have the same first name.
- d) Someone knows more than two other people.

- 23.** Translate in two ways each of these statements into logical expressions using predicates, quantifiers, and logical connectives. First, let the domain consist of the students in your class and second, let it consist of all people.

- a) Someone in your class can speak Hindi.
- b) Everyone in your class is friendly.
- c) There is a person in your class who was not born in California.
- d) A student in your class has been in a movie.
- e) No student in your class has taken a course in logic programming.

- 24.** Translate in two ways each of these statements into logical expressions using predicates, quantifiers, and logical connectives. First, let the domain consist of the students in your class and second, let it consist of all people.

- a) Everyone in your class has a cellular phone.
- b) Somebody in your class has seen a foreign movie.
- c) There is a person in your class who cannot swim.
- d) All students in your class can solve quadratic equations.
- e) Some student in your class does not want to be rich.

- 25.** Translate each of these statements into logical expressions using predicates, quantifiers, and logical connectives.

- a) No one is perfect.
- b) Not everyone is perfect.
- c) All your friends are perfect.
- d) At least one of your friends is perfect.

- e) Everyone is your friend and is perfect.
- f) Not everybody is your friend or someone is not perfect.

- 26.** Translate each of these statements into logical expressions in three different ways by varying the domain and by using predicates with one and with two variables.

- a) Someone in your school has visited Uzbekistan.
- b) Everyone in your class has studied calculus and C++.
- c) No one in your school owns both a bicycle and a motorcycle.
- d) There is a person in your school who is not happy.
- e) Everyone in your school was born in the twentieth century.

- 27.** Translate each of these statements into logical expressions in three different ways by varying the domain and by using predicates with one and with two variables.

- a) A student in your school has lived in Vietnam.
- b) There is a student in your school who cannot speak Hindi.
- c) A student in your school knows Java, Prolog, and C++.
- d) Everyone in your class enjoys Thai food.
- e) Someone in your class does not play hockey.

- 28.** Translate each of these statements into logical expressions using predicates, quantifiers, and logical connectives.

- a) Something is not in the correct place.
- b) All tools are in the correct place and are in excellent condition.
- c) Everything is in the correct place and in excellent condition.
- d) Nothing is in the correct place and is in excellent condition.
- e) One of your tools is not in the correct place, but it is in excellent condition.

- 29.** Express each of these statements using logical operators, predicates, and quantifiers.

- a) Some propositions are tautologies.
- b) The negation of a contradiction is a tautology.
- c) The disjunction of two contingencies can be a tautology.
- d) The conjunction of two tautologies is a tautology.

- 30.** Suppose the domain of the propositional function $P(x, y)$ consists of pairs x and y , where x is 1, 2, or 3 and y is 1, 2, or 3. Write out these propositions using disjunctions and conjunctions.

- a) $\exists x P(x, 3)$
- b) $\forall y P(1, y)$
- c) $\exists y \neg P(2, y)$
- d) $\forall x \neg P(x, 2)$

- 31.** Suppose that the domain of $Q(x, y, z)$ consists of triples x, y, z , where $x = 0, 1$, or 2, $y = 0$ or 1, and $z = 0$ or 1. Write out these propositions using disjunctions and conjunctions.

- a) $\forall y Q(0, y, 0)$
- b) $\exists x Q(x, 1, 1)$
- c) $\exists z \neg Q(0, 0, z)$
- d) $\exists x \neg Q(x, 0, 1)$

- 32.** Express each of these statements using quantifiers. Then form the negation of the statement so that no negation is to the left of a quantifier. Next, express the negation in simple English. (Do not simply use the phrase “It is not the case that.”)
- All dogs have fleas.
 - There is a horse that can add.
 - Every koala can climb.
 - No monkey can speak French.
 - There exists a pig that can swim and catch fish.
- 33.** Express each of these statements using quantifiers. Then form the negation of the statement, so that no negation is to the left of a quantifier. Next, express the negation in simple English. (Do not simply use the phrase “It is not the case that.”)
- Some old dogs can learn new tricks.
 - No rabbit knows calculus.
 - Every bird can fly.
 - There is no dog that can talk.
 - There is no one in this class who knows French and Russian.
- 34.** Express the negation of these propositions using quantifiers, and then express the negation in English.
- Some drivers do not obey the speed limit.
 - All Swedish movies are serious.
 - No one can keep a secret.
 - There is someone in this class who does not have a good attitude.
- 35.** Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all integers.
- $\forall x(x^2 \geq x)$
 - $\forall x(x > 0 \vee x < 0)$
 - $\forall x(x = 1)$
- 36.** Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all real numbers.
- $\forall x(x^2 \neq x)$
 - $\forall x(x^2 \neq 2)$
 - $\forall x(|x| > 0)$
- 37.** Express each of these statements using predicates and quantifiers.
- A passenger on an airline qualifies as an elite flyer if the passenger flies more than 25,000 miles in a year or takes more than 25 flights during that year.
 - A man qualifies for the marathon if his best previous time is less than 3 hours and a woman qualifies for the marathon if her best previous time is less than 3.5 hours.
 - A student must take at least 60 course hours, or at least 45 course hours and write a master’s thesis, and receive a grade no lower than a B in all required courses, to receive a master’s degree.
 - There is a student who has taken more than 21 credit hours in a semester and received all A’s.
- Exercises 38–42 deal with the translation between system specification and logical expressions involving quantifiers.
- 38.** Translate these system specifications into English where the predicate $S(x, y)$ is “ x is in state y ” and where the domain for x and y consists of all systems and all possible states, respectively.
- $\exists x S(x, \text{open})$
 - $\forall x(S(x, \text{malfunctioning}) \vee S(x, \text{diagnostic}))$
 - $\exists x S(x, \text{open}) \vee \exists x S(x, \text{diagnostic})$
 - $\exists x \neg S(x, \text{available})$
 - $\forall x \neg S(x, \text{working})$
- 39.** Translate these specifications into English where $F(p)$ is “Printer p is out of service,” $B(p)$ is “Printer p is busy,” $L(j)$ is “Print job j is lost,” and $Q(j)$ is “Print job j is queued.”
- $\exists p(F(p) \wedge B(p)) \rightarrow \exists j L(j)$
 - $\forall p B(p) \rightarrow \exists j Q(j)$
 - $\exists j(Q(j) \wedge L(j)) \rightarrow \exists p F(p)$
 - $(\forall p B(p) \wedge \forall j Q(j)) \rightarrow \exists j L(j)$
- 40.** Express each of these system specifications using predicates, quantifiers, and logical connectives.
- When there is less than 30 megabytes free on the hard disk, a warning message is sent to all users.
 - No directories in the file system can be opened and no files can be closed when system errors have been detected.
 - The file system cannot be backed up if there is a user currently logged on.
 - Video on demand can be delivered when there are at least 8 megabytes of memory available and the connection speed is at least 56 kilobits per second.
- 41.** Express each of these system specifications using predicates, quantifiers, and logical connectives.
- At least one mail message, among the nonempty set of messages, can be saved if there is a disk with more than 10 kilobytes of free space.
 - Whenever there is an active alert, all queued messages are transmitted.
 - The diagnostic monitor tracks the status of all systems except the main console.
 - Each participant on the conference call whom the host of the call did not put on a special list was billed.
- 42.** Express each of these system specifications using predicates, quantifiers, and logical connectives.
- Every user has access to an electronic mailbox.
 - The system mailbox can be accessed by everyone in the group if the file system is locked.
 - The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.
 - At least one router is functioning normally if the throughput is between 100 kbps and 500 kbps and the proxy server is not in diagnostic mode.

43. Determine whether $\forall x(P(x) \rightarrow Q(x))$ and $\forall x P(x) \rightarrow \forall x Q(x)$ are logically equivalent. Justify your answer.

44. Determine whether $\forall x(P(x) \leftrightarrow Q(x))$ and $\forall x P(x) \leftrightarrow \forall x Q(x)$ are logically equivalent. Justify your answer.

45. Show that $\exists x(P(x) \vee Q(x))$ and $\exists x P(x) \vee \exists x Q(x)$ are logically equivalent.

Exercises 46–49 establish rules for **null quantification** that we can use when a quantified variable does not appear in part of a statement.

46. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.

- a) $(\forall x P(x)) \vee A \equiv \forall x(P(x) \vee A)$
- b) $(\exists x P(x)) \vee A \equiv \exists x(P(x) \vee A)$

47. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.

- a) $(\forall x P(x)) \wedge A \equiv \forall x(P(x) \wedge A)$
- b) $(\exists x P(x)) \wedge A \equiv \exists x(P(x) \wedge A)$

48. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.

- a) $\forall x(A \rightarrow P(x)) \equiv A \rightarrow \forall x P(x)$
- b) $\exists x(A \rightarrow P(x)) \equiv A \rightarrow \exists x P(x)$

49. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.

- a) $\forall x(P(x) \rightarrow A) \equiv \exists x P(x) \rightarrow A$
- b) $\exists x(P(x) \rightarrow A) \equiv \forall x P(x) \rightarrow A$

50. Show that $\forall x P(x) \vee \forall x Q(x)$ and $\forall x(P(x) \vee Q(x))$ are not logically equivalent.

51. Show that $\exists x P(x) \wedge \exists x Q(x)$ and $\exists x(P(x) \wedge Q(x))$ are not logically equivalent.

52. As mentioned in the text, the notation $\exists!x P(x)$ denotes “There exists a unique x such that $P(x)$ is true.”

If the domain consists of all integers, what are the truth values of these statements?

- a) $\exists!x(x > 1)$
- b) $\exists!x(x^2 = 1)$
- c) $\exists!x(x + 3 = 2x)$
- d) $\exists!x(x = x + 1)$

53. What are the truth values of these statements?

- a) $\exists!x P(x) \rightarrow \exists x P(x)$
- b) $\forall x P(x) \rightarrow \exists!x P(x)$
- c) $\exists!x \neg P(x) \rightarrow \neg \forall x P(x)$

54. Write out $\exists!x P(x)$, where the domain consists of the integers 1, 2, and 3, in terms of negations, conjunctions, and disjunctions.

55. Given the Prolog facts in Example 28, what would Prolog return given these queries?

- a) `?instructor(chan,math273)`
- b) `?instructor(patel,cs301)`
- c) `?enrolled(X,cs301)`
- d) `?enrolled(kiko,Y)`
- e) `?teaches(grossman,Y)`

56. Given the Prolog facts in Example 28, what would Prolog return when given these queries?

- a) `?enrolled(kevin,ee222)`
- b) `?enrolled(kiko,math273)`
- c) `?instructor(grossman,X)`
- d) `?instructor(X,cs301)`
- e) `?teaches(X,kevin)`

57. Suppose that Prolog facts are used to define the predicates $mother(M, Y)$ and $father(F, X)$, which represent that M is the mother of Y and F is the father of X , respectively. Give a Prolog rule to define the predicate $sibling(X, Y)$, which represents that X and Y are siblings (that is, have the same mother and the same father).

58. Suppose that Prolog facts are used to define the predicates $mother(M, Y)$ and $father(F, X)$, which represent that M is the mother of Y and F is the father of X , respectively. Give a Prolog rule to define the predicate $grandfather(X, Y)$, which represents that X is the grandfather of Y . [Hint: You can write a disjunction in Prolog either by using a semicolon to separate predicates or by putting these predicates on separate lines.]

Exercises 59–62 are based on questions found in the book *Symbolic Logic* by Lewis Carroll.

59. Let $P(x)$, $Q(x)$, and $R(x)$ be the statements “ x is a professor,” “ x is ignorant,” and “ x is vain,” respectively. Express each of these statements using quantifiers; logical connectives; and $P(x)$, $Q(x)$, and $R(x)$, where the domain consists of all people.

- a) No professors are ignorant.
- b) All ignorant people are vain.
- c) No professors are vain.
- d) Does (c) follow from (a) and (b)?

60. Let $P(x)$, $Q(x)$, and $R(x)$ be the statements “ x is a clear explanation,” “ x is satisfactory,” and “ x is an excuse,” respectively. Suppose that the domain for x consists of all English text. Express each of these statements using quantifiers, logical connectives, and $P(x)$, $Q(x)$, and $R(x)$.

- a) All clear explanations are satisfactory.
- b) Some excuses are unsatisfactory.
- c) Some excuses are not clear explanations.

*d) Does (c) follow from (a) and (b)?

61. Let $P(x)$, $Q(x)$, $R(x)$, and $S(x)$ be the statements “ x is a baby,” “ x is logical,” “ x is able to manage a crocodile,” and “ x is despised,” respectively. Suppose that the domain consists of all people. Express each of these statements using quantifiers; logical connectives; and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.

- a) Babies are illogical.
- b) Nobody is despised who can manage a crocodile.
- c) Illogical persons are despised.
- d) Babies cannot manage crocodiles.

*e) Does (d) follow from (a), (b), and (c)? If not, is there a correct conclusion?

- 62.** Let $P(x)$, $Q(x)$, $R(x)$, and $S(x)$ be the statements “ x is a duck,” “ x is one of my poultry,” “ x is an officer,” and “ x is willing to waltz,” respectively. Express each of these statements using quantifiers; logical connectives; and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.
- a) No ducks are willing to waltz.

- b) No officers ever decline to waltz.
 c) All my poultry are ducks.
 d) My poultry are not officers.
 *e) Does (d) follow from (a), (b), and (c)? If not, is there a correct conclusion?

1.5 Nested Quantifiers

Introduction

In Section 1.4 we defined the existential and universal quantifiers and showed how they can be used to represent mathematical statements. We also explained how they can be used to translate English sentences into logical expressions. However, in Section 1.4 we avoided **nested quantifiers**, where one quantifier is within the scope of another, such as

$$\forall x \exists y (x + y = 0).$$

Note that everything within the scope of a quantifier can be thought of as a propositional function. For example,

$$\forall x \exists y (x + y = 0)$$

is the same thing as $\forall x Q(x)$, where $Q(x)$ is $\exists y P(x, y)$, where $P(x, y)$ is $x + y = 0$.

Nested quantifiers commonly occur in mathematics and computer science. Although nested quantifiers can sometimes be difficult to understand, the rules we have already studied in Section 1.4 can help us use them. In this section we will gain experience working with nested quantifiers. We will see how to use nested quantifiers to express mathematical statements such as “The sum of two positive integers is always positive.” We will show how nested quantifiers can be used to translate English sentences such as “Everyone has exactly one best friend” into logical statements. Moreover, we will gain experience working with the negations of statements involving nested quantifiers.

Understanding Statements Involving Nested Quantifiers

To understand statements involving nested quantifiers, we need to unravel what the quantifiers and predicates that appear mean. This is illustrated in Examples 1 and 2.

EXAMPLE 1 Assume that the domain for the variables x and y consists of all real numbers. The statement

$$\forall x \forall y (x + y = y + x)$$



says that $x + y = y + x$ for all real numbers x and y . This is the commutative law for addition of real numbers. Likewise, the statement

$$\forall x \exists y (x + y = 0)$$

says that for every real number x there is a real number y such that $x + y = 0$. This states that every real number has an additive inverse. Similarly, the statement

$$\forall x \forall y \forall z (x + (y + z) = (x + y) + z)$$

is the associative law for addition of real numbers.



EXAMPLE 2 Translate into English the statement

$$\forall x \forall y ((x > 0) \wedge (y < 0) \rightarrow (xy < 0)),$$

where the domain for both variables consists of all real numbers.

Solution: This statement says that for every real number x and for every real number y , if $x > 0$ and $y < 0$, then $xy < 0$. That is, this statement says that for real numbers x and y , if x is positive and y is negative, then xy is negative. This can be stated more succinctly as “The product of a positive real number and a negative real number is always a negative real number.” 

THINKING OF QUANTIFICATION AS LOOPS In working with quantifications of more than one variable, it is sometimes helpful to think in terms of nested loops. (Of course, if there are infinitely many elements in the domain of some variable, we cannot actually loop through all values. Nevertheless, this way of thinking is helpful in understanding nested quantifiers.) For example, to see whether $\forall x \forall y P(x, y)$ is true, we loop through the values for x , and for each x we loop through the values for y . If we find that $P(x, y)$ is true for all values for x and y , we have determined that $\forall x \forall y P(x, y)$ is true. If we ever hit a value x for which we hit a value y for which $P(x, y)$ is false, we have shown that $\forall x \forall y P(x, y)$ is false.

Similarly, to determine whether $\forall x \exists y P(x, y)$ is true, we loop through the values for x . For each x we loop through the values for y until we find a y for which $P(x, y)$ is true. If for every x we hit such a y , then $\forall x \exists y P(x, y)$ is true; if for some x we never hit such a y , then $\forall x \exists y P(x, y)$ is false.

To see whether $\exists x \forall y P(x, y)$ is true, we loop through the values for x until we find an x for which $P(x, y)$ is always true when we loop through all values for y . Once we find such an x , we know that $\exists x \forall y P(x, y)$ is true. If we never hit such an x , then we know that $\exists x \forall y P(x, y)$ is false.

Finally, to see whether $\exists x \exists y P(x, y)$ is true, we loop through the values for x , where for each x we loop through the values for y until we hit an x for which we hit a y for which $P(x, y)$ is true. The statement $\exists x \exists y P(x, y)$ is false only if we never hit an x for which we hit a y such that $P(x, y)$ is true.

The Order of Quantifiers

Many mathematical statements involve multiple quantifications of propositional functions involving more than one variable. It is important to note that the order of the quantifiers is important, unless all the quantifiers are universal quantifiers or all are existential quantifiers.

These remarks are illustrated by Examples 3–5.

EXAMPLE 3 Let $P(x, y)$ be the statement “ $x + y = y + x$.” What are the truth values of the quantifications $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ where the domain for all variables consists of all real numbers?

Solution: The quantification

$$\forall x \forall y P(x, y)$$



denotes the proposition

“For all real numbers x , for all real numbers y , $x + y = y + x$.”

Because $P(x, y)$ is true for all real numbers x and y (it is the commutative law for addition, which is an axiom for the real numbers—see Appendix 1), the proposition $\forall x \forall y P(x, y)$ is true. Note that the statement $\forall y \forall x P(x, y)$ says “For all real numbers y , for all real numbers x , $x + y = y + x$.” This has the same meaning as the statement “For all real numbers x , for all real numbers y , $x + y = y + x$.” That is, $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ have the same meaning,

and both are true. This illustrates the principle that the order of nested universal quantifiers in a statement without other quantifiers can be changed without changing the meaning of the quantified statement. 

EXAMPLE 4 Let $Q(x, y)$ denote “ $x + y = 0$.” What are the truth values of the quantifications $\exists y \forall x Q(x, y)$ and $\forall x \exists y Q(x, y)$, where the domain for all variables consists of all real numbers?

Solution: The quantification

$$\exists y \forall x Q(x, y)$$

denotes the proposition

“There is a real number y such that for every real number x , $Q(x, y)$.”

No matter what value of y is chosen, there is only one value of x for which $x + y = 0$. Because there is no real number y such that $x + y = 0$ for all real numbers x , the statement $\exists y \forall x Q(x, y)$ is false.

The quantification

$$\forall x \exists y Q(x, y)$$

denotes the proposition

“For every real number x there is a real number y such that $Q(x, y)$.”

Given a real number x , there is a real number y such that $x + y = 0$; namely, $y = -x$. Hence, the statement $\forall x \exists y Q(x, y)$ is true. 

Example 4 illustrates that the order in which quantifiers appear makes a difference. The statements $\exists y \forall x P(x, y)$ and $\forall x \exists y P(x, y)$ are not logically equivalent. The statement $\exists y \forall x P(x, y)$ is true if and only if there is a y that makes $P(x, y)$ true for every x . So, for this statement to be true, there must be a particular value of y for which $P(x, y)$ is true regardless of the choice of x . On the other hand, $\forall x \exists y P(x, y)$ is true if and only if for every value of x there is a value of y for which $P(x, y)$ is true. So, for this statement to be true, no matter which x you choose, there must be a value of y (possibly depending on the x you choose) for which $P(x, y)$ is true. In other words, in the second case, y can depend on x , whereas in the first case, y is a constant independent of x .

From these observations, it follows that if $\exists y \forall x P(x, y)$ is true, then $\forall x \exists y P(x, y)$ must also be true. However, if $\forall x \exists y P(x, y)$ is true, it is not necessary for $\exists y \forall x P(x, y)$ to be true. (See Supplementary Exercises 30 and 31.)

Table 1 summarizes the meanings of the different possible quantifications involving two variables.

Quantifications of more than two variables are also common, as Example 5 illustrates.

EXAMPLE 5 Let $Q(x, y, z)$ be the statement “ $x + y = z$.” What are the truth values of the statements $\forall x \forall y \exists z Q(x, y, z)$ and $\exists z \forall x \forall y Q(x, y, z)$, where the domain of all variables consists of all real numbers?

Solution: Suppose that x and y are assigned values. Then, there exists a real number z such that $x + y = z$. Consequently, the quantification

$$\forall x \forall y \exists z Q(x, y, z),$$

which is the statement

“For all real numbers x and for all real numbers y there is a real number z such that $x + y = z$,”

TABLE 1 Quantifications of Two Variables.

<i>Statement</i>	<i>When True?</i>	<i>When False?</i>
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y .	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y .
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y .	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y .

is true. The order of the quantification here is important, because the quantification

$$\exists z \forall x \forall y Q(x, y, z),$$

which is the statement

“There is a real number z such that for all real numbers x and for all real numbers y it is true that $x + y = z$,”

is false, because there is no value of z that satisfies the equation $x + y = z$ for all values of x and y . 

Translating Mathematical Statements into Statements Involving Nested Quantifiers

Mathematical statements expressed in English can be translated into logical expressions, as Examples 6–8 show.

EXAMPLE 6 Translate the statement “The sum of two positive integers is always positive” into a logical expression.

Solution: To translate this statement into a logical expression, we first rewrite it so that the implied quantifiers and a domain are shown: “For every two integers, if these integers are both positive, then the sum of these integers is positive.” Next, we introduce the variables x and y to obtain “For all positive integers x and y , $x + y$ is positive.” Consequently, we can express this statement as



$$\forall x \forall y ((x > 0) \wedge (y > 0) \rightarrow (x + y > 0)),$$

where the domain for both variables consists of all integers. Note that we could also translate this using the positive integers as the domain. Then the statement “The sum of two positive integers is always positive” becomes “For every two positive integers, the sum of these integers is positive. We can express this as

$$\forall x \forall y (x + y > 0),$$

where the domain for both variables consists of all positive integers. 

EXAMPLE 7 Translate the statement “Every real number except zero has a multiplicative inverse.” (A **multiplicative inverse** of a real number x is a real number y such that $xy = 1$.)

Solution: We first rewrite this as “For every real number x except zero, x has a multiplicative inverse.” We can rewrite this as “For every real number x , if $x \neq 0$, then there exists a real number y such that $xy = 1$.” This can be rewritten as

$$\forall x((x \neq 0) \rightarrow \exists y(xy = 1)).$$

One example that you may be familiar with is the concept of limit, which is important in calculus.

EXAMPLE 8 (*Requires calculus*) Use quantifiers to express the definition of the limit of a real-valued function $f(x)$ of a real variable x at a point a in its domain.

Solution: Recall that the definition of the statement

$$\lim_{x \rightarrow a} f(x) = L$$

is: For every real number $\epsilon > 0$ there exists a real number $\delta > 0$ such that $|f(x) - L| < \epsilon$ whenever $0 < |x - a| < \delta$. This definition of a limit can be phrased in terms of quantifiers by

$$\forall \epsilon \exists \delta \forall x(0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon),$$

where the domain for the variables δ and ϵ consists of all positive real numbers and for x consists of all real numbers.

This definition can also be expressed as

$$\forall \epsilon > 0 \exists \delta > 0 \forall x(0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon)$$

when the domain for the variables ϵ and δ consists of all real numbers, rather than just the positive real numbers. [Here, restricted quantifiers have been used. Recall that $\forall x > 0 P(x)$ means that for all x with $x > 0$, $P(x)$ is true.]

Translating from Nested Quantifiers into English

Expressions with nested quantifiers expressing statements in English can be quite complicated. The first step in translating such an expression is to write out what the quantifiers and predicates in the expression mean. The next step is to express this meaning in a simpler sentence. This process is illustrated in Examples 9 and 10.

EXAMPLE 9 Translate the statement

$$\forall x(C(x) \vee \exists y(C(y) \wedge F(x, y)))$$

into English, where $C(x)$ is “ x has a computer,” $F(x, y)$ is “ x and y are friends,” and the domain for both x and y consists of all students in your school.

Solution: The statement says that for every student x in your school, x has a computer or there is a student y such that y has a computer and x and y are friends. In other words, every student in your school has a computer or has a friend who has a computer.

EXAMPLE 10 Translate the statement

$$\exists x \forall y \forall z((F(x, y) \wedge F(x, z)) \wedge (y \neq z)) \rightarrow \neg F(y, z))$$

into English, where $F(a, b)$ means a and b are friends and the domain for x , y , and z consists of all students in your school.

Solution: We first examine the expression $(F(x, y) \wedge F(x, z) \wedge (y \neq z)) \rightarrow \neg F(y, z)$. This expression says that if students x and y are friends, and students x and z are friends, and furthermore, if y and z are not the same student, then y and z are not friends. It follows that the original statement, which is triply quantified, says that there is a student x such that for all students y and all students z other than y , if x and y are friends and x and z are friends, then y and z are not friends. In other words, there is a student none of whose friends are also friends with each other. 

Translating English Sentences into Logical Expressions

In Section 1.4 we showed how quantifiers can be used to translate sentences into logical expressions. However, we avoided sentences whose translation into logical expressions required the use of nested quantifiers. We now address the translation of such sentences.

EXAMPLE 11 Express the statement “If a person is female and is a parent, then this person is someone’s mother” as a logical expression involving predicates, quantifiers with a domain consisting of all people, and logical connectives.

Solution: The statement “If a person is female and is a parent, then this person is someone’s mother” can be expressed as “For every person x , if person x is female and person x is a parent, then there exists a person y such that person x is the mother of person y .” We introduce the propositional functions $F(x)$ to represent “ x is female,” $P(x)$ to represent “ x is a parent,” and $M(x, y)$ to represent “ x is the mother of y .” The original statement can be represented as

$$\forall x((F(x) \wedge P(x)) \rightarrow \exists y M(x, y)).$$

Using the null quantification rule in part (b) of Exercise 47 in Section 1.4, we can move $\exists y$ to the left so that it appears just after $\forall x$, because y does not appear in $F(x) \wedge P(x)$. We obtain the logically equivalent expression

$$\forall x \exists y ((F(x) \wedge P(x)) \rightarrow M(x, y)).$$


EXAMPLE 12 Express the statement “Everyone has exactly one best friend” as a logical expression involving predicates, quantifiers with a domain consisting of all people, and logical connectives.

Solution: The statement “Everyone has exactly one best friend” can be expressed as “For every person x , person x has exactly one best friend.” Introducing the universal quantifier, we see that this statement is the same as “ $\forall x$ (person x has exactly one best friend),” where the domain consists of all people.

To say that x has exactly one best friend means that there is a person y who is the best friend of x , and furthermore, that for every person z , if person z is not person y , then z is not the best friend of x . When we introduce the predicate $B(x, y)$ to be the statement “ y is the best friend of x ,” the statement that x has exactly one best friend can be represented as

$$\exists y(B(x, y) \wedge \forall z((z \neq y) \rightarrow \neg B(x, z))).$$

Consequently, our original statement can be expressed as

$$\forall x \exists y(B(x, y) \wedge \forall z((z \neq y) \rightarrow \neg B(x, z))).$$

[Note that we can write this statement as $\forall x \exists! y B(x, y)$, where $\exists!$ is the “uniqueness quantifier” defined in Section 1.4.] 

EXAMPLE 13 Use quantifiers to express the statement “There is a woman who has taken a flight on every airline in the world.”

Solution: Let $P(w, f)$ be “ w has taken f ” and $Q(f, a)$ be “ f is a flight on a .” We can express the statement as

$$\exists w \forall a \exists f (P(w, f) \wedge Q(f, a)),$$

where the domains of discourse for w , f , and a consist of all the women in the world, all airplane flights, and all airlines, respectively.

The statement could also be expressed as

$$\exists w \forall a \exists f R(w, f, a),$$

where $R(w, f, a)$ is “ w has taken f on a .” Although this is more compact, it somewhat obscures the relationships among the variables. Consequently, the first solution is usually preferable. 

Negating Nested Quantifiers



Statements involving nested quantifiers can be negated by successively applying the rules for negating statements involving a single quantifier. This is illustrated in Examples 14–16.

EXAMPLE 14 Express the negation of the statement $\forall x \exists y (xy = 1)$ so that no negation precedes a quantifier.



Solution: By successively applying De Morgan’s laws for quantifiers in Table 2 of Section 1.4, we can move the negation in $\neg \forall x \exists y (xy = 1)$ inside all the quantifiers. We find that $\neg \forall x \exists y (xy = 1)$ is equivalent to $\exists x \neg \exists y (xy = 1)$, which is equivalent to $\exists x \forall y \neg (xy = 1)$. Because $\neg (xy = 1)$ can be expressed more simply as $xy \neq 1$, we conclude that our negated statement can be expressed as $\exists x \forall y (xy \neq 1)$. 

EXAMPLE 15 Use quantifiers to express the statement that “There does not exist a woman who has taken a flight on every airline in the world.”

Solution: This statement is the negation of the statement “There is a woman who has taken a flight on every airline in the world” from Example 13. By Example 13, our statement can be expressed as $\neg \exists w \forall a \exists f (P(w, f) \wedge Q(f, a))$, where $P(w, f)$ is “ w has taken f ” and $Q(f, a)$ is “ f is a flight on a .” By successively applying De Morgan’s laws for quantifiers in Table 2 of Section 1.4 to move the negation inside successive quantifiers and by applying De Morgan’s law for negating a conjunction in the last step, we find that our statement is equivalent to each of this sequence of statements:

$$\begin{aligned} \forall w \neg \forall a \exists f (P(w, f) \wedge Q(f, a)) &\equiv \forall w \exists a \neg \exists f (P(w, f) \wedge Q(f, a)) \\ &\equiv \forall w \exists a \forall f \neg (P(w, f) \wedge Q(f, a)) \\ &\equiv \forall w \exists a \forall f (\neg P(w, f) \vee \neg Q(f, a)). \end{aligned}$$

This last statement states “For every woman there is an airline such that for all flights, this woman has not taken that flight or that flight is not on this airline.” 

EXAMPLE 16 (*Requires calculus*) Use quantifiers and predicates to express the fact that $\lim_{x \rightarrow a} f(x)$ does not exist where $f(x)$ is a real-valued function of a real variable x and a belongs to the domain of f .

Solution: To say that $\lim_{x \rightarrow a} f(x)$ does not exist means that for all real numbers L , $\lim_{x \rightarrow a} f(x) \neq L$. By using Example 8, the statement $\lim_{x \rightarrow a} f(x) \neq L$ can be expressed as

$$\neg \forall \epsilon > 0 \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon).$$

Successively applying the rules for negating quantified expressions, we construct this sequence of equivalent statements

$$\begin{aligned} &\neg \forall \epsilon > 0 \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \neg \exists \delta > 0 \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \forall \delta > 0 \neg \forall x (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \forall \delta > 0 \exists x \neg (0 < |x - a| < \delta \rightarrow |f(x) - L| < \epsilon) \\ &\equiv \exists \epsilon > 0 \forall \delta > 0 \exists x (0 < |x - a| < \delta \wedge |f(x) - L| \geq \epsilon). \end{aligned}$$

In the last step we used the equivalence $\neg(p \rightarrow q) \equiv p \wedge \neg q$, which follows from the fifth equivalence in Table 7 of Section 1.3.

Because the statement “ $\lim_{x \rightarrow a} f(x)$ does not exist” means for all real numbers L , $\lim_{x \rightarrow a} f(x) \neq L$, this can be expressed as

$$\forall L \exists \epsilon > 0 \forall \delta > 0 \exists x (0 < |x - a| < \delta \wedge |f(x) - L| \geq \epsilon).$$

This last statement says that for every real number L there is a real number $\epsilon > 0$ such that for every real number $\delta > 0$, there exists a real number x such that $0 < |x - a| < \delta$ and $|f(x) - L| \geq \epsilon$. 

Exercises

1. Translate these statements into English, where the domain for each variable consists of all real numbers.
 - a) $\forall x \exists y (x < y)$
 - b) $\forall x \forall y (((x \geq 0) \wedge (y \geq 0)) \rightarrow (xy \geq 0))$
 - c) $\forall x \forall y \exists z (xy = z)$
2. Translate these statements into English, where the domain for each variable consists of all real numbers.
 - a) $\exists x \forall y (xy = y)$
 - b) $\forall x \forall y (((x \geq 0) \wedge (y < 0)) \rightarrow (x - y > 0))$
 - c) $\forall x \forall y \exists z (x = y + z)$
3. Let $Q(x, y)$ be the statement “ x has sent an e-mail message to y ,” where the domain for both x and y consists of all students in your class. Express each of these quantifications in English.

a) $\exists x \exists y Q(x, y)$	b) $\exists x \forall y Q(x, y)$
c) $\forall x \exists y Q(x, y)$	d) $\exists y \forall x Q(x, y)$
e) $\forall y \exists x Q(x, y)$	f) $\forall x \forall y Q(x, y)$
4. Let $P(x, y)$ be the statement “Student x has taken class y ,” where the domain for x consists of all students in your class and for y consists of all computer science courses at your school. Express each of these quantifications in English.

a) $\exists x \exists y P(x, y)$	b) $\exists x \forall y P(x, y)$
c) $\forall x \exists y P(x, y)$	d) $\exists y \forall x P(x, y)$
e) $\forall y \exists x P(x, y)$	f) $\forall x \forall y P(x, y)$
5. Let $W(x, y)$ mean that student x has visited website y , where the domain for x consists of all students in your school and the domain for y consists of all websites. Express each of these statements by a simple English sentence.

a) $W(\text{Sarah Smith}, \text{www.att.com})$	b) $\exists x W(x, \text{www.imdb.org})$
c) $\exists y W(\text{José Orez}, y)$	d) $\exists y (W(\text{Ashok Puri}, y) \wedge W(\text{Cindy Yoon}, y))$
e) $\exists y \forall z (y \neq (\text{David Belcher}) \wedge (W(\text{David Belcher}, z) \rightarrow W(y, z)))$	f) $\exists x \exists y \forall z ((x \neq y) \wedge (W(x, z) \leftrightarrow W(y, z)))$
6. Let $C(x, y)$ mean that student x is enrolled in class y , where the domain for x consists of all students in your school and the domain for y consists of all classes being

- given at your school. Express each of these statements by a simple English sentence.
- $C(\text{Randy Goldberg}, \text{CS 252})$
 - $\exists x C(x, \text{Math 695})$
 - $\exists y C(\text{Carol Sitea}, y)$
 - $\exists x(C(x, \text{Math 222}) \wedge C(x, \text{CS 252}))$
 - $\exists x \exists y \forall z((x \neq y) \wedge (C(x, z) \rightarrow C(y, z)))$
 - $\exists x \exists y \forall z((x \neq y) \wedge (C(x, z) \leftrightarrow C(y, z)))$
7. Let $T(x, y)$ mean that student x likes cuisine y , where the domain for x consists of all students at your school and the domain for y consists of all cuisines. Express each of these statements by a simple English sentence.
- $\neg T(\text{Abdallah Hussein}, \text{Japanese})$
 - $\exists x T(x, \text{Korean}) \wedge \forall x T(x, \text{Mexican})$
 - $\exists y(T(\text{Monique Arsenault}, y) \vee T(\text{Jay Johnson}, y))$
 - $\forall x \forall z \exists y((x \neq z) \rightarrow \neg(T(x, y) \wedge T(z, y)))$
 - $\exists x \exists z \forall y(T(x, y) \leftrightarrow T(z, y))$
 - $\forall x \forall z \exists y(T(x, y) \leftrightarrow T(z, y))$
8. Let $Q(x, y)$ be the statement “student x has been a contestant on quiz show y .” Express each of these sentences in terms of $Q(x, y)$, quantifiers, and logical connectives, where the domain for x consists of all students at your school and for y consists of all quiz shows on television.
- There is a student at your school who has been a contestant on a television quiz show.
 - No student at your school has ever been a contestant on a television quiz show.
 - There is a student at your school who has been a contestant on *Jeopardy* and on *Wheel of Fortune*.
 - Every television quiz show has had a student from your school as a contestant.
 - At least two students from your school have been contestants on *Jeopardy*.
9. Let $L(x, y)$ be the statement “ x loves y ,” where the domain for both x and y consists of all people in the world. Use quantifiers to express each of these statements.
- Everybody loves Jerry.
 - Everybody loves somebody.
 - There is somebody whom everybody loves.
 - Nobody loves everybody.
 - There is somebody whom Lydia does not love.
 - There is somebody whom no one loves.
 - There is exactly one person whom everybody loves.
 - There are exactly two people whom Lynn loves.
 - Everyone loves himself or herself.
 - There is someone who loves no one besides himself or herself.
10. Let $F(x, y)$ be the statement “ x can fool y ,” where the domain consists of all people in the world. Use quantifiers to express each of these statements.
- Everybody can fool Fred.
 - Evelyn can fool everybody.
 - Everybody can fool somebody.
 - There is no one who can fool everybody.
 - Everyone can be fooled by somebody.
 - No one can fool both Fred and Jerry.
 - Nancy can fool exactly two people.
- h) There is exactly one person whom everybody can fool.
- No one can fool himself or herself.
 - There is someone who can fool exactly one person besides himself or herself.
11. Let $S(x)$ be the predicate “ x is a student,” $F(x)$ the predicate “ x is a faculty member,” and $A(x, y)$ the predicate “ x has asked y a question,” where the domain consists of all people associated with your school. Use quantifiers to express each of these statements.
- Lois has asked Professor Michaels a question.
 - Every student has asked Professor Gross a question.
 - Every faculty member has either asked Professor Miller a question or been asked a question by Professor Miller.
 - Some student has not asked any faculty member a question.
 - There is a faculty member who has never been asked a question by a student.
 - Some student has asked every faculty member a question.
 - There is a faculty member who has asked every other faculty member a question.
 - Some student has never been asked a question by a faculty member.
12. Let $I(x)$ be the statement “ x has an Internet connection” and $C(x, y)$ be the statement “ x and y have chatted over the Internet,” where the domain for the variables x and y consists of all students in your class. Use quantifiers to express each of these statements.
- Jerry does not have an Internet connection.
 - Rachel has not chatted over the Internet with Chelsea.
 - Jan and Sharon have never chatted over the Internet.
 - No one in the class has chatted with Bob.
 - Sanjay has chatted with everyone except Joseph.
 - Someone in your class does not have an Internet connection.
 - Not everyone in your class has an Internet connection.
 - Exactly one student in your class has an Internet connection.
 - Everyone except one student in your class has an Internet connection.
 - Everyone in your class with an Internet connection has chatted over the Internet with at least one other student in your class.
 - Someone in your class has an Internet connection but has not chatted with anyone else in your class.
 - There are two students in your class who have not chatted with each other over the Internet.
 - There is a student in your class who has chatted with everyone in your class over the Internet.
 - There are at least two students in your class who have not chatted with the same person in your class.
 - There are two students in the class who between them have chatted with everyone else in the class.

- 13.** Let $M(x, y)$ be “ x has sent y an e-mail message” and $T(x, y)$ be “ x has telephoned y ,” where the domain consists of all students in your class. Use quantifiers to express each of these statements. (Assume that all e-mail messages that were sent are received, which is not the way things often work.)
- a) Chou has never sent an e-mail message to Koko.
 - b) Arlene has never sent an e-mail message to or telephoned Sarah.
 - c) José has never received an e-mail message from Deborah.
 - d) Every student in your class has sent an e-mail message to Ken.
 - e) No one in your class has telephoned Nina.
 - f) Everyone in your class has either telephoned Avi or sent him an e-mail message.
 - g) There is a student in your class who has sent everyone else in your class an e-mail message.
 - h) There is someone in your class who has either sent an e-mail message or telephoned everyone else in your class.
 - i) There are two different students in your class who have sent each other e-mail messages.
 - j) There is a student who has sent himself or herself an e-mail message.
 - k) There is a student in your class who has not received an e-mail message from anyone else in the class and who has not been called by any other student in the class.
 - l) Every student in the class has either received an e-mail message or received a telephone call from another student in the class.
 - m) There are at least two students in your class such that one student has sent the other e-mail and the second student has telephoned the first student.
 - n) There are two different students in your class who between them have sent an e-mail message to or telephoned everyone else in the class.
- 14.** Use quantifiers and predicates with more than one variable to express these statements.
- a) There is a student in this class who can speak Hindi.
 - b) Every student in this class plays some sport.
 - c) Some student in this class has visited Alaska but has not visited Hawaii.
 - d) All students in this class have learned at least one programming language.
 - e) There is a student in this class who has taken every course offered by one of the departments in this school.
 - f) Some student in this class grew up in the same town as exactly one other student in this class.
 - g) Every student in this class has chatted with at least one other student in at least one chat group.
- 15.** Use quantifiers and predicates with more than one variable to express these statements.
- a) Every computer science student needs a course in discrete mathematics.
 - b) There is a student in this class who owns a personal computer.
 - c) Every student in this class has taken at least one computer science course.
 - d) There is a student in this class who has taken at least one course in computer science.
 - e) Every student in this class has been in every building on campus.
 - f) There is a student in this class who has been in every room of at least one building on campus.
 - g) Every student in this class has been in at least one room of every building on campus.
- 16.** A discrete mathematics class contains 1 mathematics major who is a freshman, 12 mathematics majors who are sophomores, 15 computer science majors who are sophomores, 2 mathematics majors who are juniors, 2 computer science majors who are juniors, and 1 computer science major who is a senior. Express each of these statements in terms of quantifiers and then determine its truth value.
- a) There is a student in the class who is a junior.
 - b) Every student in the class is a computer science major.
 - c) There is a student in the class who is neither a mathematics major nor a junior.
 - d) Every student in the class is either a sophomore or a computer science major.
 - e) There is a major such that there is a student in the class in every year of study with that major.
- 17.** Express each of these system specifications using predicates, quantifiers, and logical connectives, if necessary.
- a) Every user has access to exactly one mailbox.
 - b) There is a process that continues to run during all error conditions only if the kernel is working correctly.
 - c) All users on the campus network can access all websites whose url has a .edu extension.
 - *d) There are exactly two systems that monitor every remote server.
- 18.** Express each of these system specifications using predicates, quantifiers, and logical connectives, if necessary.
- a) At least one console must be accessible during every fault condition.
 - b) The e-mail address of every user can be retrieved whenever the archive contains at least one message sent by every user on the system.
 - c) For every security breach there is at least one mechanism that can detect that breach if and only if there is a process that has not been compromised.
 - d) There are at least two paths connecting every two distinct endpoints on the network.
 - e) No one knows the password of every user on the system except for the system administrator, who knows all passwords.[
- 19.** Express each of these statements using mathematical and logical operators, predicates, and quantifiers, where the domain consists of all integers.
- a) The sum of two negative integers is negative.
 - b) The difference of two positive integers is not necessarily positive.

- c) The sum of the squares of two integers is greater than or equal to the square of their sum.
d) The absolute value of the product of two integers is the product of their absolute values.
20. Express each of these statements using predicates, quantifiers, logical connectives, and mathematical operators where the domain consists of all integers.
- The product of two negative integers is positive.
 - The average of two positive integers is positive.
 - The difference of two negative integers is not necessarily negative.
 - The absolute value of the sum of two integers does not exceed the sum of the absolute values of these integers.
21. Use predicates, quantifiers, logical connectives, and mathematical operators to express the statement that every positive integer is the sum of the squares of four integers.
22. Use predicates, quantifiers, logical connectives, and mathematical operators to express the statement that there is a positive integer that is not the sum of three squares.
23. Express each of these mathematical statements using predicates, quantifiers, logical connectives, and mathematical operators.
- The product of two negative real numbers is positive.
 - The difference of a real number and itself is zero.
 - Every positive real number has exactly two square roots.
 - A negative real number does not have a square root that is a real number.
24. Translate each of these nested quantifications into an English statement that expresses a mathematical fact. The domain in each case consists of all real numbers.
- $\exists x \forall y (x + y = y)$
 - $\forall x \forall y (((x \geq 0) \wedge (y < 0)) \rightarrow (x - y > 0))$
 - $\exists x \exists y (((x \leq 0) \wedge (y \leq 0)) \wedge (x - y > 0))$
 - $\forall x \forall y ((x \neq 0) \wedge (y \neq 0) \leftrightarrow (xy \neq 0))$
25. Translate each of these nested quantifications into an English statement that expresses a mathematical fact. The domain in each case consists of all real numbers.
- $\exists x \forall y (xy = y)$
 - $\forall x \forall y (((x < 0) \wedge (y < 0)) \rightarrow (xy > 0))$
 - $\exists x \exists y ((x^2 > y) \wedge (x < y))$
 - $\forall x \forall y \exists z (x + y = z)$
26. Let $Q(x, y)$ be the statement " $x + y = x - y$." If the domain for both variables consists of all integers, what are the truth values?
- $Q(1, 1)$
 - $Q(2, 0)$
 - $\forall y Q(1, y)$
 - $\exists x Q(x, 2)$
 - $\exists x \exists y Q(x, y)$
 - $\forall x \exists y Q(x, y)$
 - $\exists y \forall x Q(x, y)$
 - $\forall x \forall y Q(x, y)$
27. Determine the truth value of each of these statements if the domain for all variables consists of all integers.
- $\forall n \exists m (n^2 < m)$
 - $\exists n \forall m (n < m^2)$
 - $\forall n \exists m (n + m = 0)$
 - $\exists n \forall m (nm = m)$
- e) $\exists n \exists m (n^2 + m^2 = 5)$ f) $\exists n \exists m (n^2 + m^2 = 6)$
g) $\exists n \exists m (n + m = 4 \wedge n - m = 1)$
h) $\exists n \exists m (n + m = 4 \wedge n - m = 2)$
i) $\forall n \forall m \exists p (p = (m + n)/2)$
28. Determine the truth value of each of these statements if the domain of each variable consists of all real numbers.
- $\forall x \exists y (x^2 = y)$
 - $\forall x \exists y (x = y^2)$
 - $\exists x \forall y (xy = 0)$
 - $\exists x \exists y (x + y \neq y + x)$
 - $\forall x (x \neq 0 \rightarrow \exists y (xy = 1))$
 - $\exists x \forall y (y \neq 0 \rightarrow xy = 1)$
 - $\forall x \exists y (x + y = 1)$
 - $\exists x \exists y (x + 2y = 2 \wedge 2x + 4y = 5)$
 - $\forall x \exists y (x + y = 2 \wedge 2x - y = 1)$
 - $\forall x \forall y \exists z (z = (x + y)/2)$
29. Suppose the domain of the propositional function $P(x, y)$ consists of pairs x and y , where x is 1, 2, or 3 and y is 1, 2, or 3. Write out these propositions using disjunctions and conjunctions.
- $\forall x \forall y P(x, y)$
 - $\exists x \exists y P(x, y)$
 - $\exists x \forall y P(x, y)$
 - $\forall y \exists x P(x, y)$
30. Rewrite each of these statements so that negations appear only within predicates (that is, so that no negation is outside a quantifier or an expression involving logical connectives).
- $\neg \exists y \exists x P(x, y)$
 - $\neg \forall x \exists y P(x, y)$
 - $\neg \exists y (Q(y) \wedge \forall x \neg R(x, y))$
 - $\neg \exists y (\exists x R(x, y) \vee \forall x S(x, y))$
 - $\neg \exists y (\forall x \exists z T(x, y, z) \vee \exists x \forall z U(x, y, z))$
31. Express the negations of each of these statements so that all negation symbols immediately precede predicates.
- $\forall x \exists y \forall z T(x, y, z)$
 - $\forall x \exists y P(x, y) \vee \forall x \exists y Q(x, y)$
 - $\forall x \exists y (P(x, y) \wedge \exists z R(x, y, z))$
 - $\forall x \exists y (P(x, y) \rightarrow Q(x, y))$
32. Express the negations of each of these statements so that all negation symbols immediately precede predicates.
- $\exists z \forall y \forall x T(x, y, z)$
 - $\exists x \exists y P(x, y) \wedge \forall x \forall y Q(x, y)$
 - $\exists x \exists y (Q(x, y) \leftrightarrow Q(y, x))$
 - $\forall y \exists x \exists z (T(x, y, z) \vee Q(x, y))$
33. Rewrite each of these statements so that negations appear only within predicates (that is, so that no negation is outside a quantifier or an expression involving logical connectives).
- $\neg \forall x \forall y P(x, y)$
 - $\neg \forall y \exists x P(x, y)$
 - $\neg \forall y \forall x (P(x, y) \vee Q(x, y))$
 - $\neg (\exists x \exists y \neg P(x, y) \wedge \forall x \forall y Q(x, y))$
 - $\neg \forall x (\exists y \forall z P(x, y, z) \wedge \exists z \forall y P(x, y, z))$
34. Find a common domain for the variables x , y , and z for which the statement $\forall x \forall y ((x \neq y) \rightarrow \forall z ((z = x) \vee (z = y)))$ is true and another domain for which it is false.
35. Find a common domain for the variables x , y , z , and w for which the statement $\forall x \forall y \forall z \exists w ((w \neq x) \wedge (w \neq y) \wedge (w \neq z))$ is true and another common domain for these variables for which it is false.

- 36.** Express each of these statements using quantifiers. Then form the negation of the statement so that no negation is to the left of a quantifier. Next, express the negation in simple English. (Do not simply use the phrase “It is not the case that.”)
- No one has lost more than one thousand dollars playing the lottery.
 - There is a student in this class who has chatted with exactly one other student.
 - No student in this class has sent e-mail to exactly two other students in this class.
 - Some student has solved every exercise in this book.
 - No student has solved at least one exercise in every section of this book.
- 37.** Express each of these statements using quantifiers. Then form the negation of the statement so that no negation is to the left of a quantifier. Next, express the negation in simple English. (Do not simply use the phrase “It is not the case that.”)
- Every student in this class has taken exactly two mathematics classes at this school.
 - Someone has visited every country in the world except Libya.
 - No one has climbed every mountain in the Himalayas.
 - Every movie actor has either been in a movie with Kevin Bacon or has been in a movie with someone who has been in a movie with Kevin Bacon.
- 38.** Express the negations of these propositions using quantifiers, and in English.
- Every student in this class likes mathematics.
 - There is a student in this class who has never seen a computer.
 - There is a student in this class who has taken every mathematics course offered at this school.
 - There is a student in this class who has been in at least one room of every building on campus.
- 39.** Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all integers.
- $\forall x \forall y (x^2 = y^2 \rightarrow x = y)$
 - $\forall x \exists y (y^2 = x)$
 - $\forall x \forall y (xy \geq x)$
- 40.** Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all integers.
- $\forall x \exists y (x = 1/y)$
 - $\forall x \exists y (y^2 - x < 100)$
 - $\forall x \forall y (x^2 \neq y^3)$
- 41.** Use quantifiers to express the associative law for multiplication of real numbers.
- 42.** Use quantifiers to express the distributive laws of multiplication over addition for real numbers.
- 43.** Use quantifiers and logical connectives to express the fact that every linear polynomial (that is, polynomial of degree 1) with real coefficients and where the coefficient of x is nonzero, has exactly one real root.
- 44.** Use quantifiers and logical connectives to express the fact that a quadratic polynomial with real number coefficients has at most two real roots.
- 45.** Determine the truth value of the statement $\forall x \exists y (xy = 1)$ if the domain for the variables consists of
- the nonzero real numbers.
 - the nonzero integers.
 - the positive real numbers.
- 46.** Determine the truth value of the statement $\exists x \forall y (x \leq y^2)$ if the domain for the variables consists of
- the positive real numbers.
 - the integers.
 - the nonzero real numbers.
- 47.** Show that the two statements $\neg \exists x \forall y P(x, y)$ and $\forall x \exists y \neg P(x, y)$, where both quantifiers over the first variable in $P(x, y)$ have the same domain, and both quantifiers over the second variable in $P(x, y)$ have the same domain, are logically equivalent.
- *48.** Show that $\forall x P(x) \vee \forall x Q(x)$ and $\forall x \forall y (P(x) \vee Q(y))$, where all quantifiers have the same nonempty domain, are logically equivalent. (The new variable y is used to combine the quantifications correctly.)
- *49.**
 - Show that $\forall x P(x) \wedge \exists x Q(x)$ is logically equivalent to $\forall x \exists y (P(x) \wedge Q(y))$, where all quantifiers have the same nonempty domain.
 - Show that $\forall x P(x) \vee \exists x Q(x)$ is equivalent to $\forall x \exists y (P(x) \vee Q(y))$, where all quantifiers have the same nonempty domain.
- A statement is in **prenex normal form (PNF)** if and only if it is of the form
- $$Q_1 x_1 Q_2 x_2 \cdots Q_k x_k P(x_1, x_2, \dots, x_k),$$
- where each Q_i , $i = 1, 2, \dots, k$, is either the existential quantifier or the universal quantifier, and $P(x_1, \dots, x_k)$ is a predicate involving no quantifiers. For example, $\exists x \forall y (P(x, y) \wedge Q(y))$ is in prenex normal form, whereas $\exists x P(x) \vee \forall x Q(x)$ is not (because the quantifiers do not all occur first).
- Every statement formed from propositional variables, predicates, **T**, and **F** using logical connectives and quantifiers is equivalent to a statement in prenex normal form. Exercise 51 asks for a proof of this fact.
- *50.** Put these statements in prenex normal form. [Hint: Use logical equivalence from Tables 6 and 7 in Section 1.3, Table 2 in Section 1.4, Example 19 in Section 1.4, Exercises 45 and 46 in Section 1.4, and Exercises 48 and 49.]
- $\exists x P(x) \vee \exists x Q(x) \vee A$, where A is a proposition not involving any quantifiers.
 - $\neg(\forall x P(x) \vee \forall x Q(x))$
 - $\exists x P(x) \rightarrow \exists x Q(x)$
- **51.** Show how to transform an arbitrary statement to a statement in prenex normal form that is equivalent to the given statement. (Note: A formal solution of this exercise requires use of structural induction, covered in Section 5.3.)
- *52.** Express the quantification $\exists! x P(x)$, introduced in Section 1.4, using universal quantifications, existential quantifications, and logical operators.

1.6 Rules of Inference

Introduction

Later in this chapter we will study proofs. Proofs in mathematics are valid arguments that establish the truth of mathematical statements. By an **argument**, we mean a sequence of statements that end with a conclusion. By **valid**, we mean that the conclusion, or final statement of the argument, must follow from the truth of the preceding statements, or **premises**, of the argument. That is, an argument is valid if and only if it is impossible for all the premises to be true and the conclusion to be false. To deduce new statements from statements we already have, we use rules of inference which are templates for constructing valid arguments. Rules of inference are our basic tools for establishing the truth of statements.

Before we study mathematical proofs, we will look at arguments that involve only compound propositions. We will define what it means for an argument involving compound propositions to be valid. Then we will introduce a collection of rules of inference in propositional logic. These rules of inference are among the most important ingredients in producing valid arguments. After we illustrate how rules of inference are used to produce valid arguments, we will describe some common forms of incorrect reasoning, called **fallacies**, which lead to invalid arguments.

After studying rules of inference in propositional logic, we will introduce rules of inference for quantified statements. We will describe how these rules of inference can be used to produce valid arguments. These rules of inference for statements involving existential and universal quantifiers play an important role in proofs in computer science and mathematics, although they are often used without being explicitly mentioned.

Finally, we will show how rules of inference for propositions and for quantified statements can be combined. These combinations of rule of inference are often used together in complicated arguments.

Valid Arguments in Propositional Logic

Consider the following argument involving propositions (which, by definition, is a sequence of propositions):

“If you have a current password, then you can log onto the network.”

“You have a current password.”

Therefore,

“You can log onto the network.”

We would like to determine whether this is a valid argument. That is, we would like to determine whether the conclusion “You can log onto the network” must be true when the premises “If you have a current password, then you can log onto the network” and “You have a current password” are both true.

Before we discuss the validity of this particular argument, we will look at its form. Use p to represent “You have a current password” and q to represent “You can log onto the network.” Then, the argument has the form

$$\begin{array}{c} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

where \therefore is the symbol that denotes “therefore.”

We know that when p and q are propositional variables, the statement $((p \rightarrow q) \wedge p) \rightarrow q$ is a tautology (see Exercise 10(c) in Section 1.3). In particular, when both $p \rightarrow q$ and p are true, we know that q must also be true. We say this form of argument is **valid** because whenever all its premises (all statements in the argument other than the final one, the conclusion) are true, the conclusion must also be true. Now suppose that both “If you have a current password, then you can log onto the network” and “You have a current password” are true statements. When we replace p by “You have a current password” and q by “You can log onto the network,” it necessarily follows that the conclusion “You can log onto the network” is true. This argument is **valid** because its form is valid. Note that whenever we replace p and q by propositions where $p \rightarrow q$ and p are both true, then q must also be true.

What happens when we replace p and q in this argument form by propositions where not both p and $p \rightarrow q$ are true? For example, suppose that p represents “You have access to the network” and q represents “You can change your grade” and that p is true, but $p \rightarrow q$ is false. The argument we obtain by substituting these values of p and q into the argument form is

$$\begin{array}{l} \text{“If you have access to the network, then you can change your grade.”} \\ \text{“You have access to the network.”} \\ \hline \therefore \text{“You can change your grade.”} \end{array}$$

The argument we obtained is a valid argument, but because one of the premises, namely the first premise, is false, we cannot conclude that the conclusion is true. (Most likely, this conclusion is false.)

In our discussion, to analyze an argument, we replaced propositions by propositional variables. This changed an argument to an **argument form**. We saw that the validity of an argument follows from the validity of the form of the argument. We summarize the terminology used to discuss the validity of arguments with our definition of the key notions.

DEFINITION 1

An *argument* in propositional logic is a sequence of propositions. All but the final proposition in the argument are called *premises* and the final proposition is called the *conclusion*. An argument is *valid* if the truth of all its premises implies that the conclusion is true.

An *argument form* in propositional logic is a sequence of compound propositions involving propositional variables. An argument form is *valid* no matter which particular propositions are substituted for the propositional variables in its premises, the conclusion is true if the premises are all true.

From the definition of a valid argument form we see that the argument form with premises p_1, p_2, \dots, p_n and conclusion q is valid, when $(p_1 \wedge p_2 \wedge \dots \wedge p_n) \rightarrow q$ is a tautology.

The key to showing that an argument in propositional logic is valid is to show that its argument form is valid. Consequently, we would like techniques to show that argument forms are valid. We will now develop methods for accomplishing this task.

Rules of Inference for Propositional Logic

We can always use a truth table to show that an argument form is valid. We do this by showing that whenever the premises are true, the conclusion must also be true. However, this can be a tedious approach. For example, when an argument form involves 10 different propositional variables, to use a truth table to show this argument form is valid requires $2^{10} = 1024$ different rows. Fortunately, we do not have to resort to truth tables. Instead, we can first establish the validity of some relatively simple argument forms, called **rules of inference**. These rules of inference can be used as building blocks to construct more complicated valid argument forms. We will now introduce the most important rules of inference in propositional logic.

The tautology $(p \wedge (p \rightarrow q)) \rightarrow q$ is the basis of the rule of inference called **modus ponens**, or the **law of detachment**. (Modus ponens is Latin for *mode that affirms*.) This tautology leads to the following valid argument form, which we have already seen in our initial discussion about arguments (where, as before, the symbol \therefore denotes “therefore”):

$$\begin{array}{c} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$

Using this notation, the hypotheses are written in a column, followed by a horizontal bar, followed by a line that begins with the therefore symbol and ends with the conclusion. In particular, modus ponens tells us that if a conditional statement and the hypothesis of this conditional statement are both true, then the conclusion must also be true. Example 1 illustrates the use of modus ponens.

EXAMPLE 1 Suppose that the conditional statement “If it snows today, then we will go skiing” and its hypothesis, “It is snowing today,” are true. Then, by modus ponens, it follows that the conclusion of the conditional statement, “We will go skiing,” is true. 

As we mentioned earlier, a valid argument can lead to an incorrect conclusion if one or more of its premises is false. We illustrate this again in Example 2.

EXAMPLE 2 Determine whether the argument given here is valid and determine whether its conclusion must be true because of the validity of the argument.

“If $\sqrt{2} > \frac{3}{2}$, then $(\sqrt{2})^2 > (\frac{3}{2})^2$. We know that $\sqrt{2} > \frac{3}{2}$. Consequently, $(\sqrt{2})^2 = 2 > (\frac{3}{2})^2 = \frac{9}{4}$.”

Solution: Let p be the proposition “ $\sqrt{2} > \frac{3}{2}$ ” and q the proposition “ $2 > (\frac{3}{2})^2$.” The premises of the argument are $p \rightarrow q$ and p , and q is its conclusion. This argument is valid because it is constructed by using modus ponens, a valid argument form. However, one of its premises, $\sqrt{2} > \frac{3}{2}$, is false. Consequently, we cannot conclude that the conclusion is true. Furthermore, note that the conclusion of this argument is false, because $2 < \frac{9}{4}$. 

There are many useful rules of inference for propositional logic. Perhaps the most widely used of these are listed in Table 1. Exercises 9, 10, 15, and 30 in Section 1.3 ask for the verifications that these rules of inference are valid argument forms. We now give examples of arguments that use these rules of inference. In each argument, we first use propositional variables to express the propositions in the argument. We then show that the resulting argument form is a rule of inference from Table 1.

TABLE 1 Rules of Inference.

<i>Rule of Inference</i>	<i>Tautology</i>	<i>Name</i>
$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$	$(p \wedge (p \rightarrow q)) \rightarrow q$	Modus ponens
$\begin{array}{l} \neg q \\ p \rightarrow q \\ \hline \therefore \neg p \end{array}$	$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$	Modus tollens
$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$	Hypothetical syllogism
$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$	$((p \vee q) \wedge \neg p) \rightarrow q$	Disjunctive syllogism
$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array}$	$p \rightarrow (p \vee q)$	Addition
$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array}$	$(p \wedge q) \rightarrow p$	Simplification
$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction
$\begin{array}{l} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$	$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$	Resolution

EXAMPLE 3 State which rule of inference is the basis of the following argument: “It is below freezing now. Therefore, it is either below freezing or raining now.”

Solution: Let p be the proposition “It is below freezing now” and q the proposition “It is raining now.” Then this argument is of the form

$$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array}$$

This is an argument that uses the addition rule. 

EXAMPLE 4 State which rule of inference is the basis of the following argument: “It is below freezing and raining now. Therefore, it is below freezing now.”

Solution: Let p be the proposition “It is below freezing now,” and let q be the proposition “It is raining now.” This argument is of the form

$$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array}$$

This argument uses the simplification rule. 

EXAMPLE 5 State which rule of inference is used in the argument:

If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have a barbecue tomorrow.

Solution: Let p be the proposition “It is raining today,” let q be the proposition “We will not have a barbecue today,” and let r be the proposition “We will have a barbecue tomorrow.” Then this argument is of the form

$$\begin{array}{c} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Hence, this argument is a hypothetical syllogism. 

Using Rules of Inference to Build Arguments

When there are many premises, several rules of inference are often needed to show that an argument is valid. This is illustrated by Examples 6 and 7, where the steps of arguments are displayed on separate lines, with the reason for each step explicitly stated. These examples also show how arguments in English can be analyzed using rules of inference.

EXAMPLE 6 Show that the premises “It is not sunny this afternoon and it is colder than yesterday,” “We will go swimming only if it is sunny,” “If we do not go swimming, then we will take a canoe trip,” and “If we take a canoe trip, then we will be home by sunset” lead to the conclusion “We will be home by sunset.”



Solution: Let p be the proposition “It is sunny this afternoon,” q the proposition “It is colder than yesterday,” r the proposition “We will go swimming,” s the proposition “We will take a canoe trip,” and t the proposition “We will be home by sunset.” Then the premises become $\neg p \wedge q$, $r \rightarrow p$, $\neg r \rightarrow s$, and $s \rightarrow t$. The conclusion is simply t . We need to give a valid argument with premises $\neg p \wedge q$, $r \rightarrow p$, $\neg r \rightarrow s$, and $s \rightarrow t$ and conclusion t .

We construct an argument to show that our premises lead to the desired conclusion as follows.

Step	Reason
1. $\neg p \wedge q$	Premise
2. $\neg p$	Simplification using (1)
3. $r \rightarrow p$	Premise
4. $\neg r$	Modus tollens using (2) and (3)
5. $\neg r \rightarrow s$	Premise
6. s	Modus ponens using (4) and (5)
7. $s \rightarrow t$	Premise
8. t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p , q , r , s , and t , such a truth table would have 32 rows. 

EXAMPLE 7 Show that the premises “If you send me an e-mail message, then I will finish writing the program,” “If you do not send me an e-mail message, then I will go to sleep early,” and “If I go to sleep early, then I will wake up feeling refreshed” lead to the conclusion “If I do not finish writing the program, then I will wake up feeling refreshed.”

Solution: Let p be the proposition “You send me an e-mail message,” q the proposition “I will finish writing the program,” r the proposition “I will go to sleep early,” and s the proposition “I will wake up feeling refreshed.” Then the premises are $p \rightarrow q$, $\neg p \rightarrow r$, and $r \rightarrow s$. The desired conclusion is $\neg q \rightarrow s$. We need to give a valid argument with premises $p \rightarrow q$, $\neg p \rightarrow r$, and $r \rightarrow s$ and conclusion $\neg q \rightarrow s$.

This argument form shows that the premises lead to the desired conclusion.

Step	Reason
1. $p \rightarrow q$	Premise
2. $\neg q \rightarrow \neg p$	Contrapositive of (1)
3. $\neg p \rightarrow r$	Premise
4. $\neg q \rightarrow r$	Hypothetical syllogism using (2) and (3)
5. $r \rightarrow s$	Premise
6. $\neg q \rightarrow s$	Hypothetical syllogism using (4) and (5)

Resolution

Computer programs have been developed to automate the task of reasoning and proving theorems. Many of these programs make use of a rule of inference known as **resolution**. This rule of inference is based on the tautology



$$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r).$$

(Exercise 30 in Section 1.3 asks for the verification that this is a tautology.) The final disjunction in the resolution rule, $q \vee r$, is called the **resolvent**. When we let $q = r$ in this tautology, we obtain $(p \vee q) \wedge (\neg p \vee q) \rightarrow q$. Furthermore, when we let $r = F$, we obtain $(p \vee q) \wedge (\neg p) \rightarrow q$ (because $q \vee F \equiv q$), which is the tautology on which the rule of disjunctive syllogism is based.

EXAMPLE 8 Use resolution to show that the hypotheses “Jasmine is skiing or it is not snowing” and “It is snowing or Bart is playing hockey” imply that “Jasmine is skiing or Bart is playing hockey.”



Solution: Let p be the proposition “It is snowing,” q the proposition “Jasmine is skiing,” and r the proposition “Bart is playing hockey.” We can represent the hypotheses as $\neg p \vee q$ and $p \vee r$, respectively. Using resolution, the proposition $q \vee r$, “Jasmine is skiing or Bart is playing hockey,” follows.

Resolution plays an important role in programming languages based on the rules of logic, such as Prolog (where resolution rules for quantified statements are applied). Furthermore, it can be used to build automatic theorem proving systems. To construct proofs in propositional logic using resolution as the only rule of inference, the hypotheses and the conclusion must be expressed as **clauses**, where a clause is a disjunction of variables or negations of these variables. We can replace a statement in propositional logic that is not a clause by one or more equivalent statements that are clauses. For example, suppose we have a statement of the form $p \vee (q \wedge r)$. Because $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$, we can replace the single statement $p \vee (q \wedge r)$ by two statements $p \vee q$ and $p \vee r$, each of which is a clause. We can replace a statement of the form $\neg(p \vee q)$ by the two statements $\neg p$ and $\neg q$ because De Morgan’s law tells us that $\neg(p \vee q) \equiv \neg p \wedge \neg q$. We can also replace a conditional statement $p \rightarrow q$ with the equivalent disjunction $\neg p \vee q$.

EXAMPLE 9 Show that the premises $(p \wedge q) \vee r$ and $r \rightarrow s$ imply the conclusion $p \vee s$.

Solution: We can rewrite the premises $(p \wedge q) \vee r$ as two clauses, $p \vee r$ and $q \vee r$. We can also replace $r \rightarrow s$ by the equivalent clause $\neg r \vee s$. Using the two clauses $p \vee r$ and $\neg r \vee s$, we can use resolution to conclude $p \vee s$. 

Fallacies

Several common fallacies arise in incorrect arguments. These fallacies resemble rules of inference, but are based on contingencies rather than tautologies. These are discussed here to show the distinction between correct and incorrect reasoning.



The proposition $((p \rightarrow q) \wedge q) \rightarrow p$ is not a tautology, because it is false when p is false and q is true. However, there are many incorrect arguments that treat this as a tautology. In other words, they treat the argument with premises $p \rightarrow q$ and q and conclusion p as a valid argument form, which it is not. This type of incorrect reasoning is called the **fallacy of affirming the conclusion**.

EXAMPLE 10 Is the following argument valid?

If you do every problem in this book, then you will learn discrete mathematics. You learned discrete mathematics.

Therefore, you did every problem in this book.

Solution: Let p be the proposition “You did every problem in this book.” Let q be the proposition “You learned discrete mathematics.” Then this argument is of the form: if $p \rightarrow q$ and q , then p . This is an example of an incorrect argument using the fallacy of affirming the conclusion. Indeed, it is possible for you to learn discrete mathematics in some way other than by doing every problem in this book. (You may learn discrete mathematics by reading, listening to lectures, doing some, but not all, the problems in this book, and so on.) 

The proposition $((p \rightarrow q) \wedge \neg p) \rightarrow \neg q$ is not a tautology, because it is false when p is false and q is true. Many incorrect arguments use this incorrectly as a rule of inference. This type of incorrect reasoning is called the **fallacy of denying the hypothesis**.

EXAMPLE 11 Let p and q be as in Example 10. If the conditional statement $p \rightarrow q$ is true, and $\neg p$ is true, is it correct to conclude that $\neg q$ is true? In other words, is it correct to assume that you did not learn discrete mathematics if you did not do every problem in the book, assuming that if you do every problem in this book, then you will learn discrete mathematics?

Solution: It is possible that you learned discrete mathematics even if you did not do every problem in this book. This incorrect argument is of the form $p \rightarrow q$ and $\neg p$ imply $\neg q$, which is an example of the fallacy of denying the hypothesis. 

Rules of Inference for Quantified Statements

We have discussed rules of inference for propositions. We will now describe some important rules of inference for statements involving quantifiers. These rules of inference are used extensively in mathematical arguments, often without being explicitly mentioned.

Universal instantiation is the rule of inference used to conclude that $P(c)$ is true, where c is a particular member of the domain, given the premise $\forall x P(x)$. Universal instantiation is used when we conclude from the statement “All women are wise” that “Lisa is wise,” where Lisa is a member of the domain of all women.

TABLE 2 Rules of Inference for Quantified Statements.

<i>Rule of Inference</i>	<i>Name</i>
$\frac{\forall x P(x)}{\therefore P(c)}$	Universal instantiation
$\frac{P(c) \text{ for an arbitrary } c}{\therefore \forall x P(x)}$	Universal generalization
$\frac{\exists x P(x)}{\therefore P(c) \text{ for some element } c}$	Existential instantiation
$\frac{P(c) \text{ for some element } c}{\therefore \exists x P(x)}$	Existential generalization

Universal generalization is the rule of inference that states that $\forall x P(x)$ is true, given the premise that $P(c)$ is true for all elements c in the domain. Universal generalization is used when we show that $\forall x P(x)$ is true by taking an arbitrary element c from the domain and showing that $P(c)$ is true. The element c that we select must be an arbitrary, and not a specific, element of the domain. That is, when we assert from $\forall x P(x)$ the existence of an element c in the domain, we have no control over c and cannot make any other assumptions about c other than it comes from the domain. Universal generalization is used implicitly in many proofs in mathematics and is seldom mentioned explicitly. However, the error of adding unwarranted assumptions about the arbitrary element c when universal generalization is used is all too common in incorrect reasoning.

Existential instantiation is the rule that allows us to conclude that there is an element c in the domain for which $P(c)$ is true if we know that $\exists x P(x)$ is true. We cannot select an arbitrary value of c here, but rather it must be a c for which $P(c)$ is true. Usually we have no knowledge of what c is, only that it exists. Because it exists, we may give it a name (c) and continue our argument.

Existential generalization is the rule of inference that is used to conclude that $\exists x P(x)$ is true when a particular element c with $P(c)$ true is known. That is, if we know one element c in the domain for which $P(c)$ is true, then we know that $\exists x P(x)$ is true.

We summarize these rules of inference in Table 2. We will illustrate how some of these rules of inference for quantified statements are used in Examples 12 and 13.

EXAMPLE 12 Show that the premises “Everyone in this discrete mathematics class has taken a course in computer science” and “Marla is a student in this class” imply the conclusion “Marla has taken a course in computer science.”

Solution: Let $D(x)$ denote “ x is in this discrete mathematics class,” and let $C(x)$ denote “ x has taken a course in computer science.” Then the premises are $\forall x(D(x) \rightarrow C(x))$ and $D(\text{Marla})$. The conclusion is $C(\text{Marla})$.



The following steps can be used to establish the conclusion from the premises.

Step	Reason
1. $\forall x(D(x) \rightarrow C(x))$	Premise
2. $D(\text{Marla}) \rightarrow C(\text{Marla})$	Universal instantiation from (1)
3. $D(\text{Marla})$	Premise
4. $C(\text{Marla})$	Modus ponens from (2) and (3)



EXAMPLE 13 Show that the premises “A student in this class has not read the book,” and “Everyone in this class passed the first exam” imply the conclusion “Someone who passed the first exam has not read the book.”

Solution: Let $C(x)$ be “ x is in this class,” $B(x)$ be “ x has read the book,” and $P(x)$ be “ x passed the first exam.” The premises are $\exists x(C(x) \wedge \neg B(x))$ and $\forall x(C(x) \rightarrow P(x))$. The conclusion is $\exists x(P(x) \wedge \neg B(x))$. These steps can be used to establish the conclusion from the premises.

Step	Reason
1. $\exists x(C(x) \wedge \neg B(x))$	Premise
2. $C(a) \wedge \neg B(a)$	Existential instantiation from (1)
3. $C(a)$	Simplification from (2)
4. $\forall x(C(x) \rightarrow P(x))$	Premise
5. $C(a) \rightarrow P(a)$	Universal instantiation from (4)
6. $P(a)$	Modus ponens from (3) and (5)
7. $\neg B(a)$	Simplification from (2)
8. $P(a) \wedge \neg B(a)$	Conjunction from (6) and (7)
9. $\exists x(P(x) \wedge \neg B(x))$	Existential generalization from (8)

Combining Rules of Inference for Propositions and Quantified Statements

We have developed rules of inference both for propositions and for quantified statements. Note that in our arguments in Examples 12 and 13 we used both universal instantiation, a rule of inference for quantified statements, and modus ponens, a rule of inference for propositional logic. We will often need to use this combination of rules of inference. Because universal instantiation and modus ponens are used so often together, this combination of rules is sometimes called **universal modus ponens**. This rule tells us that if $\forall x(P(x) \rightarrow Q(x))$ is true, and if $P(a)$ is true for a particular element a in the domain of the universal quantifier, then $Q(a)$ must also be true. To see this, note that by universal instantiation, $P(a) \rightarrow Q(a)$ is true. Then, by modus ponens, $Q(a)$ must also be true. We can describe universal modus ponens as follows:

$$\begin{array}{c} \forall x(P(x) \rightarrow Q(x)) \\ P(a), \text{ where } a \text{ is a particular element in the domain} \\ \therefore Q(a) \end{array}$$

Universal modus ponens is commonly used in mathematical arguments. This is illustrated in Example 14.

EXAMPLE 14 Assume that “For all positive integers n , if n is greater than 4, then n^2 is less than 2^n ” is true. Use universal modus ponens to show that $100^2 < 2^{100}$.

Solution: Let $P(n)$ denote “ $n > 4$ ” and $Q(n)$ denote “ $n^2 < 2^n$.” The statement “For all positive integers n , if n is greater than 4, then n^2 is less than 2^n ” can be represented by $\forall n(P(n) \rightarrow Q(n))$, where the domain consists of all positive integers. We are assuming that $\forall n(P(n) \rightarrow Q(n))$ is true. Note that $P(100)$ is true because $100 > 4$. It follows by universal modus ponens that $Q(100)$ is true, namely that $100^2 < 2^{100}$.

Another useful combination of a rule of inference from propositional logic and a rule of inference for quantified statements is **universal modus tollens**. Universal modus tollens

combines universal instantiation and modus tollens and can be expressed in the following way:

$$\begin{aligned} & \forall x(P(x) \rightarrow Q(x)) \\ & \neg Q(a), \text{ where } a \text{ is a particular element in the domain} \\ \therefore & \neg P(a) \end{aligned}$$

The verification of universal modus tollens is left as Exercise 25. Exercises 26–29 develop additional combinations of rules of inference in propositional logic and quantified statements.

Exercises

1. Find the argument form for the following argument and determine whether it is valid. Can we conclude that the conclusion is true if the premises are true?

If Socrates is human, then Socrates is mortal.
Socrates is human.
 \therefore Socrates is mortal.

2. Find the argument form for the following argument and determine whether it is valid. Can we conclude that the conclusion is true if the premises are true?

If George does not have eight legs, then he is not a spider.
George is a spider.
 \therefore George has eight legs.

3. What rule of inference is used in each of these arguments?

- a) Alice is a mathematics major. Therefore, Alice is either a mathematics major or a computer science major.
 - b) Jerry is a mathematics major and a computer science major. Therefore, Jerry is a mathematics major.
 - c) If it is rainy, then the pool will be closed. It is rainy. Therefore, the pool is closed.
 - d) If it snows today, the university will close. The university is not closed today. Therefore, it did not snow today.
 - e) If I go swimming, then I will stay in the sun too long. If I stay in the sun too long, then I will sunburn. Therefore, if I go swimming, then I will sunburn.
4. What rule of inference is used in each of these arguments?
- a) Kangaroos live in Australia and are marsupials. Therefore, kangaroos are marsupials.
 - b) It is either hotter than 100 degrees today or the pollution is dangerous. It is less than 100 degrees outside today. Therefore, the pollution is dangerous.
 - c) Linda is an excellent swimmer. If Linda is an excellent swimmer, then she can work as a lifeguard. Therefore, Linda can work as a lifeguard.
 - d) Steve will work at a computer company this summer. Therefore, this summer Steve will work at a computer company or he will be a beach bum.

- e) If I work all night on this homework, then I can answer all the exercises. If I answer all the exercises, I will understand the material. Therefore, if I work all night on this homework, then I will understand the material.

5. Use rules of inference to show that the hypotheses “Randy works hard,” “If Randy works hard, then he is a dull boy,” and “If Randy is a dull boy, then he will not get the job” imply the conclusion “Randy will not get the job.”

6. Use rules of inference to show that the hypotheses “If it does not rain or if it is not foggy, then the sailing race will be held and the lifesaving demonstration will go on,” “If the sailing race is held, then the trophy will be awarded,” and “The trophy was not awarded” imply the conclusion “It rained.”

7. What rules of inference are used in this famous argument? “All men are mortal. Socrates is a man. Therefore, Socrates is mortal.”

8. What rules of inference are used in this argument? “No man is an island. Manhattan is an island. Therefore, Manhattan is not a man.”

9. For each of these collections of premises, what relevant conclusion or conclusions can be drawn? Explain the rules of inference used to obtain each conclusion from the premises.

- a) “If I take the day off, it either rains or snows.” “I took Tuesday off or I took Thursday off.” “It was sunny on Tuesday.” “It did not snow on Thursday.”
- b) “If I eat spicy foods, then I have strange dreams.” “I have strange dreams if there is thunder while I sleep.” “I did not have strange dreams.”
- c) “I am either clever or lucky.” “I am not lucky.” “If I am lucky, then I will win the lottery.”
- d) “Every computer science major has a personal computer.” “Ralph does not have a personal computer.” “Ann has a personal computer.”
- e) “What is good for corporations is good for the United States.” “What is good for the United States is good for you.” “What is good for corporations is for you to buy lots of stuff.”
- f) “All rodents gnaw their food.” “Mice are rodents.” “Rabbits do not gnaw their food.” “Bats are not rodents.”

- 10.** For each of these sets of premises, what relevant conclusion or conclusions can be drawn? Explain the rules of inference used to obtain each conclusion from the premises.
- "If I play hockey, then I am sore the next day." "I use the whirlpool if I am sore." "I did not use the whirlpool."
 - "If I work, it is either sunny or partly sunny." "I worked last Monday or I worked last Friday." "It was not sunny on Tuesday." "It was not partly sunny on Friday."
 - "All insects have six legs." "Dragonflies are insects." "Spiders do not have six legs." "Spiders eat dragonflies."
 - "Every student has an Internet account." "Homer does not have an Internet account." "Maggie has an Internet account."
 - "All foods that are healthy to eat do not taste good." "Tofu is healthy to eat." "You only eat what tastes good." "You do not eat tofu." "Cheeseburgers are not healthy to eat."
 - "I am either dreaming or hallucinating." "I am not dreaming." "If I am hallucinating, I see elephants running down the road."
- 11.** Show that the argument form with premises p_1, p_2, \dots, p_n and conclusion $q \rightarrow r$ is valid if the argument form with premises p_1, p_2, \dots, p_n, q , and conclusion r is valid.
- 12.** Show that the argument form with premises $(p \wedge t) \rightarrow (r \vee s)$, $q \rightarrow (u \wedge t)$, $u \rightarrow p$, and $\neg s$ and conclusion $q \rightarrow r$ is valid by first using Exercise 11 and then using rules of inference from Table 1.
- 13.** For each of these arguments, explain which rules of inference are used for each step.
- "Doug, a student in this class, knows how to write programs in JAVA. Everyone who knows how to write programs in JAVA can get a high-paying job. Therefore, someone in this class can get a high-paying job."
 - "Somebody in this class enjoys whale watching. Every person who enjoys whale watching cares about ocean pollution. Therefore, there is a person in this class who cares about ocean pollution."
 - "Each of the 93 students in this class owns a personal computer. Everyone who owns a personal computer can use a word processing program. Therefore, Zeke, a student in this class, can use a word processing program."
 - "Everyone in New Jersey lives within 50 miles of the ocean. Someone in New Jersey has never seen the ocean. Therefore, someone who lives within 50 miles of the ocean has never seen the ocean."
- 14.** For each of these arguments, explain which rules of inference are used for each step.
- "Linda, a student in this class, owns a red convertible. Everyone who owns a red convertible has gotten at least one speeding ticket. Therefore, someone in this class has gotten a speeding ticket."
 - "Each of five roommates, Melissa, Aaron, Ralph, Veneesha, and Keeshawn, has taken a course in discrete mathematics. Every student who has taken a course in discrete mathematics can take a course in algorithms. Therefore, all five roommates can take a course in algorithms next year."
 - "All movies produced by John Sayles are wonderful. John Sayles produced a movie about coal miners. Therefore, there is a wonderful movie about coal miners."
 - "There is someone in this class who has been to France. Everyone who goes to France visits the Louvre. Therefore, someone in this class has visited the Louvre."
- 15.** For each of these arguments determine whether the argument is correct or incorrect and explain why.
- All students in this class understand logic. Xavier is a student in this class. Therefore, Xavier understands logic.
 - Every computer science major takes discrete mathematics. Natasha is taking discrete mathematics. Therefore, Natasha is a computer science major.
 - All parrots like fruit. My pet bird is not a parrot. Therefore, my pet bird does not like fruit.
 - Everyone who eats granola every day is healthy. Linda is not healthy. Therefore, Linda does not eat granola every day.
- 16.** For each of these arguments determine whether the argument is correct or incorrect and explain why.
- Everyone enrolled in the university has lived in a dormitory. Mia has never lived in a dormitory. Therefore, Mia is not enrolled in the university.
 - A convertible car is fun to drive. Isaac's car is not a convertible. Therefore, Isaac's car is not fun to drive.
 - Quincy likes all action movies. Quincy likes the movie *Eight Men Out*. Therefore, *Eight Men Out* is an action movie.
 - All lobsters set at least a dozen traps. Hamilton is a lobsterman. Therefore, Hamilton sets at least a dozen traps.
- 17.** What is wrong with this argument? Let $H(x)$ be " x is happy." Given the premise $\exists x H(x)$, we conclude that $H(\text{Lola})$. Therefore, Lola is happy.
- 18.** What is wrong with this argument? Let $S(x, y)$ be " x is shorter than y ." Given the premise $\exists s S(s, \text{Max})$, it follows that $S(\text{Max}, \text{Max})$. Then by existential generalization it follows that $\exists x S(x, x)$, so that someone is shorter than himself.
- 19.** Determine whether each of these arguments is valid. If an argument is correct, what rule of inference is being used? If it is not, what logical error occurs?
- If n is a real number such that $n > 1$, then $n^2 > 1$. Suppose that $n^2 > 1$. Then $n > 1$.
 - If n is a real number with $n > 3$, then $n^2 > 9$. Suppose that $n^2 \leq 9$. Then $n \leq 3$.
 - If n is a real number with $n > 2$, then $n^2 > 4$. Suppose that $n \leq 2$. Then $n^2 \leq 4$.

- 20.** Determine whether these are valid arguments.
- If x is a positive real number, then x^2 is a positive real number. Therefore, if a^2 is positive, where a is a real number, then a is a positive real number.
 - If $x^2 \neq 0$, where x is a real number, then $x \neq 0$. Let a be a real number with $a^2 \neq 0$; then $a \neq 0$.
- 21.** Which rules of inference are used to establish the conclusion of Lewis Carroll's argument described in Example 26 of Section 1.4?
- 22.** Which rules of inference are used to establish the conclusion of Lewis Carroll's argument described in Example 27 of Section 1.4?
- 23.** Identify the error or errors in this argument that supposedly shows that if $\exists x P(x) \wedge \exists x Q(x)$ is true then $\exists x(P(x) \wedge Q(x))$ is true.
- $\exists x P(x) \vee \exists x Q(x)$ Premise
 - $\exists x P(x)$ Simplification from (1)
 - $P(c)$ Existential instantiation from (2)
 - $\exists x Q(x)$ Simplification from (1)
 - $Q(c)$ Existential instantiation from (4)
 - $P(c) \wedge Q(c)$ Conjunction from (3) and (5)
 - $\exists x(P(x) \wedge Q(x))$ Existential generalization
- 24.** Identify the error or errors in this argument that supposedly shows that if $\forall x(P(x) \vee Q(x))$ is true then $\forall x P(x) \vee \forall x Q(x)$ is true.
- $\forall x(P(x) \vee Q(x))$ Premise
 - $P(c) \vee Q(c)$ Universal instantiation from (1)
 - $P(c)$ Simplification from (2)
 - $\forall x P(x)$ Universal generalization from (3)
 - $Q(c)$ Simplification from (2)
 - $\forall x Q(x)$ Universal generalization from (5)
 - $\forall x(P(x) \vee \forall x Q(x))$ Conjunction from (4) and (6)
- 25.** Justify the rule of universal modus tollens by showing that the premises $\forall x(P(x) \rightarrow Q(x))$ and $\neg Q(a)$ for a particular element a in the domain, imply $\neg P(a)$.
- 26.** Justify the rule of **universal transitivity**, which states that if $\forall x(P(x) \rightarrow Q(x))$ and $\forall x(Q(x) \rightarrow R(x))$ are true, then $\forall x(P(x) \rightarrow R(x))$ is true, where the domains of all quantifiers are the same.
- 27.** Use rules of inference to show that if $\forall x(P(x) \rightarrow (Q(x) \wedge S(x)))$ and $\forall x(P(x) \wedge R(x))$ are true, then $\forall x(R(x) \wedge S(x))$ is true.
- 28.** Use rules of inference to show that if $\forall x(P(x) \vee Q(x))$ and $\forall x((\neg P(x) \wedge Q(x)) \rightarrow R(x))$ are true, then $\forall x(\neg R(x) \rightarrow P(x))$ is also true, where the domains of all quantifiers are the same.
- 29.** Use rules of inference to show that if $\forall x(P(x) \vee Q(x))$, $\forall x(\neg Q(x) \vee S(x))$, $\forall x(R(x) \rightarrow \neg S(x))$, and $\exists x \neg P(x)$ are true, then $\exists x \neg R(x)$ is true.
- 30.** Use resolution to show the hypotheses "Allen is a bad boy or Hillary is a good girl" and "Allen is a good boy or David is happy" imply the conclusion "Hillary is a good girl or David is happy."
- 31.** Use resolution to show that the hypotheses "It is not raining or Yvette has her umbrella," "Yvette does not have her umbrella or she does not get wet," and "It is raining or Yvette does not get wet" imply that "Yvette does not get wet."
- 32.** Show that the equivalence $p \wedge \neg p \equiv \mathbf{F}$ can be derived using resolution together with the fact that a conditional statement with a false hypothesis is true. [Hint: Let $q = r = \mathbf{F}$ in resolution.]
- 33.** Use resolution to show that the compound proposition $(p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$ is not satisfiable.
- *34.** The Logic Problem, taken from *WFF'N PROOF, The Game of Logic*, has these two assumptions:
- "Logic is difficult or not many students like logic."
 - "If mathematics is easy, then logic is not difficult."
- By translating these assumptions into statements involving propositional variables and logical connectives, determine whether each of the following are valid conclusions of these assumptions:
- That mathematics is not easy, if many students like logic.
 - That not many students like logic, if mathematics is not easy.
 - That mathematics is not easy or logic is difficult.
 - That logic is not difficult or mathematics is not easy.
 - That if not many students like logic, then either mathematics is not easy or logic is not difficult.
- *35.** Determine whether this argument, taken from Kalish and Montague [KaMo64], is valid.
- If Superman were able and willing to prevent evil, he would do so. If Superman were unable to prevent evil, he would be impotent; if he were unwilling to prevent evil, he would be malevolent. Superman does not prevent evil. If Superman exists, he is neither impotent nor malevolent. Therefore, Superman does not exist.

1.7 Introduction to Proofs

Introduction

In this section we introduce the notion of a proof and describe methods for constructing proofs. A proof is a valid argument that establishes the truth of a mathematical statement. A proof can use the hypotheses of the theorem, if any, axioms assumed to be true, and previously proven

theorems. Using these ingredients and rules of inference, the final step of the proof establishes the truth of the statement being proved.

In our discussion we move from formal proofs of theorems toward more informal proofs. The arguments we introduced in Section 1.6 to show that statements involving propositions and quantified statements are true were formal proofs, where all steps were supplied, and the rules for each step in the argument were given. However, formal proofs of useful theorems can be extremely long and hard to follow. In practice, the proofs of theorems designed for human consumption are almost always **informal proofs**, where more than one rule of inference may be used in each step, where steps may be skipped, where the axioms being assumed and the rules of inference used are not explicitly stated. Informal proofs can often explain to humans why theorems are true, while computers are perfectly happy producing formal proofs using automated reasoning systems.

The methods of proof discussed in this chapter are important not only because they are used to prove mathematical theorems, but also for their many applications to computer science. These applications include verifying that computer programs are correct, establishing that operating systems are secure, making inferences in artificial intelligence, showing that system specifications are consistent, and so on. Consequently, understanding the techniques used in proofs is essential both in mathematics and in computer science.

Some Terminology



Formally, a **theorem** is a statement that can be shown to be true. In mathematical writing, the term theorem is usually reserved for a statement that is considered at least somewhat important. Less important theorems sometimes are called **propositions**. (Theorems can also be referred to as **facts** or **results**.) A theorem may be the universal quantification of a conditional statement with one or more premises and a conclusion. However, it may be some other type of logical statement, as the examples later in this chapter will show. We demonstrate that a theorem is true with a **proof**. A proof is a valid argument that establishes the truth of a theorem. The statements used in a proof can include **axioms** (or **postulates**), which are statements we assume to be true (for example, the axioms for the real numbers, given in Appendix 1, and the axioms of plane geometry), the premises, if any, of the theorem, and previously proven theorems. Axioms may be stated using primitive terms that do not require definition, but all other terms used in theorems and their proofs must be defined. Rules of inference, together with definitions of terms, are used to draw conclusions from other assertions, tying together the steps of a proof. In practice, the final step of a proof is usually just the conclusion of the theorem. However, for clarity, we will often recap the statement of the theorem as the final step of a proof.

A less important theorem that is helpful in the proof of other results is called a **lemma** (plural *lemmas* or *lemmata*). Complicated proofs are usually easier to understand when they are proved using a series of lemmas, where each lemma is proved individually. A **corollary** is a theorem that can be established directly from a theorem that has been proved. A **conjecture** is a statement that is being proposed to be a true statement, usually on the basis of some partial evidence, a heuristic argument, or the intuition of an expert. When a proof of a conjecture is found, the conjecture becomes a theorem. Many times conjectures are shown to be false, so they are not theorems.

Understanding How Theorems Are Stated



Before we introduce methods for proving theorems, we need to understand how many mathematical theorems are stated. Many theorems assert that a property holds for all elements in a domain, such as the integers or the real numbers. Although the precise statement of such

theorems needs to include a universal quantifier, the standard convention in mathematics is to omit it. For example, the statement

“If $x > y$, where x and y are positive real numbers, then $x^2 > y^2$.”

really means

“For all positive real numbers x and y , if $x > y$, then $x^2 > y^2$.”

Furthermore, when theorems of this type are proved, the first step of the proof usually involves selecting a general element of the domain. Subsequent steps show that this element has the property in question. Finally, universal generalization implies that the theorem holds for all members of the domain.

Methods of Proving Theorems



Proving mathematical theorems can be difficult. To construct proofs we need all available ammunition, including a powerful battery of different proof methods. These methods provide the overall approach and strategy of proofs. Understanding these methods is a key component of learning how to read and construct mathematical proofs. Once we have chosen a proof method, we use axioms, definitions of terms, previously proved results, and rules of inference to complete the proof. Note that in this book we will always assume the axioms for real numbers found in Appendix 1. We will also assume the usual axioms whenever we prove a result about geometry. When you construct your own proofs, be careful not to use anything but these axioms, definitions, and previously proved results as facts!

To prove a theorem of the form $\forall x(P(x) \rightarrow Q(x))$, our goal is to show that $P(c) \rightarrow Q(c)$ is true, where c is an arbitrary element of the domain, and then apply universal generalization. In this proof, we need to show that a conditional statement is true. Because of this, we now focus on methods that show that conditional statements are true. Recall that $p \rightarrow q$ is true unless p is true but q is false. Note that to prove the statement $p \rightarrow q$, we need only show that q is true if p is true. The following discussion will give the most common techniques for proving conditional statements. Later we will discuss methods for proving other types of statements. In this section, and in Section 1.8, we will develop a large arsenal of proof techniques that can be used to prove a wide variety of theorems.

When you read proofs, you will often find the words “obviously” or “clearly.” These words indicate that steps have been omitted that the author expects the reader to be able to fill in. Unfortunately, this assumption is often not warranted and readers are not at all sure how to fill in the gaps. We will assiduously try to avoid using these words and try not to omit too many steps. However, if we included all steps in proofs, our proofs would often be excruciatingly long.

Direct Proofs

A **direct proof** of a conditional statement $p \rightarrow q$ is constructed when the first step is the assumption that p is true; subsequent steps are constructed using rules of inference, with the final step showing that q must also be true. A direct proof shows that a conditional statement $p \rightarrow q$ is true by showing that if p is true, then q must also be true, so that the combination p true and q false never occurs. In a direct proof, we assume that p is true and use axioms, definitions, and previously proven theorems, together with rules of inference, to show that q must also be true. You will find that direct proofs of many results are quite straightforward, with a fairly obvious sequence of steps leading from the hypothesis to the conclusion. However, direct proofs sometimes require particular insights and can be quite tricky. The first direct proofs we present here are quite straightforward; later in the text you will see some that are less obvious.

We will provide examples of several different direct proofs. Before we give the first example, we need to define some terminology.

DEFINITION 1

The integer n is *even* if there exists an integer k such that $n = 2k$, and n is *odd* if there exists an integer k such that $n = 2k + 1$. (Note that every integer is either even or odd, and no integer is both even and odd.) Two integers have the *same parity* when both are even or both are odd; they have *opposite parity* when one is even and the other is odd.

EXAMPLE 1

Give a direct proof of the theorem “If n is an odd integer, then n^2 is odd.”



Solution: Note that this theorem states $\forall n P((n) \rightarrow Q(n))$, where $P(n)$ is “ n is an odd integer” and $Q(n)$ is “ n^2 is odd.” As we have said, we will follow the usual convention in mathematical proofs by showing that $P(n)$ implies $Q(n)$, and not explicitly using universal instantiation. To begin a direct proof of this theorem, we assume that the hypothesis of this conditional statement is true, namely, we assume that n is odd. By the definition of an odd integer, it follows that $n = 2k + 1$, where k is some integer. We want to show that n^2 is also odd. We can square both sides of the equation $n = 2k + 1$ to obtain a new equation that expresses n^2 . When we do this, we find that $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$. By the definition of an odd integer, we can conclude that n^2 is an odd integer (it is one more than twice an integer). Consequently, we have proved that if n is an odd integer, then n^2 is an odd integer. \blacktriangleleft

EXAMPLE 2

Give a direct proof that if m and n are both perfect squares, then mn is also a perfect square. (An integer a is a **perfect square** if there is an integer b such that $a = b^2$.)

Solution: To produce a direct proof of this theorem, we assume that the hypothesis of this conditional statement is true, namely, we assume that m and n are both perfect squares. By the definition of a perfect square, it follows that there are integers s and t such that $m = s^2$ and $n = t^2$. The goal of the proof is to show that mn must also be a perfect square when m and n are; looking ahead we see how we can show this by substituting s^2 for m and t^2 for n into mn . This tells us that $mn = s^2t^2$. Hence, $mn = s^2t^2 = (ss)(tt) = (st)(st) = (st)^2$, using commutativity and associativity of multiplication. By the definition of perfect square, it follows that mn is also a perfect square, because it is the square of st , which is an integer. We have proved that if m and n are both perfect squares, then mn is also a perfect square. \blacktriangleleft

Proof by Contraposition

Direct proofs lead from the premises of a theorem to the conclusion. They begin with the premises, continue with a sequence of deductions, and end with the conclusion. However, we will see that attempts at direct proofs often reach dead ends. We need other methods of proving theorems of the form $\forall x(P(x) \rightarrow Q(x))$. Proofs of theorems of this type that are not direct proofs, that is, that do not start with the premises and end with the conclusion, are called **indirect proofs**.

An extremely useful type of indirect proof is known as **proof by contraposition**. Proofs by contraposition make use of the fact that the conditional statement $p \rightarrow q$ is equivalent to its contrapositive, $\neg q \rightarrow \neg p$. This means that the conditional statement $p \rightarrow q$ can be proved by showing that its contrapositive, $\neg q \rightarrow \neg p$, is true. In a proof by contraposition of $p \rightarrow q$, we take $\neg q$ as a premise, and using axioms, definitions, and previously proven theorems, together with rules of inference, we show that $\neg p$ must follow. We will illustrate proof by contraposition with two examples. These examples show that proof by contraposition can succeed when we cannot easily find a direct proof.

EXAMPLE 3

Prove that if n is an integer and $3n + 2$ is odd, then n is odd.

Solution: We first attempt a direct proof. To construct a direct proof, we first assume that $3n + 2$ is an odd integer. This means that $3n + 2 = 2k + 1$ for some integer k . Can we use this fact

Extra Examples

to show that n is odd? We see that $3n + 1 = 2k$, but there does not seem to be any direct way to conclude that n is odd. Because our attempt at a direct proof failed, we next try a proof by contraposition.

The first step in a proof by contraposition is to assume that the conclusion of the conditional statement “If $3n + 2$ is odd, then n is odd” is false; namely, assume that n is even. Then, by the definition of an even integer, $n = 2k$ for some integer k . Substituting $2k$ for n , we find that $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$. This tells us that $3n + 2$ is even (because it is a multiple of 2), and therefore not odd. This is the negation of the premise of the theorem. Because the negation of the conclusion of the conditional statement implies that the hypothesis is false, the original conditional statement is true. Our proof by contraposition succeeded; we have proved the theorem “If $3n + 2$ is odd, then n is odd.” 

EXAMPLE 4 Prove that if $n = ab$, where a and b are positive integers, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.

Solution: Because there is no obvious way of showing that $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$ directly from the equation $n = ab$, where a and b are positive integers, we attempt a proof by contraposition.

The first step in a proof by contraposition is to assume that the conclusion of the conditional statement “If $n = ab$, where a and b are positive integers, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$ ” is false. That is, we assume that the statement $(a \leq \sqrt{n}) \vee (b \leq \sqrt{n})$ is false. Using the meaning of disjunction together with De Morgan’s law, we see that this implies that both $a \leq \sqrt{n}$ and $b \leq \sqrt{n}$ are false. This implies that $a > \sqrt{n}$ and $b > \sqrt{n}$. We can multiply these inequalities together (using the fact that if $0 < s < t$ and $0 < u < v$, then $su < tv$) to obtain $ab > \sqrt{n} \cdot \sqrt{n} = n$. This shows that $ab \neq n$, which contradicts the statement $n = ab$.

Because the negation of the conclusion of the conditional statement implies that the hypothesis is false, the original conditional statement is true. Our proof by contraposition succeeded; we have proved that if $n = ab$, where a and b are positive integers, then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$. 

VACUOUS AND TRIVIAL PROOFS We can quickly prove that a conditional statement $p \rightarrow q$ is true when we know that p is false, because $p \rightarrow q$ must be true when p is false. Consequently, if we can show that p is false, then we have a proof, called a **vacuous proof**, of the conditional statement $p \rightarrow q$. Vacuous proofs are often used to establish special cases of theorems that state that a conditional statement is true for all positive integers [i.e., a theorem of the kind $\forall n P(n)$, where $P(n)$ is a propositional function]. Proof techniques for theorems of this kind will be discussed in Section 5.1.

EXAMPLE 5 Show that the proposition $P(0)$ is true, where $P(n)$ is “If $n > 1$, then $n^2 > n$ ” and the domain consists of all integers.

Solution: Note that $P(0)$ is “If $0 > 1$, then $0^2 > 0$.” We can show $P(0)$ using a vacuous proof. Indeed, the hypothesis $0 > 1$ is false. This tells us that $P(0)$ is automatically true. 

Remark: The fact that the conclusion of this conditional statement, $0^2 > 0$, is false is irrelevant to the truth value of the conditional statement, because a conditional statement with a false hypothesis is guaranteed to be true.

We can also quickly prove a conditional statement $p \rightarrow q$ if we know that the conclusion q is true. By showing that q is true, it follows that $p \rightarrow q$ must also be true. A proof of $p \rightarrow q$ that uses the fact that q is true is called a **trivial proof**. Trivial proofs are often important when special cases of theorems are proved (see the discussion of proof by cases in Section 1.8) and in mathematical induction, which is a proof technique discussed in Section 5.1.

EXAMPLE 6 Let $P(n)$ be “If a and b are positive integers with $a \geq b$, then $a^n \geq b^n$,” where the domain consists of all nonnegative integers. Show that $P(0)$ is true.

Solution: The proposition $P(0)$ is “If $a \geq b$, then $a^0 \geq b^0$.” Because $a^0 = b^0 = 1$, the conclusion of the conditional statement “If $a \geq b$, then $a^0 \geq b^0$ ” is true. Hence, this conditional statement, which is $P(0)$, is true. This is an example of a trivial proof. Note that the hypothesis, which is the statement “ $a \geq b$,” was not needed in this proof. 

A LITTLE PROOF STRATEGY We have described two important approaches for proving theorems of the form $\forall x(P(x) \rightarrow Q(x))$: direct proof and proof by contraposition. We have also given examples that show how each is used. However, when you are presented with a theorem of the form $\forall x(P(x) \rightarrow Q(x))$, which method should you use to attempt to prove it? We will provide a few rules of thumb here; in Section 1.8 we will discuss proof strategy at greater length. When you want to prove a statement of the form $\forall x(P(x) \rightarrow Q(x))$, first evaluate whether a direct proof looks promising. Begin by expanding the definitions in the hypotheses. Start to reason using these hypotheses, together with axioms and available theorems. If a direct proof does not seem to go anywhere, try the same thing with a proof by contraposition. Recall that in a proof by contraposition you assume that the conclusion of the conditional statement is false and use a direct proof to show this implies that the hypothesis must be false. We illustrate this strategy in Examples 7 and 8. Before we present our next example, we need a definition.

DEFINITION 2

The real number r is *rational* if there exist integers p and q with $q \neq 0$ such that $r = p/q$. A real number that is not rational is called *irrational*.

EXAMPLE 7

Prove that the sum of two rational numbers is rational. (Note that if we include the implicit quantifiers here, the theorem we want to prove is “For every real number r and every real number s , if r and s are rational numbers, then $r + s$ is rational.”)



Solution: We first attempt a direct proof. To begin, suppose that r and s are rational numbers. From the definition of a rational number, it follows that there are integers p and q , with $q \neq 0$, such that $r = p/q$, and integers t and u , with $u \neq 0$, such that $s = t/u$. Can we use this information to show that $r + s$ is rational? The obvious next step is to add $r = p/q$ and $s = t/u$, to obtain

$$r + s = \frac{p}{q} + \frac{t}{u} = \frac{pu + qt}{qu}.$$

Because $q \neq 0$ and $u \neq 0$, it follows that $qu \neq 0$. Consequently, we have expressed $r + s$ as the ratio of two integers, $pu + qt$ and qu , where $qu \neq 0$. This means that $r + s$ is rational. We have proved that the sum of two rational numbers is rational; our attempt to find a direct proof succeeded. 

EXAMPLE 8

Prove that if n is an integer and n^2 is odd, then n is odd.

Solution: We first attempt a direct proof. Suppose that n is an integer and n^2 is odd. Then, there exists an integer k such that $n^2 = 2k + 1$. Can we use this information to show that n is odd? There seems to be no obvious approach to show that n is odd because solving for n produces the equation $n = \pm\sqrt{2k + 1}$, which is not terribly useful.

Because this attempt to use a direct proof did not bear fruit, we next attempt a proof by contraposition. We take as our hypothesis the statement that n is not odd. Because every integer is odd or even, this means that n is even. This implies that there exists an integer k such that $n = 2k$. To prove the theorem, we need to show that this hypothesis implies the conclusion that n^2 is not odd, that is, that n^2 is even. Can we use the equation $n = 2k$ to achieve this? By

squaring both sides of this equation, we obtain $n^2 = 4k^2 = 2(2k^2)$, which implies that n^2 is also even because $n^2 = 2t$, where $t = 2k^2$. We have proved that if n is an integer and n^2 is odd, then n is odd. Our attempt to find a proof by contraposition succeeded. 

Proofs by Contradiction

Suppose we want to prove that a statement p is true. Furthermore, suppose that we can find a contradiction q such that $\neg p \rightarrow q$ is true. Because q is false, but $\neg p \rightarrow q$ is true, we can conclude that $\neg p$ is false, which means that p is true. How can we find a contradiction q that might help us prove that p is true in this way?

Because the statement $r \wedge \neg r$ is a contradiction whenever r is a proposition, we can prove that p is true if we can show that $\neg p \rightarrow (r \wedge \neg r)$ is true for some proposition r . Proofs of this type are called **proofs by contradiction**. Because a proof by contradiction does not prove a result directly, it is another type of indirect proof. We provide three examples of proof by contradiction. The first is an example of an application of the pigeonhole principle, a combinatorial technique that we will cover in depth in Section 6.2.

EXAMPLE 9 Show that at least four of any 22 days must fall on the same day of the week.



Solution: Let p be the proposition “At least four of 22 chosen days fall on the same day of the week.” Suppose that $\neg p$ is true. This means that at most three of the 22 days fall on the same day of the week. Because there are seven days of the week, this implies that at most 21 days could have been chosen, as for each of the days of the week, at most three of the chosen days could fall on that day. This contradicts the premise that we have 22 days under consideration. That is, if r is the statement that 22 days are chosen, then we have shown that $\neg p \rightarrow (r \wedge \neg r)$. Consequently, we know that p is true. We have proved that at least four of 22 chosen days fall on the same day of the week. 

EXAMPLE 10 Prove that $\sqrt{2}$ is irrational by giving a proof by contradiction.

Solution: Let p be the proposition “ $\sqrt{2}$ is irrational.” To start a proof by contradiction, we suppose that $\neg p$ is true. Note that $\neg p$ is the statement “It is not the case that $\sqrt{2}$ is irrational,” which says that $\sqrt{2}$ is rational. We will show that assuming that $\neg p$ is true leads to a contradiction.

If $\sqrt{2}$ is rational, there exist integers a and b with $\sqrt{2} = a/b$, where $b \neq 0$ and a and b have no common factors (so that the fraction a/b is in lowest terms.) (Here, we are using the fact that every rational number can be written in lowest terms.) Because $\sqrt{2} = a/b$, when both sides of this equation are squared, it follows that

$$2 = \frac{a^2}{b^2}.$$

Hence,

$$2b^2 = a^2.$$

By the definition of an even integer it follows that a^2 is even. We next use the fact that if a^2 is even, a must also be even, which follows by Exercise 16. Furthermore, because a is even, by the definition of an even integer, $a = 2c$ for some integer c . Thus,

$$2b^2 = 4c^2.$$

Dividing both sides of this equation by 2 gives

$$b^2 = 2c^2.$$

By the definition of even, this means that b^2 is even. Again using the fact that if the square of an integer is even, then the integer itself must be even, we conclude that b must be even as well.

We have now shown that the assumption of $\neg p$ leads to the equation $\sqrt{2} = a/b$, where a and b have no common factors, but both a and b are even, that is, 2 divides both a and b . Note that the statement that $\sqrt{2} = a/b$, where a and b have no common factors, means, in particular, that 2 does not divide both a and b . Because our assumption of $\neg p$ leads to the contradiction that 2 divides both a and b and 2 does not divide both a and b , $\neg p$ must be false. That is, the statement p , “ $\sqrt{2}$ is irrational,” is true. We have proved that $\sqrt{2}$ is irrational. 

Proof by contradiction can be used to prove conditional statements. In such proofs, we first assume the negation of the conclusion. We then use the premises of the theorem and the negation of the conclusion to arrive at a contradiction. (The reason that such proofs are valid rests on the logical equivalence of $p \rightarrow q$ and $(p \wedge \neg q) \rightarrow F$. To see that these statements are equivalent, simply note that each is false in exactly one case, namely when p is true and q is false.)

Note that we can rewrite a proof by contraposition of a conditional statement as a proof by contradiction. In a proof of $p \rightarrow q$ by contraposition, we assume that $\neg q$ is true. We then show that $\neg p$ must also be true. To rewrite a proof by contraposition of $p \rightarrow q$ as a proof by contradiction, we suppose that both p and $\neg q$ are true. Then, we use the steps from the proof of $\neg q \rightarrow \neg p$ to show that $\neg p$ is true. This leads to the contradiction $p \wedge \neg p$, completing the proof. Example 11 illustrates how a proof by contraposition of a conditional statement can be rewritten as a proof by contradiction.

EXAMPLE 11 Give a proof by contradiction of the theorem “If $3n + 2$ is odd, then n is odd.”

Solution: Let p be “ $3n + 2$ is odd” and q be “ n is odd.” To construct a proof by contradiction, assume that both p and $\neg q$ are true. That is, assume that $3n + 2$ is odd and that n is not odd. Because n is not odd, we know that it is even. Because n is even, there is an integer k such that $n = 2k$. This implies that $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$. Because $3n + 2$ is $2t$, where $t = 3k + 1$, $3n + 2$ is even. Note that the statement “ $3n + 2$ is even” is equivalent to the statement $\neg p$, because an integer is even if and only if it is not odd. Because both p and $\neg p$ are true, we have a contradiction. This completes the proof by contradiction, proving that if $3n + 2$ is odd, then n is odd. 

Note that we can also prove by contradiction that $p \rightarrow q$ is true by assuming that p and $\neg q$ are true, and showing that q must be also be true. This implies that $\neg q$ and q are both true, a contradiction. This observation tells us that we can turn a direct proof into a proof by contradiction.

PROOFS OF EQUIVALENCE To prove a theorem that is a biconditional statement, that is, a statement of the form $p \leftrightarrow q$, we show that $p \rightarrow q$ and $q \rightarrow p$ are both true. The validity of this approach is based on the tautology

$$(p \leftrightarrow q) \leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p).$$

EXAMPLE 12 Prove the theorem “If n is an integer, then n is odd if and only if n^2 is odd.”

Solution: This theorem has the form “ p if and only if q ,” where p is “ n is odd” and q is “ n^2 is odd.” (As usual, we do not explicitly deal with the universal quantification.) To prove this theorem, we need to show that $p \rightarrow q$ and $q \rightarrow p$ are true.



We have already shown (in Example 1) that $p \rightarrow q$ is true and (in Example 8) that $q \rightarrow p$ is true.

Because we have shown that both $p \rightarrow q$ and $q \rightarrow p$ are true, we have shown that the theorem is true. 

Sometimes a theorem states that several propositions are equivalent. Such a theorem states that propositions $p_1, p_2, p_3, \dots, p_n$ are equivalent. This can be written as

$$p_1 \leftrightarrow p_2 \leftrightarrow \dots \leftrightarrow p_n,$$

which states that all n propositions have the same truth values, and consequently, that for all i and j with $1 \leq i \leq n$ and $1 \leq j \leq n$, p_i and p_j are equivalent. One way to prove these mutually equivalent is to use the tautology

$$p_1 \leftrightarrow p_2 \leftrightarrow \dots \leftrightarrow p_n \leftrightarrow (p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \dots \wedge (p_n \rightarrow p_1).$$

This shows that if the n conditional statements $p_1 \rightarrow p_2, p_2 \rightarrow p_3, \dots, p_n \rightarrow p_1$ can be shown to be true, then the propositions p_1, p_2, \dots, p_n are all equivalent.

This is much more efficient than proving that $p_i \rightarrow p_j$ for all $i \neq j$ with $1 \leq i \leq n$ and $1 \leq j \leq n$. (Note that there are $n^2 - n$ such conditional statements.)

When we prove that a group of statements are equivalent, we can establish any chain of conditional statements we choose as long as it is possible to work through the chain to go from any one of these statements to any other statement. For example, we can show that p_1, p_2 , and p_3 are equivalent by showing that $p_1 \rightarrow p_3, p_3 \rightarrow p_2$, and $p_2 \rightarrow p_1$.

EXAMPLE 13 Show that these statements about the integer n are equivalent:

- p_1 : n is even.
- p_2 : $n - 1$ is odd.
- p_3 : n^2 is even.

Solution: We will show that these three statements are equivalent by showing that the conditional statements $p_1 \rightarrow p_2, p_2 \rightarrow p_3$, and $p_3 \rightarrow p_1$ are true.

We use a direct proof to show that $p_1 \rightarrow p_2$. Suppose that n is even. Then $n = 2k$ for some integer k . Consequently, $n - 1 = 2k - 1 = 2(k - 1) + 1$. This means that $n - 1$ is odd because it is of the form $2m + 1$, where m is the integer $k - 1$.

We also use a direct proof to show that $p_2 \rightarrow p_3$. Now suppose $n - 1$ is odd. Then $n - 1 = 2k + 1$ for some integer k . Hence, $n = 2k + 2$ so that $n^2 = (2k + 2)^2 = 4k^2 + 8k + 4 = 2(2k^2 + 4k + 2)$. This means that n^2 is twice the integer $2k^2 + 4k + 2$, and hence is even.

To prove $p_3 \rightarrow p_1$, we use a proof by contraposition. That is, we prove that if n is not even, then n^2 is not even. This is the same as proving that if n is odd, then n^2 is odd, which we have already done in Example 1. This completes the proof. 

COUNTEREXAMPLES In Section 1.4 we stated that to show that a statement of the form $\forall x P(x)$ is false, we need only find a **counterexample**, that is, an example x for which $P(x)$ is false. When presented with a statement of the form $\forall x P(x)$, which we believe to be false or which has resisted all proof attempts, we look for a counterexample. We illustrate the use of counterexamples in Example 14.

EXAMPLE 14 Show that the statement “Every positive integer is the sum of the squares of two integers” is false.



Solution: To show that this statement is false, we look for a counterexample, which is a particular integer that is not the sum of the squares of two integers. It does not take long to find a counterexample, because 3 cannot be written as the sum of the squares of two integers. To show this is the case, note that the only perfect squares not exceeding 3 are $0^2 = 0$ and $1^2 = 1$. Furthermore, there is no way to get 3 as the sum of two terms each of which is 0 or 1. Consequently, we have shown that “Every positive integer is the sum of the squares of two integers” is false. 

Mistakes in Proofs

There are many common errors made in constructing mathematical proofs. We will briefly describe some of these here. Among the most common errors are mistakes in arithmetic and basic algebra. Even professional mathematicians make such errors, especially when working with complicated formulae. Whenever you use such computations you should check them as carefully as possible. (You should also review any troublesome aspects of basic algebra, especially before you study Section 5.1.)



Each step of a mathematical proof needs to be correct and the conclusion needs to follow logically from the steps that precede it. Many mistakes result from the introduction of steps that do not logically follow from those that precede it. This is illustrated in Examples 15–17.

EXAMPLE 15 What is wrong with this famous supposed “proof” that $1 = 2$?

“Proof:” We use these steps, where a and b are two equal positive integers.

Step	Reason
1. $a = b$	Given
2. $a^2 = ab$	Multiply both sides of (1) by a
3. $a^2 - b^2 = ab - b^2$	Subtract b^2 from both sides of (2)
4. $(a - b)(a + b) = b(a - b)$	Factor both sides of (3)
5. $a + b = b$	Divide both sides of (4) by $a - b$
6. $2b = b$	Replace a by b in (5) because $a = b$ and simplify
7. $2 = 1$	Divide both sides of (6) by b

Solution: Every step is valid except for one, step 5 where we divided both sides by $a - b$. The error is that $a - b$ equals zero; division of both sides of an equation by the same quantity is valid as long as this quantity is not zero. 

EXAMPLE 16 What is wrong with this “proof”?

“Theorem:” If n^2 is positive, then n is positive.

“Proof:” Suppose that n^2 is positive. Because the conditional statement “If n is positive, then n^2 is positive” is true, we can conclude that n is positive.

Solution: Let $P(n)$ be “ n is positive” and $Q(n)$ be “ n^2 is positive.” Then our hypothesis is $Q(n)$. The statement “If n is positive, then n^2 is positive” is the statement $\forall n(P(n) \rightarrow Q(n))$. From the hypothesis $Q(n)$ and the statement $\forall n(P(n) \rightarrow Q(n))$ we cannot conclude $P(n)$, because we are not using a valid rule of inference. Instead, this is an example of the fallacy of affirming the conclusion. A counterexample is supplied by $n = -1$ for which $n^2 = 1$ is positive, but n is negative. 

EXAMPLE 17 What is wrong with this “proof”?

“Theorem:” If n is not positive, then n^2 is not positive. (This is the contrapositive of the “theorem” in Example 16.)

"Proof:" Suppose that n is not positive. Because the conditional statement ‘‘If n is positive, then n^2 is positive’’ is true, we can conclude that n^2 is not positive.

Solution: Let $P(n)$ and $Q(n)$ be as in the solution of Example 16. Then our hypothesis is $\neg P(n)$ and the statement ‘‘If n is positive, then n^2 is positive’’ is the statement $\forall n(P(n) \rightarrow Q(n))$. From the hypothesis $\neg P(n)$ and the statement $\forall n(P(n) \rightarrow Q(n))$ we cannot conclude $\neg Q(n)$, because we are not using a valid rule of inference. Instead, this is an example of the fallacy of denying the hypothesis. A counterexample is supplied by $n = -1$, as in Example 16. \blacktriangleleft

Finally, we briefly discuss a particularly nasty type of error. Many incorrect arguments are based on a fallacy called **begging the question**. This fallacy occurs when one or more steps of a proof are based on the truth of the statement being proved. In other words, this fallacy arises when a statement is proved using itself, or a statement equivalent to it. That is why this fallacy is also called **circular reasoning**.

EXAMPLE 18 Is the following argument correct? It supposedly shows that n is an even integer whenever n^2 is an even integer.

Suppose that n^2 is even. Then $n^2 = 2k$ for some integer k . Let $n = 2l$ for some integer l . This shows that n is even.

Solution: This argument is incorrect. The statement ‘‘let $n = 2l$ for some integer l ’’ occurs in the proof. No argument has been given to show that n can be written as $2l$ for some integer l . This is circular reasoning because this statement is equivalent to the statement being proved, namely, ‘‘ n is even.’’ Of course, the result itself is correct; only the method of proof is wrong. \blacktriangleleft

Making mistakes in proofs is part of the learning process. When you make a mistake that someone else finds, you should carefully analyze where you went wrong and make sure that you do not make the same mistake again. Even professional mathematicians make mistakes in proofs. More than a few incorrect proofs of important results have fooled people for many years before subtle errors in them were found.

Just a Beginning

We have now developed a basic arsenal of proof methods. In the next section we will introduce other important proof methods. We will also introduce several important proof techniques in Chapter 5, including mathematical induction, which can be used to prove results that hold for all positive integers. In Chapter 6 we will introduce the notion of combinatorial proofs.

In this section we introduced several methods for proving theorems of the form $\forall x(P(x) \rightarrow Q(x))$, including direct proofs and proofs by contraposition. There are many theorems of this type whose proofs are easy to construct by directly working through the hypotheses and definitions of the terms of the theorem. However, it is often difficult to prove a theorem without resorting to a clever use of a proof by contraposition or a proof by contradiction, or some other proof technique. In Section 1.8 we will address proof strategy. We will describe various approaches that can be used to find proofs when straightforward approaches do not work. Constructing proofs is an art that can be learned only through experience, including writing proofs, having your proofs critiqued, and reading and analyzing other proofs.

Exercises

1. Use a direct proof to show that the sum of two odd integers is even.
2. Use a direct proof to show that the sum of two even integers is even.
3. Show that the square of an even number is an even number using a direct proof.
4. Show that the additive inverse, or negative, of an even number is an even number using a direct proof.
5. Prove that if $m + n$ and $n + p$ are even integers, where m , n , and p are integers, then $m + p$ is even. What kind of proof did you use?
6. Use a direct proof to show that the product of two odd numbers is odd.
7. Use a direct proof to show that every odd integer is the difference of two squares.
8. Prove that if n is a perfect square, then $n + 2$ is not a perfect square.
9. Use a proof by contradiction to prove that the sum of an irrational number and a rational number is irrational.
10. Use a direct proof to show that the product of two rational numbers is rational.
11. Prove or disprove that the product of two irrational numbers is irrational.
12. Prove or disprove that the product of a nonzero rational number and an irrational number is irrational.
13. Prove that if x is irrational, then $1/x$ is irrational.
14. Prove that if x is rational and $x \neq 0$, then $1/x$ is rational.
15. Use a proof by contraposition to show that if $x + y \geq 2$, where x and y are real numbers, then $x \geq 1$ or $y \geq 1$.
16. Prove that if m and n are integers and mn is even, then m is even or n is even.
17. Show that if n is an integer and $n^3 + 5$ is odd, then n is even using
 - a proof by contraposition.
 - a proof by contradiction.
18. Prove that if n is an integer and $3n + 2$ is even, then n is even using
 - a proof by contraposition.
 - a proof by contradiction.
19. Prove the proposition $P(0)$, where $P(n)$ is the proposition “If n is a positive integer greater than 1, then $n^2 > n$.” What kind of proof did you use?
20. Prove the proposition $P(1)$, where $P(n)$ is the proposition “If n is a positive integer, then $n^2 \geq n$.” What kind of proof did you use?
21. Let $P(n)$ be the proposition “If a and b are positive real numbers, then $(a + b)^n \geq a^n + b^n$.” Prove that $P(1)$ is true. What kind of proof did you use?
22. Show that if you pick three socks from a drawer containing just blue socks and black socks, you must get either a pair of blue socks or a pair of black socks.
23. Show that at least ten of any 64 days chosen must fall on the same day of the week.
24. Show that at least three of any 25 days chosen must fall in the same month of the year.
25. Use a proof by contradiction to show that there is no rational number r for which $r^3 + r + 1 = 0$. [Hint: Assume that $r = a/b$ is a root, where a and b are integers and a/b is in lowest terms. Obtain an equation involving integers by multiplying by b^3 . Then look at whether a and b are each odd or even.]
26. Prove that if n is a positive integer, then n is even if and only if $7n + 4$ is even.
27. Prove that if n is a positive integer, then n is odd if and only if $5n + 6$ is odd.
28. Prove that $m^2 = n^2$ if and only if $m = n$ or $m = -n$.
29. Prove or disprove that if m and n are integers such that $mn = 1$, then either $m = 1$ and $n = 1$, or else $m = -1$ and $n = -1$.
30. Show that these three statements are equivalent, where a and b are real numbers: (i) a is less than b , (ii) the average of a and b is greater than a , and (iii) the average of a and b is less than b .
31. Show that these statements about the integer x are equivalent: (i) $3x + 2$ is even, (ii) $x + 5$ is odd, (iii) x^2 is even.
32. Show that these statements about the real number x are equivalent: (i) x is rational, (ii) $x/2$ is rational, (iii) $3x - 1$ is rational.
33. Show that these statements about the real number x are equivalent: (i) x is irrational, (ii) $3x + 2$ is irrational, (iii) $x/2$ is irrational.
34. Is this reasoning for finding the solutions of the equation $\sqrt{2x^2 - 1} = x$ correct? (1) $\sqrt{2x^2 - 1} = x$ is given; (2) $2x^2 - 1 = x^2$, obtained by squaring both sides of (1); (3) $x^2 - 1 = 0$, obtained by subtracting x^2 from both sides of (2); (4) $(x - 1)(x + 1) = 0$, obtained by factoring the left-hand side of $x^2 - 1$; (5) $x = 1$ or $x = -1$, which follows because $ab = 0$ implies that $a = 0$ or $b = 0$.
35. Are these steps for finding the solutions of $\sqrt{x+3} = 3 - x$ correct? (1) $\sqrt{x+3} = 3 - x$ is given; (2) $x + 3 = x^2 - 6x + 9$, obtained by squaring both sides of (1); (3) $0 = x^2 - 7x + 6$, obtained by subtracting $x + 3$ from both sides of (2); (4) $0 = (x - 1)(x - 6)$, obtained by factoring the right-hand side of (3); (5) $x = 1$ or $x = 6$, which follows from (4) because $ab = 0$ implies that $a = 0$ or $b = 0$.
36. Show that the propositions p_1 , p_2 , p_3 , and p_4 can be shown to be equivalent by showing that $p_1 \leftrightarrow p_4$, $p_2 \leftrightarrow p_3$, and $p_1 \leftrightarrow p_3$.
37. Show that the propositions p_1 , p_2 , p_3 , p_4 , and p_5 can be shown to be equivalent by proving that the conditional statements $p_1 \rightarrow p_4$, $p_3 \rightarrow p_1$, $p_1 \rightarrow p_2$, $p_2 \rightarrow p_5$, and $p_5 \rightarrow p_3$ are true.

- 38.** Find a counterexample to the statement that every positive integer can be written as the sum of the squares of three integers.
- 39.** Prove that at least one of the real numbers a_1, a_2, \dots, a_n is greater than or equal to the average of these numbers. What kind of proof did you use?
- 40.** Use Exercise 39 to show that if the first 10 positive integers are placed around a circle, in any order, there exist three integers in consecutive locations around the circle that have a sum greater than or equal to 17.
- 41.** Prove that if n is an integer, these four statements are equivalent: (i) n is even, (ii) $n + 1$ is odd, (iii) $3n + 1$ is odd, (iv) $3n$ is even.
- 42.** Prove that these four statements about the integer n are equivalent: (i) n^2 is odd, (ii) $1 - n$ is even, (iii) n^3 is odd, (iv) $n^2 + 1$ is even.

1.8 Proof Methods and Strategy

Introduction



In Section 1.7 we introduced many methods of proof and illustrated how each method can be used. In this section we continue this effort. We will introduce several other commonly used proof methods, including the method of proving a theorem by considering different cases separately. We will also discuss proofs where we prove the existence of objects with desired properties.

In Section 1.7 we briefly discussed the strategy behind constructing proofs. This strategy includes selecting a proof method and then successfully constructing an argument step by step, based on this method. In this section, after we have developed a versatile arsenal of proof methods, we will study some aspects of the art and science of proofs. We will provide advice on how to find a proof of a theorem. We will describe some tricks of the trade, including how proofs can be found by working backward and by adapting existing proofs.

When mathematicians work, they formulate conjectures and attempt to prove or disprove them. We will briefly describe this process here by proving results about tiling checkerboards with dominoes and other types of pieces. Looking at tilings of this kind, we will be able to quickly formulate conjectures and prove theorems without first developing a theory.

We will conclude the section by discussing the role of open questions. In particular, we will discuss some interesting problems either that have been solved after remaining open for hundreds of years or that still remain open.

Exhaustive Proof and Proof by Cases

Sometimes we cannot prove a theorem using a single argument that holds for all possible cases. We now introduce a method that can be used to prove a theorem, by considering different cases separately. This method is based on a rule of inference that we will now introduce. To prove a conditional statement of the form

$$(p_1 \vee p_2 \vee \cdots \vee p_n) \rightarrow q$$

the tautology

$$[(p_1 \vee p_2 \vee \cdots \vee p_n) \rightarrow q] \leftrightarrow [(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge \cdots \wedge (p_n \rightarrow q)]$$

can be used as a rule of inference. This shows that the original conditional statement with a hypothesis made up of a disjunction of the propositions p_1, p_2, \dots, p_n can be proved by proving each of the n conditional statements $p_i \rightarrow q$, $i = 1, 2, \dots, n$, individually. Such an argument is called a **proof by cases**. Sometimes to prove that a conditional statement $p \rightarrow q$ is true, it is convenient to use a disjunction $p_1 \vee p_2 \vee \cdots \vee p_n$ instead of p as the hypothesis of the conditional statement, where p and $p_1 \vee p_2 \vee \cdots \vee p_n$ are equivalent.

EXHAUSTIVE PROOF Some theorems can be proved by examining a relatively small number of examples. Such proofs are called **exhaustive proofs**, or **proofs by exhaustion** because these proofs proceed by exhausting all possibilities. An exhaustive proof is a special type of proof by cases where each case involves checking a single example. We now provide some illustrations of exhaustive proofs.

EXAMPLE 1

Prove that $(n + 1)^3 \geq 3^n$ if n is a positive integer with $n \leq 4$.



Solution: We use a proof by exhaustion. We only need verify the inequality $(n + 1)^3 \geq 3^n$ when $n = 1, 2, 3$, and 4 . For $n = 1$, we have $(n + 1)^3 = 2^3 = 8$ and $3^n = 3^1 = 3$; for $n = 2$, we have $(n + 1)^3 = 3^3 = 27$ and $3^n = 3^2 = 9$; for $n = 3$, we have $(n + 1)^3 = 4^3 = 64$ and $3^n = 3^3 = 27$; and for $n = 4$, we have $(n + 1)^3 = 5^3 = 125$ and $3^n = 3^4 = 81$. In each of these four cases, we see that $(n + 1)^3 \geq 3^n$. We have used the method of exhaustion to prove that $(n + 1)^3 \geq 3^n$ if n is a positive integer with $n \leq 4$.

EXAMPLE 2

Prove that the only consecutive positive integers not exceeding 100 that are perfect powers are 8 and 9. (An integer is a **perfect power** if it equals n^a , where a is an integer greater than 1.)

Solution: We use a proof by exhaustion. In particular, we can prove this fact by examining positive integers n not exceeding 100, first checking whether n is a perfect power, and if it is, checking whether $n + 1$ is also a perfect power. A quicker way to do this is simply to look at all perfect powers not exceeding 100 and checking whether the next largest integer is also a perfect power. The squares of positive integers not exceeding 100 are 1, 4, 9, 16, 25, 36, 49, 64, 81, and 100. The cubes of positive integers not exceeding 100 are 1, 8, 27, and 64. The fourth powers of positive integers not exceeding 100 are 1, 16, and 81. The fifth powers of positive integers not exceeding 100 are 1 and 32. The sixth powers of positive integers not exceeding 100 are 1 and 64. There are no powers of positive integers higher than the sixth power not exceeding 100, other than 1. Looking at this list of perfect powers not exceeding 100, we see that $n = 8$ is the only perfect power n for which $n + 1$ is also a perfect power. That is, $2^3 = 8$ and $3^2 = 9$ are the only two consecutive perfect powers not exceeding 100.

Proofs by exhaustion can tire out people and computers when the number of cases challenges the available processing power!

People can carry out exhaustive proofs when it is necessary to check only a relatively small number of instances of a statement. Computers do not complain when they are asked to check a much larger number of instances of a statement, but they still have limitations. Note that not even a computer can check all instances when it is impossible to list all instances to check.

PROOF BY CASES A proof by cases must cover all possible cases that arise in a theorem. We illustrate proof by cases with a couple of examples. In each example, you should check that all possible cases are covered.

EXAMPLE 3

Prove that if n is an integer, then $n^2 \geq n$.



Solution: We can prove that $n^2 \geq n$ for every integer by considering three cases, when $n = 0$, when $n \geq 1$, and when $n \leq -1$. We split the proof into three cases because it is straightforward to prove the result by considering zero, positive integers, and negative integers separately.

Case (i): When $n = 0$, because $0^2 = 0$, we see that $0^2 \geq 0$. It follows that $n^2 \geq n$ is true in this case.

Case (ii): When $n \geq 1$, when we multiply both sides of the inequality $n \geq 1$ by the positive integer n , we obtain $n \cdot n \geq n \cdot 1$. This implies that $n^2 \geq n$ for $n \geq 1$.

Case (iii): In this case $n \leq -1$. However, $n^2 \geq 0$. It follows that $n^2 \geq n$.

Because the inequality $n^2 \geq n$ holds in all three cases, we can conclude that if n is an integer, then $n^2 \geq n$.

EXAMPLE 4 Use a proof by cases to show that $|xy| = |x||y|$, where x and y are real numbers. (Recall that $|a|$, the absolute value of a , equals a when $a \geq 0$ and equals $-a$ when $a \leq 0$.)

Solution: In our proof of this theorem, we remove absolute values using the fact that $|a| = a$ when $a \geq 0$ and $|a| = -a$ when $a < 0$. Because both $|x|$ and $|y|$ occur in our formula, we will need four cases: (i) x and y both nonnegative, (ii) x nonnegative and y is negative, (iii) x negative and y nonnegative, and (iv) x negative and y negative. We denote by p_1 , p_2 , p_3 , and p_4 , the proposition stating the assumption for each of these four cases, respectively.

(Note that we can remove the absolute value signs by making the appropriate choice of signs within each case.)

Case (i): We see that $p_1 \rightarrow q$ because $xy \geq 0$ when $x \geq 0$ and $y \geq 0$, so that $|xy| = xy = |x||y|$.

Case (ii): To see that $p_2 \rightarrow q$, note that if $x \geq 0$ and $y < 0$, then $xy \leq 0$, so that $|xy| = -xy = x(-y) = |x||y|$. (Here, because $y < 0$, we have $|y| = -y$.)

Case (iii): To see that $p_3 \rightarrow q$, we follow the same reasoning as the previous case with the roles of x and y reversed.

Case (iv): To see that $p_4 \rightarrow q$, note that when $x < 0$ and $y < 0$, it follows that $xy > 0$. Hence, $|xy| = xy = (-x)(-y) = |x||y|$.

Because $|xy| = |x||y|$ holds in each of the four cases and these cases exhaust all possibilities, we can conclude that $|xy| = |x||y|$, whenever x and y are real numbers. 

LEVERAGING PROOF BY CASES The examples we have presented illustrating proof by cases provide some insight into when to use this method of proof. In particular, when it is not possible to consider all cases of a proof at the same time, a proof by cases should be considered. When should you use such a proof? Generally, look for a proof by cases when there is no obvious way to begin a proof, but when extra information in each case helps move the proof forward. Example 5 illustrates how the method of proof by cases can be used effectively.

EXAMPLE 5 Formulate a conjecture about the final decimal digit of the square of an integer and prove your result.

Solution: The smallest perfect squares are 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, and so on. We notice that the digits that occur as the final digit of a square are 0, 1, 4, 5, 6, and 9, with 2, 3, 7, and 8 never appearing as the final digit of a square. We conjecture this theorem: The final decimal digit of a perfect square is 0, 1, 4, 5, 6 or 9. How can we prove this theorem?

We first note that we can express an integer n as $10a + b$, where a and b are positive integers and b is 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. Here a is the integer obtained by subtracting the final decimal digit of n from n and dividing by 10. Next, note that $(10a + b)^2 = 100a^2 + 20ab + b^2 = 10(10a^2 + 2b) + b^2$, so that the final decimal digit of n^2 is the same as the final decimal digit of b^2 . Furthermore, note that the final decimal digit of b^2 is the same as the final decimal digit of $(10 - b)^2 = 100 - 20b + b^2$. Consequently, we can reduce our proof to the consideration of six cases.

Case (i): The final digit of n is 1 or 9. Then the final decimal digit of n^2 is the final decimal digit of $1^2 = 1$ or $9^2 = 81$, namely 1.

Case (ii): The final digit of n is 2 or 8. Then the final decimal digit of n^2 is the final decimal digit of $2^2 = 4$ or $8^2 = 64$, namely 4.

Case (iii): The final digit of n is 3 or 7. Then the final decimal digit of n^2 is the final decimal digit of $3^2 = 9$ or $7^2 = 49$, namely 9.

Case (iv): The final digit of n is 4 or 6. Then the final decimal digit of n^2 is the final decimal digit of $4^2 = 16$ or $6^2 = 36$, namely 6.

Case (v): The final decimal digit of n is 5. Then the final decimal digit of n^2 is the final decimal digit of $5^2 = 25$, namely 5.

Case (vi): The final decimal digit of n is 0. Then the final decimal digit of n^2 is the final decimal digit of $0^2 = 0$, namely 0.

Because we have considered all six cases, we can conclude that the final decimal digit of n^2 , where n is an integer is either 0, 1, 2, 4, 5, 6, or 9. 

Sometimes we can eliminate all but a few examples in a proof by cases, as Example 6 illustrates.

EXAMPLE 6 Show that there are no solutions in integers x and y of $x^2 + 3y^2 = 8$.

Solution: We can quickly reduce a proof to checking just a few simple cases because $x^2 > 8$ when $|x| \geq 3$ and $3y^2 > 8$ when $|y| \geq 2$. This leaves the cases when x equals $-2, -1, 0, 1$, or 2 and y equals $-1, 0$, or 1 . We can finish using an exhaustive proof. To dispense with the remaining cases, we note that possible values for x^2 are $0, 1$, and 4 , and possible values for $3y^2$ are 0 and 3 , and the largest sum of possible values for x^2 and $3y^2$ is 7 . Consequently, it is impossible for $x^2 + 3y^2 = 8$ to hold when x and y are integers. 

WITHOUT LOSS OF GENERALITY In the proof in Example 4, we dismissed case (iii), where $x < 0$ and $y \geq 0$, because it is the same as case (ii), where $x \geq 0$ and $y < 0$, with the roles of x and y reversed. To shorten the proof, we could have proved cases (ii) and (iii) together by assuming, **without loss of generality**, that $x \geq 0$ and $y < 0$. Implicit in this statement is that we can complete the case with $x < 0$ and $y \geq 0$ using the same argument as we used for the case with $x \geq 0$ and $y < 0$, but with the obvious changes.

In general, when the phrase “without loss of generality” is used in a proof (often abbreviated as WLOG), we assert that by proving one case of a theorem, no additional argument is required to prove other specified cases. That is, other cases follow by making straightforward changes to the argument, or by filling in some straightforward initial step. Proofs by cases can often be made much more efficient when the notion of without loss of generality is employed. Of course, incorrect use of this principle can lead to unfortunate errors. Sometimes assumptions are made that lead to a loss in generality. Such assumptions can be made that do not take into account that one case may be substantially different from others. This can lead to an incomplete, and possibly unsalvageable, proof. In fact, many incorrect proofs of famous theorems turned out to rely on arguments that used the idea of “without loss of generality” to establish cases that could not be quickly proved from simpler cases.

We now illustrate a proof where without loss of generality is used effectively together with other proof techniques.

EXAMPLE 7 Show that if x and y are integers and both xy and $x + y$ are even, then both x and y are even.

Solution: We will use proof by contraposition, the notion of without loss of generality, and proof by cases. First, suppose that x and y are not both even. That is, assume that x is odd or that y is odd (or both). Without loss of generality, we assume that x is odd, so that $x = 2m + 1$ for some integer k .

To complete the proof, we need to show that xy is odd or $x + y$ is odd. Consider two cases: (i) y even, and (ii) y odd. In (i), $y = 2n$ for some integer n , so that $x + y = (2m + 1) + 2n = 2(m + n) + 1$ is odd. In (ii), $y = 2n + 1$ for some integer n , so that $xy = (2m + 1)(2n + 1) = 4mn + 2m + 2n + 1 = 2(2mn + m + n) + 1$ is odd. This completes the proof by contraposition. (Note that our use of without loss of generality within the proof is justified because the proof when y is odd can be obtained by simply interchanging the roles of x and y in the proof we have given.) 

COMMON ERRORS WITH EXHAUSTIVE PROOF AND PROOF BY CASES A common error of reasoning is to draw incorrect conclusions from examples. No matter how many separate examples are considered, a theorem is not proved by considering examples unless every possible



In a proof by cases be sure not to omit any cases and check that you have proved all cases correctly!

case is covered. The problem of proving a theorem is analogous to showing that a computer program always produces the output desired. No matter how many input values are tested, unless all input values are tested, we cannot conclude that the program always produces the correct output.

EXAMPLE 8 Is it true that every positive integer is the sum of 18 fourth powers of integers?

Solution: To determine whether a positive integer n can be written as the sum of 18 fourth powers of integers, we might begin by examining whether n is the sum of 18 fourth powers of integers for the smallest positive integers. Because the fourth powers of integers are 0, 1, 16, 81, ..., if we can select 18 terms from these numbers that add up to n , then n is the sum of 18 fourth powers. We can show that all positive integers up to 78 can be written as the sum of 18 fourth powers. (The details are left to the reader.) However, if we decided this was enough checking, we would come to the wrong conclusion. It is not true that every positive integer is the sum of 18 fourth powers because 79 is not the sum of 18 fourth powers (as the reader can verify). 

Another common error involves making unwarranted assumptions that lead to incorrect proofs by cases where not all cases are considered. This is illustrated in Example 9.

EXAMPLE 9 What is wrong with this “proof?”

“Theorem:” If x is a real number, then x^2 is a positive real number.

“Proof:” Let p_1 be “ x is positive,” let p_2 be “ x is negative,” and let q be “ x^2 is positive.” To show that $p_1 \rightarrow q$ is true, note that when x is positive, x^2 is positive because it is the product of two positive numbers, x and x . To show that $p_2 \rightarrow q$, note that when x is negative, x^2 is positive because it is the product of two negative numbers, x and x . This completes the proof. 

Solution: The problem with this “proof” is that we missed the case of $x = 0$. When $x = 0$, $x^2 = 0$ is not positive, so the supposed theorem is false. If p is “ x is a real number,” then we can prove results where p is the hypothesis with three cases, p_1 , p_2 , and p_3 , where p_1 is “ x is positive,” p_2 is “ x is negative,” and p_3 is “ $x = 0$ ” because of the equivalence $p \leftrightarrow p_1 \vee p_2 \vee p_3$. 

Existence Proofs

Many theorems are assertions that objects of a particular type exist. A theorem of this type is a proposition of the form $\exists x P(x)$, where P is a predicate. A proof of a proposition of the form $\exists x P(x)$ is called an **existence proof**. There are several ways to prove a theorem of this type. Sometimes an existence proof of $\exists x P(x)$ can be given by finding an element a , called a **witness**, such that $P(a)$ is true. This type of existence proof is called **constructive**. It is also possible to give an existence proof that is **nonconstructive**; that is, we do not find an element a such that $P(a)$ is true, but rather prove that $\exists x P(x)$ is true in some other way. One common method of giving a nonconstructive existence proof is to use proof by contradiction and show that the negation of the existential quantification implies a contradiction. The concept of a constructive existence proof is illustrated by Example 10 and the concept of a nonconstructive existence proof is illustrated by Example 11.

EXAMPLE 10 A Constructive Existence Proof Show that there is a positive integer that can be written as the sum of cubes of positive integers in two different ways.



Solution: After considerable computation (such as a computer search) we find that

$$1729 = 10^3 + 9^3 = 12^3 + 1^3.$$

Because we have displayed a positive integer that can be written as the sum of cubes in two different ways, we are done.

There is an interesting story pertaining to this example. The English mathematician G. H. Hardy, when visiting the ailing Indian prodigy Ramanujan in the hospital, remarked that 1729, the number of the cab he took, was rather dull. Ramanujan replied “No, it is a very interesting number; it is the smallest number expressible as the sum of cubes in two different ways.” ◀

EXAMPLE 11

A Nonconstructive Existence Proof Show that there exist irrational numbers x and y such that x^y is rational.

Solution: By Example 10 in Section 1.7 we know that $\sqrt{2}$ is irrational. Consider the number $\sqrt{2}^{\sqrt{2}}$. If it is rational, we have two irrational numbers x and y with x^y rational, namely, $x = \sqrt{2}$ and $y = \sqrt{2}$. On the other hand if $\sqrt{2}^{\sqrt{2}}$ is irrational, then we can let $x = \sqrt{2}^{\sqrt{2}}$ and $y = \sqrt{2}$ so that $x^y = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2} \cdot \sqrt{2})} = \sqrt{2}^2 = 2$.

This proof is an example of a nonconstructive existence proof because we have not found irrational numbers x and y such that x^y is rational. Rather, we have shown that either the pair $x = \sqrt{2}$, $y = \sqrt{2}$ or the pair $x = \sqrt{2}^{\sqrt{2}}$, $y = \sqrt{2}$ have the desired property, but we do not know which of these two pairs works! ◀



GODFREY HAROLD HARDY (1877–1947) Hardy, born in Cranleigh, Surrey, England, was the older of two children of Isaac Hardy and Sophia Hall Hardy. His father was the geography and drawing master at the Cranleigh School and also gave singing lessons and played soccer. His mother gave piano lessons and helped run a boardinghouse for young students. Hardy’s parents were devoted to their children’s education. Hardy demonstrated his numerical ability at the early age of two when he began writing down numbers into the millions. He had a private mathematics tutor rather than attending regular classes at the Cranleigh School. He moved to Winchester College, a private high school, when he was 13 and was awarded a scholarship. He excelled in his studies and demonstrated a strong interest in mathematics. He entered Trinity College, Cambridge, in 1896 on a scholarship and won several prizes during his time there, graduating in 1899.

Hardy held the position of lecturer in mathematics at Trinity College at Cambridge University from 1906 to 1919, when he was appointed to the Sullivan chair of geometry at Oxford. He had become unhappy with Cambridge over the dismissal of the famous philosopher and mathematician Bertrand Russell from Trinity for antiwar activities and did not like a heavy load of administrative duties. In 1931 he returned to Cambridge as the Sadleirian professor of pure mathematics, where he remained until his retirement in 1942. He was a pure mathematician and held an elitist view of mathematics, hoping that his research could never be applied. Ironically, he is perhaps best known as one of the developers of the Hardy–Weinberg law, which predicts patterns of inheritance. His work in this area appeared as a letter to the journal *Science* in which he used simple algebraic ideas to demonstrate errors in an article on genetics. Hardy worked primarily in number theory and function theory, exploring such topics as the Riemann zeta function, Fourier series, and the distribution of primes. He made many important contributions to many important problems, such as Waring’s problem about representing positive integers as sums of k th powers and the problem of representing odd integers as sums of three primes. Hardy is also remembered for his collaborations with John E. Littlewood, a colleague at Cambridge, with whom he wrote more than 100 papers, and the famous Indian mathematical prodigy Srinivasa Ramanujan. His collaboration with Littlewood led to the joke that there were only three important English mathematicians at that time, Hardy, Littlewood, and Hardy–Littlewood, although some people thought that Hardy had invented a fictitious person, Littlewood, because Littlewood was seldom seen outside Cambridge. Hardy had the wisdom of recognizing Ramanujan’s genius from unconventional but extremely creative writings Ramanujan sent him, while other mathematicians failed to see the genius. Hardy brought Ramanujan to Cambridge and collaborated on important joint papers, establishing new results on the number of partitions of an integer. Hardy was interested in mathematics education, and his book *A Course of Pure Mathematics* had a profound effect on undergraduate instruction in mathematics in the first half of the twentieth century. Hardy also wrote *A Mathematician’s Apology*, in which he gives his answer to the question of whether it is worthwhile to devote one’s life to the study of mathematics. It presents Hardy’s view of what mathematics is and what a mathematician does.

Hardy had a strong interest in sports. He was an avid cricket fan and followed scores closely. One peculiar trait he had was that he did not like his picture taken (only five snapshots are known) and disliked mirrors, covering them with towels immediately upon entering a hotel room.

Nonconstructive existence proofs often are quite subtle, as Example 12 illustrates.

EXAMPLE 12



Chomp is a game played by two players. In this game, cookies are laid out on a rectangular grid. The cookie in the top left position is poisoned, as shown in Figure 1(a). The two players take turns making moves; at each move, a player is required to eat a remaining cookie, together with all cookies to the right and/or below it (see Figure 1(b), for example). The loser is the player who has no choice but to eat the poisoned cookie. We ask whether one of the two players has a winning strategy. That is, can one of the players always make moves that are guaranteed to lead to a win?

Solution: We will give a nonconstructive existence proof of a winning strategy for the first player. That is, we will show that the first player always has a winning strategy without explicitly describing the moves this player must follow.

First, note that the game ends and cannot finish in a draw because with each move at least one cookie is eaten, so after no more than $m \times n$ moves the game ends, where the initial grid is $m \times n$. Now, suppose that the first player begins the game by eating just the cookie in the bottom right corner. There are two possibilities, this is the first move of a winning strategy for the first player, or the second player can make a move that is the first move of a winning strategy for the second player. In this second case, instead of eating just the cookie in the bottom right corner, the first player could have made the same move that the second player made as the first



SRINIVASA RAMANUJAN (1887–1920) The famous mathematical prodigy Ramanujan was born and raised in southern India near the city of Madras (now called Chennai). His father was a clerk in a cloth shop. His mother contributed to the family income by singing at a local temple. Ramanujan studied at the local English language school, displaying his talent and interest for mathematics. At the age of 13 he mastered a textbook used by college students. When he was 15, a university student lent him a copy of *Synopsis of Pure Mathematics*. Ramanujan decided to work out the over 6000 results in this book, stated without proof or explanation, writing on sheets later collected to form notebooks. He graduated from high school in 1904, winning a scholarship to the University of Madras. Enrolling in a fine arts curriculum, he neglected his subjects other than mathematics and lost his scholarship. He failed to pass examinations at the university four times from 1904 to 1907, doing well only in mathematics. During this time he filled his notebooks with original writings, sometimes rediscovering already published work and at other times making new discoveries.

Without a university degree, it was difficult for Ramanujan to find a decent job. To survive, he had to depend on the goodwill of his friends. He tutored students in mathematics, but his unconventional ways of thinking and failure to stick to the syllabus caused problems. He was married in 1909 in an arranged marriage to a young woman nine years his junior. Needing to support himself and his wife, he moved to Madras and sought a job. He showed his notebooks of mathematical writings to his potential employers, but the books bewildered them. However, a professor at the Presidency College recognized his genius and supported him, and in 1912 he found work as an accounts clerk, earning a small salary.

Ramanujan continued his mathematical work during this time and published his first paper in 1910 in an Indian journal. He realized that his work was beyond that of Indian mathematicians and decided to write to leading English mathematicians. The first mathematicians he wrote to turned down his request for help. But in January 1913 he wrote to G. H. Hardy, who was inclined to turn Ramanujan down, but the mathematical statements in the letter, although stated without proof, puzzled Hardy. He decided to examine them closely with the help of his colleague and collaborator J. E. Littlewood. They decided, after careful study, that Ramanujan was probably a genius, because his statements “could only be written down by a mathematician of the highest class; they must be true, because if they were not true, no one would have the imagination to invent them.”

Hardy arranged a scholarship for Ramanujan, bringing him to England in 1914. Hardy personally tutored him in mathematical analysis, and they collaborated for five years, proving significant theorems about the number of partitions of integers. During this time, Ramanujan made important contributions to number theory and also worked on continued fractions, infinite series, and elliptic functions. Ramanujan had amazing insight involving certain types of functions and series, but his purported theorems on prime numbers were often wrong, illustrating his vague idea of what constitutes a correct proof. He was one of the youngest members ever appointed a Fellow of the Royal Society. Unfortunately, in 1917 Ramanujan became extremely ill. At the time, it was thought that he had trouble with the English climate and had contracted tuberculosis. It is now thought that he suffered from a vitamin deficiency, brought on by Ramanujan’s strict vegetarianism and shortages in wartime England. He returned to India in 1919, continuing to do mathematics even when confined to his bed. He was religious and thought his mathematical talent came from his family deity, Namagiri. He considered mathematics and religion to be linked. He said that “an equation for me has no meaning unless it expresses a thought of God.” His short life came to an end in April 1920, when he was 32 years old. Ramanujan left several notebooks of unpublished results. The writings in these notebooks illustrate Ramanujan’s insights but are quite sketchy. Several mathematicians have devoted many years of study to explaining and justifying the results in these notebooks.

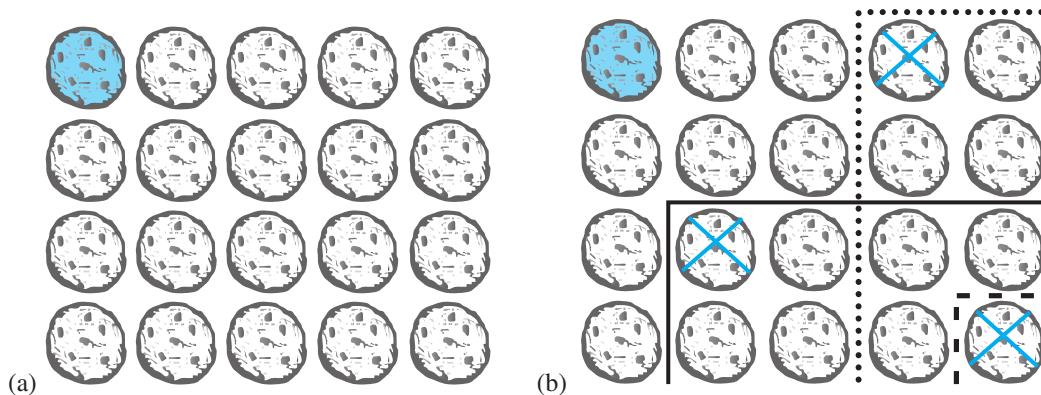


FIGURE 1 (a) Chomp (Top Left Cookie Poisoned). (b) Three Possible Moves.

move of a winning strategy (and then continued to follow that winning strategy). This would guarantee a win for the first player.

Note that we showed that a winning strategy exists, but we did not specify an actual winning strategy. Consequently, the proof is a nonconstructive existence proof. In fact, no one has been able to describe a winning strategy for that Chomp that applies for all rectangular grids by describing the moves that the first player should follow. However, winning strategies can be described for certain special cases, such as when the grid is square and when the grid only has two rows of cookies (see Exercises 15 and 16 in Section 5.2). \blacktriangleleft

Uniqueness Proofs

Some theorems assert the existence of a unique element with a particular property. In other words, these theorems assert that there is exactly one element with this property. To prove a statement of this type we need to show that an element with this property exists and that no other element has this property. The two parts of a **uniqueness proof** are:

Existence: We show that an element x with the desired property exists.

Uniqueness: We show that if $y \neq x$, then y does not have the desired property.

Equivalently, we can show that if x and y both have the desired property, then $x = y$.

Remark: Showing that there is a unique element x such that $P(x)$ is the same as proving the statement $\exists x(P(x) \wedge \forall y(y \neq x \rightarrow \neg P(y)))$.

We illustrate the elements of a uniqueness proof in Example 13.

EXAMPLE 13 Show that if a and b are real numbers and $a \neq 0$, then there is a unique real number r such that $ar + b = 0$.

Solution: First, note that the real number $r = -b/a$ is a solution of $ar + b = 0$ because $a(-b/a) + b = -b + b = 0$. Consequently, a real number r exists for which $ar + b = 0$. This is the existence part of the proof.

Second, suppose that s is a real number such that $as + b = 0$. Then $ar + b = as + b$, where $r = -b/a$. Subtracting b from both sides, we find that $ar = as$. Dividing both sides of this last equation by a , which is nonzero, we see that $r = s$. This means that if $s \neq r$, then $as + b \neq 0$. This establishes the uniqueness part of the proof. \blacktriangleleft

Proof Strategies

Finding proofs can be a challenging business. When you are confronted with a statement to prove, you should first replace terms by their definitions and then carefully analyze what the hypotheses and the conclusion mean. After doing so, you can attempt to prove the result using one of the available methods of proof. Generally, if the statement is a conditional statement, you should first try a direct proof; if this fails, you can try an indirect proof. If neither of these approaches works, you might try a proof by contradiction.

FORWARD AND BACKWARD REASONING Whichever method you choose, you need a starting point for your proof. To begin a direct proof of a conditional statement, you start with the premises. Using these premises, together with axioms and known theorems, you can construct a proof using a sequence of steps that leads to the conclusion. This type of reasoning, called *forward reasoning*, is the most common type of reasoning used to prove relatively simple results. Similarly, with indirect reasoning you can start with the negation of the conclusion and, using a sequence of steps, obtain the negation of the premises.

Unfortunately, forward reasoning is often difficult to use to prove more complicated results, because the reasoning needed to reach the desired conclusion may be far from obvious. In such cases it may be helpful to use *backward reasoning*. To reason backward to prove a statement q , we find a statement p that we can prove with the property that $p \rightarrow q$. (Note that it is not helpful to find a statement r that you can prove such that $q \rightarrow r$, because it is the fallacy of begging the question to conclude from $q \rightarrow r$ and r that q is true.) Backward reasoning is illustrated in Examples 14 and 15.

EXAMPLE 14

Given two positive real numbers x and y , their **arithmetic mean** is $(x + y)/2$ and their **geometric mean** is \sqrt{xy} . When we compare the arithmetic and geometric means of pairs of distinct positive real numbers, we find that the arithmetic mean is always greater than the geometric mean. [For example, when $x = 4$ and $y = 6$, we have $5 = (4 + 6)/2 > \sqrt{4 \cdot 6} = \sqrt{24}$.] Can we prove that this inequality is always true?

Solution: To prove that $(x + y)/2 > \sqrt{xy}$ when x and y are distinct positive real numbers, we can work backward. We construct a sequence of equivalent inequalities. The equivalent inequalities are

$$\begin{aligned} (x + y)/2 &> \sqrt{xy}, \\ (x + y)^2/4 &> xy, \\ (x + y)^2 &> 4xy, \\ x^2 + 2xy + y^2 &> 4xy, \\ x^2 - 2xy + y^2 &> 0, \\ (x - y)^2 &> 0. \end{aligned}$$



Because $(x - y)^2 > 0$ when $x \neq y$, it follows that the final inequality is true. Because all these inequalities are equivalent, it follows that $(x + y)/2 > \sqrt{xy}$ when $x \neq y$. Once we have carried out this backward reasoning, we can easily reverse the steps to construct a proof using forward reasoning. We now give this proof.

Suppose that x and y are distinct positive real numbers. Then $(x - y)^2 > 0$ because the square of a nonzero real number is positive (see Appendix 1). Because $(x - y)^2 = x^2 - 2xy + y^2$, this implies that $x^2 - 2xy + y^2 > 0$. Adding $4xy$ to both sides, we obtain $x^2 + 2xy + y^2 > 4xy$. Because $x^2 + 2xy + y^2 = (x + y)^2$, this means that $(x + y)^2 \geq 4xy$. Dividing both sides of this equation by 4, we see that $(x + y)^2/4 > xy$. Finally, taking square roots of both sides (which preserves the inequality because both sides are positive) yields

$(x + y)/2 > \sqrt{xy}$. We conclude that if x and y are distinct positive real numbers, then their arithmetic mean $(x + y)/2$ is greater than their geometric mean \sqrt{xy} . 

EXAMPLE 15 Suppose that two people play a game taking turns removing one, two, or three stones at a time from a pile that begins with 15 stones. The person who removes the last stone wins the game. Show that the first player can win the game no matter what the second player does.

Solution: To prove that the first player can always win the game, we work backward. At the last step, the first player can win if this player is left with a pile containing one, two, or three stones. The second player will be forced to leave one, two, or three stones if this player has to remove stones from a pile containing four stones. Consequently, one way for the first person to win is to leave four stones for the second player on the next-to-last move. The first person can leave four stones when there are five, six, or seven stones left at the beginning of this player's move, which happens when the second player has to remove stones from a pile with eight stones. Consequently, to force the second player to leave five, six, or seven stones, the first player should leave eight stones for the second player at the second-to-last move for the first player. This means that there are nine, ten, or eleven stones when the first player makes this move. Similarly, the first player should leave twelve stones when this player makes the first move. We can reverse this argument to show that the first player can always make moves so that this player wins the game no matter what the second player does. These moves successively leave twelve, eight, and four stones for the second player. 

ADAPTING EXISTING PROOFS An excellent way to look for possible approaches that can be used to prove a statement is to take advantage of existing proofs of similar results. Often an existing proof can be adapted to prove other facts. Even when this is not the case, some of the ideas used in existing proofs may be helpful. Because existing proofs provide clues for new proofs, you should read and understand the proofs you encounter in your studies. This process is illustrated in Example 16.

EXAMPLE 16 In Example 10 of Section 1.7 we proved that $\sqrt{2}$ is irrational. We now conjecture that $\sqrt{3}$ is irrational. Can we adapt the proof in Example 10 in Section 1.7 to show that $\sqrt{3}$ is irrational?



Solution: To adapt the proof in Example 10 in Section 1.7, we begin by mimicking the steps in that proof, but with $\sqrt{2}$ replaced with $\sqrt{3}$. First, we suppose that $\sqrt{3} = d/c$ where the fraction c/d is in lowest terms. Squaring both sides tells us that $3 = c^2/d^2$, so that $3d^2 = c^2$. Can we use this equation to show that 3 must be a factor of both c and d , similar to how we used the equation $2b^2 = a^2$ in Example 10 in Section 1.7 to show that 2 must be a factor of both a and b ? (Recall that an integer s is a factor of the integer t if t/s is an integer. An integer n is even if and only if 2 is a factor of n .) It turns out that we can, but we need some ammunition from number theory, which we will develop in Chapter 4. We sketch out the remainder of the proof, but leave the justification of these steps until Chapter 4. Because 3 is a factor of c^2 , it must also be a factor of c . Furthermore, because 3 is a factor of c , 9 is a factor of c^2 , which means that 9 is a factor of $3d^2$. This implies that 3 is a factor of d^2 , which means that 3 is a factor of d . This makes 3 a factor of both c and d , which contradicts the assumption that c/d is in lowest terms. After we have filled in the justification for these steps, we will have shown that $\sqrt{3}$ is irrational by adapting the proof that $\sqrt{2}$ is irrational. Note that this proof can be extended to show that \sqrt{n} is irrational whenever n is a positive integer that is not a perfect square. We leave the details of this to Chapter 4. 

A good tip is to look for existing proofs that you might adapt when you are confronted with proving a new theorem, particularly when the new theorem seems similar to one you have already proved.

Looking for Counterexamples

In Section 1.7 we introduced the use of counterexamples to show that certain statements are false. When confronted with a conjecture, you might first try to prove this conjecture, and if your attempts are unsuccessful, you might try to find a counterexample, first by looking at the simplest, smallest examples. If you cannot find a counterexample, you might again try to prove the statement. In any case, looking for counterexamples is an extremely important pursuit, which often provides insights into problems. We will illustrate the role of counterexamples in Example 17.

EXAMPLE 17 In Example 14 in Section 1.7 we showed that the statement “Every positive integer is the sum of two squares of integers” is false by finding a counterexample. That is, there are positive integers that cannot be written as the sum of the squares of two integers. Although we cannot write every positive integer as the sum of the squares of two integers, maybe we can write every positive integer as the sum of the squares of three integers. That is, is the statement “Every positive integer is the sum of the squares of three integers” true or false?



Solution: Because we know that not every positive integer can be written as the sum of two squares of integers, we might initially be skeptical that every positive integer can be written as the sum of three squares of integers. So, we first look for a counterexample. That is, we can show that the statement “Every positive integer is the sum of three squares of integers” is false if we can find a particular integer that is not the sum of the squares of three integers. To look for a counterexample, we try to write successive positive integers as a sum of three squares. We find that $1 = 0^2 + 0^2 + 1^2$, $2 = 0^2 + 1^2 + 1^2$, $3 = 1^2 + 1^2 + 1^2$, $4 = 0^2 + 0^2 + 2^2$, $5 = 0^2 + 1^2 + 2^2$, $6 = 1^2 + 1^2 + 2^2$, but we cannot find a way to write 7 as the sum of three squares. To show that there are not three squares that add up to 7, we note that the only possible squares we can use are those not exceeding 7, namely, 0, 1, and 4. Because no three terms where each term is 0, 1, or 4 add up to 7, it follows that 7 is a counterexample. We conclude that the statement “Every positive integer is the sum of the squares of three integers” is false.

We have shown that not every positive integer is the sum of the squares of three integers. The next question to ask is whether every positive integer is the sum of the squares of four positive integers. Some experimentation provides evidence that the answer is yes. For example, $7 = 1^2 + 1^2 + 1^2 + 2^2$, $25 = 4^2 + 2^2 + 2^2 + 1^2$, and $87 = 9^2 + 2^2 + 1^2 + 1^2$. It turns out the conjecture “Every positive integer is the sum of the squares of four integers” is true. For a proof, see [Ro10].

Proof Strategy in Action

Mathematics is generally taught as if mathematical facts were carved in stone. Mathematics texts (including the bulk of this book) formally present theorems and their proofs. Such presentations do not convey the discovery process in mathematics. This process begins with exploring concepts and examples, asking questions, formulating conjectures, and attempting to settle these conjectures either by proof or by counterexample. These are the day-to-day activities of mathematicians. Believe it or not, the material taught in textbooks was originally developed in this way.



People formulate conjectures on the basis of many types of possible evidence. The examination of special cases can lead to a conjecture, as can the identification of possible patterns. Altering the hypotheses and conclusions of known theorems also can lead to plausible conjectures. At other times, conjectures are made based on intuition or a belief that a result holds. No matter how a conjecture was made, once it has been formulated, the goal is to prove or disprove it. When mathematicians believe that a conjecture may be true, they try to find a proof. If they cannot find a proof, they may look for a counterexample. When they cannot find a counterexample, they may switch gears and once again try to prove the conjecture. Although many conjectures are quickly settled, a few conjectures resist attack for hundreds of years and lead to

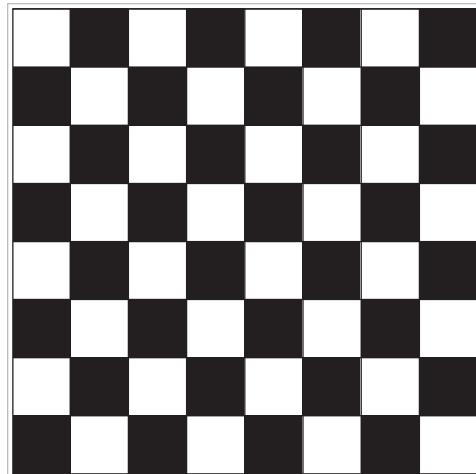


FIGURE 2 The Standard Checkerboard.

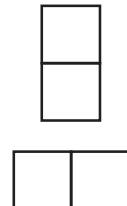


FIGURE 3
Two Dominoes.

the development of new parts of mathematics. We will mention a few famous conjectures later in this section.

Tilings



We can illustrate aspects of proof strategy through a brief study of tilings of checkerboards. Looking at tilings of checkerboards is a fruitful way to quickly discover many different results and construct their proofs using a variety of proof methods. There are almost an endless number of conjectures that can be made and studied in this area too. To begin, we need to define some terms. A **checkerboard** is a rectangle divided into squares of the same size by horizontal and vertical lines. The game of checkers is played on a board with 8 rows and 8 columns; this board is called the **standard checkerboard** and is shown in Figure 2. In this section we use the term **board** to refer to a checkerboard of any rectangular size as well as parts of checkerboards obtained by removing one or more squares. A **domino** is a rectangular piece that is one square by two squares, as shown in Figure 3. We say that a board is **tiled** by dominoes when all its squares are covered with no overlapping dominoes and no dominoes overhanging the board. We now develop some results about tiling boards using dominoes.

EXAMPLE 18 Can we tile the standard checkerboard using dominoes?

Solution: We can find many ways to tile the standard checkerboard using dominoes. For example, we can tile it by placing 32 dominoes horizontally, as shown in Figure 4. The existence of one such tiling completes a constructive existence proof. Of course, there are a large number of other ways to do this tiling. We can place 32 dominoes vertically on the board or we can place some tiles vertically and some horizontally. But for a constructive existence proof we needed to find just one such tiling. 

EXAMPLE 19 Can we tile a board obtained by removing one of the four corner squares of a standard checkerboard?



Solution: To answer this question, note that a standard checkerboard has 64 squares, so removing a square produces a board with 63 squares. Now suppose that we could tile a board obtained from the standard checkerboard by removing a corner square. The board has an even number of

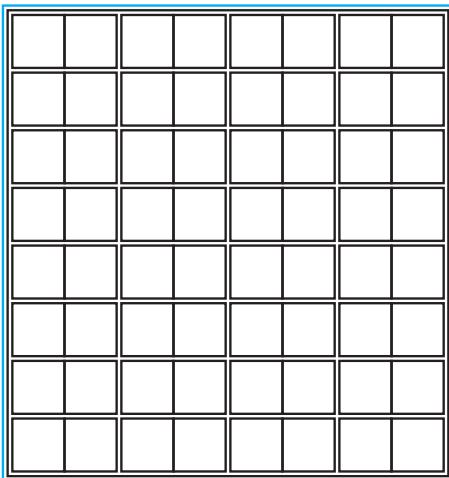


FIGURE 4 Tiling the Standard Checkerboard.

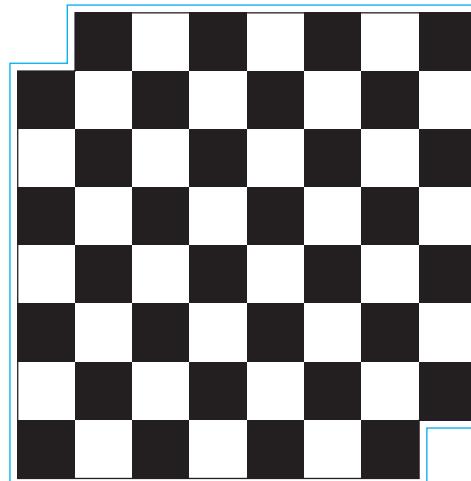


FIGURE 5 The Standard Checkerboard with the Upper Left and Lower Right Squares Removed.

squares because each domino covers two squares and no two dominoes overlap and no dominoes overhang the board. Consequently, we can prove by contradiction that a standard checkerboard with one square removed cannot be tiled using dominoes because such a board has an odd number of squares. \blacktriangleleft

We now consider a trickier situation.

EXAMPLE 20 Can we tile the board obtained by deleting the upper left and lower right corner squares of a standard checkerboard, shown in Figure 5?

Solution: A board obtained by deleting two squares of a standard checkerboard contains $64 - 2 = 62$ squares. Because 62 is even, we cannot quickly rule out the existence of a tiling of the standard checkerboard with its upper left and lower right squares removed, unlike Example 19, where we ruled out the existence of a tiling of the standard checkerboard with one corner square removed. Trying to construct a tiling of this board by successively placing dominoes might be a first approach, as the reader should attempt. However, no matter how much we try, we cannot find such a tiling. Because our efforts do not produce a tiling, we are led to conjecture that no tiling exists.

We might try to prove that no tiling exists by showing that we reach a dead end however we successively place dominoes on the board. To construct such a proof, we would have to consider all possible cases that arise as we run through all possible choices of successively placing dominoes. For example, we have two choices for covering the square in the second column of the first row, next to the removed top left corner. We could cover it with a horizontally placed tile or a vertically placed tile. Each of these two choices leads to further choices, and so on. It does not take long to see that this is not a fruitful plan of attack for a person, although a computer could be used to complete such a proof by exhaustion. (Exercise 45 asks you to supply such a proof to show that a 4×4 checkerboard with opposite corners removed cannot be tiled.)

We need another approach. Perhaps there is an easier way to prove there is no tiling of a standard checkerboard with two opposite corners removed. As with many proofs, a key observation can help. We color the squares of this checkerboard using alternating white and black squares, as in Figure 2. Observe that a domino in a tiling of such a board covers one white square and one black square. Next, note that this board has unequal numbers of white square and black

squares. We can use these observations to prove by contradiction that a standard checkerboard with opposite corners removed cannot be tiled using dominoes. We now present such a proof.

Proof: Suppose we can use dominoes to tile a standard checkerboard with opposite corners removed. Note that the standard checkerboard with opposite corners removed contains $64 - 2 = 62$ squares. The tiling would use $62/2 = 31$ dominoes. Note that each domino in this tiling covers one white and one black square. Consequently, the tiling covers 31 white squares and 31 black squares. However, when we remove two opposite corner squares, either 32 of the remaining squares are white and 30 are black or else 30 are white and 32 are black. This contradicts the assumption that we can use dominoes to cover a standard checkerboard with opposite corners removed, completing the proof. \blacktriangleleft

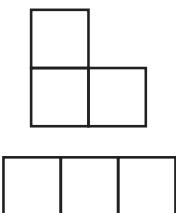


FIGURE 6 A Right Triomino and a Straight Triomino.

We can use other types of pieces besides dominoes in tilings. Instead of dominoes we can study tilings that use identically shaped pieces constructed from congruent squares that are connected along their edges. Such pieces are called **polyominoes**, a term coined in 1953 by the mathematician Solomon Golomb, the author of an entertaining book about them [Go94]. We will consider two polyominoes with the same number of squares the same if we can rotate and/or flip one of the polyominoes to get the other one. For example, there are two types of triominoes (see Figure 6), which are polyominoes made up of three squares connected by their sides. One type of triomino, the **straight triomino**, has three horizontally connected squares; the other type, **right triominoes**, resembles the letter L in shape, flipped and/or rotated, if necessary. We will study the tilings of a checkerboard by straight triominoes here; we will study tilings by right triominoes in Section 5.1.

EXAMPLE 21 Can you use straight triominoes to tile a standard checkerboard?

Solution: The standard checkerboard contains 64 squares and each triomino covers three squares. Consequently, if triominoes tile a board, the number of squares of the board must be a multiple of 3. Because 64 is not a multiple of 3, triominoes cannot be used to cover an 8×8 checkerboard. \blacktriangleleft

In Example 22, we consider the problem of using straight triominoes to tile a standard checkerboard with one corner missing.

EXAMPLE 22 Can we use straight triominoes to tile a standard checkerboard with one of its four corners removed? An 8×8 checkerboard with one corner removed contains $64 - 1 = 63$ squares. Any tiling by straight triominoes of one of these boards uses $63/3 = 21$ triominoes. However, when we experiment, we cannot find a tiling of one of these boards using straight triominoes. A proof by exhaustion does not appear promising. Can we adapt our proof from Example 20 to prove that no such tiling exists?

Solution: We will color the squares of the checkerboard in an attempt to adapt the proof by contradiction we gave in Example 20 of the impossibility of using dominoes to tile a standard checkerboard with opposite corners removed. Because we are using straight triominoes rather than dominoes, we color the squares using three colors rather than two colors, as shown in Figure 7. Note that there are 21 blue squares, 21 black squares, and 22 white squares in this coloring. Next, we make the crucial observation that when a straight triomino covers three squares of the checkerboard, it covers one blue square, one black square, and one white square. Next, note that each of the three colors appears in a corner square. Thus without loss of generality, we may assume that we have rotated the coloring so that the missing square is colored blue. Therefore, we assume that the remaining board contains 20 blue squares, 21 black squares, and 22 white squares.

If we could tile this board using straight triominoes, then we would use $63/3 = 21$ straight triominoes. These triominoes would cover 21 blue squares, 21 black squares, and 21 white

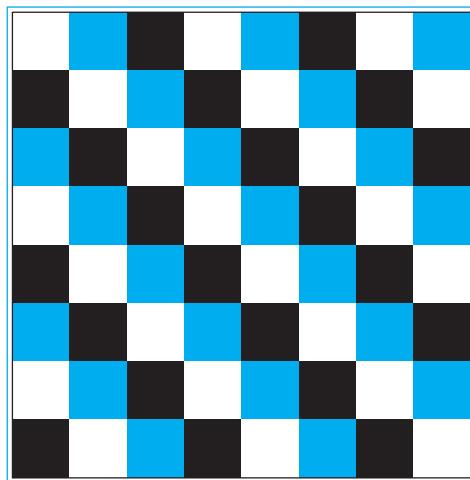


FIGURE 7 Coloring the Squares of the Standard Checkerboard with Three Colors.

squares. This contradicts the fact that this board contains 20 blue squares, 21 black squares, and 22 white squares. Therefore we cannot tile this board using straight triominoes. \blacktriangleleft

The Role of Open Problems

Many advances in mathematics have been made by people trying to solve famous unsolved problems. In the past 20 years, many unsolved problems have finally been resolved, such as the proof of a conjecture in number theory made more than 300 years ago. This conjecture asserts the truth of the statement known as **Fermat's last theorem**.

THEOREM 1

FERMAT'S LAST THEOREM

The equation

$$x^n + y^n = z^n$$

has no solutions in integers x , y , and z with $xyz \neq 0$ whenever n is an integer with $n > 2$.



Remark: The equation $x^2 + y^2 = z^2$ has infinitely many solutions in integers x , y , and z ; these solutions are called Pythagorean triples and correspond to the lengths of the sides of right triangles with integer lengths. See Exercise 32.

This problem has a fascinating history. In the seventeenth century, Fermat jotted in the margin of his copy of the works of Diophantus that he had a “wondrous proof” that there are no integer solutions of $x^n + y^n = z^n$ when n is an integer greater than 2 with $xyz \neq 0$. However, he never published a proof (Fermat published almost nothing), and no proof could be found in the papers he left when he died. Mathematicians looked for a proof for three centuries without success, although many people were convinced that a relatively simple proof could be found. (Proofs of special cases were found, such as the proof of the case when $n = 3$ by Euler and the proof of the $n = 4$ case by Fermat himself.) Over the years, several established mathematicians thought that they had proved this theorem. In the nineteenth century, one of these failed attempts led to the development of the part of number theory called algebraic number theory. A correct

proof, requiring hundreds of pages of advanced mathematics, was not found until the 1990s, when Andrew Wiles used recently developed ideas from a sophisticated area of number theory called the theory of elliptic curves to prove Fermat's last theorem. Wiles's quest to find a proof of Fermat's last theorem using this powerful theory, described in a program in the *Nova* series on public television, took close to ten years! Moreover, his proof was based on major contributions of many mathematicians. (The interested reader should consult [Ro10] for more information about Fermat's last theorem and for additional references concerning this problem and its resolution.)

We now state an open problem that is simple to describe, but that seems quite difficult to resolve.

EXAMPLE 23



Watch out! Working on the $3x + 1$ problem can be addictive.

The $3x + 1$ Conjecture Let T be the transformation that sends an even integer x to $x/2$ and an odd integer x to $3x + 1$. A famous conjecture, sometimes known as the **$3x + 1$ conjecture**, states that for all positive integers x , when we repeatedly apply the transformation T , we will eventually reach the integer 1. For example, starting with $x = 13$, we find $T(13) = 3 \cdot 13 + 1 = 40$, $T(40) = 40/2 = 20$, $T(20) = 20/2 = 10$, $T(10) = 10/2 = 5$, $T(5) = 3 \cdot 5 + 1 = 16$, $T(16) = 8$, $T(8) = 4$, $T(4) = 2$, and $T(2) = 1$. The $3x + 1$ conjecture has been verified using computers for all integers x up to $5.6 \cdot 10^{13}$.

The $3x + 1$ conjecture has an interesting history and has attracted the attention of mathematicians since the 1950s. The conjecture has been raised many times and goes by many other names, including the Collatz problem, Hasse's algorithm, Ulam's problem, the Syracuse problem, and Kakutani's problem. Many mathematicians have been diverted from their work to spend time attacking this conjecture. This led to the joke that this problem was part of a conspiracy to slow down American mathematical research. See the article by Jeffrey Lagarias [La10] for a fascinating discussion of this problem and the results that have been found by mathematicians attacking it.

In Chapter 4 we will describe additional open questions about prime numbers. Students already familiar with the basic notions about primes might want to explore Section 4.3, where these open questions are discussed. We will mention other important open questions throughout the book.

Additional Proof Methods

Build up your arsenal of proof methods as you work through this book.

In this chapter we introduced the basic methods used in proofs. We also described how to leverage these methods to prove a variety of results. We will use these proof methods in all subsequent chapters. In particular, we will use them in Chapters 2, 3, and 4 to prove results about sets, functions, algorithms, and number theory and in Chapters 9, 10, and 11 to prove results in graph theory. Among the theorems we will prove is the famous halting theorem which states that there is a problem that cannot be solved using any procedure. However, there are many important proof methods besides those we have covered. We will introduce some of these methods later in this book. In particular, in Section 5.1 we will discuss mathematical induction, which is an extremely useful method for proving statements of the form $\forall n P(n)$, where the domain consists of all positive integers. In Section 5.3 we will introduce structural induction, which can be used to prove results about recursively defined sets. We will use the Cantor diagonalization method, which can be used to prove results about the size of infinite sets, in Section 2.5. In Chapter 6 we will introduce the notion of combinatorial proofs, which can be used to prove results by counting arguments. The reader should note that entire books have been devoted to the activities discussed in this section, including many excellent works by George Pólya ([Po61], [Po71], [Po90]).

Finally, note that we have not given a procedure that can be used for proving theorems in mathematics. It is a deep theorem of mathematical logic that there is no such procedure.

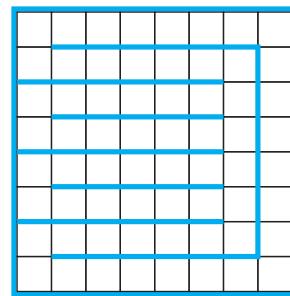
Exercises

1. Prove that $n^2 + 1 \geq 2^n$ when n is a positive integer with $1 \leq n \leq 4$.
2. Prove that there are no positive perfect cubes less than 1000 that are the sum of the cubes of two positive integers.
3. Prove that if x and y are real numbers, then $\max(x, y) + \min(x, y) = x + y$. [Hint: Use a proof by cases, with the two cases corresponding to $x \geq y$ and $x < y$, respectively.]
4. Use a proof by cases to show that $\min(a, \min(b, c)) = \min(\min(a, b), c)$ whenever a, b , and c are real numbers.
5. Prove using the notion of without loss of generality that $\min(x, y) = (x + y - |x - y|)/2$ and $\max(x, y) = (x + y + |x - y|)/2$ whenever x and y are real numbers.
6. Prove using the notion of without loss of generality that $5x + 5y$ is an odd integer when x and y are integers of opposite parity.
7. Prove the **triangle inequality**, which states that if x and y are real numbers, then $|x| + |y| \geq |x + y|$ (where $|x|$ represents the absolute value of x , which equals x if $x \geq 0$ and equals $-x$ if $x < 0$).
8. Prove that there is a positive integer that equals the sum of the positive integers not exceeding it. Is your proof constructive or nonconstructive?
9. Prove that there are 100 consecutive positive integers that are not perfect squares. Is your proof constructive or nonconstructive?
10. Prove that either $2 \cdot 10^{500} + 15$ or $2 \cdot 10^{500} + 16$ is not a perfect square. Is your proof constructive or nonconstructive?
11. Prove that there exists a pair of consecutive integers such that one of these integers is a perfect square and the other is a perfect cube.
12. Show that the product of two of the numbers $65^{1000} - 8^{2001} + 3^{177}$, $79^{1212} - 9^{2399} + 2^{2001}$, and $24^{4493} - 5^{8192} + 7^{1777}$ is nonnegative. Is your proof constructive or nonconstructive? [Hint: Do not try to evaluate these numbers!]
13. Prove or disprove that there is a rational number x and an irrational number y such that x^y is irrational.
14. Prove or disprove that if a and b are rational numbers, then a^b is also rational.
15. Show that each of these statements can be used to express the fact that there is a unique element x such that $P(x)$ is true. [Note that we can also write this statement as $\exists!x P(x)$.]
 - $\exists x \forall y (P(y) \leftrightarrow x = y)$
 - $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \rightarrow x = y)$
 - $\exists x (P(x) \wedge \forall y (P(y) \rightarrow x = y))$
16. Show that if a, b , and c are real numbers and $a \neq 0$, then there is a unique solution of the equation $ax + b = c$.
17. Suppose that a and b are odd integers with $a \neq b$. Show there is a unique integer c such that $|a - c| = |b - c|$.
18. Show that if r is an irrational number, there is a unique integer n such that the distance between r and n is less than $1/2$.
19. Show that if n is an odd integer, then there is a unique integer k such that n is the sum of $k - 2$ and $k + 3$.
20. Prove that given a real number x there exist unique numbers n and ϵ such that $x = n + \epsilon$, n is an integer, and $0 \leq \epsilon < 1$.
21. Prove that given a real number x there exist unique numbers n and ϵ such that $x = n - \epsilon$, n is an integer, and $0 \leq \epsilon < 1$.
22. Use forward reasoning to show that if x is a nonzero real number, then $x^2 + 1/x^2 \geq 2$. [Hint: Start with the inequality $(x - 1/x)^2 \geq 0$ which holds for all nonzero real numbers x .]
23. The **harmonic mean** of two real numbers x and y equals $2xy/(x + y)$. By computing the harmonic and geometric means of different pairs of positive real numbers, formulate a conjecture about their relative sizes and prove your conjecture.
24. The **quadratic mean** of two real numbers x and y equals $\sqrt{(x^2 + y^2)/2}$. By computing the arithmetic and quadratic means of different pairs of positive real numbers, formulate a conjecture about their relative sizes and prove your conjecture.
- *25. Write the numbers $1, 2, \dots, 2n$ on a blackboard, where n is an odd integer. Pick any two of the numbers, j and k , write $|j - k|$ on the board and erase j and k . Continue this process until only one integer is written on the board. Prove that this integer must be odd.
- *26. Suppose that five ones and four zeros are arranged around a circle. Between any two equal bits you insert a 0 and between any two unequal bits you insert a 1 to produce nine new bits. Then you erase the nine original bits. Show that when you iterate this procedure, you can never get nine zeros. [Hint: Work backward, assuming that you did end up with nine zeros.]
27. Formulate a conjecture about the decimal digits that appear as the final decimal digit of the fourth power of an integer. Prove your conjecture using a proof by cases.
28. Formulate a conjecture about the final two decimal digits of the square of an integer. Prove your conjecture using a proof by cases.
29. Prove that there is no positive integer n such that $n^2 + n^3 = 100$.
30. Prove that there are no solutions in integers x and y to the equation $2x^2 + 5y^2 = 14$.
31. Prove that there are no solutions in positive integers x and y to the equation $x^4 + y^4 = 625$.
32. Prove that there are infinitely many solutions in positive integers x , y , and z to the equation $x^2 + y^2 = z^2$. [Hint: Let $x = m^2 - n^2$, $y = 2mn$, and $z = m^2 + n^2$, where m and n are integers.]

- 33.** Adapt the proof in Example 4 in Section 1.7 to prove that if $n = abc$, where a , b , and c are positive integers, then $a \leq \sqrt[3]{n}$, $b \leq \sqrt[3]{n}$, or $c \leq \sqrt[3]{n}$.
- 34.** Prove that $\sqrt[3]{2}$ is irrational.
- 35.** Prove that between every two rational numbers there is an irrational number.
- 36.** Prove that between every rational number and every irrational number there is an irrational number.
- *37.** Let $S = x_1y_1 + x_2y_2 + \cdots + x_ny_n$, where x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n are orderings of two different sequences of positive real numbers, each containing n elements.
- Show that S takes its maximum value over all orderings of the two sequences when both sequences are sorted (so that the elements in each sequence are in nondecreasing order).
 - Show that S takes its minimum value over all orderings of the two sequences when one sequence is sorted into nondecreasing order and the other is sorted into nonincreasing order.
- 38.** Prove or disprove that if you have an 8-gallon jug of water and two empty jugs with capacities of 5 gallons and 3 gallons, respectively, then you can measure 4 gallons by successively pouring some of or all of the water in a jug into another jug.
- 39.** Verify the $3x + 1$ conjecture for these integers.
- 6
 - 7
 - 17
 - 21
- 40.** Verify the $3x + 1$ conjecture for these integers.
- 16
 - 11
 - 35
 - 113
- 41.** Prove or disprove that you can use dominoes to tile the standard checkerboard with two adjacent corners removed (that is, corners that are not opposite).
- 42.** Prove or disprove that you can use dominoes to tile a standard checkerboard with all four corners removed.
- 43.** Prove that you can use dominoes to tile a rectangular checkerboard with an even number of squares.
- 44.** Prove or disprove that you can use dominoes to tile a 5×5 checkerboard with three corners removed.
- 45.** Use a proof by exhaustion to show that a tiling using dominoes of a 4×4 checkerboard with opposite corners removed does not exist. [Hint: First show that you can assume that the squares in the upper left and lower right corners are removed. Number the squares of the original

checkerboard from 1 to 16, starting in the first row, moving right in this row, then starting in the leftmost square in the second row and moving right, and so on. Remove squares 1 and 16. To begin the proof, note that square 2 is covered either by a domino laid horizontally, which covers squares 2 and 3, or vertically, which covers squares 2 and 6. Consider each of these cases separately, and work through all the subcases that arise.]

- *46.** Prove that when a white square and a black square are removed from an 8×8 checkerboard (colored as in the text) you can tile the remaining squares of the checkerboard using dominoes. [Hint: Show that when one black and one white square are removed, each part of the partition of the remaining cells formed by inserting the barriers shown in the figure can be covered by dominoes.]



- 47.** Show that by removing two white squares and two black squares from an 8×8 checkerboard (colored as in the text) you can make it impossible to tile the remaining squares using dominoes.
- *48.** Find all squares, if they exist, on an 8×8 checkerboard such that the board obtained by removing one of these squares can be tiled using straight triominoes. [Hint: First use arguments based on coloring and rotations to eliminate as many squares as possible from consideration.]
- *49.**
 - Draw each of the five different tetrominoes, where a tetromino is a polyomino consisting of four squares.
 - For each of the five different tetrominoes, prove or disprove that you can tile a standard checkerboard using these tetrominoes.
- *50.** Prove or disprove that you can tile a 10×10 checkerboard using straight tetrominoes.

Key Terms and Results

TERMS

proposition: a statement that is true or false

propositional variable: a variable that represents a proposition

truth value: true or false

$\neg p$ (negation of p): the proposition with truth value opposite to the truth value of p

logical operators: operators used to combine propositions

compound proposition: a proposition constructed by combining propositions using logical operators

truth table: a table displaying all possible truth values of propositions

$p \vee q$ (disjunction of p and q): the proposition “ p or q ,” which is true if and only if at least one of p and q is true

$p \wedge q$ (conjunction of p and q): the proposition “ p and q ,” which is true if and only if both p and q are true

$p \oplus q$ (exclusive or of p and q): the proposition “ p XOR q ,” which is true when exactly one of p and q is true

$p \rightarrow q$ (p implies q): the proposition “if p , then q ,” which is false if and only if p is true and q is false

converse of $p \rightarrow q$: the conditional statement $q \rightarrow p$

contrapositive of $p \rightarrow q$: the conditional statement $\neg q \rightarrow \neg p$

inverse of $p \rightarrow q$: the conditional statement $\neg p \rightarrow \neg q$

$p \leftrightarrow q$ (biconditional): the proposition “ p if and only if q ,” which is true if and only if p and q have the same truth value

bit: either a 0 or a 1

Boolean variable: a variable that has a value of 0 or 1

bit operation: an operation on a bit or bits

bit string: a list of bits

bitwise operations: operations on bit strings that operate on each bit in one string and the corresponding bit in the other string

logic gate: a logic element that performs a logical operation on one or more bits to produce an output bit

logic circuit: a switching circuit made up of logic gates that produces one or more output bits

tautology: a compound proposition that is always true

contradiction: a compound proposition that is always false

contingency: a compound proposition that is sometimes true and sometimes false

consistent compound propositions: compound propositions for which there is an assignment of truth values to the variables that makes all these propositions true

satisfiable compound proposition: a compound proposition for which there is an assignment of truth values to its variables that makes it true

logically equivalent compound propositions: compound propositions that always have the same truth values

predicate: part of a sentence that attributes a property to the subject

propositional function: a statement containing one or more variables that becomes a proposition when each of its variables is assigned a value or is bound by a quantifier

domain (or universe) of discourse: the values a variable in a propositional function may take

$\exists x P(x)$ (existential quantification of $P(x)$): the proposition that is true if and only if there exists an x in the domain such that $P(x)$ is true

$\forall x P(x)$ (universal quantification of $P(x)$): the proposition that is true if and only if $P(x)$ is true for every x in the domain

logically equivalent expressions: expressions that have the same truth value no matter which propositional functions and domains are used

free variable: a variable not bound in a propositional function

bound variable: a variable that is quantified

scope of a quantifier: portion of a statement where the quantifier binds its variable

argument: a sequence of statements

argument form: a sequence of compound propositions involving propositional variables

premise: a statement, in an argument, or argument form, other than the final one

conclusion: the final statement in an argument or argument form

valid argument form: a sequence of compound propositions involving propositional variables where the truth of all the premises implies the truth of the conclusion

valid argument: an argument with a valid argument form

rule of inference: a valid argument form that can be used in the demonstration that arguments are valid

fallacy: an invalid argument form often used incorrectly as a rule of inference (or sometimes, more generally, an incorrect argument)

circular reasoning or begging the question: reasoning where one or more steps are based on the truth of the statement being proved

theorem: a mathematical assertion that can be shown to be true

conjecture: a mathematical assertion proposed to be true, but that has not been proved

proof: a demonstration that a theorem is true

axiom: a statement that is assumed to be true and that can be used as a basis for proving theorems

lemma: a theorem used to prove other theorems

corollary: a proposition that can be proved as a consequence of a theorem that has just been proved

vacuous proof: a proof that $p \rightarrow q$ is true based on the fact that p is false

trivial proof: a proof that $p \rightarrow q$ is true based on the fact that q is true

direct proof: a proof that $p \rightarrow q$ is true that proceeds by showing that q must be true when p is true

proof by contraposition: a proof that $p \rightarrow q$ is true that proceeds by showing that p must be false when q is false

proof by contradiction: a proof that p is true based on the truth of the conditional statement $\neg p \rightarrow q$, where q is a contradiction

exhaustive proof: a proof that establishes a result by checking a list of all possible cases

proof by cases: a proof broken into separate cases, where these cases cover all possibilities

without loss of generality: an assumption in a proof that makes it possible to prove a theorem by reducing the number of cases to consider in the proof

counterexample: an element x such that $P(x)$ is false

constructive existence proof: a proof that an element with a specified property exists that explicitly finds such an element

nonconstructive existence proof: a proof that an element with a specified property exists that does not explicitly find such an element

rational number: a number that can be expressed as the ratio of two integers p and q such that $q \neq 0$

uniqueness proof: a proof that there is exactly one element satisfying a specified property

RESULTS

The logical equivalences given in Tables 6, 7, and 8 in Section 1.3.

De Morgan's laws for quantifiers.

Rules of inference for propositional calculus.

Rules of inference for quantified statements.

Review Questions

1. a) Define the negation of a proposition.
b) What is the negation of “This is a boring course”?
2. a) Define (using truth tables) the disjunction, conjunction, exclusive or, conditional, and biconditional of the propositions p and q .
b) What are the disjunction, conjunction, exclusive or, conditional, and biconditional of the propositions “I'll go to the movies tonight” and “I'll finish my discrete mathematics homework”?
3. a) Describe at least five different ways to write the conditional statement $p \rightarrow q$ in English.
b) Define the converse and contrapositive of a conditional statement.
c) State the converse and the contrapositive of the conditional statement “If it is sunny tomorrow, then I will go for a walk in the woods.”
4. a) What does it mean for two propositions to be logically equivalent?
b) Describe the different ways to show that two compound propositions are logically equivalent.
c) Show in at least two different ways that the compound propositions $\neg p \vee (r \rightarrow \neg q)$ and $\neg p \vee \neg q \vee \neg r$ are equivalent.
5. (Depends on the Exercise Set in Section 1.3)
 - a) Given a truth table, explain how to use disjunctive normal form to construct a compound proposition with this truth table.
 - b) Explain why part (a) shows that the operators \wedge , \vee , and \neg are functionally complete.
 - c) Is there an operator such that the set containing just this operator is functionally complete?
6. What are the universal and existential quantifications of a predicate $P(x)$? What are their negations?
7. a) What is the difference between the quantification $\exists x \forall y P(x, y)$ and $\forall y \exists x P(x, y)$, where $P(x, y)$ is a predicate?
8. a) Give an example of a predicate $P(x, y)$ such that $\exists x \forall y P(x, y)$ and $\forall y \exists x P(x, y)$ have different truth values.
9. Describe what is meant by a valid argument in propositional logic and show that the argument “If the earth is flat, then you can sail off the edge of the earth,” “You cannot sail off the edge of the earth,” therefore, “The earth is not flat” is a valid argument.
10. a) Use rules of inference to show that if the premises “All zebras have stripes” and “Mark is a zebra” are true, then the conclusion “Mark has stripes” is true.
b) Describe what is meant by a direct proof, a proof by contraposition, and a proof by contradiction of a conditional statement $p \rightarrow q$.
c) Give a direct proof, a proof by contraposition and a proof by contradiction of the statement: “If n is even, then $n + 4$ is even.”
11. a) Describe a way to prove the biconditional $p \leftrightarrow q$.
b) Prove the statement: “The integer $3n + 2$ is odd if and only if the integer $9n + 5$ is even, where n is an integer.”
12. To prove that the statements p_1 , p_2 , p_3 , and p_4 are equivalent, is it sufficient to show that the conditional statements $p_4 \rightarrow p_2$, $p_3 \rightarrow p_1$, and $p_1 \rightarrow p_2$ are valid? If not, provide another collection of conditional statements that can be used to show that the four statements are equivalent.
13. a) Suppose that a statement of the form $\forall x P(x)$ is false. How can this be proved?
b) Show that the statement “For every positive integer n , $n^2 \geq 2n$ ” is false.
14. What is the difference between a constructive and non-constructive existence proof? Give an example of each.
15. What are the elements of a proof that there is a unique element x such that $P(x)$, where $P(x)$ is a propositional function?
16. Explain how a proof by cases can be used to prove a result about absolute values, such as the fact that $|xy| = |x||y|$ for all real numbers x and y .

Supplementary Exercises

1. Let p be the proposition “I will do every exercise in this book” and q be the proposition “I will get an “A” in this course.” Express each of these as a combination of p and q .
 - a) I will get an “A” in this course only if I do every exercise in this book.
 - b) I will get an “A” in this course and I will do every exercise in this book.
 - c) Either I will not get an “A” in this course or I will not do every exercise in this book.
 - d) For me to get an “A” in this course it is necessary and sufficient that I do every exercise in this book.

2. Find the truth table of the compound proposition $(p \vee q) \rightarrow (p \wedge \neg r)$.
 3. Show that these compound propositions are tautologies.
 - a) $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$
 - b) $((p \vee q) \wedge \neg p) \rightarrow q$
 4. Give the converse, the contrapositive, and the inverse of these conditional statements.
 - a) If it rains today, then I will drive to work.
 - b) If $|x| = x$, then $x \geq 0$.
 - c) If n is greater than 3, then n^2 is greater than 9.
 5. Given a conditional statement $p \rightarrow q$, find the converse of its inverse, the converse of its converse, and the converse of its contrapositive.
 6. Given a conditional statement $p \rightarrow q$, find the inverse of its inverse, the inverse of its converse, and the inverse of its contrapositive.
 7. Find a compound proposition involving the propositional variables p, q, r , and s that is true when exactly three of these propositional variables are true and is false otherwise.
 8. Show that these statements are inconsistent: “If Sergei takes the job offer then he will get a signing bonus.” “If Sergei takes the job offer, then he will receive a higher salary.” “If Sergei gets a signing bonus, then he will not receive a higher salary.” “Sergei takes the job offer.”
 9. Show that these statements are inconsistent: “If Miranda does not take a course in discrete mathematics, then she will not graduate.” “If Miranda does not graduate, then she is not qualified for the job.” “If Miranda reads this book, then she is qualified for the job.” “Miranda does not take a course in discrete mathematics but she reads this book.”
- Teachers in the Middle Ages supposedly tested the realtime propositional logic ability of a student via a technique known as an **obligato game**. In an obligato game, a number of rounds is set and in each round the teacher gives the student successive assertions that the student must either accept or reject as they are given. When the student accepts an assertion, it is added as a commitment; when the student rejects an assertion its negation is added as a commitment. The student passes the test if the consistency of all commitments is maintained throughout the test.
10. Suppose that in a three-round obligato game, the teacher first gives the student the proposition $p \rightarrow q$, then the proposition $\neg(p \vee r) \vee q$, and finally the proposition q . For which of the eight possible sequences of three answers will the student pass the test?
 11. Suppose that in a four-round obligato game, the teacher first gives the student the proposition $\neg(p \rightarrow (q \wedge r))$, then the proposition $p \vee \neg q$, then the proposition $\neg r$, and finally, the proposition $(p \wedge r) \vee (q \rightarrow p)$. For which of the 16 possible sequences of four answers will the student pass the test?
 12. Explain why every obligato game has a winning strategy.
- Exercises 13 and 14 are set on the island of knights and knaves described in Example 7 in Section 1.2.
13. Suppose that you meet three people Aaron, Bohan, and Crystal. Can you determine what Aaron, Bohan, and Crystal are if Aaron says “All of us are knaves” and Bohan says “Exactly one of us is a knave.”?
 14. Suppose that you meet three people, Anita, Boris, and Carmen. What are Anita, Boris, and Carmen if Anita says “I am a knave and Boris is a knight” and Boris says “Exactly one of the three of us is a knight”?
 15. (Adapted from [Sm78]) Suppose that on an island there are three types of people, knights, knaves, and normals (also known as spies). Knights always tell the truth, knaves always lie, and normals sometimes lie and sometimes tell the truth. Detectives questioned three inhabitants of the island—Amy, Brenda, and Claire—as part of the investigation of a crime. The detectives knew that one of the three committed the crime, but not which one. They also knew that the criminal was a knight, and that the other two were not. Additionally, the detectives recorded these statements: Amy: “I am innocent.” Brenda: “What Amy says is true.” Claire: “Brenda is not a normal.” After analyzing their information, the detectives positively identified the guilty party. Who was it?
 16. Show that if S is a proposition, where S is the conditional statement “If S is true, then unicorns live,” then “Unicorns live” is true. Show that it follows that S cannot be a proposition. (This paradox is known as *Löb’s paradox*.)
 17. Show that the argument with premises “The tooth fairy is a real person” and “The tooth fairy is not a real person” and conclusion “You can find gold at the end of the rainbow” is a valid argument. Does this show that the conclusion is true?
 18. Suppose that the truth value of the proposition p_i is **T** whenever i is an odd positive integer and is **F** whenever i is an even positive integer. Find the truth values of $\bigvee_{i=1}^{100} (p_i \wedge p_{i+1})$ and $\bigwedge_{i=1}^{100} (p_i \vee p_{i+1})$.
 - *19. Model 16 × 16 Sudoku puzzles (with 4 × 4 blocks) as satisfiability problems.
 20. Let $P(x)$ be the statement “Student x knows calculus” and let $Q(y)$ be the statement “Class y contains a student who knows calculus.” Express each of these as quantifications of $P(x)$ and $Q(y)$.
 - a) Some students know calculus.
 - b) Not every student knows calculus.
 - c) Every class has a student in it who knows calculus.
 - d) Every student in every class knows calculus.
 - e) There is at least one class with no students who know calculus.
 21. Let $P(m, n)$ be the statement “ m divides n ,” where the domain for both variables consists of all positive integers. (By “ m divides n ” we mean that $n = km$ for some integer k .) Determine the truth values of each of these statements.

a) $P(4, 5)$	b) $P(2, 4)$
c) $\forall m \forall n P(m, n)$	d) $\exists m \forall n P(m, n)$
e) $\exists n \forall m P(m, n)$	f) $\forall n P(1, n)$
 22. Find a domain for the quantifiers in $\exists x \exists y (x \neq y \wedge \forall z ((z = x) \vee (z = y)))$ such that this statement is true.

- 23.** Find a domain for the quantifiers in $\exists x \exists y (x \neq y \wedge \forall z((z = x) \vee (z = y)))$ such that this statement is false.
- 24.** Use existential and universal quantifiers to express the statement “No one has more than three grandmothers” using the propositional function $G(x, y)$, which represents “ x is the grandmother of y .”
- 25.** Use existential and universal quantifiers to express the statement “Everyone has exactly two biological parents” using the propositional function $P(x, y)$, which represents “ x is the biological parent of y .”
- 26.** The quantifier \exists_n denotes “there exists exactly n ,” so that $\exists_n x P(x)$ means there exist exactly n values in the domain such that $P(x)$ is true. Determine the true value of these statements where the domain consists of all real numbers.
- a) $\exists_0 x (x^2 = -1)$ b) $\exists_1 x (|x| = 0)$
 c) $\exists_2 x (x^2 = 2)$ d) $\exists_3 x (x = |x|)$
- 27.** Express each of these statements using existential and universal quantifiers and propositional logic where \exists_n is defined in Exercise 26.
- a) $\exists_0 x P(x)$ b) $\exists_1 x P(x)$
 c) $\exists_2 x P(x)$ d) $\exists_3 x P(x)$
- 28.** Let $P(x, y)$ be a propositional function. Show that $\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$ is a tautology.
- 29.** Let $P(x)$ and $Q(x)$ be propositional functions. Show that $\exists x (P(x) \rightarrow Q(x))$ and $\forall x P(x) \rightarrow \exists x Q(x)$ always have the same truth value.
- 30.** If $\forall y \exists x P(x, y)$ is true, does it necessarily follow that $\exists x \forall y P(x, y)$ is true?
- 31.** If $\forall x \exists y P(x, y)$ is true, does it necessarily follow that $\exists x \forall y P(x, y)$ is true?
- 32.** Find the negations of these statements.
- a) If it snows today, then I will go skiing tomorrow.
 b) Every person in this class understands mathematical induction.
 c) Some students in this class do not like discrete mathematics.
 d) In every mathematics class there is some student who falls asleep during lectures.
- 33.** Express this statement using quantifiers: “Every student in this class has taken some course in every department in the school of mathematical sciences.”
- 34.** Express this statement using quantifiers: “There is a building on the campus of some college in the United States in which every room is painted white.”
- 35.** Express the statement “There is exactly one student in this class who has taken exactly one mathematics class at this school” using the uniqueness quantifier. Then express this statement using quantifiers, without using the uniqueness quantifier.
- 36.** Describe a rule of inference that can be used to prove that there are exactly two elements x and y in a domain such that $P(x)$ and $P(y)$ are true. Express this rule of inference as a statement in English.
- 37.** Use rules of inference to show that if the premises $\forall x(P(x) \rightarrow Q(x))$, $\forall x(Q(x) \rightarrow R(x))$, and $\neg R(a)$, where a is in the domain, are true, then the conclusion $\neg P(a)$ is true.
- 38.** Prove that if x^3 is irrational, then x is irrational.
- 39.** Prove that if x is irrational and $x \geq 0$, then \sqrt{x} is irrational.
- 40.** Prove that given a nonnegative integer n , there is a unique nonnegative integer m such that $m^2 \leq n < (m+1)^2$.
- 41.** Prove that there exists an integer m such that $m^2 > 10^{1000}$. Is your proof constructive or nonconstructive?
- 42.** Prove that there is a positive integer that can be written as the sum of squares of positive integers in two different ways. (Use a computer or calculator to speed up your work.)
- 43.** Disprove the statement that every positive integer is the sum of the cubes of eight nonnegative integers.
- 44.** Disprove the statement that every positive integer is the sum of at most two squares and a cube of nonnegative integers.
- 45.** Disprove the statement that every positive integer is the sum of 36 fifth powers of nonnegative integers.
- 46.** Assuming the truth of the theorem that states that \sqrt{n} is irrational whenever n is a positive integer that is not a perfect square, prove that $\sqrt{2} + \sqrt{3}$ is irrational.

Computer Projects

Write programs with the specified input and output.

- Given the truth values of the propositions p and q , find the truth values of the conjunction, disjunction, exclusive or, conditional statement, and biconditional of these propositions.
- Given two bit strings of length n , find the bitwise AND, bitwise OR, and bitwise XOR of these strings.
- * Given a compound proposition, determine whether it is satisfiable by checking its truth value for all positive assignments of truth values to its propositional variables.
- Given the truth values of the propositions p and q in fuzzy logic, find the truth value of the disjunction and the conjunction of p and q (see Exercises 46 and 47 of Section 1.1).
- * Given positive integers m and n , interactively play the game of Chomp.
- * Given a portion of a checkerboard, look for tilings of this checkerboard with various types of polyominoes, including dominoes, the two types of triominoes, and larger polyominoes.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Look for positive integers that are not the sum of the cubes of nine different positive integers.
2. Look for positive integers greater than 79 that are not the sum of the fourth powers of 18 positive integers.
3. Find as many positive integers as you can that can be written as the sum of cubes of positive integers, in two different ways, sharing this property with 1729.
- *4. Try to find winning strategies for the game of Chomp for different initial configurations of cookies.
5. Construct the 12 different pentominoes, where a pentomino is a polyomino consisting of five squares.
6. Find all the rectangles of 60 squares that can be tiled using every one of the 12 different pentominoes.

Writing Projects

Respond to these with essays using outside sources.

1. Discuss logical paradoxes, including the paradox of Epimenides the Cretan, Jourdain's card paradox, and the barber paradox, and how they are resolved.
2. Describe how fuzzy logic is being applied to practical applications. Consult one or more of the recent books on fuzzy logic written for general audiences.
3. Describe some of the practical problems that can be modeled as satisfiability problems.
4. Describe some of the techniques that have been devised to help people solve Sudoku puzzles without the use of a computer.
5. Describe the basic rules of *WFF'N PROOF*, *The Game of Modern Logic*, developed by Layman Allen. Give examples of some of the games included in *WFF'N PROOF*.
6. Read some of the writings of Lewis Carroll on symbolic logic. Describe in detail some of the models he used to represent logical arguments and the rules of inference he used in these arguments.
7. Extend the discussion of Prolog given in Section 1.4, explaining in more depth how Prolog employs resolution.
8. Discuss some of the techniques used in computational logic, including Skolem's rule.
9. "Automated theorem proving" is the task of using computers to mechanically prove theorems. Discuss the goals and applications of automated theorem proving and the progress made in developing automated theorem provers.
10. Describe how DNA computing has been used to solve instances of the satisfiability problem.
11. Look up some of the incorrect proofs of famous open questions and open questions that were solved since 1970 and describe the type of error made in each proof.
12. Discuss what is known about winning strategies in the game of Chomp.
13. Describe various aspects of proof strategy discussed by George Pólya in his writings on reasoning, including [Po62], [Po71], and [Po90].
14. Describe a few problems and results about tilings with polyominoes, as described in [Go94] and [Ma91], for example.

2

Basic Structures: Sets, Functions, Sequences, Sums, and Matrices

- [2.1 Sets](#)
- [2.2 Set Operations](#)
- [2.3 Functions](#)
- [2.4 Sequences and Summations](#)
- [2.5 Cardinality of Sets](#)
- [2.6 Matrices](#)

Much of discrete mathematics is devoted to the study of discrete structures, used to represent discrete objects. Many important discrete structures are built using sets, which are collections of objects. Among the discrete structures built from sets are combinations, unordered collections of objects used extensively in counting; relations, sets of ordered pairs that represent relationships between objects; graphs, sets of vertices and edges that connect vertices; and finite state machines, used to model computing machines. These are some of the topics we will study in later chapters.

The concept of a function is extremely important in discrete mathematics. A function assigns to each element of a first set exactly one element of a second set, where the two sets are not necessarily distinct. Functions play important roles throughout discrete mathematics. They are used to represent the computational complexity of algorithms, to study the size of sets, to count objects, and in a myriad of other ways. Useful structures such as sequences and strings are special types of functions. In this chapter, we will introduce the notion of a sequence, which represents ordered lists of elements. Furthermore, we will introduce some important types of sequences and we will show how to define the terms of a sequence using earlier terms. We will also address the problem of identifying a sequence from its first few terms.

In our study of discrete mathematics, we will often add consecutive terms of a sequence of numbers. Because adding terms from a sequence, as well as other indexed sets of numbers, is such a common occurrence, a special notation has been developed for adding such terms. In this chapter, we will introduce the notation used to express summations. We will develop formulae for certain types of summations that appear throughout the study of discrete mathematics. For instance, we will encounter such summations in the analysis of the number of steps used by an algorithm to sort a list of numbers so that its terms are in increasing order.

The relative sizes of infinite sets can be studied by introducing the notion of the size, or cardinality, of a set. We say that a set is countable when it is finite or has the same size as the set of positive integers. In this chapter we will establish the surprising result that the set of rational numbers is countable, while the set of real numbers is not. We will also show how the concepts we discuss can be used to show that there are functions that cannot be computed using a computer program in any programming language.

Matrices are used in discrete mathematics to represent a variety of discrete structures. We will review the basic material about matrices and matrix arithmetic needed to represent relations and graphs. The matrix arithmetic we study will be used to solve a variety of problems involving these structures.

2.1 Sets

Introduction

In this section, we study the fundamental discrete structure on which all other discrete structures are built, namely, the set. Sets are used to group objects together. Often, but not always, the objects in a set have similar properties. For instance, all the students who are currently enrolled in your school make up a set. Likewise, all the students currently taking a course in discrete mathematics at any school make up a set. In addition, those students enrolled in your school who are taking a course in discrete mathematics form a set that can be obtained by taking the elements common to the first two collections. The language of sets is a means to study such

collections in an organized fashion. We now provide a definition of a set. This definition is an intuitive definition, which is not part of a formal theory of sets.

DEFINITION 1

A *set* is an unordered collection of objects, called *elements* or *members* of the set. A set is said to *contain* its elements. We write $a \in A$ to denote that a is an element of the set A . The notation $a \notin A$ denotes that a is not an element of the set A .

It is common for sets to be denoted using uppercase letters. Lowercase letters are usually used to denote elements of sets.

There are several ways to describe a set. One way is to list all the members of a set, when this is possible. We use a notation where all members of the set are listed between braces. For example, the notation $\{a, b, c, d\}$ represents the set with the four elements a, b, c , and d . This way of describing a set is known as the **roster method**.

EXAMPLE 1 The set V of all vowels in the English alphabet can be written as $V = \{a, e, i, o, u\}$.

EXAMPLE 2 The set O of odd positive integers less than 10 can be expressed by $O = \{1, 3, 5, 7, 9\}$.

EXAMPLE 3 Although sets are usually used to group together elements with common properties, there is nothing that prevents a set from having seemingly unrelated elements. For instance, $\{a, 2, \text{Fred}, \text{New Jersey}\}$ is the set containing the four elements $a, 2$, Fred, and New Jersey.

Sometimes the roster method is used to describe a set without listing all its members. Some members of the set are listed, and then *ellipses* (...) are used when the general pattern of the elements is obvious.

EXAMPLE 4 The set of positive integers less than 100 can be denoted by $\{1, 2, 3, \dots, 99\}$.



Another way to describe a set is to use **set builder** notation. We characterize all those elements in the set by stating the property or properties they must have to be members. For instance, the set O of all odd positive integers less than 10 can be written as

$$O = \{x \mid x \text{ is an odd positive integer less than } 10\},$$

or, specifying the universe as the set of positive integers, as

$$O = \{x \in \mathbf{Z}^+ \mid x \text{ is odd and } x < 10\}.$$

We often use this type of notation to describe sets when it is impossible to list all the elements of the set. For instance, the set \mathbf{Q}^+ of all positive rational numbers can be written as

$$\mathbf{Q}^+ = \{x \in \mathbf{R} \mid x = \frac{p}{q}, \text{ for some positive integers } p \text{ and } q\}.$$

These sets, each denoted using a boldface letter, play an important role in discrete mathematics:

$\mathbf{N} = \{0, 1, 2, 3, \dots\}$, the set of **natural numbers**

$\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$, the set of **integers**

$\mathbf{Z}^+ = \{1, 2, 3, \dots\}$, the set of **positive integers**

$\mathbf{Q} = \{p/q \mid p \in \mathbf{Z}, q \in \mathbf{Z}, \text{ and } q \neq 0\}$, the set of **rational numbers**

\mathbf{R} , the set of **real numbers**

\mathbf{R}^+ , the set of **positive real numbers**

\mathbf{C} , the set of **complex numbers**.

Beware that mathematicians disagree whether 0 is a natural number. We consider it quite natural.

(Note that some people do not consider 0 a natural number, so be careful to check how the term *natural numbers* is used when you read other books.)

Recall the notation for **intervals** of real numbers. When a and b are real numbers with $a < b$, we write

$$[a, b] = \{x \mid a \leq x \leq b\}$$

$$[a, b) = \{x \mid a \leq x < b\}$$

$$(a, b] = \{x \mid a < x \leq b\}$$

$$(a, b) = \{x \mid a < x < b\}$$

Note that $[a, b]$ is called the **closed interval** from a to b and (a, b) is called the **open interval** from a to b .

Sets can have other sets as members, as Example 5 illustrates.

EXAMPLE 5

The set $\{\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}\}$ is a set containing four elements, each of which is a set. The four elements of this set are **N**, the set of natural numbers; **Z**, the set of integers; **Q**, the set of rational numbers; and **R**, the set of real numbers. 

Remark: Note that the concept of a datatype, or type, in computer science is built upon the concept of a set. In particular, a **datatype** or **type** is the name of a set, together with a set of operations that can be performed on objects from that set. For example, *boolean* is the name of the set $\{0, 1\}$ together with operators on one or more elements of this set, such as AND, OR, and NOT.

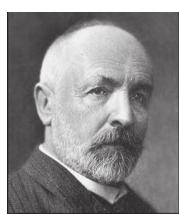
Because many mathematical statements assert that two differently specified collections of objects are really the same set, we need to understand what it means for two sets to be equal.

DEFINITION 2

Two sets are *equal* if and only if they have the same elements. Therefore, if A and B are sets, then A and B are equal if and only if $\forall x(x \in A \leftrightarrow x \in B)$. We write $A = B$ if A and B are equal sets.

EXAMPLE 6

The sets $\{1, 3, 5\}$ and $\{3, 5, 1\}$ are equal, because they have the same elements. Note that the order in which the elements of a set are listed does not matter. Note also that it does not matter if an element of a set is listed more than once, so $\{1, 3, 3, 3, 5, 5, 5, 5\}$ is the same as the set $\{1, 3, 5\}$ because they have the same elements. 



GEORG CANTOR (1845–1918) Georg Cantor was born in St. Petersburg, Russia, where his father was a successful merchant. Cantor developed his interest in mathematics in his teens. He began his university studies in Zurich in 1862, but when his father died he left Zurich. He continued his university studies at the University of Berlin in 1863, where he studied under the eminent mathematicians Weierstrass, Kummer, and Kronecker. He received his doctor's degree in 1867, after having written a dissertation on number theory. Cantor assumed a position at the University of Halle in 1869, where he continued working until his death.

Cantor is considered the founder of set theory. His contributions in this area include the discovery that the set of real numbers is uncountable. He is also noted for his many important contributions to analysis. Cantor also was interested in philosophy and wrote papers relating his theory of sets with metaphysics.

Cantor married in 1874 and had five children. His melancholy temperament was balanced by his wife's happy disposition. Although he received a large inheritance from his father, he was poorly paid as a professor. To mitigate this, he tried to obtain a better-paying position at the University of Berlin. His appointment there was blocked by Kronecker, who did not agree with Cantor's views on set theory. Cantor suffered from mental illness throughout the later years of his life. He died in 1918 from a heart attack.

THE EMPTY SET There is a special set that has no elements. This set is called the **empty set**, or **null set**, and is denoted by \emptyset . The empty set can also be denoted by $\{ \}$ (that is, we represent the empty set with a pair of braces that encloses all the elements in this set). Often, a set of elements with certain properties turns out to be the null set. For instance, the set of all positive integers that are greater than their squares is the null set.

$\{\emptyset\}$ has one more element than \emptyset .

A set with one element is called a **singleton set**. A common error is to confuse the empty set \emptyset with the set $\{\emptyset\}$, which is a singleton set. The single element of the set $\{\emptyset\}$ is the empty set itself! A useful analogy for remembering this difference is to think of folders in a computer file system. The empty set can be thought of as an empty folder and the set consisting of just the empty set can be thought of as a folder with exactly one folder inside, namely, the empty folder.

NAIVE SET THEORY Note that the term *object* has been used in the definition of a set, Definition 1, without specifying what an object is. This description of a set as a collection of objects, based on the intuitive notion of an object, was first stated in 1895 by the German mathematician Georg Cantor. The theory that results from this intuitive definition of a set, and the use of the intuitive notion that for any property whatever, there is a set consisting of exactly the objects with this property, leads to **paradoxes**, or logical inconsistencies. This was shown by the English philosopher Bertrand Russell in 1902 (see Exercise 46 for a description of one of these paradoxes). These logical inconsistencies can be avoided by building set theory beginning with axioms. However, we will use Cantor's original version of set theory, known as **naive set theory**, in this book because all sets considered in this book can be treated consistently using Cantor's original theory. Students will find familiarity with naive set theory helpful if they go on to learn about axiomatic set theory. They will also find the development of axiomatic set theory much more abstract than the material in this text. We refer the interested reader to [Su72] to learn more about axiomatic set theory.



Venn Diagrams



Sets can be represented graphically using Venn diagrams, named after the English mathematician John Venn, who introduced their use in 1881. In Venn diagrams the **universal set** U , which contains all the objects under consideration, is represented by a rectangle. (Note that the universal set varies depending on which objects are of interest.) Inside this rectangle, circles or other geometrical figures are used to represent sets. Sometimes points are used to represent the particular elements of the set. Venn diagrams are often used to indicate the relationships between sets. We show how a Venn diagram can be used in Example 7.

EXAMPLE 7 Draw a Venn diagram that represents V , the set of vowels in the English alphabet.

Solution: We draw a rectangle to indicate the universal set U , which is the set of the 26 letters of the English alphabet. Inside this rectangle we draw a circle to represent V . Inside this circle we indicate the elements of V with points (see Figure 1).

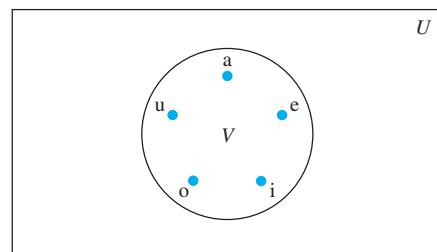


FIGURE 1 Venn Diagram for the Set of Vowels.

Subsets

It is common to encounter situations where the elements of one set are also the elements of a second set. We now introduce some terminology and notation to express such relationships between sets.

DEFINITION 3

The set A is a *subset* of B if and only if every element of A is also an element of B . We use the notation $A \subseteq B$ to indicate that A is a subset of the set B .

We see that $A \subseteq B$ if and only if the quantification

$$\forall x(x \in A \rightarrow x \in B)$$

is true. Note that to show that A is not a subset of B we need only find one element $x \in A$ with $x \notin B$. Such an x is a counterexample to the claim that $x \in A$ implies $x \in B$.

We have these useful rules for determining whether one set is a subset of another:

Showing that A is a Subset of B To show that $A \subseteq B$, show that if x belongs to A then x also belongs to B .

Showing that A is Not a Subset of B To show that $A \not\subseteq B$, find a single $x \in A$ such that $x \notin B$.

EXAMPLE 8

The set of all odd positive integers less than 10 is a subset of the set of all positive integers less than 10, the set of rational numbers is a subset of the set of real numbers, the set of all computer science majors at your school is a subset of the set of all students at your school, and the set of all people in China is a subset of the set of all people in China (that is, it is a subset of itself). Each of these facts follows immediately by noting that an element that belongs to the first set in each pair of sets also belongs to the second set in that pair. 

EXAMPLE 9

The set of integers with squares less than 100 is not a subset of the set of nonnegative integers because -1 is in the former set [as $(-1)^2 < 100$], but not the later set. The set of people who have taken discrete mathematics at your school is not a subset of the set of all computer science majors at your school if there is at least one student who has taken discrete mathematics who is not a computer science major. 

Links



BERTRAND RUSSELL (1872–1970) Bertrand Russell was born into a prominent English family active in the progressive movement and having a strong commitment to liberty. He became an orphan at an early age and was placed in the care of his father's parents, who had him educated at home. He entered Trinity College, Cambridge, in 1890, where he excelled in mathematics and in moral science. He won a fellowship on the basis of his work on the foundations of geometry. In 1910 Trinity College appointed him to a lectureship in logic and the philosophy of mathematics.

Russell fought for progressive causes throughout his life. He held strong pacifist views, and his protests against World War I led to dismissal from his position at Trinity College. He was imprisoned for 6 months in 1918 because of an article he wrote that was branded as seditious. Russell fought for women's suffrage in Great Britain. In 1961, at the age of 89, he was imprisoned for the second time for his protests advocating nuclear disarmament.

Russell's greatest work was in his development of principles that could be used as a foundation for all of mathematics. His most famous work is *Principia Mathematica*, written with Alfred North Whitehead, which attempts to deduce all of mathematics using a set of primitive axioms. He wrote many books on philosophy, physics, and his political ideas. Russell won the Nobel Prize for literature in 1950.

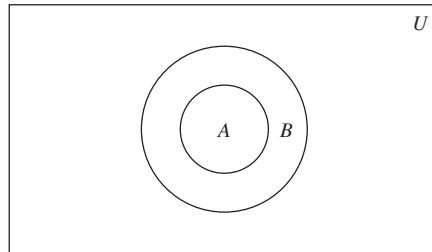


FIGURE 2 Venn Diagram Showing that A Is a Subset of B .

Theorem 1 shows that every nonempty set S is guaranteed to have at least two subsets, the empty set and the set S itself, that is, $\emptyset \subseteq S$ and $S \subseteq S$.

THEOREM 1

For every set S , (i) $\emptyset \subseteq S$ and (ii) $S \subseteq S$.

Proof: We will prove (i) and leave the proof of (ii) as an exercise.

Let S be a set. To show that $\emptyset \subseteq S$, we must show that $\forall x(x \in \emptyset \rightarrow x \in S)$ is true. Because the empty set contains no elements, it follows that $x \in \emptyset$ is always false. It follows that the conditional statement $x \in \emptyset \rightarrow x \in S$ is always true, because its hypothesis is always false and a conditional statement with a false hypothesis is true. Therefore, $\forall x(x \in \emptyset \rightarrow x \in S)$ is true. This completes the proof of (i). Note that this is an example of a vacuous proof. \triangleleft

When we wish to emphasize that a set A is a subset of a set B but that $A \neq B$, we write $A \subset B$ and say that A is a **proper subset** of B . For $A \subset B$ to be true, it must be the case that $A \subseteq B$ and there must exist an element x of B that is not an element of A . That is, A is a proper subset of B if and only if

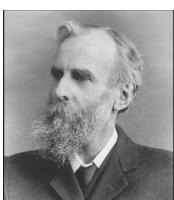
$$\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$$

is true. Venn diagrams can be used to illustrate that a set A is a subset of a set B . We draw the universal set U as a rectangle. Within this rectangle we draw a circle for B . Because A is a subset of B , we draw the circle for A within the circle for B . This relationship is shown in Figure 2.

A useful way to show that two sets have the same elements is to show that each set is a subset of the other. In other words, we can show that if A and B are sets with $A \subseteq B$ and $B \subseteq A$, then $A = B$. That is, $A = B$ if and only if $\forall x(x \in A \rightarrow x \in B)$ and $\forall x(x \in B \rightarrow x \in A)$ or equivalently if and only if $\forall x(x \in A \leftrightarrow x \in B)$, which is what it means for the A and B to be equal. Because this method of showing two sets are equal is so useful, we highlight it here.



JOHN VENN (1834–1923) John Venn was born into a London suburban family noted for its philanthropy. He attended London schools and got his mathematics degree from Caius College, Cambridge, in 1857. He was elected a fellow of this college and held his fellowship there until his death. He took holy orders in 1859 and, after a brief stint of religious work, returned to Cambridge, where he developed programs in the moral sciences. Besides his mathematical work, Venn had an interest in history and wrote extensively about his college and family.



Venn's book *Symbolic Logic* clarifies ideas originally presented by Boole. In this book, Venn presents a systematic development of a method that uses geometric figures, known now as *Venn diagrams*. Today these diagrams are primarily used to analyze logical arguments and to illustrate relationships between sets. In addition to his work on symbolic logic, Venn made contributions to probability theory described in his widely used textbook on that subject.

Showing Two Sets are Equal To show that two sets A and B are equal, show that $A \subseteq B$ and $B \subseteq A$.

Sets may have other sets as members. For instance, we have the sets

$$A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \quad \text{and} \quad B = \{x \mid x \text{ is a subset of the set } \{a, b\}\}.$$

Note that these two sets are equal, that is, $A = B$. Also note that $\{a\} \in A$, but $a \notin A$.

The Size of a Set

Sets are used extensively in counting problems, and for such applications we need to discuss the sizes of sets.

DEFINITION 4

Let S be a set. If there are exactly n distinct elements in S where n is a nonnegative integer, we say that S is a *finite set* and that n is the *cardinality* of S . The cardinality of S is denoted by $|S|$.

Remark: The term *cardinality* comes from the common usage of the term *cardinal number* as the size of a finite set.

EXAMPLE 10 Let A be the set of odd positive integers less than 10. Then $|A| = 5$.

EXAMPLE 11 Let S be the set of letters in the English alphabet. Then $|S| = 26$.

EXAMPLE 12 Because the null set has no elements, it follows that $|\emptyset| = 0$.

We will also be interested in sets that are not finite.

DEFINITION 5

A set is said to be *infinite* if it is not finite.

EXAMPLE 13 The set of positive integers is infinite.



We will extend the notion of cardinality to infinite sets in Section 2.5, a challenging topic full of surprising results.

Power Sets

Many problems involve testing all combinations of elements of a set to see if they satisfy some property. To consider all such combinations of elements of a set S , we build a new set that has as its members all the subsets of S .

DEFINITION 6

Given a set S , the *power set* of S is the set of all subsets of the set S . The power set of S is denoted by $\mathcal{P}(S)$.

EXAMPLE 14 What is the power set of the set $\{0, 1, 2\}$?



Solution: The power set $\mathcal{P}(\{0, 1, 2\})$ is the set of all subsets of $\{0, 1, 2\}$. Hence,

$$\mathcal{P}(\{0, 1, 2\}) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}.$$

Note that the empty set and the set itself are members of this set of subsets.

EXAMPLE 15 What is the power set of the empty set? What is the power set of the set $\{\emptyset\}$?

Solution: The empty set has exactly one subset, namely, itself. Consequently,

$$\mathcal{P}(\emptyset) = \{\emptyset\}.$$

The set $\{\emptyset\}$ has exactly two subsets, namely, \emptyset and the set $\{\emptyset\}$ itself. Therefore,

$$\mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}. \quad \blacktriangleleft$$

If a set has n elements, then its power set has 2^n elements. We will demonstrate this fact in several ways in subsequent sections of the text.

Cartesian Products

The order of elements in a collection is often important. Because sets are unordered, a different structure is needed to represent ordered collections. This is provided by **ordered n -tuples**.

DEFINITION 7

The *ordered n -tuple* (a_1, a_2, \dots, a_n) is the ordered collection that has a_1 as its first element, a_2 as its second element, \dots , and a_n as its n th element.

We say that two ordered n -tuples are equal if and only if each corresponding pair of their elements is equal. In other words, $(a_1, a_2, \dots, a_n) = (b_1, b_2, \dots, b_n)$ if and only if $a_i = b_i$, for $i = 1, 2, \dots, n$. In particular, ordered 2-tuples are called **ordered pairs**. The ordered pairs (a, b) and (c, d) are equal if and only if $a = c$ and $b = d$. Note that (a, b) and (b, a) are not equal unless $a = b$.



RENÉ DESCARTES (1596–1650) René Descartes was born into a noble family near Tours, France, about 200 miles southwest of Paris. He was the third child of his father's first wife; she died several days after his birth. Because of René's poor health, his father, a provincial judge, let his son's formal lessons slide until, at the age of 8, René entered the Jesuit college at La Flèche. The rector of the school took a liking to him and permitted him to stay in bed until late in the morning because of his frail health. From then on, Descartes spent his mornings in bed; he considered these times his most productive hours for thinking.

Descartes left school in 1612, moving to Paris, where he spent 2 years studying mathematics. He earned a law degree in 1616 from the University of Poitiers. At 18 Descartes became disgusted with studying and decided to see the world. He moved to Paris and became a successful gambler. However, he grew tired of bawdy living and moved to the suburb of Saint-Germain, where he devoted himself to mathematical study. When his gambling friends found him, he decided to leave France and undertake a military career. However, he never did any fighting. One day, while escaping the cold in an overheated room at a military encampment, he had several feverish dreams, which revealed his future career as a mathematician and philosopher.

After ending his military career, he traveled throughout Europe. He then spent several years in Paris, where he studied mathematics and philosophy and constructed optical instruments. Descartes decided to move to Holland, where he spent 20 years wandering around the country, accomplishing his most important work. During this time he wrote several books, including the *Discours*, which contains his contributions to analytic geometry, for which he is best known. He also made fundamental contributions to philosophy.

In 1649 Descartes was invited by Queen Christina to visit her court in Sweden to tutor her in philosophy. Although he was reluctant to live in what he called "the land of bears amongst rocks and ice," he finally accepted the invitation and moved to Sweden. Unfortunately, the winter of 1649–1650 was extremely bitter. Descartes caught pneumonia and died in mid-February.

Many of the discrete structures we will study in later chapters are based on the notion of the *Cartesian product* of sets (named after René Descartes). We first define the Cartesian product of two sets.

DEFINITION 8

Let A and B be sets. The *Cartesian product* of A and B , denoted by $A \times B$, is the set of all ordered pairs (a, b) , where $a \in A$ and $b \in B$. Hence,

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

EXAMPLE 16

Let A represent the set of all students at a university, and let B represent the set of all courses offered at the university. What is the Cartesian product $A \times B$ and how can it be used?



Solution: The Cartesian product $A \times B$ consists of all the ordered pairs of the form (a, b) , where a is a student at the university and b is a course offered at the university. One way to use the set $A \times B$ is to represent all possible enrollments of students in courses at the university.

EXAMPLE 17

What is the Cartesian product of $A = \{1, 2\}$ and $B = \{a, b, c\}$?

Solution: The Cartesian product $A \times B$ is

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}.$$

Note that the Cartesian products $A \times B$ and $B \times A$ are not equal, unless $A = \emptyset$ or $B = \emptyset$ (so that $A \times B = \emptyset$) or $A = B$ (see Exercises 31 and 38). This is illustrated in Example 18.

EXAMPLE 18

Show that the Cartesian product $B \times A$ is not equal to the Cartesian product $A \times B$, where A and B are as in Example 17.

Solution: The Cartesian product $B \times A$ is

$$B \times A = \{(a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2)\}.$$

This is not equal to $A \times B$, which was found in Example 17.

The Cartesian product of more than two sets can also be defined.

DEFINITION 9

The *Cartesian product* of the sets A_1, A_2, \dots, A_n , denoted by $A_1 \times A_2 \times \dots \times A_n$, is the set of ordered n -tuples (a_1, a_2, \dots, a_n) , where a_i belongs to A_i for $i = 1, 2, \dots, n$. In other words,

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \dots, n\}.$$

EXAMPLE 19 What is the Cartesian product $A \times B \times C$, where $A = \{0, 1\}$, $B = \{1, 2\}$, and $C = \{0, 1, 2\}$?

Solution: The Cartesian product $A \times B \times C$ consists of all ordered triples (a, b, c) , where $a \in A$, $b \in B$, and $c \in C$. Hence,

$$A \times B \times C = \{(0, 1, 0), (0, 1, 1), (0, 1, 2), (0, 2, 0), (0, 2, 1), (0, 2, 2), (1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}.$$

Remark: Note that when A , B , and C are sets, $(A \times B) \times C$ is not the same as $A \times B \times C$ (see Exercise 39).

We use the notation A^2 to denote $A \times A$, the Cartesian product of the set A with itself. Similarly, $A^3 = A \times A \times A$, $A^4 = A \times A \times A \times A$, and so on. More generally,

$$A^n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A \text{ for } i = 1, 2, \dots, n\}.$$

EXAMPLE 20 Suppose that $A = \{1, 2\}$. It follows that $A^2 = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$ and $A^3 = \{(1, 1, 1), (1, 1, 2), (1, 2, 1), (1, 2, 2), (2, 1, 1), (2, 1, 2), (2, 2, 1), (2, 2, 2)\}$.

A subset R of the Cartesian product $A \times B$ is called a **relation** from the set A to the set B . The elements of R are ordered pairs, where the first element belongs to A and the second to B . For example, $R = \{(a, 0), (a, 1), (a, 3), (b, 1), (b, 2), (c, 0), (c, 3)\}$ is a relation from the set $\{a, b, c\}$ to the set $\{0, 1, 2, 3\}$. A relation from a set A to itself is called a relation on A .

EXAMPLE 21 What are the ordered pairs in the less than or equal to relation, which contains (a, b) if $a \leq b$, on the set $\{0, 1, 2, 3\}$?

Solution: The ordered pair (a, b) belongs to R if and only if both a and b belong to $\{0, 1, 2, 3\}$ and $a \leq b$. Consequently, the ordered pairs in R are $(0, 0), (0, 1), (0, 2), (0, 3), (1, 1), (1, 2), (1, 3), (2, 2), (2, 3)$, and $(3, 3)$.

We will study relations and their properties at length in Chapter 9.

Using Set Notation with Quantifiers

Sometimes we restrict the domain of a quantified statement explicitly by making use of a particular notation. For example, $\forall x \in S(P(x))$ denotes the universal quantification of $P(x)$ over all elements in the set S . In other words, $\forall x \in S(P(x))$ is shorthand for $\forall x(x \in S \rightarrow P(x))$. Similarly, $\exists x \in S(P(x))$ denotes the existential quantification of $P(x)$ over all elements in S . That is, $\exists x \in S(P(x))$ is shorthand for $\exists x(x \in S \wedge P(x))$.

EXAMPLE 22 What do the statements $\forall x \in \mathbf{R}(x^2 \geq 0)$ and $\exists x \in \mathbf{Z}(x^2 = 1)$ mean?

Solution: The statement $\forall x \in \mathbf{R}(x^2 \geq 0)$ states that for every real number x , $x^2 \geq 0$. This statement can be expressed as “The square of every real number is nonnegative.” This is a true statement.

The statement $\exists x \in \mathbf{Z}(x^2 = 1)$ states that there exists an integer x such that $x^2 = 1$. This statement can be expressed as “There is an integer whose square is 1.” This is also a true statement because $x = 1$ is such an integer (as is -1).

Truth Sets and Quantifiers

We will now tie together concepts from set theory and from predicate logic. Given a predicate P , and a domain D , we define the **truth set** of P to be the set of elements x in D for which $P(x)$ is true. The truth set of $P(x)$ is denoted by $\{x \in D \mid P(x)\}$.

EXAMPLE 23 What are the truth sets of the predicates $P(x)$, $Q(x)$, and $R(x)$, where the domain is the set of integers and $P(x)$ is “ $|x| = 1$,” $Q(x)$ is “ $x^2 = 2$,” and $R(x)$ is “ $|x| = x$.”

Solution: The truth set of P , $\{x \in \mathbf{Z} \mid |x| = 1\}$, is the set of integers for which $|x| = 1$. Because $|x| = 1$ when $x = 1$ or $x = -1$, and for no other integers x , we see that the truth set of P is the set $\{-1, 1\}$.

The truth set of Q , $\{x \in \mathbf{Z} \mid x^2 = 2\}$, is the set of integers for which $x^2 = 2$. This is the empty set because there are no integers x for which $x^2 = 2$.

The truth set of R , $\{x \in \mathbf{Z} \mid |x| = x\}$, is the set of integers for which $|x| = x$. Because $|x| = x$ if and only if $x \geq 0$, it follows that the truth set of R is \mathbf{N} , the set of nonnegative integers. 

Note that $\forall x P(x)$ is true over the domain U if and only if the truth set of P is the set U . Likewise, $\exists x P(x)$ is true over the domain U if and only if the truth set of P is nonempty.

Exercises

1. List the members of these sets.
 - a) $\{x \mid x \text{ is a real number such that } x^2 = 1\}$
 - b) $\{x \mid x \text{ is a positive integer less than } 12\}$
 - c) $\{x \mid x \text{ is the square of an integer and } x < 100\}$
 - d) $\{x \mid x \text{ is an integer such that } x^2 = 2\}$
 2. Use set builder notation to give a description of each of these sets.
 - a) $\{0, 3, 6, 9, 12\}$
 - b) $\{-3, -2, -1, 0, 1, 2, 3\}$
 - c) $\{m, n, o, p\}$
 3. For each of these pairs of sets, determine whether the first is a subset of the second, the second is a subset of the first, or neither is a subset of the other.
 - a) the set of airline flights from New York to New Delhi, the set of nonstop airline flights from New York to New Delhi
 - b) the set of people who speak English, the set of people who speak Chinese
 - c) the set of flying squirrels, the set of living creatures that can fly
 4. For each of these pairs of sets, determine whether the first is a subset of the second, the second is a subset of the first, or neither is a subset of the other.
 - a) the set of people who speak English, the set of people who speak English with an Australian accent
 - b) the set of fruits, the set of citrus fruits
 - c) the set of students studying discrete mathematics, the set of students studying data structures
 5. Determine whether each of these pairs of sets are equal.

<ol style="list-style-type: none"> a) $\{1, 3, 3, 3, 5, 5, 5, 5, 5\}$ b) $\{\{1\}\}, \{1, \{1\}\}$ 	<ol style="list-style-type: none"> c) $\emptyset, \{\emptyset\}$ d) $\{2, \{2\}\}$ e) $\{\{2\}, \{2, \{2\}\}\}$ f) $\{\{\{2\}\}\}$
--	--
6. Suppose that $A = \{2, 4, 6\}$, $B = \{2, 6\}$, $C = \{4, 6\}$, and $D = \{4, 6, 8\}$. Determine which of these sets are subsets of which other of these sets.
 7. For each of the following sets, determine whether 2 is an element of that set.

<ol style="list-style-type: none"> a) $\{x \in \mathbf{R} \mid x \text{ is an integer greater than } 1\}$ b) $\{x \in \mathbf{R} \mid x \text{ is the square of an integer}\}$ 	<ol style="list-style-type: none"> c) $\{2, \{2\}\}$ d) $\{\{2\}, \{\{2\}\}\}$ e) $\{\{2\}, \{2, \{2\}\}\}$ f) $\{\{\{2\}\}\}$
--	--
 8. For each of the sets in Exercise 7, determine whether $\{2\}$ is an element of that set.
 9. Determine whether each of these statements is true or false.

<ol style="list-style-type: none"> a) $0 \in \emptyset$ c) $\{0\} \subset \emptyset$ e) $\{0\} \in \{0\}$ g) $\{\emptyset\} \subseteq \{\emptyset\}$ 	<ol style="list-style-type: none"> b) $\emptyset \in \{0\}$ d) $\emptyset \subset \{0\}$ f) $\{0\} \subset \{0\}$
--	---
 10. Determine whether these statements are true or false.

<ol style="list-style-type: none"> a) $\emptyset \in \{\emptyset\}$ c) $\{\emptyset\} \in \{\emptyset\}$ e) $\{\emptyset\} \subset \{\emptyset, \{\emptyset\}\}$ g) $\{\{\emptyset\}\} \subset \{\{\emptyset\}, \{\emptyset\}\}$ 	<ol style="list-style-type: none"> b) $\emptyset \in \{\emptyset, \{\emptyset\}\}$ d) $\{\emptyset\} \in \{\{\emptyset\}\}$ f) $\{\{\emptyset\}\} \subset \{\emptyset, \{\emptyset\}\}$
--	---
 11. Determine whether each of these statements is true or false.

<ol style="list-style-type: none"> a) $x \in \{x\}$ c) $\{x\} \subseteq \{x\}$ e) $\{x\} \in \{x\}$ 	<ol style="list-style-type: none"> b) $\{x\} \subseteq \{x\}$ d) $\{x\} \in \{\{x\}\}$ f) $\emptyset \subseteq \{x\}$
---	---
 12. Use a Venn diagram to illustrate the subset of odd integers in the set of all positive integers not exceeding 10.

- 13.** Use a Venn diagram to illustrate the set of all months of the year whose names do not contain the letter R in the set of all months of the year.
- 14.** Use a Venn diagram to illustrate the relationship $A \subseteq B$ and $B \subseteq C$.
- 15.** Use a Venn diagram to illustrate the relationships $A \subset B$ and $B \subset C$.
- 16.** Use a Venn diagram to illustrate the relationships $A \subset B$ and $A \subset C$.
- 17.** Suppose that A , B , and C are sets such that $A \subseteq B$ and $B \subseteq C$. Show that $A \subseteq C$.
- 18.** Find two sets A and B such that $A \in B$ and $A \subseteq B$.
- 19.** What is the cardinality of each of these sets?
- a) $\{a\}$
 - b) $\{\{a\}\}$
 - c) $\{a, \{a\}\}$
 - d) $\{a, \{a\}, \{a, \{a\}\}\}$
- 20.** What is the cardinality of each of these sets?
- a) \emptyset
 - b) $\{\emptyset\}$
 - c) $\{\emptyset, \{\emptyset\}\}$
 - d) $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$
- 21.** Find the power set of each of these sets, where a and b are distinct elements.
- a) $\{a\}$
 - b) $\{a, b\}$
 - c) $\{\emptyset, \{\emptyset\}\}$
- 22.** Can you conclude that $A = B$ if A and B are two sets with the same power set?
- 23.** How many elements does each of these sets have where a and b are distinct elements?
- a) $P(\{a, b, \{a, b\}\})$
 - b) $P(\{\emptyset, a, \{a\}, \{\{a\}\}\})$
 - c) $P(P(\emptyset))$
- 24.** Determine whether each of these sets is the power set of a set, where a and b are distinct elements.
- a) \emptyset
 - b) $\{\emptyset, \{a\}\}$
 - c) $\{\emptyset, \{a\}, \{\emptyset, a\}\}$
 - d) $\{\emptyset, \{a\}, \{b\}, \{a, b\}\}$
- 25.** Prove that $P(A) \subseteq P(B)$ if and only if $A \subseteq B$.
- 26.** Show that if $A \subseteq C$ and $B \subseteq D$, then $A \times B \subseteq C \times D$
- 27.** Let $A = \{a, b, c, d\}$ and $B = \{y, z\}$. Find
- a) $A \times B$.
 - b) $B \times A$.
- 28.** What is the Cartesian product $A \times B$, where A is the set of courses offered by the mathematics department at a university and B is the set of mathematics professors at this university? Give an example of how this Cartesian product can be used.
- 29.** What is the Cartesian product $A \times B \times C$, where A is the set of all airlines and B and C are both the set of all cities in the United States? Give an example of how this Cartesian product can be used.
- 30.** Suppose that $A \times B = \emptyset$, where A and B are sets. What can you conclude?
- 31.** Let A be a set. Show that $\emptyset \times A = A \times \emptyset = \emptyset$.
- 32.** Let $A = \{a, b, c\}$, $B = \{x, y\}$, and $C = \{0, 1\}$. Find
- a) $A \times B \times C$.
 - b) $C \times B \times A$.
 - c) $C \times A \times B$.
 - d) $B \times B \times B$.
- 33.** Find A^2 if
- a) $A = \{0, 1, 3\}$.
 - b) $A = \{1, 2, a, b\}$.
- 34.** Find A^3 if
- a) $A = \{a\}$.
 - b) $A = \{0, a\}$.
- 35.** How many different elements does $A \times B$ have if A has m elements and B has n elements?
- 36.** How many different elements does $A \times B \times C$ have if A has m elements, B has n elements, and C has p elements?
- 37.** How many different elements does A^n have when A has m elements and n is a positive integer?
- 38.** Show that $A \times B \neq B \times A$, when A and B are nonempty, unless $A = B$.
- 39.** Explain why $A \times B \times C$ and $(A \times B) \times C$ are not the same.
- 40.** Explain why $(A \times B) \times (C \times D)$ and $A \times (B \times C) \times D$ are not the same.
- 41.** Translate each of these quantifications into English and determine its truth value.
- a) $\forall x \in \mathbf{R} (x^2 \neq -1)$
 - b) $\exists x \in \mathbf{Z} (x^2 = 2)$
 - c) $\forall x \in \mathbf{Z} (x^2 > 0)$
 - d) $\exists x \in \mathbf{R} (x^2 = x)$
- 42.** Translate each of these quantifications into English and determine its truth value.
- a) $\exists x \in \mathbf{R} (x^3 = -1)$
 - b) $\exists x \in \mathbf{Z} (x + 1 > x)$
 - c) $\forall x \in \mathbf{Z} (x - 1 \in \mathbf{Z})$
 - d) $\forall x \in \mathbf{Z} (x^2 \in \mathbf{Z})$
- 43.** Find the truth set of each of these predicates where the domain is the set of integers.
- a) $P(x): x^2 < 3$
 - b) $Q(x): x^2 > x$
 - c) $R(x): 2x + 1 = 0$
- 44.** Find the truth set of each of these predicates where the domain is the set of integers.
- a) $P(x): x^3 \geq 1$
 - b) $Q(x): x^2 = 2$
 - c) $R(x): x < x^2$
- *45.** The defining property of an ordered pair is that two ordered pairs are equal if and only if their first elements are equal and their second elements are equal. Surprisingly, instead of taking the ordered pair as a primitive concept, we can construct ordered pairs using basic notions from set theory. Show that if we define the ordered pair (a, b) to be $\{\{a\}, \{a, b\}\}$, then $(a, b) = (c, d)$ if and only if $a = c$ and $b = d$. [Hint: First show that $\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}$ if and only if $a = c$ and $b = d$.]
- *46.** This exercise presents **Russell's paradox**. Let S be the set that contains a set x if the set x does not belong to itself, so that $S = \{x \mid x \notin x\}$.
- a) Show the assumption that S is a member of S leads to a contradiction.
 - b) Show the assumption that S is not a member of S leads to a contradiction.
- By parts (a) and (b) it follows that the set S cannot be defined as it was. This paradox can be avoided by restricting the types of elements that sets can have.
- *47.** Describe a procedure for listing all the subsets of a finite set.

2.2 Set Operations

Introduction

Two, or more, sets can be combined in many different ways. For instance, starting with the set of mathematics majors at your school and the set of computer science majors at your school, we can form the set of students who are mathematics majors or computer science majors, the set of students who are joint majors in mathematics and computer science, the set of all students not majoring in mathematics, and so on.



DEFINITION 1

Let A and B be sets. The *union* of the sets A and B , denoted by $A \cup B$, is the set that contains those elements that are either in A or in B , or in both.

An element x belongs to the union of the sets A and B if and only if x belongs to A or x belongs to B . This tells us that

$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$

The Venn diagram shown in Figure 1 represents the union of two sets A and B . The area that represents $A \cup B$ is the shaded area within either the circle representing A or the circle representing B .

We will give some examples of the union of sets.

EXAMPLE 1

The union of the sets $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{1, 2, 3, 5\}$; that is, $\{1, 3, 5\} \cup \{1, 2, 3\} = \{1, 2, 3, 5\}$.

EXAMPLE 2

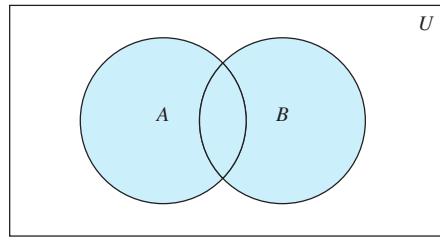
The union of the set of all computer science majors at your school and the set of all mathematics majors at your school is the set of students at your school who are majoring either in mathematics or in computer science (or in both).

DEFINITION 2

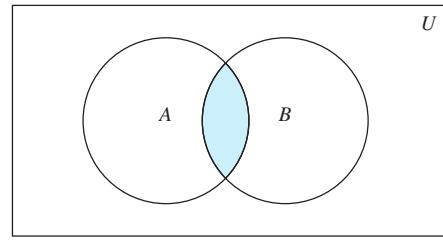
Let A and B be sets. The *intersection* of the sets A and B , denoted by $A \cap B$, is the set containing those elements in both A and B .

An element x belongs to the intersection of the sets A and B if and only if x belongs to A and x belongs to B . This tells us that

$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$



$A \cup B$ is shaded.



$A \cap B$ is shaded.

FIGURE 1 Venn Diagram of the Union of A and B .

FIGURE 2 Venn Diagram of the Intersection of A and B .

The Venn diagram shown in Figure 2 represents the intersection of two sets A and B . The shaded area that is within both the circles representing the sets A and B is the area that represents the intersection of A and B .

We give some examples of the intersection of sets.

EXAMPLE 3 The intersection of the sets $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{1, 3\}$; that is, $\{1, 3, 5\} \cap \{1, 2, 3\} = \{1, 3\}$.

EXAMPLE 4 The intersection of the set of all computer science majors at your school and the set of all mathematics majors is the set of all students who are joint majors in mathematics and computer science.

DEFINITION 3

Two sets are called *disjoint* if their intersection is the empty set.

EXAMPLE 5 Let $A = \{1, 3, 5, 7, 9\}$ and $B = \{2, 4, 6, 8, 10\}$. Because $A \cap B = \emptyset$, A and B are disjoint.

Be careful not to overcount!

We are often interested in finding the cardinality of a union of two finite sets A and B . Note that $|A| + |B|$ counts each element that is in A but not in B or in B but not in A exactly once, and each element that is in both A and B exactly twice. Thus, if the number of elements that are in both A and B is subtracted from $|A| + |B|$, elements in $A \cap B$ will be counted only once. Hence,

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

The generalization of this result to unions of an arbitrary number of sets is called the **principle of inclusion–exclusion**. The principle of inclusion–exclusion is an important technique used in enumeration. We will discuss this principle and other counting techniques in detail in Chapters 6 and 8.

There are other important ways to combine sets.

DEFINITION 4

Let A and B be sets. The *difference* of A and B , denoted by $A - B$, is the set containing those elements that are in A but not in B . The difference of A and B is also called the *complement of B with respect to A* .

Remark: The difference of sets A and B is sometimes denoted by $A \setminus B$.

An element x belongs to the difference of A and B if and only if $x \in A$ and $x \notin B$. This tells us that

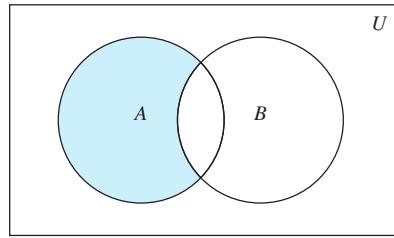
$$A - B = \{x \mid x \in A \wedge x \notin B\}.$$

The Venn diagram shown in Figure 3 represents the difference of the sets A and B . The shaded area inside the circle that represents A and outside the circle that represents B is the area that represents $A - B$.

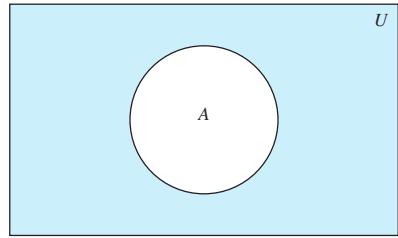
We give some examples of differences of sets.

EXAMPLE 6 The difference of $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{5\}$; that is, $\{1, 3, 5\} - \{1, 2, 3\} = \{5\}$. This is different from the difference of $\{1, 2, 3\}$ and $\{1, 3, 5\}$, which is the set $\{2\}$.

EXAMPLE 7 The difference of the set of computer science majors at your school and the set of mathematics majors at your school is the set of all computer science majors at your school who are not also mathematics majors.



A - B is shaded.

FIGURE 3 Venn Diagram for the Difference of A and B. \bar{A} is shaded.**FIGURE 4** Venn Diagram for the Complement of the Set A.

Once the universal set U has been specified, the **complement** of a set can be defined.

DEFINITION 5

Let U be the universal set. The *complement* of the set A , denoted by \bar{A} , is the complement of A with respect to U . Therefore, the complement of the set A is $U - A$.

An element belongs to \bar{A} if and only if $x \notin A$. This tells us that

$$\bar{A} = \{x \in U \mid x \notin A\}.$$

In Figure 4 the shaded area outside the circle representing A is the area representing \bar{A} .

We give some examples of the complement of a set.

EXAMPLE 8 Let $A = \{a, e, i, o, u\}$ (where the universal set is the set of letters of the English alphabet). Then $\bar{A} = \{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z\}$. 

EXAMPLE 9 Let A be the set of positive integers greater than 10 (with universal set the set of all positive integers). Then $\bar{A} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. 

It is left to the reader (Exercise 19) to show that we can express the difference of A and B as the intersection of A and the complement of B . That is,

$$A - B = A \cap \bar{B}.$$

Set Identities

Set identities and propositional equivalences are just special cases of identities for Boolean algebra.

Table 1 lists the most important set identities. We will prove several of these identities here, using three different methods. These methods are presented to illustrate that there are often many different approaches to the solution of a problem. The proofs of the remaining identities will be left as exercises. The reader should note the similarity between these set identities and the logical equivalences discussed in Section 1.3. (Compare Table 6 of Section 1.6 and Table 1.) In fact, the set identities given can be proved directly from the corresponding logical equivalences. Furthermore, both are special cases of identities that hold for Boolean algebra (discussed in Chapter 12).

One way to show that two sets are equal is to show that each is a subset of the other. Recall that to show that one set is a subset of a second set, we can show that if an element belongs to the first set, then it must also belong to the second set. We generally use a direct proof to do this. We illustrate this type of proof by establishing the first of De Morgan's laws.

TABLE 1 Set Identities.

<i>Identity</i>	<i>Name</i>
$A \cap U = A$ $A \cup \emptyset = A$	Identity laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$(\overline{A}) = A$	Complementation law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive laws
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Complement laws

EXAMPLE 10 Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

This identity says that the complement of the intersection of two sets is the union of their complements.



Solution: We will prove that the two sets $\overline{A \cap B}$ and $\overline{A} \cup \overline{B}$ are equal by showing that each set is a subset of the other.

First, we will show that $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$. We do this by showing that if x is in $\overline{A \cap B}$, then it must also be in $\overline{A} \cup \overline{B}$. Now suppose that $x \in \overline{A \cap B}$. By the definition of complement, $x \notin A \cap B$. Using the definition of intersection, we see that the proposition $\neg((x \in A) \wedge (x \in B))$ is true.

By applying De Morgan's law for propositions, we see that $\neg(x \in A) \vee \neg(x \in B)$. Using the definition of negation of propositions, we have $x \notin A$ or $x \notin B$. Using the definition of the complement of a set, we see that this implies that $x \in \overline{A}$ or $x \in \overline{B}$. Consequently, by the definition of union, we see that $x \in \overline{A} \cup \overline{B}$. We have now shown that $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$.

Next, we will show that $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$. We do this by showing that if x is in $\overline{A} \cup \overline{B}$, then it must also be in $\overline{A \cap B}$. Now suppose that $x \in \overline{A} \cup \overline{B}$. By the definition of union, we know that $x \in \overline{A}$ or $x \in \overline{B}$. Using the definition of complement, we see that $x \notin A$ or $x \notin B$. Consequently, the proposition $\neg(x \in A) \vee \neg(x \in B)$ is true.

By De Morgan's law for propositions, we conclude that $\neg((x \in A) \wedge (x \in B))$ is true. By the definition of intersection, it follows that $\neg(x \in A \cap B)$. We now use the definition of complement to conclude that $x \in \overline{A \cap B}$. This shows that $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$.

Because we have shown that each set is a subset of the other, the two sets are equal, and the identity is proved.

We can more succinctly express the reasoning used in Example 10 using set builder notation, as Example 11 illustrates.

EXAMPLE 11 Use set builder notation and logical equivalences to establish the first De Morgan law $\overline{A \cap B} = A \cup \overline{B}$.

Solution: We can prove this identity with the following steps.

$$\begin{aligned}
 \overline{A \cap B} &= \{x \mid x \notin A \cap B\} && \text{by definition of complement} \\
 &= \{x \mid \neg(x \in (A \cap B))\} && \text{by definition of does not belong symbol} \\
 &= \{x \mid \neg(x \in A \wedge x \in B)\} && \text{by definition of intersection} \\
 &= \{x \mid \neg(x \in A) \vee \neg(x \in B)\} && \text{by the first De Morgan law for logical equivalences} \\
 &= \{x \mid x \notin A \vee x \notin B\} && \text{by definition of does not belong symbol} \\
 &= \{x \mid x \in \overline{A} \vee x \in \overline{B}\} && \text{by definition of complement} \\
 &= \{x \mid x \in \overline{A} \cup \overline{B}\} && \text{by definition of union} \\
 &= \overline{A} \cup \overline{B} && \text{by meaning of set builder notation}
 \end{aligned}$$

Note that besides the definitions of complement, union, set membership, and set builder notation, this proof uses the second De Morgan law for logical equivalences. 

Proving a set identity involving more than two sets by showing each side of the identity is a subset of the other often requires that we keep track of different cases, as illustrated by the proof in Example 12 of one of the distributive laws for sets.

EXAMPLE 12 Prove the second distributive law from Table 1, which states that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ for all sets A , B , and C .

Solution: We will prove this identity by showing that each side is a subset of the other side.

Suppose that $x \in A \cap (B \cup C)$. Then $x \in A$ and $x \in B \cup C$. By the definition of union, it follows that $x \in A$, and $x \in B$ or $x \in C$ (or both). In other words, we know that the compound proposition $(x \in A) \wedge ((x \in B) \vee (x \in C))$ is true. By the distributive law for conjunction over disjunction, it follows that $((x \in A) \wedge (x \in B)) \vee ((x \in A) \wedge (x \in C))$. We conclude that either $x \in A$ and $x \in B$, or $x \in A$ and $x \in C$. By the definition of intersection, it follows that $x \in A \cap B$ or $x \in A \cap C$. Using the definition of union, we conclude that $x \in (A \cap B) \cup (A \cap C)$. We conclude that $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.

Now suppose that $x \in (A \cap B) \cup (A \cap C)$. Then, by the definition of union, $x \in A \cap B$ or $x \in A \cap C$. By the definition of intersection, it follows that $x \in A$ and $x \in B$ or that $x \in A$ and $x \in C$. From this we see that $x \in A$, and $x \in B$ or $x \in C$. Consequently, by the definition of union we see that $x \in A$ and $x \in B \cup C$. Furthermore, by the definition of intersection, it follows that $x \in A \cap (B \cup C)$. We conclude that $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. This completes the proof of the identity. 

Set identities can also be proved using **membership tables**. We consider each combination of sets that an element can belong to and verify that elements in the same combinations of sets belong to both the sets in the identity. To indicate that an element is in a set, a 1 is used; to indicate that an element is not in a set, a 0 is used. (The reader should note the similarity between membership tables and truth tables.)

EXAMPLE 13 Use a membership table to show that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Solution: The membership table for these combinations of sets is shown in Table 2. This table has eight rows. Because the columns for $A \cap (B \cup C)$ and $(A \cap B) \cup (A \cap C)$ are the same, the identity is valid. 

Additional set identities can be established using those that we have already proved. Consider Example 14.

TABLE 2 A Membership Table for the Distributive Property.

A	B	C	$B \cup C$	$A \cap (B \cup C)$	$A \cap B$	$A \cap C$	$(A \cap B) \cup (A \cap C)$
1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1
1	0	1	1	1	0	1	1
1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

EXAMPLE 14 Let A , B , and C be sets. Show that

$$\overline{A \cup (B \cap C)} = (\overline{C} \cup \overline{B}) \cap \overline{A}.$$

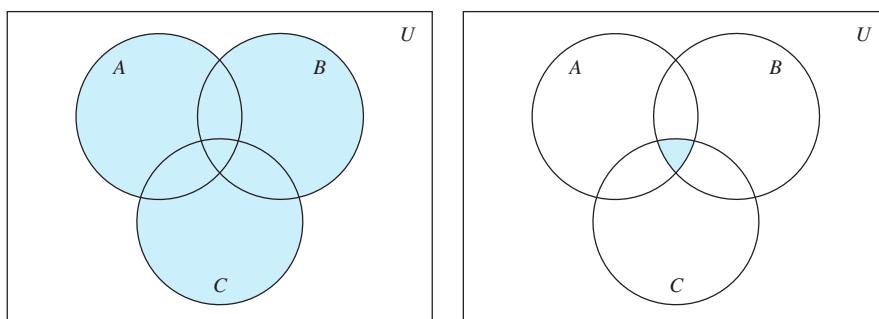
Solution: We have

$$\begin{aligned} \overline{A \cup (B \cap C)} &= \overline{A} \cap \overline{(B \cap C)} && \text{by the first De Morgan law} \\ &= \overline{A} \cap (\overline{B} \cup \overline{C}) && \text{by the second De Morgan law} \\ &= (\overline{B} \cup \overline{C}) \cap \overline{A} && \text{by the commutative law for intersections} \\ &= (\overline{C} \cup \overline{B}) \cap \overline{A} && \text{by the commutative law for unions.} \end{aligned}$$



Generalized Unions and Intersections

Because unions and intersections of sets satisfy associative laws, the sets $A \cup B \cup C$ and $A \cap B \cap C$ are well defined; that is, the meaning of this notation is unambiguous when A , B , and C are sets. That is, we do not have to use parentheses to indicate which operation comes first because $A \cup (B \cup C) = (A \cup B) \cup C$ and $A \cap (B \cap C) = (A \cap B) \cap C$. Note that $A \cup B \cup C$ contains those elements that are in at least one of the sets A , B , and C , and that $A \cap B \cap C$ contains those elements that are in all of A , B , and C . These combinations of the three sets, A , B , and C , are shown in Figure 5.

(a) $A \cup B \cup C$ is shaded.(b) $A \cap B \cap C$ is shaded.**FIGURE 5** The Union and Intersection of A , B , and C .

EXAMPLE 15 Let $A = \{0, 2, 4, 6, 8\}$, $B = \{0, 1, 2, 3, 4\}$, and $C = \{0, 3, 6, 9\}$. What are $A \cup B \cup C$ and $A \cap B \cap C$?

Solution: The set $A \cup B \cup C$ contains those elements in at least one of A , B , and C . Hence,

$$A \cup B \cup C = \{0, 1, 2, 3, 4, 6, 8, 9\}.$$

The set $A \cap B \cap C$ contains those elements in all three of A , B , and C . Thus,

$$A \cap B \cap C = \{0\}.$$

We can also consider unions and intersections of an arbitrary number of sets. We introduce these definitions.

DEFINITION 6

The *union* of a collection of sets is the set that contains those elements that are members of at least one set in the collection.

We use the notation

$$A_1 \cup A_2 \cup \dots \cup A_n = \bigcup_{i=1}^n A_i$$

to denote the union of the sets A_1, A_2, \dots, A_n .

DEFINITION 7

The *intersection* of a collection of sets is the set that contains those elements that are members of all the sets in the collection.

We use the notation

$$A_1 \cap A_2 \cap \dots \cap A_n = \bigcap_{i=1}^n A_i$$

to denote the intersection of the sets A_1, A_2, \dots, A_n . We illustrate generalized unions and intersections with Example 16.

EXAMPLE 16 For $i = 1, 2, \dots$, let $A_i = \{i, i + 1, i + 2, \dots\}$. Then,

$$\bigcup_{i=1}^n A_i = \bigcup_{i=1}^n \{i, i + 1, i + 2, \dots\} = \{1, 2, 3, \dots\},$$

and

$$\bigcap_{i=1}^n A_i = \bigcap_{i=1}^n \{i, i + 1, i + 2, \dots\} = \{n, n + 1, n + 2, \dots\} = A_n.$$

We can extend the notation we have introduced for unions and intersections to other families of sets. In particular, we use the notation

$$A_1 \cup A_2 \cup \dots \cup A_n \cup \dots = \bigcup_{i=1}^{\infty} A_i$$

to denote the union of the sets $A_1, A_2, \dots, A_n, \dots$. Similarly, the intersection of these sets is denoted by

$$A_1 \cap A_2 \cap \dots \cap A_n \cap \dots = \bigcap_{i=1}^{\infty} A_i.$$

More generally, when I is a set, the notations $\bigcap_{i \in I} A_i$ and $\bigcup_{i \in I} A_i$ are used to denote the intersection and union of the sets A_i for $i \in I$, respectively. Note that we have $\bigcap_{i \in I} A_i = \{x \mid \forall i \in I (x \in A_i)\}$ and $\bigcup_{i \in I} A_i = \{x \mid \exists i \in I (x \in A_i)\}$.

EXAMPLE 17 Suppose that $A_i = \{1, 2, 3, \dots, i\}$ for $i = 1, 2, 3, \dots$. Then,

$$\bigcup_{i=1}^{\infty} A_i = \bigcup_{i=1}^{\infty} \{1, 2, 3, \dots, i\} = \{1, 2, 3, \dots\} = \mathbf{Z}^+$$

and

$$\bigcap_{i=1}^{\infty} A_i = \bigcap_{i=1}^{\infty} \{1, 2, 3, \dots, i\} = \{1\}.$$

To see that the union of these sets is the set of positive integers, note that every positive integer n is in at least one of the sets, because it belongs to $A_n = \{1, 2, \dots, n\}$, and every element of the sets in the union is a positive integer. To see that the intersection of these sets is the set $\{1\}$, note that the only element that belongs to all the sets A_1, A_2, \dots is 1. To see this note that $A_1 = \{1\}$ and $1 \in A_i$ for $i = 1, 2, \dots$ 

Computer Representation of Sets

There are various ways to represent sets using a computer. One method is to store the elements of the set in an unordered fashion. However, if this is done, the operations of computing the union, intersection, or difference of two sets would be time-consuming, because each of these operations would require a large amount of searching for elements. We will present a method for storing elements using an arbitrary ordering of the elements of the universal set. This method of representing sets makes computing combinations of sets easy.

Assume that the universal set U is finite (and of reasonable size so that the number of elements of U is not larger than the memory size of the computer being used). First, specify an arbitrary ordering of the elements of U , for instance a_1, a_2, \dots, a_n . Represent a subset A of U with the bit string of length n , where the i th bit in this string is 1 if a_i belongs to A and is 0 if a_i does not belong to A . Example 18 illustrates this technique.

EXAMPLE 18 Let $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, and the ordering of elements of U has the elements in increasing order; that is, $a_i = i$. What bit strings represent the subset of all odd integers in U , the subset of all even integers in U , and the subset of integers not exceeding 5 in U ?

Solution: The bit string that represents the set of odd integers in U , namely, $\{1, 3, 5, 7, 9\}$, has a one bit in the first, third, fifth, seventh, and ninth positions, and a zero elsewhere. It is

$$10\ 1010\ 1010.$$

(We have split this bit string of length ten into blocks of length four for easy reading.) Similarly, we represent the subset of all even integers in U , namely, $\{2, 4, 6, 8, 10\}$, by the string

$$01\ 0101\ 0101.$$

The set of all integers in U that do not exceed 5, namely, $\{1, 2, 3, 4, 5\}$, is represented by the string

$$11\ 1110\ 0000.$$

Using bit strings to represent sets, it is easy to find complements of sets and unions, intersections, and differences of sets. To find the bit string for the complement of a set from the bit string for that set, we simply change each 1 to a 0 and each 0 to 1, because $x \in A$ if and only if $x \notin \bar{A}$. Note that this operation corresponds to taking the negation of each bit when we associate a bit with a truth value—with 1 representing true and 0 representing false.

EXAMPLE 19 We have seen that the bit string for the set $\{1, 3, 5, 7, 9\}$ (with universal set $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$) is

$$10\ 1010\ 1010.$$

What is the bit string for the complement of this set?

Solution: The bit string for the complement of this set is obtained by replacing 0s with 1s and vice versa. This yields the string

$$01\ 0101\ 0101,$$

which corresponds to the set $\{2, 4, 6, 8, 10\}$.

To obtain the bit string for the union and intersection of two sets we perform bitwise Boolean operations on the bit strings representing the two sets. The bit in the i th position of the bit string of the union is 1 if either of the bits in the i th position in the two strings is 1 (or both are 1), and is 0 when both bits are 0. Hence, the bit string for the union is the bitwise *OR* of the bit strings for the two sets. The bit in the i th position of the bit string of the intersection is 1 when the bits in the corresponding position in the two strings are both 1, and is 0 when either of the two bits is 0 (or both are). Hence, the bit string for the intersection is the bitwise *AND* of the bit strings for the two sets.

EXAMPLE 20 The bit strings for the sets $\{1, 2, 3, 4, 5\}$ and $\{1, 3, 5, 7, 9\}$ are $11\ 1110\ 0000$ and $10\ 1010\ 1010$, respectively. Use bit strings to find the union and intersection of these sets.

Solution: The bit string for the union of these sets is

$$11\ 1110\ 0000 \vee 10\ 1010\ 1010 = 11\ 1110\ 1010,$$

which corresponds to the set $\{1, 2, 3, 4, 5, 7, 9\}$. The bit string for the intersection of these sets is

$$11\ 1110\ 0000 \wedge 10\ 1010\ 1010 = 10\ 1010\ 0000,$$

which corresponds to the set $\{1, 3, 5\}$.

Exercises

1. Let A be the set of students who live within one mile of school and let B be the set of students who walk to classes. Describe the students in each of these sets.

- a) $A \cap B$ b) $A \cup B$
c) $A - B$ d) $B - A$

2. Suppose that A is the set of sophomores at your school and B is the set of students in discrete mathematics at your school. Express each of these sets in terms of A and B .

- a) the set of sophomores taking discrete mathematics in your school
b) the set of sophomores at your school who are not taking discrete mathematics
c) the set of students at your school who either are sophomores or are taking discrete mathematics
d) the set of students at your school who either are not sophomores or are not taking discrete mathematics

3. Let $A = \{1, 2, 3, 4, 5\}$ and $B = \{0, 3, 6\}$. Find

- a) $A \cup B$. b) $A \cap B$.
c) $A - B$. d) $B - A$.

4. Let $A = \{a, b, c, d, e\}$ and $B = \{a, b, c, d, e, f, g, h\}$. Find

- a) $A \cup B$. b) $A \cap B$.
c) $A - B$. d) $B - A$.

In Exercises 5–10 assume that A is a subset of some underlying universal set U .

5. Prove the complementation law in Table 1 by showing that $\overline{\overline{A}} = A$.

6. Prove the identity laws in Table 1 by showing that

- a) $A \cup \emptyset = A$. b) $A \cap U = A$.

7. Prove the domination laws in Table 1 by showing that

- a) $A \cup U = U$. b) $A \cap \emptyset = \emptyset$.

8. Prove the idempotent laws in Table 1 by showing that

- a) $A \cup A = A$. b) $A \cap A = A$.

9. Prove the complement laws in Table 1 by showing that

- a) $A \cup \overline{A} = U$. b) $A \cap \overline{A} = \emptyset$.

10. Show that

- a) $A - \emptyset = A$. b) $\emptyset - A = \emptyset$.

11. Let A and B be sets. Prove the commutative laws from Table 1 by showing that

- a) $A \cup B = B \cup A$.
b) $A \cap B = B \cap A$.

12. Prove the first absorption law from Table 1 by showing that if A and B are sets, then $A \cup (A \cap B) = A$.

13. Prove the second absorption law from Table 1 by showing that if A and B are sets, then $A \cap (A \cup B) = A$.

14. Find the sets A and B if $A - B = \{1, 5, 7, 8\}$, $B - A = \{2, 10\}$, and $A \cap B = \{3, 6, 9\}$.

15. Prove the second De Morgan law in Table 1 by showing that if A and B are sets, then $\overline{A \cup B} = \overline{A} \cap \overline{B}$

- a) by showing each side is a subset of the other side.

- b) using a membership table.

16. Let A and B be sets. Show that

- a) $(A \cap B) \subseteq A$. b) $A \subseteq (A \cup B)$.
c) $A - B \subseteq A$. d) $A \cap (B - A) = \emptyset$.
e) $A \cup (B - A) = A \cup B$.

17. Show that if A , B , and C are sets, then $\overline{A \cap B \cap C} = \overline{A} \cup \overline{B} \cup \overline{C}$

- a) by showing each side is a subset of the other side.
b) using a membership table.

18. Let A , B , and C be sets. Show that

- a) $(A \cup B) \subseteq (A \cup B \cup C)$.
b) $(A \cap B \cap C) \subseteq (A \cap B)$.
c) $(A - B) - C \subseteq A - C$.
d) $(A - C) \cap (C - B) = \emptyset$.
e) $(B - A) \cup (C - A) = (B \cup C) - A$.

19. Show that if A and B are sets, then

- a) $A - B = A \cap \overline{B}$.
b) $(A \cap B) \cup (A \cap \overline{B}) = A$.

20. Show that if A and B are sets with $A \subseteq B$, then

- a) $A \cup B = B$.
b) $A \cap B = A$.

21. Prove the first associative law from Table 1 by showing that if A , B , and C are sets, then $A \cup (B \cup C) = (A \cup B) \cup C$.

22. Prove the second associative law from Table 1 by showing that if A , B , and C are sets, then $A \cap (B \cap C) = (A \cap B) \cap C$.

23. Prove the first distributive law from Table 1 by showing that if A , B , and C are sets, then $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

24. Let A , B , and C be sets. Show that $(A - B) - C = (A - C) - (B - C)$.

25. Let $A = \{0, 2, 4, 6, 8, 10\}$, $B = \{0, 1, 2, 3, 4, 5, 6\}$, and $C = \{4, 5, 6, 7, 8, 9, 10\}$. Find

- a) $A \cap B \cap C$. b) $A \cup B \cup C$.
c) $(A \cup B) \cap C$. d) $(A \cap B) \cup C$.

26. Draw the Venn diagrams for each of these combinations of the sets A , B , and C .

- a) $A \cap (B \cup C)$. b) $\overline{A} \cap \overline{B} \cap \overline{C}$
c) $(A - B) \cup (A - C) \cup (B - C)$.

27. Draw the Venn diagrams for each of these combinations of the sets A , B , and C .

- a) $A \cap (\overline{B} - C)$. b) $(A \cap B) \cup (A \cap C)$
c) $(A \cap \overline{B}) \cup (A \cap \overline{C})$.

28. Draw the Venn diagrams for each of these combinations of the sets A , B , C , and D .

- a) $(A \cap B) \cup (C \cap D)$. b) $\overline{A} \cup \overline{B} \cup \overline{C} \cup \overline{D}$
c) $A - (B \cap C \cap D)$.

29. What can you say about the sets A and B if we know that

- a) $A \cup B = A$? b) $A \cap B = A$?
c) $A - B = A$? d) $A \cap B = B \cap A$?
e) $A - B = B - A$?

- 30.** Can you conclude that $A = B$ if A , B , and C are sets such that

- a) $A \cup C = B \cup C$? b) $A \cap C = B \cap C$?
c) $A \cup C = B \cup C$ and $A \cap C = B \cap C$?

- 31.** Let A and B be subsets of a universal set U . Show that $A \subseteq B$ if and only if $\overline{B} \subseteq \overline{A}$.

The **symmetric difference** of A and B , denoted by $A \oplus B$, is the set containing those elements in either A or B , but not in both A and B .

- 32.** Find the symmetric difference of $\{1, 3, 5\}$ and $\{1, 2, 3\}$.

- 33.** Find the symmetric difference of the set of computer science majors at a school and the set of mathematics majors at this school.

- 34.** Draw a Venn diagram for the symmetric difference of the sets A and B .

- 35.** Show that $A \oplus B = (A \cup B) - (A \cap B)$.

- 36.** Show that $A \oplus B = (A - B) \cup (B - A)$.

- 37.** Show that if A is a subset of a universal set U , then

- a) $A \oplus A = \emptyset$. b) $A \oplus \emptyset = A$.
c) $A \oplus U = \overline{A}$. d) $A \oplus \overline{A} = U$.

- 38.** Show that if A and B are sets, then

- a) $A \oplus B = B \oplus A$. b) $(A \oplus B) \oplus B = A$.

- 39.** What can you say about the sets A and B if $A \oplus B = A$?

- *40.** Determine whether the symmetric difference is associative; that is, if A , B , and C are sets, does it follow that $A \oplus (B \oplus C) = (A \oplus B) \oplus C$?

- *41.** Suppose that A , B , and C are sets such that $A \oplus C = B \oplus C$. Must it be the case that $A = B$?

- 42.** If A , B , C , and D are sets, does it follow that $(A \oplus B) \oplus (C \oplus D) = (A \oplus C) \oplus (B \oplus D)$?

- 43.** If A , B , C , and D are sets, does it follow that $(A \oplus B) \oplus (C \oplus D) = (A \oplus D) \oplus (B \oplus C)$?

- 44.** Show that if A and B are finite sets, then $A \cup B$ is a finite set.

- 45.** Show that if A is an infinite set, then whenever B is a set, $A \cup B$ is also an infinite set.

- *46.** Show that if A , B , and C are sets, then

$$\begin{aligned}|A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| \\&\quad - |A \cap C| - |B \cap C| + |A \cap B \cap C|.\end{aligned}$$

(This is a special case of the inclusion–exclusion principle, which will be studied in Chapter 8.)

- 47.** Let $A_i = \{1, 2, 3, \dots, i\}$ for $i = 1, 2, 3, \dots$. Find

a) $\bigcup_{i=1}^n A_i$. b) $\bigcap_{i=1}^n A_i$.

- 48.** Let $A_i = \{\dots, -2, -1, 0, 1, \dots, i\}$. Find

a) $\bigcup_{i=1}^n A_i$. b) $\bigcap_{i=1}^n A_i$.

- 49.** Let A_i be the set of all nonempty bit strings (that is, bit strings of length at least one) of length not exceeding i .

Find

a) $\bigcup_{i=1}^n A_i$. b) $\bigcap_{i=1}^n A_i$.

- 50.** Find $\bigcup_{i=1}^{\infty} A_i$ and $\bigcap_{i=1}^{\infty} A_i$ if for every positive integer i ,

- a) $A_i = \{i, i+1, i+2, \dots\}$.
b) $A_i = \{0, i\}$.
c) $A_i = (0, i)$, that is, the set of real numbers x with $0 < x < i$.
d) $A_i = (i, \infty)$, that is, the set of real numbers x with $x > i$.

- 51.** Find $\bigcup_{i=1}^{\infty} A_i$ and $\bigcap_{i=1}^{\infty} A_i$ if for every positive integer i ,

- a) $A_i = \{-i, -i+1, \dots, -1, 0, 1, \dots, i-1, i\}$.
b) $A_i = \{-i, i\}$.
c) $A_i = [-i, i]$, that is, the set of real numbers x with $-i \leq x \leq i$.
d) $A_i = [i, \infty)$, that is, the set of real numbers x with $x \geq i$.

- 52.** Suppose that the universal set is $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Express each of these sets with bit strings where the i th bit in the string is 1 if i is in the set and 0 otherwise.

- a) $\{3, 4, 5\}$
b) $\{1, 3, 6, 10\}$
c) $\{2, 3, 4, 7, 8, 9\}$

- 53.** Using the same universal set as in the last problem, find the set specified by each of these bit strings.

- a) 11 1100 1111
b) 01 0111 1000
c) 10 0000 0001

- 54.** What subsets of a finite universal set do these bit strings represent?

- a) the string with all zeros
b) the string with all ones

- 55.** What is the bit string corresponding to the difference of two sets?

- 56.** What is the bit string corresponding to the symmetric difference of two sets?

- 57.** Show how bitwise operations on bit strings can be used to find these combinations of $A = \{a, b, c, d, e\}$, $B = \{b, c, d, g, p, t, v\}$, $C = \{c, e, i, o, u, x, y, z\}$, and $D = \{d, e, h, i, n, o, t, u, x, y\}$.

- a) $A \cup B$ b) $A \cap B$
c) $(A \cup D) \cap (B \cup C)$ d) $A \cup B \cup C \cup D$

- 58.** How can the union and intersection of n sets that all are subsets of the universal set U be found using bit strings?

The **successor** of the set A is the set $A \cup \{A\}$.

- 59.** Find the successors of the following sets.

- a) $\{1, 2, 3\}$ b) \emptyset
c) $\{\emptyset\}$ d) $\{\emptyset, \{\emptyset\}\}$

- 60.** How many elements does the successor of a set with n elements have?

Sometimes the number of times that an element occurs in an unordered collection matters. **Multisets** are unordered collections of elements where an element can occur as a member more than once. The notation $\{m_1 \cdot a_1, m_2 \cdot a_2, \dots, m_r \cdot a_r\}$ denotes the multiset with element a_1 occurring m_1 times, element a_2 occurring m_2 times, and so on. The numbers m_i , $i = 1, 2, \dots, r$ are called the **multiplicities** of the elements a_i , $i = 1, 2, \dots, r$.

Let P and Q be multisets. The **union** of the multisets P and Q is the multiset where the multiplicity of an element is the maximum of its multiplicities in P and Q . The **intersection** of P and Q is the multiset where the multiplicity of an element is the minimum of its multiplicities in P and Q . The **difference** of P and Q is the multiset where the multiplicity of an element is the multiplicity of the element in P less its multiplicity in Q unless this difference is negative, in which case the multiplicity is 0. The **sum** of P and Q is the multiset where the multiplicity of an element is the sum of multiplicities in P and Q . The union, intersection, and difference of P and Q are denoted by $P \cup Q$, $P \cap Q$, and $P - Q$, respectively (where these operations should not be confused with the analogous operations for sets). The sum of P and Q is denoted by $P + Q$.

- 61.** Let A and B be the multisets $\{3 \cdot a, 2 \cdot b, 1 \cdot c\}$ and $\{2 \cdot a, 3 \cdot b, 4 \cdot d\}$, respectively. Find
- $A \cup B$.
 - $A \cap B$.
 - $A - B$.
 - $B - A$.
 - $A + B$.
- 62.** Suppose that A is the multiset that has as its elements the types of computer equipment needed by one department of a university and the multiplicities are the number of pieces of each type needed, and B is the analogous multiset for a second department of the university. For instance, A could be the multiset $\{107 \cdot \text{personal computers}, 44 \cdot \text{routers}, 6 \cdot \text{servers}\}$ and B could be the multiset $\{14 \cdot \text{personal computers}, 6 \cdot \text{routers}, 2 \cdot \text{mainframes}\}$.
- What combination of A and B represents the equipment the university should buy assuming both departments use the same equipment?

- What combination of A and B represents the equipment that will be used by both departments if both departments use the same equipment?
- What combination of A and B represents the equipment that the second department uses, but the first department does not, if both departments use the same equipment?
- What combination of A and B represents the equipment that the university should purchase if the departments do not share equipment?

Fuzzy sets are used in artificial intelligence. Each element in the universal set U has a **degree of membership**, which is a real number between 0 and 1 (including 0 and 1), in a fuzzy set S . The fuzzy set S is denoted by listing the elements with their degrees of membership (elements with 0 degree of membership are not listed). For instance, we write $\{0.6 \text{ Alice}, 0.9 \text{ Brian}, 0.4 \text{ Fred}, 0.1 \text{ Oscar}, 0.5 \text{ Rita}\}$ for the set F (of famous people) to indicate that Alice has a 0.6 degree of membership in F , Brian has a 0.9 degree of membership in F , Fred has a 0.4 degree of membership in F , Oscar has a 0.1 degree of membership in F , and Rita has a 0.5 degree of membership in F (so that Brian is the most famous and Oscar is the least famous of these people). Also suppose that R is the set of rich people with $R = \{0.4 \text{ Alice}, 0.8 \text{ Brian}, 0.2 \text{ Fred}, 0.9 \text{ Oscar}, 0.7 \text{ Rita}\}$.

- The **complement** of a fuzzy set S is the set \bar{S} , with the degree of the membership of an element in \bar{S} equal to 1 minus the degree of membership of this element in S . Find \bar{F} (the fuzzy set of people who are not famous) and \bar{R} (the fuzzy set of people who are not rich).
- The **union** of two fuzzy sets S and T is the fuzzy set $S \cup T$, where the degree of membership of an element in $S \cup T$ is the maximum of the degrees of membership of this element in S and in T . Find the fuzzy set $F \cup R$ of rich or famous people.
- The **intersection** of two fuzzy sets S and T is the fuzzy set $S \cap T$, where the degree of membership of an element in $S \cap T$ is the minimum of the degrees of membership of this element in S and in T . Find the fuzzy set $F \cap R$ of rich and famous people.

2.3 Functions

Introduction

In many instances we assign to each element of a set a particular element of a second set (which may be the same as the first). For example, suppose that each student in a discrete mathematics class is assigned a letter grade from the set $\{A, B, C, D, F\}$. And suppose that the grades are A for Adams, C for Chou, B for Goodfriend, A for Rodriguez, and F for Stevens. This assignment of grades is illustrated in Figure 1.

This assignment is an example of a function. The concept of a function is extremely important in mathematics and computer science. For example, in discrete mathematics functions are used in the definition of such discrete structures as sequences and strings. Functions are also used to represent how long it takes a computer to solve problems of a given size. Many computer programs and subroutines are designed to calculate values of functions. Recursive functions,

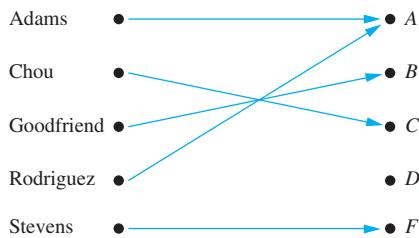


FIGURE 1 Assignment of Grades in a Discrete Mathematics Class.

which are functions defined in terms of themselves, are used throughout computer science; they will be studied in Chapter 5. This section reviews the basic concepts involving functions needed in discrete mathematics.

DEFINITION 1

Let A and B be nonempty sets. A *function* f from A to B is an assignment of exactly one element of B to each element of A . We write $f(a) = b$ if b is the unique element of B assigned by the function f to the element a of A . If f is a function from A to B , we write $f : A \rightarrow B$.

Remark: Functions are sometimes also called **mappings** or **transformations**.

Functions are specified in many different ways. Sometimes we explicitly state the assignments, as in Figure 1. Often we give a formula, such as $f(x) = x + 1$, to define a function. Other times we use a computer program to specify a function.



A function $f : A \rightarrow B$ can also be defined in terms of a relation from A to B . Recall from Section 2.1 that a relation from A to B is just a subset of $A \times B$. A relation from A to B that contains one, and only one, ordered pair (a, b) for every element $a \in A$, defines a function f from A to B . This function is defined by the assignment $f(a) = b$, where (a, b) is the unique ordered pair in the relation that has a as its first element.

DEFINITION 2

If f is a function from A to B , we say that A is the *domain* of f and B is the *codomain* of f . If $f(a) = b$, we say that b is the *image* of a and a is a *preimage* of b . The *range*, or *image*, of f is the set of all images of elements of A . Also, if f is a function from A to B , we say that f *maps* A to B .

Figure 2 represents a function f from A to B .

When we define a function we specify its domain, its codomain, and the mapping of elements of the domain to elements in the codomain. Two functions are **equal** when they have the same domain, have the same codomain, and map each element of their common domain to the same element in their common codomain. Note that if we change either the domain or the codomain

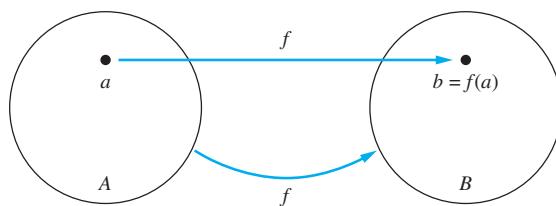


FIGURE 2 The Function f Maps A to B .

of a function, then we obtain a different function. If we change the mapping of elements, then we also obtain a different function.

Examples 1–5 provide examples of functions. In each case, we describe the domain, the codomain, the range, and the assignment of values to elements of the domain.

EXAMPLE 1 What are the domain, codomain, and range of the function that assigns grades to students described in the first paragraph of the introduction of this section?

Solution: Let G be the function that assigns a grade to a student in our discrete mathematics class. Note that $G(\text{Adams}) = A$, for instance. The domain of G is the set {Adams, Chou, Goodfriend, Rodriguez, Stevens}, and the codomain is the set {A, B, C, D, F}. The range of G is the set {A, B, C, F}, because each grade except D is assigned to some student. 

EXAMPLE 2 Let R be the relation with ordered pairs (Abdul, 22), (Brenda, 24), (Carla, 21), (Desire, 22), (Eddie, 24), and (Felicia, 22). Here each pair consists of a graduate student and this student's age. Specify a function determined by this relation.

Solution: If f is a function specified by R , then $f(\text{Abdul}) = 22$, $f(\text{Brenda}) = 24$, $f(\text{Carla}) = 21$, $f(\text{Desire}) = 22$, $f(\text{Eddie}) = 24$, and $f(\text{Felicia}) = 22$. (Here, $f(x)$ is the age of x , where x is a student.) For the domain, we take the set {Abdul, Brenda, Carla, Desire, Eddie, Felicia}. We also need to specify a codomain, which needs to contain all possible ages of students. Because it is highly likely that all students are less than 100 years old, we can take the set of positive integers less than 100 as the codomain. (Note that we could choose a different codomain, such as the set of all positive integers or the set of positive integers between 10 and 90, but that would change the function. Using this codomain will also allow us to extend the function by adding the names and ages of more students later.) The range of the function we have specified is the set of different ages of these students, which is the set {21, 22, 24}. 

EXAMPLE 3  Let f be the function that assigns the last two bits of a bit string of length 2 or greater to that string. For example, $f(11010) = 10$. Then, the domain of f is the set of all bit strings of length 2 or greater, and both the codomain and range are the set {00, 01, 10, 11}. 

EXAMPLE 4 Let $f: \mathbf{Z} \rightarrow \mathbf{Z}$ assign the square of an integer to this integer. Then, $f(x) = x^2$, where the domain of f is the set of all integers, the codomain of f is the set of all integers, and the range of f is the set of all integers that are perfect squares, namely, {0, 1, 4, 9, ...}. 

EXAMPLE 5 The domain and codomain of functions are often specified in programming languages. For instance, the Java statement

```
int floor(float real){...}
```

and the C++ function statement

```
int function (float x){...}
```

both tell us that the domain of the floor function is the set of real numbers (represented by floating point numbers) and its codomain is the set of integers. 

A function is called **real-valued** if its codomain is the set of real numbers, and it is called **integer-valued** if its codomain is the set of integers. Two real-valued functions or two integer-valued functions with the same domain can be added, as well as multiplied.

DEFINITION 3

Let f_1 and f_2 be functions from A to \mathbf{R} . Then $f_1 + f_2$ and $f_1 f_2$ are also functions from A to \mathbf{R} defined for all $x \in A$ by

$$(f_1 + f_2)(x) = f_1(x) + f_2(x), \\ (f_1 f_2)(x) = f_1(x) f_2(x).$$

Note that the functions $f_1 + f_2$ and $f_1 f_2$ have been defined by specifying their values at x in terms of the values of f_1 and f_2 at x .

EXAMPLE 6 Let f_1 and f_2 be functions from \mathbf{R} to \mathbf{R} such that $f_1(x) = x^2$ and $f_2(x) = x - x^2$. What are the functions $f_1 + f_2$ and $f_1 f_2$?

Solution: From the definition of the sum and product of functions, it follows that

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x^2 + (x - x^2) = x$$

and

$$(f_1 f_2)(x) = x^2(x - x^2) = x^3 - x^4.$$

When f is a function from A to B , the image of a subset of A can also be defined.

DEFINITION 4

Let f be a function from A to B and let S be a subset of A . The *image* of S under the function f is the subset of B that consists of the images of the elements of S . We denote the image of S by $f(S)$, so

$$f(S) = \{t \mid \exists s \in S (t = f(s))\}.$$

We also use the shorthand $\{f(s) \mid s \in S\}$ to denote this set.

Remark: The notation $f(S)$ for the image of the set S under the function f is potentially ambiguous. Here, $f(S)$ denotes a set, and not the value of the function f for the set S .

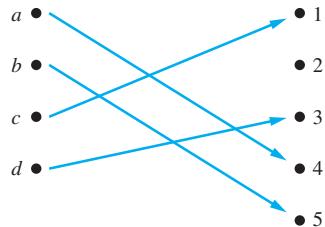
EXAMPLE 7 Let $A = \{a, b, c, d, e\}$ and $B = \{1, 2, 3, 4\}$ with $f(a) = 2$, $f(b) = 1$, $f(c) = 4$, $f(d) = 1$, and $f(e) = 1$. The image of the subset $S = \{b, c, d\}$ is the set $f(S) = \{1, 4\}$.

One-to-One and Onto Functions

Some functions never assign the same value to two different domain elements. These functions are said to be **one-to-one**.

DEFINITION 5

A function f is said to be *one-to-one*, or an *injunction*, if and only if $f(a) = f(b)$ implies that $a = b$ for all a and b in the domain of f . A function is said to be *injective* if it is one-to-one.

**FIGURE 3** A One-to-One Function.

Note that a function f is one-to-one if and only if $f(a) \neq f(b)$ whenever $a \neq b$. This way of expressing that f is one-to-one is obtained by taking the contrapositive of the implication in the definition.

Remark: We can express that f is one-to-one using quantifiers as $\forall a \forall b (f(a) = f(b) \rightarrow a = b)$ or equivalently $\forall a \forall b (a \neq b \rightarrow f(a) \neq f(b))$, where the universe of discourse is the domain of the function.



We illustrate this concept by giving examples of functions that are one-to-one and other functions that are not one-to-one.

EXAMPLE 8

Determine whether the function f from $\{a, b, c, d\}$ to $\{1, 2, 3, 4, 5\}$ with $f(a) = 4$, $f(b) = 5$, $f(c) = 1$, and $f(d) = 3$ is one-to-one.



Solution: The function f is one-to-one because f takes on different values at the four elements of its domain. This is illustrated in Figure 3.

EXAMPLE 9

Determine whether the function $f(x) = x^2$ from the set of integers to the set of integers is one-to-one.

Solution: The function $f(x) = x^2$ is not one-to-one because, for instance, $f(1) = f(-1) = 1$, but $1 \neq -1$.

Note that the function $f(x) = x^2$ with its domain restricted to \mathbf{Z}^+ is one-to-one. (Technically, when we restrict the domain of a function, we obtain a new function whose values agree with those of the original function for the elements of the restricted domain. The restricted function is not defined for elements of the original domain outside of the restricted domain.)

EXAMPLE 10

Determine whether the function $f(x) = x + 1$ from the set of real numbers to itself is one-to-one.

Solution: The function $f(x) = x + 1$ is a one-to-one function. To demonstrate this, note that $x + 1 \neq y + 1$ when $x \neq y$.

EXAMPLE 11

Suppose that each worker in a group of employees is assigned a job from a set of possible jobs, each to be done by a single worker. In this situation, the function f that assigns a job to each worker is one-to-one. To see this, note that if x and y are two different workers, then $f(x) \neq f(y)$ because the two workers x and y must be assigned different jobs.

We now give some conditions that guarantee that a function is one-to-one.

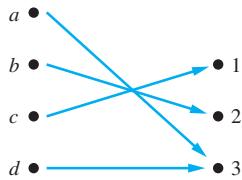


FIGURE 4 An Onto Function.

DEFINITION 6

A function f whose domain and codomain are subsets of the set of real numbers is called *increasing* if $f(x) \leq f(y)$, and *strictly increasing* if $f(x) < f(y)$, whenever $x < y$ and x and y are in the domain of f . Similarly, f is called *decreasing* if $f(x) \geq f(y)$, and *strictly decreasing* if $f(x) > f(y)$, whenever $x < y$ and x and y are in the domain of f . (The word *strictly* in this definition indicates a strict inequality.)

Remark: A function f is increasing if $\forall x \forall y (x < y \rightarrow f(x) \leq f(y))$, strictly increasing if $\forall x \forall y (x < y \rightarrow f(x) < f(y))$, decreasing if $\forall x \forall y (x < y \rightarrow f(x) \geq f(y))$, and strictly decreasing if $\forall x \forall y (x < y \rightarrow f(x) > f(y))$, where the universe of discourse is the domain of f .

From these definitions, it can be shown (see Exercises 26 and 27) that a function that is either strictly increasing or strictly decreasing must be one-to-one. However, a function that is increasing, but not strictly increasing, or decreasing, but not strictly decreasing, is not one-to-one.

For some functions the range and the codomain are equal. That is, every member of the codomain is the image of some element of the domain. Functions with this property are called **onto** functions.

DEFINITION 7

A function f from A to B is called *onto*, or a *surjection*, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$. A function f is called *surjective* if it is onto.

Remark: A function f is onto if $\forall y \exists x (f(x) = y)$, where the domain for x is the domain of the function and the domain for y is the codomain of the function.

We now give examples of onto functions and functions that are not onto.

EXAMPLE 12

Let f be the function from $\{a, b, c, d\}$ to $\{1, 2, 3\}$ defined by $f(a) = 3$, $f(b) = 2$, $f(c) = 1$, and $f(d) = 3$. Is f an onto function?



Solution: Because all three elements of the codomain are images of elements in the domain, we see that f is onto. This is illustrated in Figure 4. Note that if the codomain were $\{1, 2, 3, 4\}$, then f would not be onto.

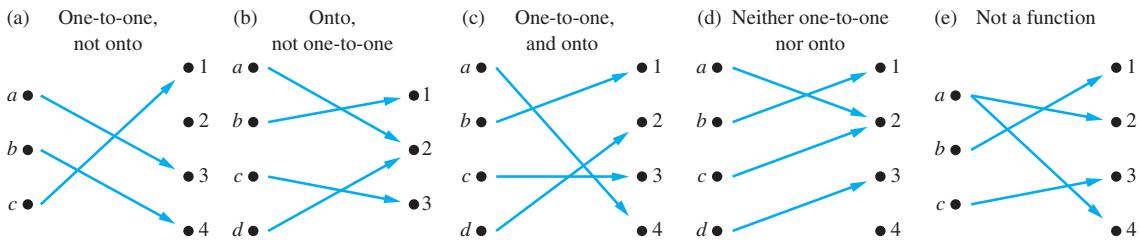
EXAMPLE 13

Is the function $f(x) = x^2$ from the set of integers to the set of integers onto?

Solution: The function f is not onto because there is no integer x with $x^2 = -1$, for instance.

EXAMPLE 14

Is the function $f(x) = x + 1$ from the set of integers to the set of integers onto?

**FIGURE 5** Examples of Different Types of Correspondences.

Solution: This function is onto, because for every integer y there is an integer x such that $f(x) = y$. To see this, note that $f(x) = y$ if and only if $x + 1 = y$, which holds if and only if $x = y - 1$.

EXAMPLE 15 Consider the function f in Example 11 that assigns jobs to workers. The function f is onto if for every job there is a worker assigned this job. The function f is not onto when there is at least one job that has no worker assigned it.

DEFINITION 8

The function f is a *one-to-one correspondence*, or a *bijection*, if it is both one-to-one and onto. We also say that such a function is *bijequivate*.

Examples 16 and 17 illustrate the concept of a bijection.

EXAMPLE 16 Let f be the function from $\{a, b, c, d\}$ to $\{1, 2, 3, 4\}$ with $f(a) = 4$, $f(b) = 2$, $f(c) = 1$, and $f(d) = 3$. Is f a bijection?

Solution: The function f is one-to-one and onto. It is one-to-one because no two values in the domain are assigned the same function value. It is onto because all four elements of the codomain are images of elements in the domain. Hence, f is a bijection.

Figure 5 displays four functions where the first is one-to-one but not onto, the second is onto but not one-to-one, the third is both one-to-one and onto, and the fourth is neither one-to-one nor onto. The fifth correspondence in Figure 5 is not a function, because it sends an element to two different elements.

Suppose that f is a function from a set A to itself. If A is finite, then f is one-to-one if and only if it is onto. (This follows from the result in Exercise 72.) This is not necessarily the case if A is infinite (as will be shown in Section 2.5).

EXAMPLE 17 Let A be a set. The *identity function* on A is the function $\iota_A : A \rightarrow A$, where

$$\iota_A(x) = x$$

for all $x \in A$. In other words, the identity function ι_A is the function that assigns each element to itself. The function ι_A is one-to-one and onto, so it is a bijection. (Note that ι is the Greek letter iota.)

For future reference, we summarize what needs to be shown to establish whether a function is one-to-one and whether it is onto. It is instructive to review Examples 8–17 in light of this summary.

Suppose that $f : A \rightarrow B$.

To show that f is injective Show that if $f(x) = f(y)$ for arbitrary $x, y \in A$ with $x \neq y$, then $x = y$.

To show that f is not injective Find particular elements $x, y \in A$ such that $x \neq y$ and $f(x) = f(y)$.

To show that f is surjective Consider an arbitrary element $y \in B$ and find an element $x \in A$ such that $f(x) = y$.

To show that f is not surjective Find a particular $y \in B$ such that $f(x) \neq y$ for all $x \in A$.

Inverse Functions and Compositions of Functions

Now consider a one-to-one correspondence f from the set A to the set B . Because f is an onto function, every element of B is the image of some element in A . Furthermore, because f is also a one-to-one function, every element of B is the image of a *unique* element of A . Consequently, we can define a new function from B to A that reverses the correspondence given by f . This leads to Definition 9.

DEFINITION 9

Let f be a one-to-one correspondence from the set A to the set B . The *inverse function* of f is the function that assigns to an element b belonging to B the unique element a in A such that $f(a) = b$. The inverse function of f is denoted by f^{-1} . Hence, $f^{-1}(b) = a$ when $f(a) = b$.

Remark: Be sure not to confuse the function f^{-1} with the function $1/f$, which is the function that assigns to each x in the domain the value $1/f(x)$. Notice that the latter makes sense only when $f(x)$ is a non-zero real number.

Figure 6 illustrates the concept of an inverse function.

If a function f is not a one-to-one correspondence, we cannot define an inverse function of f . When f is not a one-to-one correspondence, either it is not one-to-one or it is not onto. If f is not one-to-one, some element b in the codomain is the image of more than one element in the domain. If f is not onto, for some element b in the codomain, no element a in the domain exists for which $f(a) = b$. Consequently, if f is not a one-to-one correspondence, we cannot assign to each element b in the codomain a unique element a in the domain such that $f(a) = b$ (because for some b there is either more than one such a or no such a).

A one-to-one correspondence is called **invertible** because we can define an inverse of this function. A function is **not invertible** if it is not a one-to-one correspondence, because the inverse of such a function does not exist.

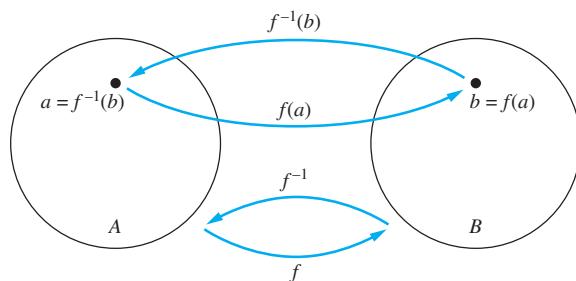


FIGURE 6 The Function f^{-1} Is the Inverse of Function f .

EXAMPLE 18 Let f be the function from $\{a, b, c\}$ to $\{1, 2, 3\}$ such that $f(a) = 2$, $f(b) = 3$, and $f(c) = 1$. Is f invertible, and if it is, what is its inverse?

Solution: The function f is invertible because it is a one-to-one correspondence. The inverse function f^{-1} reverses the correspondence given by f , so $f^{-1}(1) = c$, $f^{-1}(2) = a$, and $f^{-1}(3) = b$.

EXAMPLE 19 Let $f : \mathbf{Z} \rightarrow \mathbf{Z}$ be such that $f(x) = x + 1$. Is f invertible, and if it is, what is its inverse?

Solution: The function f has an inverse because it is a one-to-one correspondence, as follows from Examples 10 and 14. To reverse the correspondence, suppose that y is the image of x , so that $y = x + 1$. Then $x = y - 1$. This means that $y - 1$ is the unique element of \mathbf{Z} that is sent to y by f . Consequently, $f^{-1}(y) = y - 1$.

EXAMPLE 20 Let f be the function from \mathbf{R} to \mathbf{R} with $f(x) = x^2$. Is f invertible?

Solution: Because $f(-2) = f(2) = 4$, f is not one-to-one. If an inverse function were defined, it would have to assign two elements to 4. Hence, f is not invertible. (Note we can also show that f is not invertible because it is not onto.)

Sometimes we can restrict the domain or the codomain of a function, or both, to obtain an invertible function, as Example 21 illustrates.

EXAMPLE 21 Show that if we restrict the function $f(x) = x^2$ in Example 20 to a function from the set of all nonnegative real numbers to the set of all nonnegative real numbers, then f is invertible.

Solution: The function $f(x) = x^2$ from the set of nonnegative real numbers to the set of nonnegative real numbers is one-to-one. To see this, note that if $f(x) = f(y)$, then $x^2 = y^2$, so $x^2 - y^2 = (x + y)(x - y) = 0$. This means that $x + y = 0$ or $x - y = 0$, so $x = -y$ or $x = y$. Because both x and y are nonnegative, we must have $x = y$. So, this function is one-to-one. Furthermore, $f(x) = x^2$ is onto when the codomain is the set of all nonnegative real numbers, because each nonnegative real number has a square root. That is, if y is a nonnegative real number, there exists a nonnegative real number x such that $x = \sqrt{y}$, which means that $x^2 = y$. Because the function $f(x) = x^2$ from the set of nonnegative real numbers to the set of nonnegative real numbers is one-to-one and onto, it is invertible. Its inverse is given by the rule $f^{-1}(y) = \sqrt{y}$.

DEFINITION 10

Let g be a function from the set A to the set B and let f be a function from the set B to the set C . The *composition* of the functions f and g , denoted for all $a \in A$ by $f \circ g$, is defined by

$$(f \circ g)(a) = f(g(a)).$$

In other words, $f \circ g$ is the function that assigns to the element a of A the element assigned by f to $g(a)$. That is, to find $(f \circ g)(a)$ we first apply the function g to a to obtain $g(a)$ and then we apply the function f to the result $g(a)$ to obtain $(f \circ g)(a) = f(g(a))$. Note that the composition $f \circ g$ cannot be defined unless the range of g is a subset of the domain of f . In Figure 7 the composition of functions is shown.

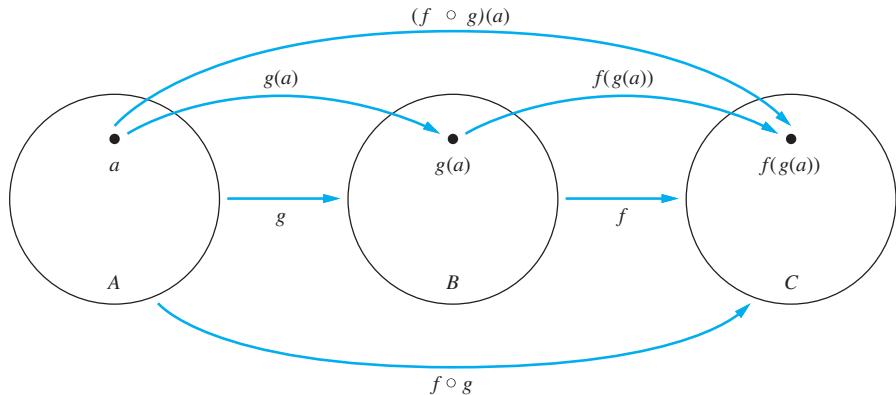


FIGURE 7 The Composition of the Functions f and g .

EXAMPLE 22 Let g be the function from the set $\{a, b, c\}$ to itself such that $g(a) = b$, $g(b) = c$, and $g(c) = a$. Let f be the function from the set $\{a, b, c\}$ to the set $\{1, 2, 3\}$ such that $f(a) = 3$, $f(b) = 2$, and $f(c) = 1$. What is the composition of f and g , and what is the composition of g and f ?

Solution: The composition $f \circ g$ is defined by $(f \circ g)(a) = f(g(a)) = f(b) = 2$, $(f \circ g)(b) = f(g(b)) = f(c) = 1$, and $(f \circ g)(c) = f(g(c)) = f(a) = 3$.

Note that $g \circ f$ is not defined, because the range of f is not a subset of the domain of g .

EXAMPLE 23 Let f and g be the functions from the set of integers to the set of integers defined by $f(x) = 2x + 3$ and $g(x) = 3x + 2$. What is the composition of f and g ? What is the composition of g and f ?

Solution: Both the compositions $f \circ g$ and $g \circ f$ are defined. Moreover,

$$(f \circ g)(x) = f(g(x)) = f(3x + 2) = 2(3x + 2) + 3 = 6x + 7$$

and

$$(g \circ f)(x) = g(f(x)) = g(2x + 3) = 3(2x + 3) + 2 = 6x + 11. \quad \blacktriangleleft$$

Remark: Note that even though $f \circ g$ and $g \circ f$ are defined for the functions f and g in Example 23, $f \circ g$ and $g \circ f$ are not equal. In other words, the commutative law does not hold for the composition of functions.

When the composition of a function and its inverse is formed, in either order, an identity function is obtained. To see this, suppose that f is a one-to-one correspondence from the set A to the set B . Then the inverse function f^{-1} exists and is a one-to-one correspondence from B to A . The inverse function reverses the correspondence of the original function, so $f^{-1}(b) = a$ when $f(a) = b$, and $f(a) = b$ when $f^{-1}(b) = a$. Hence,

$$(f^{-1} \circ f)(a) = f^{-1}(f(a)) = f^{-1}(b) = a,$$

and

$$(f \circ f^{-1})(b) = f(f^{-1}(b)) = f(a) = b.$$

Consequently $f^{-1} \circ f = \iota_A$ and $f \circ f^{-1} = \iota_B$, where ι_A and ι_B are the identity functions on the sets A and B , respectively. That is, $(f^{-1})^{-1} = f$.

The Graphs of Functions

We can associate a set of pairs in $A \times B$ to each function from A to B . This set of pairs is called the **graph** of the function and is often displayed pictorially to aid in understanding the behavior of the function.

DEFINITION 11

Let f be a function from the set A to the set B . The *graph* of the function f is the set of ordered pairs $\{(a, b) \mid a \in A \text{ and } f(a) = b\}$.

From the definition, the graph of a function f from A to B is the subset of $A \times B$ containing the ordered pairs with the second entry equal to the element of B assigned by f to the first entry. Also, note that the graph of a function f from A to B is the same as the relation from A to B determined by the function f , as described on page 139.

EXAMPLE 24 Display the graph of the function $f(n) = 2n + 1$ from the set of integers to the set of integers.

Solution: The graph of f is the set of ordered pairs of the form $(n, 2n + 1)$, where n is an integer. This graph is displayed in Figure 8. 

EXAMPLE 25 Display the graph of the function $f(x) = x^2$ from the set of integers to the set of integers.

Solution: The graph of f is the set of ordered pairs of the form $(x, f(x)) = (x, x^2)$, where x is an integer. This graph is displayed in Figure 9. 

Some Important Functions

Next, we introduce two important functions in discrete mathematics, namely, the floor and ceiling functions. Let x be a real number. The floor function rounds x down to the closest integer less than or equal to x , and the ceiling function rounds x up to the closest integer greater than or equal to x . These functions are often used when objects are counted. They play an important role in the analysis of the number of steps used by procedures to solve problems of a particular size.

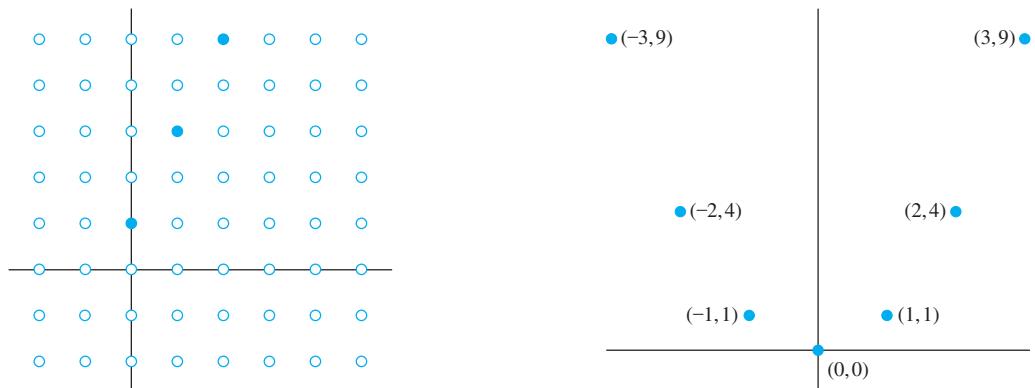


FIGURE 8 The Graph of $f(n) = 2n + 1$ from \mathbb{Z} to \mathbb{Z} .

FIGURE 9 The Graph of $f(x) = x^2$ from \mathbb{Z} to \mathbb{Z} .

DEFINITION 12

The *floor function* assigns to the real number x the largest integer that is less than or equal to x . The value of the floor function at x is denoted by $\lfloor x \rfloor$. The *ceiling function* assigns to the real number x the smallest integer that is greater than or equal to x . The value of the ceiling function at x is denoted by $\lceil x \rceil$.

Remark: The floor function is often also called the *greatest integer function*. It is often denoted by $[x]$.

EXAMPLE 26 These are some values of the floor and ceiling functions:

$$\lfloor \frac{1}{2} \rfloor = 0, \lceil \frac{1}{2} \rceil = 1, \lfloor -\frac{1}{2} \rfloor = -1, \lceil -\frac{1}{2} \rceil = 0, \lfloor 3.1 \rfloor = 3, \lceil 3.1 \rceil = 4, \lfloor 7 \rfloor = 7, \lceil 7 \rceil = 7.$$



We display the graphs of the floor and ceiling functions in Figure 10. In Figure 10(a) we display the graph of the floor function $\lfloor x \rfloor$. Note that this function has the same value throughout the interval $[n, n + 1)$, namely n , and then it jumps up to $n + 1$ when $x = n + 1$. In Figure 10(b) we display the graph of the ceiling function $\lceil x \rceil$. Note that this function has the same value throughout the interval $(n, n + 1]$, namely $n + 1$, and then jumps to $n + 2$ when x is a little larger than $n + 1$.

The floor and ceiling functions are useful in a wide variety of applications, including those involving data storage and data transmission. Consider Examples 27 and 28, typical of basic calculations done when database and data communications problems are studied.

EXAMPLE 27

Data stored on a computer disk or transmitted over a data network are usually represented as a string of bytes. Each byte is made up of 8 bits. How many bytes are required to encode 100 bits of data?

Solution: To determine the number of bytes needed, we determine the smallest integer that is at least as large as the quotient when 100 is divided by 8, the number of bits in a byte. Consequently, $\lceil 100/8 \rceil = \lceil 12.5 \rceil = 13$ bytes are required.

EXAMPLE 28

In asynchronous transfer mode (ATM) (a communications protocol used on backbone networks), data are organized into cells of 53 bytes. How many ATM cells can be transmitted in 1 minute over a connection that transmits data at the rate of 500 kilobits per second?

Solution: In 1 minute, this connection can transmit $500,000 \cdot 60 = 30,000,000$ bits. Each ATM cell is 53 bytes long, which means that it is $53 \cdot 8 = 424$ bits long. To determine the number

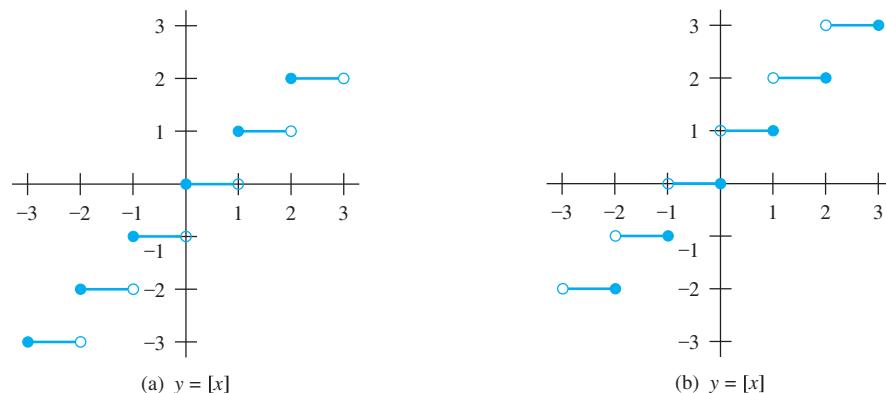


FIGURE 10 Graphs of the (a) Floor and (b) Ceiling Functions.

TABLE 1 Useful Properties of the Floor and Ceiling Functions.

(n is an integer, x is a real number)

- | |
|---|
| (1a) $\lfloor x \rfloor = n$ if and only if $n \leq x < n + 1$ |
| (1b) $\lceil x \rceil = n$ if and only if $n - 1 < x \leq n$ |
| (1c) $\lfloor x \rfloor = n$ if and only if $x - 1 < n \leq x$ |
| (1d) $\lceil x \rceil = n$ if and only if $x \leq n < x + 1$ |
| (2) $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$ |
| (3a) $\lfloor -x \rfloor = -\lceil x \rceil$ |
| (3b) $\lceil -x \rceil = -\lfloor x \rfloor$ |
| (4a) $\lfloor x + n \rfloor = \lfloor x \rfloor + n$ |
| (4b) $\lceil x + n \rceil = \lceil x \rceil + n$ |

of cells that can be transmitted in 1 minute, we determine the largest integer not exceeding the quotient when 30,000,000 is divided by 424. Consequently, $\lfloor 30,000,000/424 \rfloor = 70,754$ ATM cells can be transmitted in 1 minute over a 500 kilobit per second connection. 

Table 1, with x denoting a real number, displays some simple but important properties of the floor and ceiling functions. Because these functions appear so frequently in discrete mathematics, it is useful to look over these identities. Each property in this table can be established using the definitions of the floor and ceiling functions. Properties (1a), (1b), (1c), and (1d) follow directly from these definitions. For example, (1a) states that $\lfloor x \rfloor = n$ if and only if the integer n is less than or equal to x and $n + 1$ is larger than x . This is precisely what it means for n to be the greatest integer not exceeding x , which is the definition of $\lfloor x \rfloor = n$. Properties (1b), (1c), and (1d) can be established similarly. We will prove property (4a) using a direct proof.

Proof: Suppose that $\lfloor x \rfloor = m$, where m is a positive integer. By property (1a), it follows that $m \leq x < m + 1$. Adding n to all three quantities in this chain of two inequalities shows that $m + n \leq x + n < m + n + 1$. Using property (1a) again, we see that $\lfloor x + n \rfloor = m + n = \lfloor x \rfloor + n$. This completes the proof. Proofs of the other properties are left as exercises. 

The floor and ceiling functions enjoy many other useful properties besides those displayed in Table 1. There are also many statements about these functions that may appear to be correct, but actually are not. We will consider statements about the floor and ceiling functions in Examples 29 and 30.

A useful approach for considering statements about the floor function is to let $x = n + \epsilon$, where $n = \lfloor x \rfloor$ is an integer, and ϵ , the fractional part of x , satisfies the inequality $0 \leq \epsilon < 1$. Similarly, when considering statements about the ceiling function, it is useful to write $x = n - \epsilon$, where $n = \lceil x \rceil$ is an integer and $0 \leq \epsilon < 1$.

EXAMPLE 29 Prove that if x is a real number, then $\lfloor 2x \rfloor = \lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor$.



Solution: To prove this statement we let $x = n + \epsilon$, where n is an integer and $0 \leq \epsilon < 1$. There are two cases to consider, depending on whether ϵ is less than, or greater than or equal to $\frac{1}{2}$. (The reason we choose these two cases will be made clear in the proof.)

We first consider the case when $0 \leq \epsilon < \frac{1}{2}$. In this case, $2x = 2n + 2\epsilon$ and $\lfloor 2x \rfloor = 2n$ because $0 \leq 2\epsilon < 1$. Similarly, $x + \frac{1}{2} = n + (\frac{1}{2} + \epsilon)$, so $\lfloor x + \frac{1}{2} \rfloor = n$, because $0 < \frac{1}{2} + \epsilon < 1$. Consequently, $\lfloor 2x \rfloor = 2n$ and $\lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor = n + n = 2n$.

Next, we consider the case when $\frac{1}{2} \leq \epsilon < 1$. In this case, $2x = 2n + 2\epsilon = (2n + 1) + (2\epsilon - 1)$. Because $0 \leq 2\epsilon - 1 < 1$, it follows that $\lfloor 2x \rfloor = 2n + 1$. Because $\lfloor x + \frac{1}{2} \rfloor = \lfloor n + (\frac{1}{2} + \epsilon) \rfloor = \lfloor n + 1 + (\epsilon - \frac{1}{2}) \rfloor$ and $0 \leq \epsilon - \frac{1}{2} < 1$, it follows that $\lfloor x + \frac{1}{2} \rfloor = n + 1$. Consequently, $\lfloor 2x \rfloor = 2n + 1$ and $\lfloor x \rfloor + \lfloor x + \frac{1}{2} \rfloor = n + (n + 1) = 2n + 1$. This concludes the proof. 

EXAMPLE 30 Prove or disprove that $\lceil x + y \rceil = \lceil x \rceil + \lceil y \rceil$ for all real numbers x and y .

Solution: Although this statement may appear reasonable, it is false. A counterexample is supplied by $x = \frac{1}{2}$ and $y = \frac{1}{2}$. With these values we find that $\lceil x + y \rceil = \lceil \frac{1}{2} + \frac{1}{2} \rceil = \lceil 1 \rceil = 1$, but $\lceil x \rceil + \lceil y \rceil = \lceil \frac{1}{2} \rceil + \lceil \frac{1}{2} \rceil = 1 + 1 = 2$. 

There are certain types of functions that will be used throughout the text. These include polynomial, logarithmic, and exponential functions. A brief review of the properties of these functions needed in this text is given in Appendix 2. In this book the notation $\log x$ will be used to denote the logarithm to the base 2 of x , because 2 is the base that we will usually use for logarithms. We will denote logarithms to the base b , where b is any real number greater than 1, by $\log_b x$, and the natural logarithm by $\ln x$.

Another function we will use throughout this text is the **factorial function** $f : \mathbb{N} \rightarrow \mathbb{Z}^+$, denoted by $f(n) = n!$. The value of $f(n) = n!$ is the product of the first n positive integers, so $f(n) = 1 \cdot 2 \cdots (n-1) \cdot n$ [and $f(0) = 0! = 1$].

EXAMPLE 31 We have $f(1) = 1! = 1$, $f(2) = 2! = 1 \cdot 2 = 2$, $f(6) = 6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$, and $f(20) = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13 \cdot 14 \cdot 15 \cdot 16 \cdot 17 \cdot 18 \cdot 19 \cdot 20 = 2,432,902,008,176,640,000$. 

Example 31 illustrates that the factorial function grows extremely rapidly as n grows. The rapid growth of the factorial function is made clearer by Stirling's formula, a result from higher mathematics that tell us that $n! \sim \sqrt{2\pi n}(n/e)^n$. Here, we have used the notation $f(n) \sim g(n)$, which means that the ratio $f(n)/g(n)$ approaches 1 as n grows without bound (that is, $\lim_{n \rightarrow \infty} f(n)/g(n) = 1$). The symbol \sim is read "is asymptotic to." Stirling's formula is named after James Stirling, a Scottish mathematician of the eighteenth century.



JAMES STIRLING (1692–1770) James Stirling was born near the town of Stirling, Scotland. His family strongly supported the Jacobite cause of the Stuarts as an alternative to the British crown. The first information known about James is that he entered Balliol College, Oxford, on a scholarship in 1711. However, he later lost his scholarship when he refused to pledge his allegiance to the British crown. The first Jacobean rebellion took place in 1715, and Stirling was accused of communicating with rebels. He was charged with cursing King George, but he was acquitted of these charges. Even though he could not graduate from Oxford because of his politics, he remained there for several years. Stirling published his first work, which extended Newton's work on plane curves, in 1717. He traveled to Venice, where a chair of mathematics had been promised to him, an appointment that unfortunately fell through. Nevertheless, Stirling stayed in Venice, continuing his mathematical work. He attended the University of Padua in 1721, and in 1722 he returned to Glasgow. Stirling apparently fled Italy after learning the secrets of the Italian glass industry, avoiding the efforts of Italian glass makers to assassinate him to protect their secrets.

In late 1724 Stirling moved to London, staying there 10 years teaching mathematics and actively engaging in research. In 1730 he published *Methodus Differentialis*, his most important work, presenting results on infinite series, summations, interpolation, and quadrature. It is in this book that his asymptotic formula for $n!$ appears. Stirling also worked on gravitation and the shape of the earth; he stated, but did not prove, that the earth is an oblate spheroid. Stirling returned to Scotland in 1735, when he was appointed manager of a Scottish mining company. He was very successful in this role and even published a paper on the ventilation of mine shafts. He continued his mathematical research, but at a reduced pace, during his years in the mining industry. Stirling is also noted for surveying the River Clyde with the goal of creating a series of locks to make it navigable. In 1752 the citizens of Glasgow presented him with a silver teakettle as a reward for this work.

Partial Functions

A program designed to evaluate a function may not produce the correct value of the function for all elements in the domain of this function. For example, a program may not produce a correct value because evaluating the function may lead to an infinite loop or an overflow. Similarly, in abstract mathematics, we often want to discuss functions that are defined only for a subset of the real numbers, such as $1/x$, \sqrt{x} , and $\arcsin(x)$. Also, we may want to use such notions as the “youngest child” function, which is undefined for a couple having no children, or the “time of sunrise,” which is undefined for some days above the Arctic Circle. To study such situations, we use the concept of a partial function.

DEFINITION 13

A *partial function* f from a set A to a set B is an assignment to each element a in a subset of A , called the *domain of definition* of f , of a unique element b in B . The sets A and B are called the *domain* and *codomain* of f , respectively. We say that f is *undefined* for elements in A that are not in the domain of definition of f . When the domain of definition of f equals A , we say that f is a *total function*.

Remark: We write $f : A \rightarrow B$ to denote that f is a partial function from A to B . Note that this is the same notation as is used for functions. The context in which the notation is used determines whether f is a partial function or a total function.

EXAMPLE 32 The function $f : \mathbf{Z} \rightarrow \mathbf{R}$ where $f(n) = \sqrt{n}$ is a partial function from \mathbf{Z} to \mathbf{R} where the domain of definition is the set of nonnegative integers. Note that f is undefined for negative integers. ◀

Exercises

1. Why is f not a function from \mathbf{R} to \mathbf{R} if
 - $f(x) = 1/x$?
 - $f(x) = \sqrt{x}$?
 - $f(x) = \pm\sqrt{(x^2 + 1)}$?
2. Determine whether f is a function from \mathbf{Z} to \mathbf{R} if
 - $f(n) = \pm n$.
 - $f(n) = \sqrt{n^2 + 1}$.
 - $f(n) = 1/(n^2 - 4)$.
3. Determine whether f is a function from the set of all bit strings to the set of integers if
 - $f(S)$ is the position of a 0 bit in S .
 - $f(S)$ is the number of 1 bits in S .
 - $f(S)$ is the smallest integer i such that the i th bit of S is 1 and $f(S) = 0$ when S is the empty string, the string with no bits.
4. Find the domain and range of these functions. Note that in each case, to find the domain, determine the set of elements assigned values by the function.
 - the function that assigns to each nonnegative integer its last digit
 - the function that assigns the next largest integer to a positive integer
 - the function that assigns to a bit string the number of one bits in the string
 - the function that assigns to a bit string the number of bits in the string
5. Find the domain and range of these functions. Note that in each case, to find the domain, determine the set of elements assigned values by the function.
 - the function that assigns to each bit string the number of ones in the string minus the number of zeros in the string
 - the function that assigns to each bit string twice the number of zeros in that string
 - the function that assigns the number of bits left over when a bit string is split into bytes (which are blocks of 8 bits)
 - the function that assigns to each positive integer the largest perfect square not exceeding this integer
6. Find the domain and range of these functions.
 - the function that assigns to each pair of positive integers the first integer of the pair
 - the function that assigns to each positive integer its largest decimal digit
 - the function that assigns to a bit string the number of ones minus the number of zeros in the string
 - the function that assigns to each positive integer the largest integer not exceeding the square root of the integer
 - the function that assigns to a bit string the longest string of ones in the string

7. Find the domain and range of these functions.
- the function that assigns to each pair of positive integers the maximum of these two integers
 - the function that assigns to each positive integer the number of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 that do not appear as decimal digits of the integer
 - the function that assigns to a bit string the number of times the block 11 appears
 - the function that assigns to a bit string the numerical position of the first 1 in the string and that assigns the value 0 to a bit string consisting of all 0s
8. Find these values.
- $\lfloor 1.1 \rfloor$
 - $\lceil 1.1 \rceil$
 - $\lfloor -0.1 \rfloor$
 - $\lceil -0.1 \rceil$
 - $\lceil 2.99 \rceil$
 - $\lceil -2.99 \rceil$
 - $\lfloor \frac{1}{2} + \lceil \frac{1}{2} \rceil \rfloor$
 - $\lceil \lfloor \frac{1}{2} \rfloor + \lceil \frac{1}{2} \rceil + \frac{1}{2} \rceil$
9. Find these values.
- $\lceil \frac{3}{4} \rceil$
 - $\lfloor \frac{7}{8} \rfloor$
 - $\lceil -\frac{3}{4} \rceil$
 - $\lfloor -\frac{7}{8} \rfloor$
 - $\lceil 3 \rceil$
 - $\lfloor -1 \rfloor$
 - $\lfloor \frac{1}{2} + \lceil \frac{3}{2} \rceil \rfloor$
 - $\lfloor \frac{1}{2} \cdot \lceil \frac{5}{2} \rceil \rfloor$
10. Determine whether each of these functions from $\{a, b, c, d\}$ to itself is one-to-one.
- $f(a) = b, f(b) = a, f(c) = c, f(d) = d$
 - $f(a) = b, f(b) = b, f(c) = d, f(d) = c$
 - $f(a) = d, f(b) = b, f(c) = c, f(d) = d$
11. Which functions in Exercise 10 are onto?
12. Determine whether each of these functions from \mathbf{Z} to \mathbf{Z} is one-to-one.
- $f(n) = n - 1$
 - $f(n) = n^2 + 1$
 - $f(n) = n^3$
 - $f(n) = \lceil n/2 \rceil$
13. Which functions in Exercise 12 are onto?
14. Determine whether $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$ is onto if
- $f(m, n) = 2m - n$.
 - $f(m, n) = m^2 - n^2$.
 - $f(m, n) = m + n + 1$.
 - $f(m, n) = |m| - |n|$.
 - $f(m, n) = m^2 - 4$.
15. Determine whether the function $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$ is onto if
- $f(m, n) = m + n$.
 - $f(m, n) = m^2 + n^2$.
 - $f(m, n) = m$.
 - $f(m, n) = |n|$.
 - $f(m, n) = m - n$.
16. Consider these functions from the set of students in a discrete mathematics class. Under what conditions is the function one-to-one if it assigns to a student his or her
- mobile phone number.
 - student identification number.
 - final grade in the class.
 - home town.
17. Consider these functions from the set of teachers in a school. Under what conditions is the function one-to-one if it assigns to a teacher his or her
- office.
 - assigned bus to chaperone in a group of buses taking students on a field trip.
 - salary.
 - social security number.
18. Specify a codomain for each of the functions in Exercise 16. Under what conditions is each of these functions with the codomain you specified onto?
19. Specify a codomain for each of the functions in Exercise 17. Under what conditions is each of the functions with the codomain you specified onto?
20. Give an example of a function from \mathbf{N} to \mathbf{N} that is
- one-to-one but not onto.
 - onto but not one-to-one.
 - both onto and one-to-one (but different from the identity function).
 - neither one-to-one nor onto.
21. Give an explicit formula for a function from the set of integers to the set of positive integers that is
- one-to-one, but not onto.
 - onto, but not one-to-one.
 - one-to-one and onto.
 - neither one-to-one nor onto.
22. Determine whether each of these functions is a bijection from \mathbf{R} to \mathbf{R} .
- $f(x) = -3x + 4$
 - $f(x) = -3x^2 + 7$
 - $f(x) = (x+1)/(x+2)$
 - $f(x) = x^5 + 1$
23. Determine whether each of these functions is a bijection from \mathbf{R} to \mathbf{R} .
- $f(x) = 2x + 1$
 - $f(x) = x^2 + 1$
 - $f(x) = x^3$
 - $f(x) = (x^2 + 1)/(x^2 + 2)$
24. Let $f: \mathbf{R} \rightarrow \mathbf{R}$ and let $f(x) > 0$ for all $x \in \mathbf{R}$. Show that $f(x)$ is strictly increasing if and only if the function $g(x) = 1/f(x)$ is strictly decreasing.
25. Let $f: \mathbf{R} \rightarrow \mathbf{R}$ and let $f(x) > 0$ for all $x \in \mathbf{R}$. Show that $f(x)$ is strictly decreasing if and only if the function $g(x) = 1/f(x)$ is strictly increasing.
26. a) Prove that a strictly increasing function from \mathbf{R} to itself is one-to-one.
b) Give an example of an increasing function from \mathbf{R} to itself that is not one-to-one.
27. a) Prove that a strictly decreasing function from \mathbf{R} to itself is one-to-one.
b) Give an example of a decreasing function from \mathbf{R} to itself that is not one-to-one.
28. Show that the function $f(x) = e^x$ from the set of real numbers to the set of real numbers is not invertible, but if the codomain is restricted to the set of positive real numbers, the resulting function is invertible.

- 29.** Show that the function $f(x) = |x|$ from the set of real numbers to the set of nonnegative real numbers is not invertible, but if the domain is restricted to the set of non-negative real numbers, the resulting function is invertible.
- 30.** Let $S = \{-1, 0, 2, 4, 7\}$. Find $f(S)$ if
- $f(x) = 1$.
 - $f(x) = 2x + 1$.
 - $f(x) = \lceil x/5 \rceil$.
 - $f(x) = \lfloor (x^2 + 1)/3 \rfloor$.
- 31.** Let $f(x) = \lfloor x^2/3 \rfloor$. Find $f(S)$ if
- $S = \{-2, -1, 0, 1, 2, 3\}$.
 - $S = \{0, 1, 2, 3, 4, 5\}$.
 - $S = \{1, 5, 7, 11\}$.
 - $S = \{2, 6, 10, 14\}$.
- 32.** Let $f(x) = 2x$ where the domain is the set of real numbers. What is
- $f(\mathbb{Z})$?
 - $f(\mathbb{N})$?
 - $f(\mathbb{R})$?
- 33.** Suppose that g is a function from A to B and f is a function from B to C .
- Show that if both f and g are one-to-one functions, then $f \circ g$ is also one-to-one.
 - Show that if both f and g are onto functions, then $f \circ g$ is also onto.
- *34.** If f and $f \circ g$ are one-to-one, does it follow that g is one-to-one? Justify your answer.
- *35.** If f and $f \circ g$ are onto, does it follow that g is onto? Justify your answer.
- 36.** Find $f \circ g$ and $g \circ f$, where $f(x) = x^2 + 1$ and $g(x) = x + 2$, are functions from \mathbf{R} to \mathbf{R} .
- 37.** Find $f + g$ and fg for the functions f and g given in Exercise 36.
- 38.** Let $f(x) = ax + b$ and $g(x) = cx + d$, where a, b, c , and d are constants. Determine necessary and sufficient conditions on the constants a, b, c , and d so that $f \circ g = g \circ f$.
- 39.** Show that the function $f(x) = ax + b$ from \mathbf{R} to \mathbf{R} is invertible, where a and b are constants, with $a \neq 0$, and find the inverse of f .
- 40.** Let f be a function from the set A to the set B . Let S and T be subsets of A . Show that
- $f(S \cup T) = f(S) \cup f(T)$.
 - $f(S \cap T) \subseteq f(S) \cap f(T)$.
- 41.** a) Give an example to show that the inclusion in part (b) in Exercise 40 may be proper.
b) Show that if f is one-to-one, the inclusion in part (b) in Exercise 40 is an equality.
- Let f be a function from the set A to the set B . Let S be a subset of B . We define the **inverse image** of S to be the subset of A whose elements are precisely all pre-images of all elements of S . We denote the inverse image of S by $f^{-1}(S)$, so $f^{-1}(S) = \{a \in A \mid f(a) \in S\}$. (*Beware:* The notation f^{-1} is used in two different ways. Do not confuse the notation introduced here with the notation $f^{-1}(y)$ for the value at y of the inverse of the invertible function f . Notice also that $f^{-1}(S)$, the inverse image of the set S , makes sense for all functions f , not just invertible functions.)
- 42.** Let f be the function from \mathbf{R} to \mathbf{R} defined by
 $f(x) = x^2$. Find
- $f^{-1}(\{1\})$.
 - $f^{-1}(\{x \mid 0 < x < 1\})$.
 - $f^{-1}(\{x \mid x > 4\})$.
- 43.** Let $g(x) = \lfloor x \rfloor$. Find
- $g^{-1}(\{0\})$.
 - $g^{-1}(\{-1, 0, 1\})$.
 - $g^{-1}(\{x \mid 0 < x < 1\})$.
- 44.** Let f be a function from A to B . Let S and T be subsets of B . Show that
- $f^{-1}(S \cup T) = f^{-1}(S) \cup f^{-1}(T)$.
 - $f^{-1}(S \cap T) = f^{-1}(S) \cap f^{-1}(T)$.
- 45.** Let f be a function from A to B . Let S be a subset of B . Show that $f^{-1}(\overline{S}) = \overline{f^{-1}(S)}$.
- 46.** Show that $\lfloor x + \frac{1}{2} \rfloor$ is the closest integer to the number x , except when x is midway between two integers, when it is the larger of these two integers.
- 47.** Show that $\lceil x - \frac{1}{2} \rceil$ is the closest integer to the number x , except when x is midway between two integers, when it is the smaller of these two integers.
- 48.** Show that if x is a real number, then $\lceil x \rceil - \lfloor x \rfloor = 1$ if x is not an integer and $\lceil x \rceil - \lfloor x \rfloor = 0$ if x is an integer.
- 49.** Show that if x is a real number, then $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$.
- 50.** Show that if x is a real number and m is an integer, then $\lceil x + m \rceil = \lceil x \rceil + m$.
- 51.** Show that if x is a real number and n is an integer, then
- $x < n$ if and only if $\lfloor x \rfloor < n$.
 - $n < x$ if and only if $n < \lceil x \rceil$.
- 52.** Show that if x is a real number and n is an integer, then
- $x \leq n$ if and only if $\lceil x \rceil \leq n$.
 - $n \leq x$ if and only if $n \leq \lfloor x \rfloor$.
- 53.** Prove that if n is an integer, then $\lfloor n/2 \rfloor = n/2$ if n is even and $(n-1)/2$ if n is odd.
- 54.** Prove that if x is a real number, then $\lfloor -x \rfloor = -\lceil x \rceil$ and $\lceil -x \rceil = -\lfloor x \rfloor$.
- 55.** The function INT is found on some calculators, where $\text{INT}(x) = \lfloor x \rfloor$ when x is a nonnegative real number and $\text{INT}(x) = \lceil x \rceil$ when x is a negative real number. Show that this INT function satisfies the identity $\text{INT}(-x) = -\text{INT}(x)$.
- 56.** Let a and b be real numbers with $a < b$. Use the floor and/or ceiling functions to express the number of integers n that satisfy the inequality $a \leq n \leq b$.
- 57.** Let a and b be real numbers with $a < b$. Use the floor and/or ceiling functions to express the number of integers n that satisfy the inequality $a < n < b$.
- 58.** How many bytes are required to encode n bits of data where n equals
- 4?
 - 10?
 - 500?
 - 3000?

- 59.** How many bytes are required to encode n bits of data where n equals
 a) 7? b) 17? c) 1001? d) 28,800?
- 60.** How many ATM cells (described in Example 28) can be transmitted in 10 seconds over a link operating at the following rates?
 a) 128 kilobits per second (1 kilobit = 1000 bits)
 b) 300 kilobits per second
 c) 1 megabit per second (1 megabit = 1,000,000 bits)
- 61.** Data are transmitted over a particular Ethernet network in blocks of 1500 octets (blocks of 8 bits). How many blocks are required to transmit the following amounts of data over this Ethernet network? (Note that a byte is a synonym for an octet, a kilobyte is 1000 bytes, and a megabyte is 1,000,000 bytes.)
 a) 150 kilobytes of data
 b) 384 kilobytes of data
 c) 1.544 megabytes of data
 d) 45.3 megabytes of data
- 62.** Draw the graph of the function $f(n) = 1 - n^2$ from \mathbf{Z} to \mathbf{Z} .
- 63.** Draw the graph of the function $f(x) = \lfloor 2x \rfloor$ from \mathbf{R} to \mathbf{R} .
- 64.** Draw the graph of the function $f(x) = \lfloor x/2 \rfloor$ from \mathbf{R} to \mathbf{R} .
- 65.** Draw the graph of the function $f(x) = \lfloor x \rfloor + \lfloor x/2 \rfloor$ from \mathbf{R} to \mathbf{R} .
- 66.** Draw the graph of the function $f(x) = \lceil x \rceil + \lfloor x/2 \rfloor$ from \mathbf{R} to \mathbf{R} .
- 67.** Draw graphs of each of these functions.
 a) $f(x) = \lfloor x + \frac{1}{2} \rfloor$ b) $f(x) = \lfloor 2x + 1 \rfloor$
 c) $f(x) = \lceil x/3 \rceil$ d) $f(x) = \lceil 1/x \rceil$
 e) $f(x) = \lceil x - 2 \rceil + \lfloor x + 2 \rfloor$
 f) $f(x) = \lfloor 2x \rfloor \lceil x/2 \rceil$ g) $f(x) = \lceil \lfloor x - \frac{1}{2} \rfloor + \frac{1}{2} \rceil$
- 68.** Draw graphs of each of these functions.
 a) $f(x) = \lceil 3x - 2 \rceil$ b) $f(x) = \lceil 0.2x \rceil$
 c) $f(x) = \lfloor -1/x \rfloor$ d) $f(x) = \lfloor x^2 \rfloor$
 e) $f(x) = \lceil x/2 \rceil \lfloor x/2 \rfloor$ f) $f(x) = \lfloor x/2 \rfloor + \lceil x/2 \rceil$
 g) $f(x) = \lfloor 2 \lceil x/2 \rceil + \frac{1}{2} \rfloor$
- 69.** Find the inverse function of $f(x) = x^3 + 1$.
- 70.** Suppose that f is an invertible function from Y to Z and g is an invertible function from X to Y . Show that the inverse of the composition $f \circ g$ is given by $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.
- 71.** Let S be a subset of a universal set U . The **characteristic function** f_S of S is the function from U to the set $\{0, 1\}$ such that $f_S(x) = 1$ if x belongs to S and $f_S(x) = 0$ if x does not belong to S . Let A and B be sets. Show that for all $x \in U$,
 a) $f_{A \cap B}(x) = f_A(x) \cdot f_B(x)$
 b) $f_{A \cup B}(x) = f_A(x) + f_B(x) - f_A(x) \cdot f_B(x)$
 c) $f_{\bar{A}}(x) = 1 - f_A(x)$
 d) $f_{A \oplus B}(x) = f_A(x) + f_B(x) - 2f_A(x)f_B(x)$
- 72.** Suppose that f is a function from A to B , where A and B are finite sets with $|A| = |B|$. Show that f is one-to-one if and only if it is onto.
- 73.** Prove or disprove each of these statements about the floor and ceiling functions.
 a) $\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$ for all real numbers x .
 b) $\lfloor 2x \rfloor = 2\lfloor x \rfloor$ whenever x is a real number.
 c) $\lceil x \rceil + \lceil y \rceil - \lceil x + y \rceil = 0$ or 1 whenever x and y are real numbers.
 d) $\lceil xy \rceil = \lceil x \rceil \lceil y \rceil$ for all real numbers x and y .
 e) $\lceil \frac{x}{2} \rceil = \left\lfloor \frac{x+1}{2} \right\rfloor$ for all real numbers x .
- 74.** Prove or disprove each of these statements about the floor and ceiling functions.
 a) $\lfloor \lceil x \rceil \rfloor = \lceil x \rceil$ for all real numbers x .
 b) $\lfloor x + y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$ for all real numbers x and y .
 c) $\lceil \lceil x/2 \rceil / 2 \rceil = \lceil x/4 \rceil$ for all real numbers x .
 d) $\lfloor \sqrt{\lceil x \rceil} \rfloor = \lfloor \sqrt{x} \rfloor$ for all positive real numbers x .
 e) $\lfloor x \rfloor + \lfloor y \rfloor + \lfloor x + y \rfloor \leq \lfloor 2x \rfloor + \lfloor 2y \rfloor$ for all real numbers x and y .
- 75.** Prove that if x is a positive real number, then
 a) $\lfloor \sqrt{\lfloor x \rfloor} \rfloor = \lfloor \sqrt{x} \rfloor$.
 b) $\lceil \sqrt{\lceil x \rceil} \rceil = \lceil \sqrt{x} \rceil$.
- 76.** Let x be a real number. Show that $\lfloor 3x \rfloor = \lfloor x \rfloor + \lfloor x + \frac{1}{3} \rfloor + \lfloor x + \frac{2}{3} \rfloor$.
- 77.** For each of these partial functions, determine its domain, codomain, domain of definition, and the set of values for which it is undefined. Also, determine whether it is a total function.
 a) $f: \mathbf{Z} \rightarrow \mathbf{R}, f(n) = 1/n$
 b) $f: \mathbf{Z} \rightarrow \mathbf{Z}, f(n) = \lceil n/2 \rceil$
 c) $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Q}, f(m, n) = m/n$
 d) $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}, f(m, n) = mn$
 e) $f: \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}, f(m, n) = m - n$ if $m > n$
- 78.** a) Show that a partial function from A to B can be viewed as a function f^* from A to $B \cup \{u\}$, where u is not an element of B and
- $$f^*(a) = \begin{cases} f(a) & \text{if } a \text{ belongs to the domain} \\ u & \text{of definition of } f \\ u & \text{if } f \text{ is undefined at } a. \end{cases}$$
- b) Using the construction in (a), find the function f^* corresponding to each partial function in Exercise 77.
- 79.** a) Show that if a set S has cardinality m , where m is a positive integer, then there is a one-to-one correspondence between S and the set $\{1, 2, \dots, m\}$.
 b) Show that if S and T are two sets each with m elements, where m is a positive integer, then there is a one-to-one correspondence between S and T .
- *80.** Show that a set S is infinite if and only if there is a proper subset A of S such that there is a one-to-one correspondence between A and S .

2.4 Sequences and Summations

Introduction

Sequences are ordered lists of elements, used in discrete mathematics in many ways. For example, they can be used to represent solutions to certain counting problems, as we will see in Chapter 8. They are also an important data structure in computer science. We will often need to work with sums of terms of sequences in our study of discrete mathematics. This section reviews the use of summation notation, basic properties of summations, and formulas for the sums of terms of some particular types of sequences.

The terms of a sequence can be specified by providing a formula for each term of the sequence. In this section we describe another way to specify the terms of a sequence using a recurrence relation, which expresses each term as a combination of the previous terms. We will introduce one method, known as iteration, for finding a closed formula for the terms of a sequence specified via a recurrence relation. Identifying a sequence when the first few terms are provided is a useful skill when solving problems in discrete mathematics. We will provide some tips, including a useful tool on the Web, for doing so.

Sequences

A sequence is a discrete structure used to represent an ordered list. For example, 1, 2, 3, 5, 8 is a sequence with five terms and $1, 3, 9, 27, 81, \dots, 3^n, \dots$ is an infinite sequence.

DEFINITION 1

A *sequence* is a function from a subset of the set of integers (usually either the set $\{0, 1, 2, \dots\}$ or the set $\{1, 2, 3, \dots\}$) to a set S . We use the notation a_n to denote the image of the integer n . We call a_n a *term* of the sequence.

We use the notation $\{a_n\}$ to describe the sequence. (Note that a_n represents an individual term of the sequence $\{a_n\}$. Be aware that the notation $\{a_n\}$ for a sequence conflicts with the notation for a set. However, the context in which we use this notation will always make it clear when we are dealing with sets and when we are dealing with sequences. Moreover, although we have used the letter a in the notation for a sequence, other letters or expressions may be used depending on the sequence under consideration. That is, the choice of the letter a is arbitrary.)

We describe sequences by listing the terms of the sequence in order of increasing subscripts.

EXAMPLE 1 Consider the sequence $\{a_n\}$, where

$$a_n = \frac{1}{n}.$$

The list of the terms of this sequence, beginning with a_1 , namely,

$$a_1, a_2, a_3, a_4, \dots,$$

starts with

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots$$

DEFINITION 2

A *geometric progression* is a sequence of the form

$$a, ar, ar^2, \dots, ar^n, \dots$$

where the *initial term* a and the *common ratio* r are real numbers.

Remark: A geometric progression is a discrete analogue of the exponential function $f(x) = ar^x$.

EXAMPLE 2

The sequences $\{b_n\}$ with $b_n = (-1)^n$, $\{c_n\}$ with $c_n = 2 \cdot 5^n$, and $\{d_n\}$ with $d_n = 6 \cdot (1/3)^n$ are geometric progressions with initial term and common ratio equal to 1 and -1 ; 2 and 5; and 6 and $1/3$, respectively, if we start at $n = 0$. The list of terms $b_0, b_1, b_2, b_3, b_4, \dots$ begins with

$$1, -1, 1, -1, 1, \dots;$$

the list of terms $c_0, c_1, c_2, c_3, c_4, \dots$ begins with

$$2, 10, 50, 250, 1250, \dots;$$

and the list of terms $d_0, d_1, d_2, d_3, d_4, \dots$ begins with

$$6, 2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \dots$$

DEFINITION 3

An *arithmetic progression* is a sequence of the form

$$a, a + d, a + 2d, \dots, a + nd, \dots$$

where the *initial term* a and the *common difference* d are real numbers.

Remark: An arithmetic progression is a discrete analogue of the linear function $f(x) = dx + a$.

EXAMPLE 3

The sequences $\{s_n\}$ with $s_n = -1 + 4n$ and $\{t_n\}$ with $t_n = 7 - 3n$ are both arithmetic progressions with initial terms and common differences equal to -1 and 4 , and 7 and -3 , respectively, if we start at $n = 0$. The list of terms $s_0, s_1, s_2, s_3, \dots$ begins with

$$-1, 3, 7, 11, \dots,$$

and the list of terms $t_0, t_1, t_2, t_3, \dots$ begins with

$$7, 4, 1, -2, \dots.$$

Sequences of the form a_1, a_2, \dots, a_n are often used in computer science. These finite sequences are also called **strings**. This string is also denoted by $a_1a_2 \dots a_n$. (Recall that bit strings, which are finite sequences of bits, were introduced in Section 1.1.) The **length** of a string is the number of terms in this string. The **empty string**, denoted by λ , is the string that has no terms. The empty string has length zero.

EXAMPLE 4

The string $abcd$ is a string of length four.

Recurrence Relations

In Examples 1–3 we specified sequences by providing explicit formulas for their terms. There are many other ways to specify a sequence. For example, another way to specify a sequence is

to provide one or more initial terms together with a rule for determining subsequent terms from those that precede them.

DEFINITION 4

A *recurrence relation* for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, a_0, a_1, \dots, a_{n-1} , for all integers n with $n \geq n_0$, where n_0 is a nonnegative integer. A sequence is called a *solution* of a recurrence relation if its terms satisfy the recurrence relation. (A recurrence relation is said to *recursively define* a sequence. We will explain this alternative terminology in Chapter 5.)

EXAMPLE 5 Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 1, 2, 3, \dots$, and suppose that $a_0 = 2$. What are a_1, a_2 , and a_3 ?

Solution: We see from the recurrence relation that $a_1 = a_0 + 3 = 2 + 3 = 5$. It then follows that $a_2 = 5 + 3 = 8$ and $a_3 = 8 + 3 = 11$.

EXAMPLE 6 Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} - a_{n-2}$ for $n = 2, 3, 4, \dots$, and suppose that $a_0 = 3$ and $a_1 = 5$. What are a_2 and a_3 ?

Solution: We see from the recurrence relation that $a_2 = a_1 - a_0 = 5 - 3 = 2$ and $a_3 = a_2 - a_1 = 2 - 5 = -3$. We can find a_4, a_5 , and each successive term in a similar way.

The **initial conditions** for a recursively defined sequence specify the terms that precede the first term where the recurrence relation takes effect. For instance, the initial condition in Example 5 is $a_0 = 2$, and the initial conditions in Example 6 are $a_0 = 3$ and $a_1 = 5$. Using mathematical induction, a proof technique introduced in Chapter 5, it can be shown that a recurrence relation together with its initial conditions determines a unique solution.

Next, we define a particularly useful sequence defined by a recurrence relation, known as the **Fibonacci sequence**, after the Italian mathematician Fibonacci who was born in the 12th century (see Chapter 5 for his biography). We will study this sequence in depth in Chapters 5 and 8, where we will see why it is important for many applications, including modeling the population growth of rabbits.

Hop along to Chapter 8 to learn how to find a formula for the Fibonacci numbers.

DEFINITION 5



The *Fibonacci sequence*, f_0, f_1, f_2, \dots , is defined by the initial conditions $f_0 = 0, f_1 = 1$, and the recurrence relation

$$f_n = f_{n-1} + f_{n-2}$$

for $n = 2, 3, 4, \dots$

EXAMPLE 7 Find the Fibonacci numbers f_2, f_3, f_4, f_5 , and f_6 .

Solution: The recurrence relation for the Fibonacci sequence tells us that we find successive terms by adding the previous two terms. Because the initial conditions tell us that $f_0 = 0$ and $f_1 = 1$, using the recurrence relation in the definition we find that

$$\begin{aligned} f_2 &= f_1 + f_0 = 1 + 0 = 1, \\ f_3 &= f_2 + f_1 = 1 + 1 = 2, \\ f_4 &= f_3 + f_2 = 2 + 1 = 3, \\ f_5 &= f_4 + f_3 = 3 + 2 = 5, \\ f_6 &= f_5 + f_4 = 5 + 3 = 8. \end{aligned}$$

EXAMPLE 8 Suppose that $\{a_n\}$ is the sequence of integers defined by $a_n = n!$, the value of the factorial function at the integer n , where $n = 1, 2, 3, \dots$. Because $n! = n((n-1)(n-2)\dots 2 \cdot 1) = n(n-1)! = na_{n-1}$, we see that the sequence of factorials satisfies the recurrence relation $a_n = na_{n-1}$, together with the initial condition $a_1 = 1$. 

We say that we have solved the recurrence relation together with the initial conditions when we find an explicit formula, called a **closed formula**, for the terms of the sequence.

EXAMPLE 9 Determine whether the sequence $\{a_n\}$, where $a_n = 3n$ for every nonnegative integer n , is a solution of the recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ for $n = 2, 3, 4, \dots$. Answer the same question where $a_n = 2^n$ and where $a_n = 5$.

Solution: Suppose that $a_n = 3n$ for every nonnegative integer n . Then, for $n \geq 2$, we see that $2a_{n-1} - a_{n-2} = 2(3(n-1)) - 3(n-2) = 3n = a_n$. Therefore, $\{a_n\}$, where $a_n = 3n$, is a solution of the recurrence relation.

Suppose that $a_n = 2^n$ for every nonnegative integer n . Note that $a_0 = 1, a_1 = 2$, and $a_2 = 4$. Because $2a_1 - a_0 = 2 \cdot 2 - 1 = 3 \neq a_2$, we see that $\{a_n\}$, where $a_n = 2^n$, is not a solution of the recurrence relation.

Suppose that $a_n = 5$ for every nonnegative integer n . Then for $n \geq 2$, we see that $a_n = 2a_{n-1} - a_{n-2} = 2 \cdot 5 - 5 = 5 = a_n$. Therefore, $\{a_n\}$, where $a_n = 5$, is a solution of the recurrence relation. 

Many methods have been developed for solving recurrence relations. Here, we will introduce a straightforward method known as iteration via several examples. In Chapter 8 we will study recurrence relations in depth. In that chapter we will show how recurrence relations can be used to solve counting problems and we will introduce several powerful methods that can be used to solve many different recurrence relations.

EXAMPLE 10 Solve the recurrence relation and initial condition in Example 5.

Solution: We can successively apply the recurrence relation in Example 5, starting with the initial condition $a_1 = 2$, and working upward until we reach a_n to deduce a closed formula for the sequence. We see that

$$\begin{aligned} a_2 &= 2 + 3 \\ a_3 &= (2 + 3) + 3 = 2 + 3 \cdot 2 \\ a_4 &= (2 + 2 \cdot 3) + 3 = 2 + 3 \cdot 3 \\ &\vdots \\ a_n &= a_{n-1} + 3 = (2 + 3 \cdot (n-2)) + 3 = 2 + 3(n-1). \end{aligned}$$

We can also successively apply the recurrence relation in Example 5, starting with the term a_n and working downward until we reach the initial condition $a_1 = 2$ to deduce this same formula. The steps are

$$\begin{aligned} a_n &= a_{n-1} + 3 \\ &= (a_{n-2} + 3) + 3 = a_{n-2} + 3 \cdot 2 \\ &= (a_{n-3} + 3) + 3 \cdot 2 = a_{n-3} + 3 \cdot 3 \\ &\vdots \\ &= a_2 + 3(n-2) = (a_1 + 3) + 3(n-2) = 2 + 3(n-1). \end{aligned}$$

At each iteration of the recurrence relation, we obtain the next term in the sequence by adding 3 to the previous term. We obtain the n th term after $n - 1$ iterations of the recurrence relation. Hence, we have added $3(n - 1)$ to the initial term $a_0 = 2$ to obtain a_n . This gives us the closed formula $a_n = 2 + 3(n - 1)$. Note that this sequence is an arithmetic progression. 

The technique used in Example 10 is called **iteration**. We have iterated, or repeatedly used, the recurrence relation. The first approach is called **forward substitution** – we found successive terms beginning with the initial condition and ending with a_n . The second approach is called **backward substitution**, because we began with a_n and iterated to express it in terms of falling terms of the sequence until we found it in terms of a_1 . Note that when we use iteration, we essential guess a formula for the terms of the sequence. To prove that our guess is correct, we need to use mathematical induction, a technique we discuss in Chapter 5.

In Chapter 8 we will show that recurrence relations can be used to model a wide variety of problems. We provide one such example here, showing how to use a recurrence relation to find compound interest.

EXAMPLE 11 Compound Interest Suppose that a person deposits \$10,000 in a savings account at a bank yielding 11% per year with interest compounded annually. How much will be in the account after 30 years?



Solution: To solve this problem, let P_n denote the amount in the account after n years. Because the amount in the account after n years equals the amount in the account after $n - 1$ years plus interest for the n th year, we see that the sequence $\{P_n\}$ satisfies the recurrence relation

$$P_n = P_{n-1} + 0.11P_{n-1} = (1.11)P_{n-1}.$$

The initial condition is $P_0 = 10,000$.

We can use an iterative approach to find a formula for P_n . Note that

$$\begin{aligned} P_1 &= (1.11)P_0 \\ P_2 &= (1.11)P_1 = (1.11)^2 P_0 \\ P_3 &= (1.11)P_2 = (1.11)^3 P_0 \\ &\vdots \\ P_n &= (1.11)P_{n-1} = (1.11)^n P_0. \end{aligned}$$

When we insert the initial condition $P_0 = 10,000$, the formula $P_n = (1.11)^n 10,000$ is obtained.

Inserting $n = 30$ into the formula $P_n = (1.11)^n 10,000$ shows that after 30 years the account contains

$$P_{30} = (1.11)^{30} 10,000 = \$228,922.97. \quad \blacktriangleleft$$

Special Integer Sequences

A common problem in discrete mathematics is finding a closed formula, a recurrence relation, or some other type of general rule for constructing the terms of a sequence. Sometimes only a few terms of a sequence solving a problem are known; the goal is to identify the sequence. Even though the initial terms of a sequence do not determine the entire sequence (after all, there are infinitely many different sequences that start with any finite set of initial terms), knowing the first few terms may help you make an educated conjecture about the identity of your sequence. Once you have made this conjecture, you can try to verify that you have the correct sequence.

When trying to deduce a possible formula, recurrence relation, or some other type of rule for the terms of a sequence when given the initial terms, try to find a pattern in these terms. You might also see whether you can determine how a term might have been produced from those preceding it. There are many questions you could ask, but some of the more useful are:

- Are there runs of the same value? That is, does the same value occur many times in a row?
- Are terms obtained from previous terms by adding the same amount or an amount that depends on the position in the sequence?
- Are terms obtained from previous terms by multiplying by a particular amount?
- Are terms obtained by combining previous terms in a certain way?
- Are there cycles among the terms?

EXAMPLE 12 Find formulae for the sequences with the following first five terms: (a) 1, 1/2, 1/4, 1/8, 1/16
 (b) 1, 3, 5, 7, 9 (c) 1, -1, 1, -1, 1.



Solution: (a) We recognize that the denominators are powers of 2. The sequence with $a_n = 1/2^n$, $n = 0, 1, 2, \dots$ is a possible match. This proposed sequence is a geometric progression with $a = 1$ and $r = 1/2$.

(b) We note that each term is obtained by adding 2 to the previous term. The sequence with $a_n = 2n + 1$, $n = 0, 1, 2, \dots$ is a possible match. This proposed sequence is an arithmetic progression with $a = 1$ and $d = 2$.

(c) The terms alternate between 1 and -1. The sequence with $a_n = (-1)^n$, $n = 0, 1, 2, \dots$ is a possible match. This proposed sequence is a geometric progression with $a = 1$ and $r = -1$.

Examples 13–15 illustrate how we can analyze sequences to find how the terms are constructed.

EXAMPLE 13 How can we produce the terms of a sequence if the first 10 terms are 1, 2, 2, 3, 3, 3, 4, 4, 4, 4?

Solution: In this sequence, the integer 1 appears once, the integer 2 appears twice, the integer 3 appears three times, and the integer 4 appears four times. A reasonable rule for generating this sequence is that the integer n appears exactly n times, so the next five terms of the sequence would all be 5, the following six terms would all be 6, and so on. The sequence generated this way is a possible match.

EXAMPLE 14 How can we produce the terms of a sequence if the first 10 terms are 5, 11, 17, 23, 29, 35, 41, 47, 53, 59?

Solution: Note that each of the first 10 terms of this sequence after the first is obtained by adding 6 to the previous term. (We could see this by noticing that the difference between consecutive terms is 6.) Consequently, the n th term could be produced by starting with 5 and adding 6 a total of $n - 1$ times; that is, a reasonable guess is that the n th term is $5 + 6(n - 1) = 6n - 1$. (This is an arithmetic progression with $a = 5$ and $d = 6$.)

EXAMPLE 15 How can we produce the terms of a sequence if the first 10 terms are 1, 3, 4, 7, 11, 18, 29, 47, 76, 123?

Solution: Observe that each successive term of this sequence, starting with the third term, is the sum of the two previous terms. That is, $4 = 3 + 1$, $7 = 4 + 3$, $11 = 7 + 4$, and so on. Consequently, if L_n is the n th term of this sequence, we guess that the sequence is determined by the recurrence relation $L_n = L_{n-1} + L_{n-2}$ with initial conditions $L_1 = 1$ and $L_2 = 3$ (the

TABLE 1 Some Useful Sequences.

<i>nth Term</i>	<i>First 10 Terms</i>
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, ...
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, ...
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, ...
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, ...
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, ...
f_n	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

same recurrence relation as the Fibonacci sequence, but with different initial conditions). This sequence is known as the **Lucas sequence**, after the French mathematician François Édouard Lucas. Lucas studied this sequence and the Fibonacci sequence in the nineteenth century. ▲

Another useful technique for finding a rule for generating the terms of a sequence is to compare the terms of a sequence of interest with the terms of a well-known integer sequence, such as terms of an arithmetic progression, terms of a geometric progression, perfect squares, perfect cubes, and so on. The first 10 terms of some sequences you may want to keep in mind are displayed in Table 1.

EXAMPLE 16 Conjecture a simple formula for a_n if the first 10 terms of the sequence $\{a_n\}$ are 1, 7, 25, 79, 241, 727, 2185, 6559, 19681, 59047.

Solution: To attack this problem, we begin by looking at the difference of consecutive terms, but we do not see a pattern. When we form the ratio of consecutive terms to see whether each term is a multiple of the previous term, we find that this ratio, although not a constant, is close to 3. So it is reasonable to suspect that the terms of this sequence are generated by a formula involving 3^n . Comparing these terms with the corresponding terms of the sequence $\{3^n\}$, we notice that the n th term is 2 less than the corresponding power of 3. We see that $a_n = 3^n - 2$ for $1 \leq n \leq 10$ and conjecture that this formula holds for all n . ▲

We will see throughout this text that integer sequences appear in a wide range of contexts in discrete mathematics. Sequences we have encountered or will encounter include the sequence of prime numbers (Chapter 4), the number of ways to order n discrete objects (Chapter 6), the number of moves required to solve the famous Tower of Hanoi puzzle with n disks (Chapter 8), and the number of rabbits on an island after n months (Chapter 8).

Integer sequences appear in an amazingly wide range of subject areas besides discrete mathematics, including biology, engineering, chemistry, and physics, as well as in puzzles. An amazing database of over 200,000 different integer sequences can be found in the *On-Line Encyclopedia of Integer Sequences (OEIS)*. This database was originated by Neil Sloane in the 1960s. The last printed version of this database was published in 1995 ([SIP195]); the current encyclopedia would occupy more than 750 volumes of the size of the 1995 book with more than 10,000 new submissions a year. There is also a program accessible via the Web that you can use to find sequences from the encyclopedia that match initial terms you provide.

Summations

Next, we consider the addition of the terms of a sequence. For this we introduce **summation notation**. We begin by describing the notation used to express the sum of the terms

$$a_m, a_{m+1}, \dots, a_n$$

Check out the puzzles at
the OEIS site.



from the sequence $\{a_n\}$. We use the notation

$$\sum_{j=m}^n a_j, \quad \sum_{j=m}^n a_j, \quad \text{or} \quad \sum_{m \leq j \leq n} a_j$$

(read as the sum from $j = m$ to $j = n$ of a_j) to represent

$$a_m + a_{m+1} + \cdots + a_n.$$

Here, the variable j is called the **index of summation**, and the choice of the letter j as the variable is arbitrary; that is, we could have used any other letter, such as i or k . Or, in notation,

$$\sum_{j=m}^n a_j = \sum_{i=m}^n a_i = \sum_{k=m}^n a_k.$$

Here, the index of summation runs through all integers starting with its **lower limit** m and ending with its **upper limit** n . A large uppercase Greek letter sigma, \sum , is used to denote summation.

The usual laws for arithmetic apply to summations. For example, when a and b are real numbers, we have $\sum_{j=1}^n (ax_j + by_j) = a \sum_{j=1}^n x_j + b \sum_{j=1}^n y_j$, where x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n are real numbers. (We do not present a formal proof of this identity here. Such a proof can be constructed using mathematical induction, a proof method we introduce in Chapter 5. The proof also uses the commutative and associative laws for addition and the distributive law of multiplication over addition.)

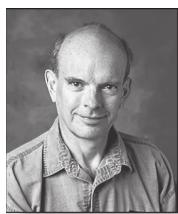
We give some examples of summation notation.

EXAMPLE 17 Use summation notation to express the sum of the first 100 terms of the sequence $\{a_j\}$, where $a_j = 1/j$ for $j = 1, 2, 3, \dots$



Solution: The lower limit for the index of summation is 1, and the upper limit is 100. We write this sum as

$$\sum_{j=1}^{100} \frac{1}{j}.$$



NEIL SLOANE (BORN 1939) Neil Sloane studied mathematics and electrical engineering at the University of Melbourne on a scholarship from the Australian state telephone company. He mastered many telephone-related jobs, such as erecting telephone poles, in his summer work. After graduating, he designed minimal-cost telephone networks in Australia. In 1962 he came to the United States and studied electrical engineering at Cornell University. His Ph.D. thesis was on what are now called neural networks. He took a job at Bell Labs in 1969, working in many areas, including network design, coding theory, and sphere packing. He now works for AT&T Labs, moving there from Bell Labs when AT&T split up in 1996. One of his favorite problems is the **kissing problem** (a name he coined), which asks how many spheres can be arranged in n dimensions so that they all touch a central sphere of the same size. (In two dimensions the answer is 6, because 6 pennies can be placed so that they touch a central penny. In three dimensions, 12 billiard balls can be placed so that they touch a central billiard ball. Two billiard balls that just touch are said to “kiss,” giving rise to the terminology “kissing problem” and “kissing number.”) Sloane, together with Andrew Odlyzko, showed that in 8 and 24 dimensions, the optimal kissing numbers are, respectively, 240 and 196,560. The kissing number is known in dimensions 1, 2, 3, 4, 8, and 24, but not in any other dimensions. Sloane’s books include *Sphere Packings, Lattices and Groups*, 3d ed., with John Conway; *The Theory of Error-Correcting Codes* with Jessie MacWilliams; *The Encyclopedia of Integer Sequences* with Simon Plouffe (which has grown into the famous OEIS website); and *The Rock-Climbing Guide to New Jersey Crags* with Paul Nick. The last book demonstrates his interest in rock climbing; it includes more than 50 climbing sites in New Jersey.

EXAMPLE 18 What is the value of $\sum_{j=1}^5 j^2$?

Solution: We have

$$\begin{aligned}\sum_{j=1}^5 j^2 &= 1^2 + 2^2 + 3^2 + 4^2 + 5^2 \\ &= 1 + 4 + 9 + 16 + 25 \\ &= 55.\end{aligned}$$

EXAMPLE 19 What is the value of $\sum_{k=4}^8 (-1)^k$?

Solution: We have

$$\begin{aligned}\sum_{k=4}^8 (-1)^k &= (-1)^4 + (-1)^5 + (-1)^6 + (-1)^7 + (-1)^8 \\ &= 1 + (-1) + 1 + (-1) + 1 \\ &= 1.\end{aligned}$$

Sometimes it is useful to shift the index of summation in a sum. This is often done when two sums need to be added but their indices of summation do not match. When shifting an index of summation, it is important to make the appropriate changes in the corresponding summand. This is illustrated by Example 20.

EXAMPLE 20 Suppose we have the sum

$$\sum_{j=1}^5 j^2$$



but want the index of summation to run between 0 and 4 rather than from 1 to 5. To do this, we let $k = j - 1$. Then the new summation index runs from 0 (because $k = 1 - 0 = 0$ when $j = 1$) to 4 (because $k = 5 - 1 = 4$ when $j = 5$), and the term j^2 becomes $(k + 1)^2$. Hence,

$$\sum_{j=1}^5 j^2 = \sum_{k=0}^4 (k + 1)^2.$$

It is easily checked that both sums are $1 + 4 + 9 + 16 + 25 = 55$.

Sums of terms of geometric progressions commonly arise (such sums are called **geometric series**). Theorem 1 gives us a formula for the sum of terms of a geometric progression.

THEOREM 1

If a and r are real numbers and $r \neq 0$, then

$$\sum_{j=0}^n ar^j = \begin{cases} \frac{ar^{n+1} - a}{r - 1} & \text{if } r \neq 1 \\ (n + 1)a & \text{if } r = 1. \end{cases}$$

Proof: Let

$$S_n = \sum_{j=0}^n ar^j.$$

To compute S , first multiply both sides of the equality by r and then manipulate the resulting sum as follows:

$$\begin{aligned}
 rS_n &= r \sum_{j=0}^n ar^j && \text{substituting summation formula for } S \\
 &= \sum_{j=0}^n ar^{j+1} && \text{by the distributive property} \\
 &= \sum_{k=1}^{n+1} ar^k && \text{shifting the index of summation, with } k = j + 1 \\
 &= \left(\sum_{k=0}^n ar^k \right) + (ar^{n+1} - a) && \text{removing } k = n + 1 \text{ term and adding } k = 0 \text{ term} \\
 &= S_n + (ar^{n+1} - a) && \text{substituting } S \text{ for summation formula}
 \end{aligned}$$

From these equalities, we see that

$$rS_n = S_n + (ar^{n+1} - a).$$

Solving for S_n shows that if $r \neq 1$, then

$$S_n = \frac{ar^{n+1} - a}{r - 1}.$$

If $r = 1$, then the $S_n = \sum_{j=0}^n ar^j = \sum_{j=0}^n a = (n + 1)a$. ◀

EXAMPLE 21 Double summations arise in many contexts (as in the analysis of nested loops in computer programs). An example of a double summation is

$$\sum_{i=1}^4 \sum_{j=1}^3 ij.$$

To evaluate the double sum, first expand the inner summation and then continue by computing the outer summation:

$$\begin{aligned}
 \sum_{i=1}^4 \sum_{j=1}^3 ij &= \sum_{i=1}^4 (i + 2i + 3i) \\
 &= \sum_{i=1}^4 6i \\
 &= 6 + 12 + 18 + 24 = 60.
 \end{aligned}$$
◀

We can also use summation notation to add all values of a function, or terms of an indexed set, where the index of summation runs over all values in a set. That is, we write

$$\sum_{s \in S} f(s)$$

to represent the sum of the values $f(s)$, for all members s of S .

TABLE 2 Some Useful Summation Formulae.

<i>Sum</i>	<i>Closed Form</i>
$\sum_{k=0}^n ar^k (r \neq 0)$	$\frac{ar^{n+1} - a}{r - 1}, r \neq 1$
$\sum_{k=1}^n k$	$\frac{n(n + 1)}{2}$
$\sum_{k=1}^n k^2$	$\frac{n(n + 1)(2n + 1)}{6}$
$\sum_{k=1}^n k^3$	$\frac{n^2(n + 1)^2}{4}$
$\sum_{k=0}^{\infty} x^k, x < 1$	$\frac{1}{1 - x}$
$\sum_{k=1}^{\infty} kx^{k-1}, x < 1$	$\frac{1}{(1 - x)^2}$

EXAMPLE 22 What is the value of $\sum_{s \in \{0, 2, 4\}} s$?

Solution: Because $\sum_{s \in \{0, 2, 4\}} s$ represents the sum of the values of s for all the members of the set $\{0, 2, 4\}$, it follows that

$$\sum_{s \in \{0, 2, 4\}} s = 0 + 2 + 4 = 6.$$

Certain sums arise repeatedly throughout discrete mathematics. Having a collection of formulae for such sums can be useful; Table 2 provides a small table of formulae for commonly occurring sums.

We derived the first formula in this table in Theorem 1. The next three formulae give us the sum of the first n positive integers, the sum of their squares, and the sum of their cubes. These three formulae can be derived in many different ways (for example, see Exercises 37 and 38). Also note that each of these formulae, once known, can easily be proved using mathematical induction, the subject of Section 5.1. The last two formulae in the table involve infinite series and will be discussed shortly.

Example 23 illustrates how the formulae in Table 2 can be useful.

EXAMPLE 23 Find $\sum_{k=50}^{100} k^2$.

Solution: First note that because $\sum_{k=1}^{100} k^2 = \sum_{k=1}^{49} k^2 + \sum_{k=50}^{100} k^2$, we have

$$\sum_{k=50}^{100} k^2 = \sum_{k=1}^{100} k^2 - \sum_{k=1}^{49} k^2.$$

Using the formula $\sum_{k=1}^n k^2 = n(n + 1)(2n + 1)/6$ from Table 2 (and proved in Exercise 38), we see that

$$\sum_{k=50}^{100} k^2 = \frac{100 \cdot 101 \cdot 201}{6} - \frac{49 \cdot 50 \cdot 99}{6} = 338,350 - 40,425 = 297,925.$$

SOME INFINITE SERIES Although most of the summations in this book are finite sums, infinite series are important in some parts of discrete mathematics. Infinite series are usually studied in a course in calculus and even the definition of these series requires the use of calculus, but sometimes they arise in discrete mathematics, because discrete mathematics deals with infinite collections of discrete elements. In particular, in our future studies in discrete mathematics, we will find the closed forms for the infinite series in Examples 24 and 25 to be quite useful.

EXAMPLE 24

(Requires calculus) Let x be a real number with $|x| < 1$. Find $\sum_{n=0}^{\infty} x^n$.



Solution: By Theorem 1 with $a = 1$ and $r = x$ we see that $\sum_{n=0}^k x^n = \frac{x^{k+1} - 1}{x - 1}$. Because $|x| < 1$, x^{k+1} approaches 0 as k approaches infinity. It follows that

$$\sum_{n=0}^{\infty} x^n = \lim_{k \rightarrow \infty} \frac{x^{k+1} - 1}{x - 1} = \frac{0 - 1}{x - 1} = \frac{1}{1-x}.$$



We can produce new summation formulae by differentiating or integrating existing formulae.

EXAMPLE 25

(Requires calculus) Differentiating both sides of the equation

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x},$$

from Example 24 we find that

$$\sum_{k=1}^{\infty} kx^{k-1} = \frac{1}{(1-x)^2}.$$



(This differentiation is valid for $|x| < 1$ by a theorem about infinite series.)

Exercises

1. Find these terms of the sequence $\{a_n\}$, where $a_n = 2 \cdot (-3)^n + 5^n$.
 - a) a_0
 - b) a_1
 - c) a_4
 - d) a_5
2. What is the term a_8 of the sequence $\{a_n\}$ if a_n equals
 - a) 2^{n-1}
 - b) 7?
 - c) $1 + (-1)^n$?
 - d) $-(-2)^n$?
3. What are the terms a_0, a_1, a_2 , and a_3 of the sequence $\{a_n\}$, where a_n equals
 - a) $2^n + 1$?
 - b) $(n+1)^{n+1}$?
 - c) $\lfloor n/2 \rfloor$?
 - d) $\lfloor n/2 \rfloor + \lceil n/2 \rceil$?
4. What are the terms a_0, a_1, a_2 , and a_3 of the sequence $\{a_n\}$, where a_n equals
 - a) $(-2)^n$?
 - b) 3?
 - c) $7 + 4^n$?
 - d) $2^n + (-2)^n$?
5. List the first 10 terms of each of these sequences.
 - a) the sequence that begins with 2 and in which each successive term is 3 more than the preceding term
 - b) the sequence that lists each positive integer three times, in increasing order
 - c) the sequence that lists the odd positive integers in increasing order, listing each odd integer twice
 - d) the sequence whose n th term is $n! - 2^n$
 - e) the sequence that begins with 3, where each succeeding term is twice the preceding term
 - f) the sequence whose first term is 2, second term is 4, and each succeeding term is the sum of the two preceding terms
 - g) the sequence whose n th term is the number of bits in the binary expansion of the number n (defined in Section 4.2)
 - h) the sequence where the n th term is the number of letters in the English word for the index n
6. List the first 10 terms of each of these sequences.
 - a) the sequence obtained by starting with 10 and obtaining each term by subtracting 3 from the previous term
 - b) the sequence whose n th term is the sum of the first n positive integers
 - c) the sequence whose n th term is $3^n - 2^n$
 - d) the sequence whose n th term is $\lfloor \sqrt{n} \rfloor$
 - e) the sequence whose first two terms are 1 and 5 and each succeeding term is the sum of the two previous terms

- f) the sequence whose n th term is the largest integer whose binary expansion (defined in Section 4.2) has n bits (Write your answer in decimal notation.)
- g) the sequence whose terms are constructed sequentially as follows: start with 1, then add 1, then multiply by 1, then add 2, then multiply by 2, and so on
- h) the sequence whose n th term is the largest integer k such that $k! \leq n$
7. Find at least three different sequences beginning with the terms 1, 2, 4 whose terms are generated by a simple formula or rule.
8. Find at least three different sequences beginning with the terms 3, 5, 7 whose terms are generated by a simple formula or rule.
9. Find the first five terms of the sequence defined by each of these recurrence relations and initial conditions.
- $a_n = 6a_{n-1}$, $a_0 = 2$
 - $a_n = a_{n-1}^2$, $a_1 = 2$
 - $a_n = a_{n-1} + 3a_{n-2}$, $a_0 = 1$, $a_1 = 2$
 - $a_n = na_{n-1} + n^2a_{n-2}$, $a_0 = 1$, $a_1 = 1$
 - $a_n = a_{n-1} + a_{n-3}$, $a_0 = 1$, $a_1 = 2$, $a_2 = 0$
10. Find the first six terms of the sequence defined by each of these recurrence relations and initial conditions.
- $a_n = -2a_{n-1}$, $a_0 = -1$
 - $a_n = a_{n-1} - a_{n-2}$, $a_0 = 2$, $a_1 = -1$
 - $a_n = 3a_{n-1}^2$, $a_0 = 1$
 - $a_n = na_{n-1} + a_{n-2}^2$, $a_0 = -1$, $a_1 = 0$
 - $a_n = a_{n-1} - a_{n-2} + a_{n-3}$, $a_0 = 1$, $a_1 = 1$, $a_2 = 2$
11. Let $a_n = 2^n + 5 \cdot 3^n$ for $n = 0, 1, 2, \dots$.
- Find a_0, a_1, a_2, a_3 , and a_4 .
 - Show that $a_2 = 5a_1 - 6a_0$, $a_3 = 5a_2 - 6a_1$, and $a_4 = 5a_3 - 6a_2$.
 - Show that $a_n = 5a_{n-1} - 6a_{n-2}$ for all integers n with $n \geq 2$.
12. Show that the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = -3a_{n-1} + 4a_{n-2}$ if
- $a_n = 0$.
 - $a_n = 1$.
 - $a_n = (-4)^n$.
 - $a_n = 2(-4)^n + 3$.
13. Is the sequence $\{a_n\}$ a solution of the recurrence relation $a_n = 8a_{n-1} - 16a_{n-2}$ if
- $a_n = 0?$
 - $a_n = 1?$
 - $a_n = 2^n?$
 - $a_n = 4^n?$
 - $a_n = n4^n?$
 - $a_n = 2 \cdot 4^n + 3n4^n?$
 - $a_n = (-4)^n?$
 - $a_n = n^24^n?$
14. For each of these sequences find a recurrence relation satisfied by this sequence. (The answers are not unique because there are infinitely many different recurrence relations satisfied by any sequence.)
- $a_n = 3$
 - $a_n = 2n$
 - $a_n = 2n + 3$
 - $a_n = 5^n$
 - $a_n = n^2$
 - $a_n = n^2 + n$
 - $a_n = n + (-1)^n$
 - $a_n = n!$
15. Show that the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = a_{n-1} + 2a_{n-2} + 2n - 9$ if
- $a_n = -n + 2$.
 - $a_n = 5(-1)^n - n + 2$.
- c) $a_n = 3(-1)^n + 2^n - n + 2$.
- d) $a_n = 7 \cdot 2^n - n + 2$.
16. Find the solution to each of these recurrence relations with the given initial conditions. Use an iterative approach such as that used in Example 10.
- $a_n = -a_{n-1}$, $a_0 = 5$
 - $a_n = a_{n-1} + 3$, $a_0 = 1$
 - $a_n = a_{n-1} - n$, $a_0 = 4$
 - $a_n = 2a_{n-1} - 3$, $a_0 = -1$
 - $a_n = (n + 1)a_{n-1}$, $a_0 = 2$
 - $a_n = 2na_{n-1}$, $a_0 = 3$
 - $a_n = -a_{n-1} + n - 1$, $a_0 = 7$
17. Find the solution to each of these recurrence relations and initial conditions. Use an iterative approach such as that used in Example 10.
- $a_n = 3a_{n-1}$, $a_0 = 2$
 - $a_n = a_{n-1} + 2$, $a_0 = 3$
 - $a_n = a_{n-1} + n$, $a_0 = 1$
 - $a_n = a_{n-1} + 2n + 3$, $a_0 = 4$
 - $a_n = 2a_{n-1} - 1$, $a_0 = 1$
 - $a_n = 3a_{n-1} + 1$, $a_0 = 1$
 - $a_n = na_{n-1}$, $a_0 = 5$
 - $a_n = 2na_{n-1}$, $a_0 = 1$
18. A person deposits \$1000 in an account that yields 9% interest compounded annually.
- Set up a recurrence relation for the amount in the account at the end of n years.
 - Find an explicit formula for the amount in the account at the end of n years.
 - How much money will the account contain after 100 years?
19. Suppose that the number of bacteria in a colony triples every hour.
- Set up a recurrence relation for the number of bacteria after n hours have elapsed.
 - If 100 bacteria are used to begin a new colony, how many bacteria will be in the colony in 10 hours?
20. Assume that the population of the world in 2010 was 6.9 billion and is growing at the rate of 1.1% a year.
-  a) Set up a recurrence relation for the population of the world n years after 2010.
- Find an explicit formula for the population of the world n years after 2010.
 - What will the population of the world be in 2030?
21. A factory makes custom sports cars at an increasing rate. In the first month only one car is made, in the second month two cars are made, and so on, with n cars made in the n th month.
- Set up a recurrence relation for the number of cars produced in the first n months by this factory.
 - How many cars are produced in the first year?
 - Find an explicit formula for the number of cars produced in the first n months by this factory.
22. An employee joined a company in 2009 with a starting salary of \$50,000. Every year this employee receives a raise of \$1000 plus 5% of the salary of the previous year.

- a) Set up a recurrence relation for the salary of this employee n years after 2009.
 b) What will the salary of this employee be in 2017?
 c) Find an explicit formula for the salary of this employee n years after 2009.
23. Find a recurrence relation for the balance $B(k)$ owed at the end of k months on a loan of \$5000 at a rate of 7% if a payment of \$100 is made each month. [Hint: Express $B(k)$ in terms of $B(k - 1)$; the monthly interest is $(0.07/12)B(k - 1)$.]
24. a) Find a recurrence relation for the balance $B(k)$ owed at the end of k months on a loan at a rate of r if a payment P is made on the loan each month. [Hint: Express $B(k)$ in terms of $B(k - 1)$ and note that the monthly interest rate is $r/12$.]
 b) Determine what the monthly payment P should be so that the loan is paid off after T months.
25. For each of these lists of integers, provide a simple formula or rule that generates the terms of an integer sequence that begins with the given list. Assuming that your formula or rule is correct, determine the next three terms of the sequence.
- a) 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, ...
 - b) 1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, ...
 - c) 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, ...
 - d) 3, 6, 12, 24, 48, 96, 192, ...
 - e) 15, 8, 1, -6, -13, -20, -27, ...
 - f) 3, 5, 8, 12, 17, 23, 30, 38, 47, ...
 - g) 2, 16, 54, 128, 250, 432, 686, ...
 - h) 2, 3, 7, 25, 121, 721, 5041, 40321, ...
26. For each of these lists of integers, provide a simple formula or rule that generates the terms of an integer sequence that begins with the given list. Assuming that your formula or rule is correct, determine the next three terms of the sequence.
- a) 3, 6, 11, 18, 27, 38, 51, 66, 83, 102, ...
 - b) 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, ...
 - c) 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, ...
 - d) 1, 2, 2, 2, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, ...
 - e) 0, 2, 8, 26, 80, 242, 728, 2186, 6560, 19682, ...
 - f) 1, 3, 15, 105, 945, 10395, 135135, 2027025, 34459425, ...
 - g) 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, ...
 - h) 2, 4, 16, 256, 65536, 4294967296, ...
- **27. Show that if a_n denotes the n th positive integer that is not a perfect square, then $a_n = n + \{\sqrt{n}\}$, where $\{x\}$ denotes the integer closest to the real number x .
- *28. Let a_n be the n th term of the sequence 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 6, ..., constructed by including the integer k exactly k times. Show that $a_n = \lfloor \sqrt{2n} + \frac{1}{2} \rfloor$.
29. What are the values of these sums?
- a) $\sum_{k=1}^5 (k+1)$
 - b) $\sum_{j=0}^4 (-2)^j$
 - c) $\sum_{i=1}^{10} 3$
 - d) $\sum_{j=0}^8 (2^{j+1} - 2^j)$
30. What are the values of these sums, where $S = \{1, 3, 5, 7\}$?
- a) $\sum_{j \in S} j$
 - b) $\sum_{j \in S} j^2$
 - c) $\sum_{j \in S} (1/j)$
 - d) $\sum_{j \in S} 1$
31. What is the value of each of these sums of terms of a geometric progression?
- a) $\sum_{j=0}^8 3 \cdot 2^j$
 - b) $\sum_{j=1}^8 2^j$
 - c) $\sum_{j=2}^8 (-3)^j$
 - d) $\sum_{j=0}^8 2 \cdot (-3)^j$
32. Find the value of each of these sums.
- a) $\sum_{j=0}^8 (1 + (-1)^j)$
 - b) $\sum_{j=0}^8 (3^j - 2^j)$
 - c) $\sum_{j=0}^8 (2 \cdot 3^j + 3 \cdot 2^j)$
 - d) $\sum_{j=0}^8 (2^{j+1} - 2^j)$
33. Compute each of these double sums.
- a) $\sum_{i=1}^2 \sum_{j=1}^3 (i+j)$
 - b) $\sum_{i=0}^2 \sum_{j=0}^3 (2i+3j)$
 - c) $\sum_{i=1}^3 \sum_{j=0}^2 i$
 - d) $\sum_{i=0}^2 \sum_{j=1}^3 ij$
34. Compute each of these double sums.
- a) $\sum_{i=1}^3 \sum_{j=1}^2 (i-j)$
 - b) $\sum_{i=0}^3 \sum_{j=0}^2 (3i+2j)$
 - c) $\sum_{i=1}^3 \sum_{j=0}^2 j$
 - d) $\sum_{i=0}^2 \sum_{j=0}^3 i^2 j^3$
35. Show that $\sum_{j=1}^n (a_j - a_{j-1}) = a_n - a_0$, where a_0, a_1, \dots, a_n is a sequence of real numbers. This type of sum is called **telescoping**.
36. Use the identity $1/(k(k+1)) = 1/k - 1/(k+1)$ and Exercise 35 to compute $\sum_{k=1}^n 1/(k(k+1))$.
37. Sum both sides of the identity $k^2 - (k-1)^2 = 2k-1$ from $k=1$ to $k=n$ and use Exercise 35 to find
- a) a formula for $\sum_{k=1}^n (2k-1)$ (the sum of the first n odd natural numbers).
 - b) a formula for $\sum_{k=1}^n k$.
- *38. Use the technique given in Exercise 35, together with the result of Exercise 37b, to derive the formula for $\sum_{k=1}^n k^2$ given in Table 2. [Hint: Take $a_k = k^3$ in the telescoping sum in Exercise 35.]
39. Find $\sum_{k=100}^{200} k$. (Use Table 2.)
40. Find $\sum_{k=99}^{200} k^3$. (Use Table 2.)
- *41. Find a formula for $\sum_{k=0}^m \lfloor \sqrt{k} \rfloor$, when m is a positive integer.
- *42. Find a formula for $\sum_{k=0}^m \lfloor \sqrt[3]{k} \rfloor$, when m is a positive integer.

There is also a special notation for products. The product of a_m, a_{m+1}, \dots, a_n is represented by $\prod_{j=m}^n a_j$, read as the product from $j=m$ to $j=n$ of a_j .

43. What are the values of the following products?

- a) $\prod_{i=0}^{10} i$
- b) $\prod_{i=5}^8 i$
- c) $\prod_{i=1}^{100} (-1)^i$
- d) $\prod_{i=1}^{10} 2$

Recall that the value of the factorial function at a positive integer n , denoted by $n!$, is the product of the positive integers from 1 to n , inclusive. Also, we specify that $0! = 1$.

44. Express $n!$ using product notation.

45. Find $\sum_{j=0}^4 j!$.

46. Find $\prod_{j=0}^4 j!$.

2.5 Cardinality of Sets

Introduction

In Definition 4 of Section 2.1 we defined the cardinality of a finite set as the number of elements in the set. We use the cardinalities of finite sets to tell us when they have the same size, or when one is bigger than the other. In this section we extend this notion to infinite sets. That is, we will define what it means for two infinite sets to have the same cardinality, providing us with a way to measure the relative sizes of infinite sets.

We will be particularly interested in countably infinite sets, which are sets with the same cardinality as the set of positive integers. We will establish the surprising result that the set of rational numbers is countably infinite. We will also provide an example of an uncountable set when we show that the set of real numbers is not countable.

The concepts developed in this section have important applications to computer science. A function is called uncomputable if no computer program can be written to find all its values, even with unlimited time and memory. We will use the concepts in this section to explain why uncomputable functions exist.

We now define what it means for two sets to have the same size, or cardinality. In Section 2.1, we discussed the cardinality of finite sets and we defined the size, or cardinality, of such sets. In Exercise 79 of Section 2.3 we showed that there is a one-to-one correspondence between any two finite sets with the same number of elements. We use this observation to extend the concept of cardinality to all sets, both finite and infinite.

DEFINITION 1

The sets A and B have the same *cardinality* if and only if there is a one-to-one correspondence from A to B . When A and B have the same cardinality, we write $|A| = |B|$.

For infinite sets the definition of cardinality provides a relative measure of the sizes of two sets, rather than a measure of the size of one particular set. We can also define what it means for one set to have a smaller cardinality than another set.

DEFINITION 2

If there is a one-to-one function from A to B , the cardinality of A is less than or the same as the cardinality of B and we write $|A| \leq |B|$. Moreover, when $|A| \leq |B|$ and A and B have different cardinality, we say that the cardinality of A is less than the cardinality of B and we write $|A| < |B|$.

Countable Sets

We will now split infinite sets into two groups, those with the same cardinality as the set of natural numbers and those with a different cardinality.

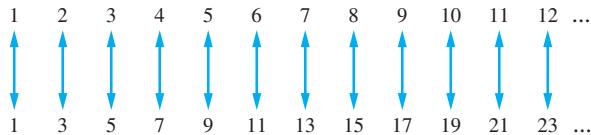


FIGURE 1 A One-to-One Correspondence Between \mathbf{Z}^+ and the Set of Odd Positive Integers.

DEFINITION 3

A set that is either finite or has the same cardinality as the set of positive integers is called *countable*. A set that is not countable is called *uncountable*. When an infinite set S is countable, we denote the cardinality of S by \aleph_0 (where \aleph is aleph, the first letter of the Hebrew alphabet). We write $|S| = \aleph_0$ and say that S has cardinality “aleph null.”

We illustrate how to show a set is countable in the next example.

EXAMPLE 1

Show that the set of odd positive integers is a countable set.

Solution: To show that the set of odd positive integers is countable, we will exhibit a one-to-one correspondence between this set and the set of positive integers. Consider the function

$$f(n) = 2n - 1$$

from \mathbf{Z}^+ to the set of odd positive integers. We show that f is a one-to-one correspondence by showing that it is both one-to-one and onto. To see that it is one-to-one, suppose that $f(n) = f(m)$. Then $2n - 1 = 2m - 1$, so $n = m$. To see that it is onto, suppose that t is an odd positive integer. Then t is 1 less than an even integer $2k$, where k is a natural number. Hence $t = 2k - 1 = f(k)$. We display this one-to-one correspondence in Figure 1. \blacktriangleleft

An infinite set is countable if and only if it is possible to list the elements of the set in a sequence (indexed by the positive integers). The reason for this is that a one-to-one correspondence f from the set of positive integers to a set S can be expressed in terms of a sequence $a_1, a_2, \dots, a_n, \dots$, where $a_1 = f(1), a_2 = f(2), \dots, a_n = f(n), \dots$

You can always get a room
at Hilbert's Grand Hotel!



HILBERT'S GRAND HOTEL We now describe a paradox that shows that something impossible with finite sets may be possible with infinite sets. The famous mathematician David Hilbert invented the notion of the **Grand Hotel**, which has a countably infinite number of rooms, each occupied by a guest. When a new guest arrives at a hotel with a finite number of rooms, and all rooms are occupied, this guest cannot be accommodated without evicting a current guest. However, we can always accommodate a new guest at the Grand Hotel, even when all rooms are already occupied, as we show in Example 2. Exercises 5 and 8 ask you to show that we can accommodate a finite number of new guests and a countable number of new guests, respectively, at the fully occupied Grand Hotel.



DAVID HILBERT (1862–1943) Hilbert, born in Königsberg, the city famous in mathematics for its seven bridges, was the son of a judge. During his tenure at Göttingen University, from 1892 to 1930, he made many fundamental contributions to a wide range of mathematical subjects. He almost always worked on one area of mathematics at a time, making important contributions, then moving to a new mathematical subject. Some areas in which Hilbert worked are the calculus of variations, geometry, algebra, number theory, logic, and mathematical physics. Besides his many outstanding original contributions, Hilbert is remembered for his famous list of 23 difficult problems. He described these problems at the 1900 International Congress of Mathematicians, as a challenge to mathematicians at the birth of the twentieth century. Since that time, they have spurred a tremendous amount and variety of research. Although many of these problems have now been solved, several remain open, including the Riemann hypothesis, which is part of Problem 8 on Hilbert's list. Hilbert was also the author of several important textbooks in number theory and geometry.

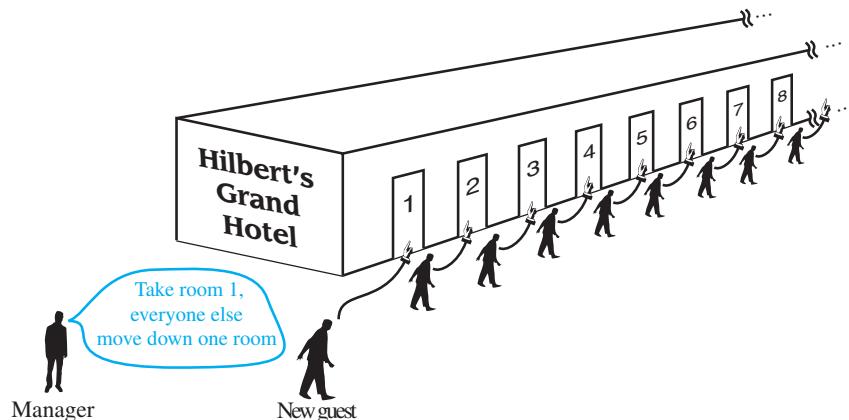


FIGURE 2 A New Guest Arrives at Hilbert's Grand Hotel.

EXAMPLE 2 How can we accommodate a new guest arriving at the fully occupied Grand Hotel without removing any of the current guests?

Solution: Because the rooms of the Grand Hotel are countable, we can list them as Room 1, Room 2, Room 3, and so on. When a new guest arrives, we move the guest in Room 1 to Room 2, the guest in Room 2 to Room 3, and in general, the guest in Room n to Room $n + 1$, for all positive integers n . This frees up Room 1, which we assign to the new guest, and all the current guests still have rooms. We illustrate this situation in Figure 2. \blacktriangleleft

When there are finitely many room in a hotel, the notion that all rooms are occupied is equivalent to the notion that no new guests can be accommodated. However, Hilbert's paradox of the Grand Hotel can be explained by noting that this equivalence no longer holds when there are infinitely many room.

EXAMPLES OF COUNTABLE AND UNCOUNTABLE SETS We will now show that certain sets of numbers are countable. We begin with the set of all integers. Note that we can show that the set of all integers is countable by listing its members.

EXAMPLE 3 Show that the set of all integers is countable.

Solution: We can list all integers in a sequence by starting with 0 and alternating between positive and negative integers: 0, 1, -1 , 2, -2 , \dots . Alternatively, we could find a one-to-one correspondence between the set of positive integers and the set of all integers. We leave it to the reader to show that the function $f(n) = n/2$ when n is even and $f(n) = -(n - 1)/2$ when n is odd is such a function. Consequently, the set of all integers is countable. \blacktriangleleft

It is not surprising that the set of odd integers and the set of all integers are both countable sets (as shown in Examples 1 and 3). Many people are amazed to learn that the set of rational numbers is countable, as Example 4 demonstrates.

EXAMPLE 4 Show that the set of positive rational numbers is countable.

Solution: It may seem surprising that the set of positive rational numbers is countable, but we will show how we can list the positive rational numbers as a sequence $r_1, r_2, \dots, r_n, \dots$. First, note that every positive rational number is the quotient p/q of two positive integers. We can

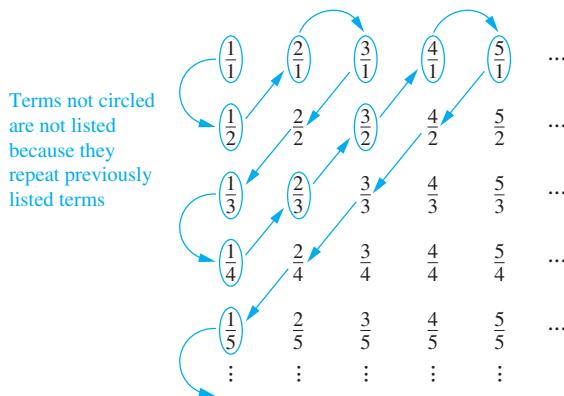


FIGURE 3 The Positive Rational Numbers Are Countable.

arrange the positive rational numbers by listing those with denominator $q = 1$ in the first row, those with denominator $q = 2$ in the second row, and so on, as displayed in Figure 3.

The key to listing the rational numbers in a sequence is to first list the positive rational numbers p/q with $p + q = 2$, followed by those with $p + q = 3$, followed by those with $p + q = 4$, and so on, following the path shown in Figure 3. Whenever we encounter a number p/q that is already listed, we do not list it again. For example, when we come to $2/2 = 1$ we do not list it because we have already listed $1/1 = 1$. The initial terms in the list of positive rational numbers we have constructed are $1, 1/2, 2, 3, 1/3, 1/4, 2/3, 3/2, 4, 5$, and so on. These numbers are shown circled; the uncircled numbers in the list are those we leave out because they are already listed. Because all positive rational numbers are listed once, as the reader can verify, we have shown that the set of positive rational numbers is countable. \blacktriangleleft

An Uncountable Set

Not all infinite sets have the same size!



We have seen that the set of positive rational numbers is a countable set. Do we have a promising candidate for an uncountable set? The first place we might look is the set of real numbers. In Example 5 we use an important proof method, introduced in 1879 by Georg Cantor and known as the **Cantor diagonalization argument**, to prove that the set of real numbers is not countable. This proof method is used extensively in mathematical logic and in the theory of computation.

EXAMPLE 5

Show that the set of real numbers is an uncountable set.



Solution: To show that the set of real numbers is uncountable, we suppose that the set of real numbers is countable and arrive at a contradiction. Then, the subset of all real numbers that fall between 0 and 1 would also be countable (because any subset of a countable set is also countable; see Exercise 16). Under this assumption, the real numbers between 0 and 1 can be listed in some order, say, r_1, r_2, r_3, \dots . Let the decimal representation of these real numbers be

$$r_1 = 0.d_{11}d_{12}d_{13}d_{14} \dots$$

$$r_2 = 0.d_{21}d_{22}d_{23}d_{24} \dots$$

$$r_3 = 0.d_{31}d_{32}d_{33}d_{34} \dots$$

$$r_4 = 0.d_{41}d_{42}d_{43}d_{44} \dots$$

⋮

where $d_{ij} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. (For example, if $r_1 = 0.23794102 \dots$, we have $d_{11} = 2, d_{12} = 3, d_{13} = 7$, and so on.) Then, form a new real number with decimal expansion

$r = 0.d_1d_2d_3d_4\dots$, where the decimal digits are determined by the following rule:

$$d_i = \begin{cases} 4 & \text{if } d_{ii} \neq 4 \\ 5 & \text{if } d_{ii} = 4. \end{cases}$$

(As an example, suppose that $r_1 = 0.23794102\dots$, $r_2 = 0.44590138\dots$, $r_3 = 0.09118764\dots$, $r_4 = 0.80553900\dots$, and so on. Then we have $r = 0.d_1d_2d_3d_4\dots = 0.4544\dots$, where $d_1 = 4$ because $d_{11} \neq 4$, $d_2 = 5$ because $d_{22} = 4$, $d_3 = 4$ because $d_{33} \neq 4$, $d_4 = 4$ because $d_{44} \neq 4$, and so on.)



A number with a decimal expansion that terminates has a second decimal expansion ending with an infinite sequence of 9s because $1 = 0.999\dots$

Every real number has a unique decimal expansion (when the possibility that the expansion has a tail end that consists entirely of the digit 9 is excluded). Therefore, the real number r is not equal to any of r_1, r_2, \dots because the decimal expansion of r differs from the decimal expansion of r_i in the i th place to the right of the decimal point, for each i .

Because there is a real number r between 0 and 1 that is not in the list, the assumption that all the real numbers between 0 and 1 could be listed must be false. Therefore, all the real numbers between 0 and 1 cannot be listed, so the set of real numbers between 0 and 1 is uncountable. Any set with an uncountable subset is uncountable (see Exercise 15). Hence, the set of real numbers is uncountable. 

RESULTS ABOUT CARDINALITY We will now discuss some results about the cardinality of sets. First, we will prove that the union of two countable sets is also countable.

THEOREM 1

If A and B are countable sets, then $A \cup B$ is also countable.

This proof uses WLOG and cases.

Proof: Suppose that A and B are both countable sets. Without loss of generality, we can assume that A and B are disjoint. (If they are not, we can replace B by $B - A$, because $A \cap (B - A) = \emptyset$ and $A \cup (B - A) = A \cup B$.) Furthermore, without loss of generality, if one of the two sets is countably infinite and other finite, we can assume that B is the one that is finite.

There are three cases to consider: (i) A and B are both finite, (ii) A is infinite and B is finite, and (iii) A and B are both countably infinite.

Case (i): Note that when A and B are finite, $A \cup B$ is also finite, and therefore, countable.

Case (ii): Because A is countably infinite, its elements can be listed in an infinite sequence $a_1, a_2, a_3, \dots, a_n, \dots$ and because B is finite, its terms can be listed as b_1, b_2, \dots, b_m for some positive integer m . We can list the elements of $A \cup B$ as $b_1, b_2, \dots, b_m, a_1, a_2, a_3, \dots, a_n, \dots$. This means that $A \cup B$ is countably infinite.

Case (iii): Because both A and B are countably infinite, we can list their elements as $a_1, a_2, a_3, \dots, a_n, \dots$ and $b_1, b_2, b_3, \dots, b_n, \dots$, respectively. By alternating terms of these two sequences we can list the elements of $A \cup B$ in the infinite sequence $a_1, b_1, a_2, b_2, a_3, b_3, \dots, a_n, b_n, \dots$. This means $A \cup B$ must be countably infinite.

We have completed the proof, as we have shown that $A \cup B$ is countable in all three cases. 

Because of its importance, we now state a key theorem in the study of cardinality.

THEOREM 2

SCHRÖDER-BERNSTEIN THEOREM If A and B are sets with $|A| \leq |B|$ and $|B| \leq |A|$, then $|A| = |B|$. In other words, if there are one-to-one functions f from A to B and g from B to A , then there is a one-to-one correspondence between A and B .

Because Theorem 2 seems to be quite straightforward, we might expect that it has an easy proof. However, even though it can be proved without using advanced mathematics, no known proof is easy to explain. Consequently, we omit a proof here. We refer the interested reader to [AiZiHo09] and [Ve06] for a proof. This result is called the Schröder-Bernstein theorem after Ernst Schröder who published a flawed proof of it in 1898 and Felix Bernstein, a student of Georg Cantor, who presented a proof in 1897. However, a proof of this theorem was found in notes of Richard Dedekind dated 1887. Dedekind was a German mathematician who made important contributions to the foundations of mathematics, abstract algebra, and number theory.

We illustrate the use of Theorem 2 with an example.

EXAMPLE 6 Show that the $|(0, 1)| = |(0, 1]|$.

Solution: It is not at all obvious how to find a one-to-one correspondence between $(0, 1)$ and $(0, 1]$ to show that $|(0, 1)| = |(0, 1]|$. Fortunately, we can use the Schröder-Bernstein theorem instead. Finding a one-to-one function from $(0, 1)$ to $(0, 1]$ is simple. Because $(0, 1) \subset (0, 1]$, $f(x) = x$ is a one-to-one function from $(0, 1)$ to $(0, 1]$. Finding a one-to-one function from $(0, 1]$ to $(0, 1)$ is also not difficult. The function $g(x) = x/2$ is clearly one-to-one and maps $(0, 1]$ to $(0, 1/2] \subset (0, 1)$. As we have found one-to-one functions from $(0, 1)$ to $(0, 1]$ and from $(0, 1]$ to $(0, 1)$, the Schröder-Bernstein theorem tells us that $|(0, 1)| = |(0, 1]|$. \blacktriangleleft

UNCOMPUTABLE FUNCTIONS We will now describe an important application of the concepts of this section to computer science. In particular, we will show that there are functions whose values cannot be computed by any computer program.

DEFINITION 4

We say that a function is **computable** if there is a computer program in some programming language that finds the values of this function. If a function is not computable we say it is **uncomputable**.

To show that there are uncomputable functions, we need to establish two results. First, we need to show that the set of all computer programs in any particular programming language is countable. This can be proved by noting that a computer programs in a particular language can be thought of as a string of characters from a finite alphabet (see Exercise 37). Next, we show that there are uncountably many different functions from a particular countably infinite set to itself. In particular, Exercise 38 shows that the set of functions from the set of positive integers to itself is uncountable. This is a consequence of the uncountability of the real numbers between 0 and 1 (see Example 5). Putting these two results together (Exercise 39) shows that there are uncomputable functions.

THE CONTINUUM HYPOTHESIS We conclude this section with a brief discussion of a famous open question about cardinality. It can be shown that the power set of \mathbf{Z}^+ and the set of real numbers \mathbf{R} have the same cardinality (see Exercise 38). In other words, we know that $|\mathcal{P}(\mathbf{Z}^+)| = |\mathbf{R}| = \mathfrak{c}$, where \mathfrak{c} denotes the cardinality of the set of real numbers.

An important theorem of Cantor (Exercise 40) states that the cardinality of a set is always less than the cardinality of its power set. Hence, $|\mathbf{Z}^+| < |\mathcal{P}(\mathbf{Z}^+)|$. We can rewrite this as $\aleph_0 < 2^{\aleph_0}$, using the notation $2^{|S|}$ to denote the cardinality of the power set of the set S . Also, note that the relationship $|\mathcal{P}(\mathbf{Z}^+)| = |\mathbf{R}|$ can be expressed as $2^{\aleph_0} = \mathfrak{c}$.

This leads us to the famous **continuum hypothesis**, which asserts that there is no cardinal number X between \aleph_0 and \mathfrak{c} . In other words, the continuum hypothesis states that there is no set A such that \aleph_0 , the cardinality of the set of positive integers, is less than $|A|$ and $|A|$ is less than \mathfrak{c} , the cardinality of the set of real numbers. It can be shown that the smallest infinite cardinal numbers form an infinite sequence $\aleph_0 < \aleph_1 < \aleph_2 < \dots$. If we assume that the continuum hypothesis is true, it would follow that $\mathfrak{c} = \aleph_1$, so that $2^{\aleph_0} = \aleph_1$.

\mathfrak{c} is the lowercase Fraktur c .

The continuum hypothesis was stated by Cantor in 1877. He labored unsuccessfully to prove it, becoming extremely dismayed that he could not. By 1900, settling the continuum hypothesis was considered to be among the most important unsolved problems in mathematics. It was the first problem posed by David Hilbert in his famous 1900 list of open problems in mathematics.

The continuum hypothesis is still an open question and remains an area for active research. However, it has been shown that it can be neither proved nor disproved under the standard set theory axioms in modern mathematics, the Zermelo-Fraenkel axioms. The Zermelo-Fraenkel axioms were formulated to avoid the paradoxes of naive set theory, such as Russell's paradox, but there is much controversy whether they should be replaced by some other set of axioms for set theory.

Exercises

1. Determine whether each of these sets is finite, countably infinite, or uncountable. For those that are countably infinite, exhibit a one-to-one correspondence between the set of positive integers and that set.
 - a) the negative integers
 - b) the even integers
 - c) the integers less than 100
 - d) the real numbers between 0 and $\frac{1}{2}$
 - e) the positive integers less than 1,000,000,000
 - f) the integers that are multiples of 7
2. Determine whether each of these sets is finite, countably infinite, or uncountable. For those that are countably infinite, exhibit a one-to-one correspondence between the set of positive integers and that set.
 - a) the integers greater than 10
 - b) the odd negative integers
 - c) the integers with absolute value less than 1,000,000
 - d) the real numbers between 0 and 2
 - e) the set $A \times \mathbf{Z}^+$ where $A = \{2, 3\}$
 - f) the integers that are multiples of 10
3. Determine whether each of these sets is countable or uncountable. For those that are countably infinite, exhibit a one-to-one correspondence between the set of positive integers and that set.
 - a) all bit strings not containing the bit 0
 - b) all positive rational numbers that cannot be written with denominators less than 4
 - c) the real numbers not containing 0 in their decimal representation
 - d) the real numbers containing only a finite number of 1s in their decimal representation
4. Determine whether each of these sets is countable or uncountable. For those that are countably infinite, exhibit a one-to-one correspondence between the set of positive integers and that set.
 - a) integers not divisible by 3
 - b) integers divisible by 5 but not by 7
 - c) the real numbers with decimal representations consisting of all 1s
 - d) the real numbers with decimal representations of all 1s or 9s
5. Show that a finite group of guests arriving at Hilbert's fully occupied Grand Hotel can be given rooms without evicting any current guest.
6. Suppose that Hilbert's Grand Hotel is fully occupied, but the hotel closes all the even numbered rooms for maintenance. Show that all guests can remain in the hotel.
7. Suppose that Hilbert's Grand Hotel is fully occupied on the day the hotel expands to a second building which also contains a countably infinite number of rooms. Show that the current guests can be spread out to fill every room of the two buildings of the hotel.
8. Show that a countably infinite number of guests arriving at Hilbert's fully occupied Grand Hotel can be given rooms without evicting any current guest.
- *9. Suppose that a countably infinite number of buses, each containing a countably infinite number of guests, arrive at Hilbert's fully occupied Grand Hotel. Show that all the arriving guests can be accommodated without evicting any current guest.
10. Give an example of two uncountable sets A and B such that $A - B$ is
 - a) finite.
 - b) countably infinite.
 - c) uncountable.
11. Give an example of two uncountable sets A and B such that $A \cap B$ is
 - a) finite.
 - b) countably infinite.
 - c) uncountable.
12. Show that if A and B are sets and $A \subset B$ then $|A| \leq |B|$.
13. Explain why the set A is countable if and only if $|A| \leq |\mathbf{Z}^+|$.
14. Show that if A and B are sets with the same cardinality, then $|A| \leq |B|$ and $|B| \leq |A|$.
15. Show that if A and B are sets, A is uncountable, and $A \subseteq B$, then B is uncountable.
16. Show that a subset of a countable set is also countable.
17. If A is an uncountable set and B is a countable set, must $A - B$ be uncountable?

- 18.** Show that if A and B are sets $|A| = |B|$, then $|\mathcal{P}(A)| = |\mathcal{P}(B)|$.
- 19.** Show that if A, B, C , and D are sets with $|A| = |B|$ and $|C| = |D|$, then $|A \times C| = |B \times D|$.
- 20.** Show that if $|A| = |B|$ and $|B| = |C|$, then $|A| = |C|$.
- 21.** Show that if A, B , and C are sets such that $|A| \leq |B|$ and $|B| \leq |C|$, then $|A| \leq |C|$.
- 22.** Suppose that A is a countable set. Show that the set B is also countable if there is an onto function f from A to B .
- 23.** Show that if A is an infinite set, then it contains a countably infinite subset.
- 24.** Show that there is no infinite set A such that $|A| < |\mathbf{Z}^+| = \aleph_0$.
- 25.** Prove that if it is possible to label each element of an infinite set S with a finite string of keyboard characters, from a finite list characters, where no two elements of S have the same label, then S is a countably infinite set.
- 26.** Use Exercise 25 to provide a proof different from that in the text that the set of rational numbers is countable. [Hint: Show that you can express a rational number as a string of digits with a slash and possibly a minus sign.]
- *27.** Show that the union of a countable number of countable sets is countable.
- 28.** Show that the set $\mathbf{Z}^+ \times \mathbf{Z}^+$ is countable.
- *29.** Show that the set of all finite bit strings is countable.
- *30.** Show that the set of real numbers that are solutions of quadratic equations $ax^2 + bx + c = 0$, where a, b , and c are integers, is countable.
- *31.** Show that $\mathbf{Z}^+ \times \mathbf{Z}^+$ is countable by showing that the polynomial function $f : \mathbf{Z}^+ \times \mathbf{Z}^+ \rightarrow \mathbf{Z}^+$ with $f(m, n) = (m+n-2)(m+n-1)/2 + m$ is one-to-one and onto.
- *32.** Show that when you substitute $(3n+1)^2$ for each occurrence of n and $(3m+1)^2$ for each occurrence of m in the right-hand side of the formula for the function $f(m, n)$ in Exercise 31, you obtain a one-to-one polynomial function $\mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z}$. It is an open question whether there is a one-to-one polynomial function $\mathbf{Q} \times \mathbf{Q} \rightarrow \mathbf{Q}$.
- 33.** Use the Schröder-Bernstein theorem to show that $(0, 1)$ and $[0, 1]$ have the same cardinality
- 34.** Show that $(0, 1)$ and \mathbf{R} have the same cardinality. [Hint: Use the Schröder-Bernstein theorem.]
- 35.** Show that there is no one-to-one correspondence from the set of positive integers to the power set of the set of positive integers. [Hint: Assume that there is such a one-to-one correspondence. Represent a subset of the set of positive integers as an infinite bit string with i th bit 1 if i belongs to the subset and 0 otherwise. Suppose that you can list these infinite strings in a sequence indexed by the positive integers. Construct a new bit string with its i th bit equal to the complement of the i th bit of the i th string in the list. Show that this new bit string cannot appear in the list.]
- *36.** Show that there is a one-to-one correspondence from the set of subsets of the positive integers to the set real numbers between 0 and 1. Use this result and Exercises 34 and 35 to conclude that $\aleph_0 < |\mathcal{P}(\mathbf{Z}^+)| = |\mathbf{R}|$. [Hint: Look at the first part of the hint for Exercise 35.]
- *37.** Show that the set of all computer programs in a particular programming language is countable. [Hint: A computer program written in a programming language can be thought of as a string of symbols from a finite alphabet.]
- *38.** Show that the set of functions from the positive integers to the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ is uncountable. [Hint: First set up a one-to-one correspondence between the set of real numbers between 0 and 1 and a subset of these functions. Do this by associating to the real number $0.d_1d_2\dots d_n\dots$ the function f with $f(n) = d_n$.]
- *39.** We say that a function is **computable** if there is a computer program that finds the values of this function. Use Exercises 37 and 38 to show that there are functions that are not computable.
- *40.** Show that if S is a set, then there does not exist an onto function f from S to $\mathcal{P}(S)$, the power set of S . Conclude that $|S| < |\mathcal{P}(S)|$. This result is known as **Cantor's theorem**. [Hint: Suppose such a function f existed. Let $T = \{s \in S \mid s \notin f(s)\}$ and show that no element s can exist for which $f(s) = T$.]

2.6 Matrices

Introduction

Matrices are used throughout discrete mathematics to express relationships between elements in sets. In subsequent chapters we will use matrices in a wide variety of models. For instance, matrices will be used in models of communications networks and transportation systems. Many algorithms will be developed that use these matrix models. This section reviews matrix arithmetic that will be used in these algorithms.

DEFINITION 1

A *matrix* is a rectangular array of numbers. A matrix with m rows and n columns is called an $m \times n$ matrix. The plural of matrix is *matrices*. A matrix with the same number of rows as columns is called *square*. Two matrices are *equal* if they have the same number of rows and the same number of columns and the corresponding entries in every position are equal.

EXAMPLE 1 The matrix $\begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$ is a 3×2 matrix. 

We now introduce some terminology about matrices. Boldface uppercase letters will be used to represent matrices.

DEFINITION 2

Let m and n be positive integers and let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

The i th *row* of \mathbf{A} is the $1 \times n$ matrix $[a_{i1}, a_{i2}, \dots, a_{in}]$. The j th *column* of \mathbf{A} is the $m \times 1$ matrix

$$\begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ \vdots \\ a_{mj} \end{bmatrix}.$$

The (i, j) th *element* or *entry* of \mathbf{A} is the element a_{ij} , that is, the number in the i th row and j th column of \mathbf{A} . A convenient shorthand notation for expressing the matrix \mathbf{A} is to write $\mathbf{A} = [a_{ij}]$, which indicates that \mathbf{A} is the matrix with its (i, j) th element equal to a_{ij} .

Matrix Arithmetic

The basic operations of matrix arithmetic will now be discussed, beginning with a definition of matrix addition.

DEFINITION 3

Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ be $m \times n$ matrices. The *sum* of \mathbf{A} and \mathbf{B} , denoted by $\mathbf{A} + \mathbf{B}$, is the $m \times n$ matrix that has $a_{ij} + b_{ij}$ as its (i, j) th element. In other words, $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]$.

The sum of two matrices of the same size is obtained by adding elements in the corresponding positions. Matrices of different sizes cannot be added, because the sum of two matrices is defined only when both matrices have the same number of rows and the same number of columns.

EXAMPLE 2

We have $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & -3 \\ 3 & 4 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 & -1 \\ 1 & -3 & 0 \\ -1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -2 \\ 3 & -1 & -3 \\ 2 & 5 & 2 \end{bmatrix}$. 

We now discuss matrix products. A product of two matrices is defined only when the number of columns in the first matrix equals the number of rows of the second matrix.

DEFINITION 4

Let \mathbf{A} be an $m \times k$ matrix and \mathbf{B} be a $k \times n$ matrix. The *product* of \mathbf{A} and \mathbf{B} , denoted by \mathbf{AB} , is the $m \times n$ matrix with its (i, j) th entry equal to the sum of the products of the corresponding elements from the i th row of \mathbf{A} and the j th column of \mathbf{B} . In other words, if $\mathbf{AB} = [c_{ij}]$, then

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}.$$

In Figure 1 the colored row of \mathbf{A} and the colored column of \mathbf{B} are used to compute the element c_{ij} of \mathbf{AB} . The product of two matrices is not defined when the number of columns in the first matrix and the number of rows in the second matrix are not the same.

We now give some examples of matrix products.

EXAMPLE 3

Let

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 4 \\ 2 & 1 & 1 \\ 3 & 1 & 0 \\ 0 & 2 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 2 & 4 \\ 1 & 1 \\ 3 & 0 \end{bmatrix}.$$

Find \mathbf{AB} if it is defined.



Solution: Because \mathbf{A} is a 4×3 matrix and \mathbf{B} is a 3×2 matrix, the product \mathbf{AB} is defined and is a 4×2 matrix. To find the elements of \mathbf{AB} , the corresponding elements of the rows of \mathbf{A} and the columns of \mathbf{B} are first multiplied and then these products are added. For instance, the element in the $(3, 1)$ th position of \mathbf{AB} is the sum of the products of the corresponding elements of the third row of \mathbf{A} and the first column of \mathbf{B} ; namely, $3 \cdot 2 + 1 \cdot 1 + 0 \cdot 3 = 7$. When all the elements of \mathbf{AB} are computed, we see that

$$\mathbf{AB} = \begin{bmatrix} 14 & 4 \\ 8 & 9 \\ 7 & 13 \\ 8 & 2 \end{bmatrix}.$$

Matrix multiplication is *not* commutative. That is, if \mathbf{A} and \mathbf{B} are two matrices, it is not necessarily true that \mathbf{AB} and \mathbf{BA} are the same. In fact, it may be that only one of these two products is defined. For instance, if \mathbf{A} is 2×3 and \mathbf{B} is 3×4 , then \mathbf{AB} is defined and is 2×4 ; however, \mathbf{BA} is not defined, because it is impossible to multiply a 3×4 matrix and a 2×3 matrix.

In general, suppose that \mathbf{A} is an $m \times n$ matrix and \mathbf{B} is an $r \times s$ matrix. Then \mathbf{AB} is defined only when $n = r$ and \mathbf{BA} is defined only when $s = m$. Moreover, even when \mathbf{AB} and \mathbf{BA} are

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ik} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mk} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ b_{k1} & b_{k2} & \dots & b_{kj} & \dots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

FIGURE 1 The Product of $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$.

both defined, they will not be the same size unless $m = n = r = s$. Hence, if both \mathbf{AB} and \mathbf{BA} are defined and are the same size, then both \mathbf{A} and \mathbf{B} must be square and of the same size. Furthermore, even with \mathbf{A} and \mathbf{B} both $n \times n$ matrices, \mathbf{AB} and \mathbf{BA} are not necessarily equal, as Example 4 demonstrates.

EXAMPLE 4 Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Does $\mathbf{AB} = \mathbf{BA}$?

Solution: We find that

$$\mathbf{AB} = \begin{bmatrix} 3 & 2 \\ 5 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{BA} = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}.$$

Hence, $\mathbf{AB} \neq \mathbf{BA}$. 

Transposes and Powers of Matrices

We now introduce an important matrix with entries that are zeros and ones.

DEFINITION 5

The *identity matrix of order n* is the $n \times n$ matrix $\mathbf{I}_n = [\delta_{ij}]$, where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Hence

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

Multiplying a matrix by an appropriately sized identity matrix does not change this matrix. In other words, when \mathbf{A} is an $m \times n$ matrix, we have

$$\mathbf{AI}_n = \mathbf{I}_m \mathbf{A} = \mathbf{A}.$$

Powers of square matrices can be defined. When \mathbf{A} is an $n \times n$ matrix, we have

$$\mathbf{A}^0 = \mathbf{I}_n, \quad \mathbf{A}^r = \underbrace{\mathbf{AAA}\cdots\mathbf{A}}_{r \text{ times}}.$$

The operation of interchanging the rows and columns of a square matrix arises in many contexts.

DEFINITION 6

Let $\mathbf{A} = [a_{ij}]$ be an $m \times n$ matrix. The *transpose* of \mathbf{A} , denoted by \mathbf{A}^t , is the $n \times m$ matrix obtained by interchanging the rows and columns of \mathbf{A} . In other words, if $\mathbf{A}^t = [b_{ij}]$, then $b_{ij} = a_{ji}$ for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$.

EXAMPLE 5

The transpose of the matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ is the matrix $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$.

Matrices that do not change when their rows and columns are interchanged are often important.

DEFINITION 7

A square matrix \mathbf{A} is called *symmetric* if $\mathbf{A} = \mathbf{A}^t$. Thus $\mathbf{A} = [a_{ij}]$ is symmetric if $a_{ij} = a_{ji}$ for all i and j with $1 \leq i \leq n$ and $1 \leq j \leq n$.

Note that a matrix is symmetric if and only if it is square and it is symmetric with respect to its main diagonal (which consists of entries that are in the i th row and i th column for some i). This symmetry is displayed in Figure 2.

EXAMPLE 6

The matrix $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ is symmetric.

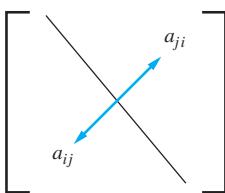


FIGURE 2 A Symmetric Matrix.

Zero–One Matrices

A matrix all of whose entries are either 0 or 1 is called a **zero–one matrix**. Zero–one matrices are often used to represent discrete structures, as we will see in Chapters 9 and 10. Algorithms using these structures are based on Boolean arithmetic with zero–one matrices. This arithmetic is based on the Boolean operations \wedge and \vee , which operate on pairs of bits, defined by

$$b_1 \wedge b_2 = \begin{cases} 1 & \text{if } b_1 = b_2 = 1 \\ 0 & \text{otherwise,} \end{cases}$$

$$b_1 \vee b_2 = \begin{cases} 1 & \text{if } b_1 = 1 \text{ or } b_2 = 1 \\ 0 & \text{otherwise.} \end{cases}$$

DEFINITION 8

Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ be $m \times n$ zero–one matrices. Then the *join* of \mathbf{A} and \mathbf{B} is the zero–one matrix with (i, j) th entry $a_{ij} \vee b_{ij}$. The join of \mathbf{A} and \mathbf{B} is denoted by $\mathbf{A} \vee \mathbf{B}$. The *meet* of \mathbf{A} and \mathbf{B} is the zero–one matrix with (i, j) th entry $a_{ij} \wedge b_{ij}$. The meet of \mathbf{A} and \mathbf{B} is denoted by $\mathbf{A} \wedge \mathbf{B}$.

EXAMPLE 7

Find the join and meet of the zero–one matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

Solution: We find that the join of \mathbf{A} and \mathbf{B} is

$$\mathbf{A} \vee \mathbf{B} = \begin{bmatrix} 1 \vee 0 & 0 \vee 1 & 1 \vee 0 \\ 0 \vee 1 & 1 \vee 1 & 0 \vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

The meet of \mathbf{A} and \mathbf{B} is

$$\mathbf{A} \wedge \mathbf{B} = \begin{bmatrix} 1 \wedge 0 & 0 \wedge 1 & 1 \wedge 0 \\ 0 \wedge 1 & 1 \wedge 1 & 0 \wedge 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad \blacktriangleleft$$

We now define the **Boolean product** of two matrices.

DEFINITION 9

Let $\mathbf{A} = [a_{ij}]$ be an $m \times k$ zero–one matrix and $\mathbf{B} = [b_{ij}]$ be a $k \times n$ zero–one matrix. Then the *Boolean product* of \mathbf{A} and \mathbf{B} , denoted by $\mathbf{A} \odot \mathbf{B}$, is the $m \times n$ matrix with (i, j) th entry c_{ij} where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \cdots \vee (a_{ik} \wedge b_{kj}).$$

Note that the Boolean product of \mathbf{A} and \mathbf{B} is obtained in an analogous way to the ordinary product of these matrices, but with addition replaced with the operation \vee and with multiplication replaced with the operation \wedge . We give an example of the Boolean products of matrices.

EXAMPLE 8 Find the Boolean product of \mathbf{A} and \mathbf{B} , where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Solution: The Boolean product $\mathbf{A} \odot \mathbf{B}$ is given by

$$\begin{aligned} \mathbf{A} \odot \mathbf{B} &= \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \\ (0 \wedge 1) \vee (1 \wedge 0) & (0 \wedge 1) \vee (1 \wedge 1) & (0 \wedge 0) \vee (1 \wedge 1) \\ (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) & (1 \wedge 0) \vee (0 \wedge 1) \end{bmatrix} \\ &= \begin{bmatrix} 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \\ 0 \vee 0 & 0 \vee 1 & 0 \vee 1 \\ 1 \vee 0 & 1 \vee 0 & 0 \vee 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad \blacktriangleleft \end{aligned}$$

We can also define the Boolean powers of a square zero–one matrix. These powers will be used in our subsequent studies of paths in graphs, which are used to model such things as communications paths in computer networks.

DEFINITION 10

Let \mathbf{A} be a square zero–one matrix and let r be a positive integer. The r th Boolean power of \mathbf{A} is the Boolean product of r factors of \mathbf{A} . The r th Boolean product of \mathbf{A} is denoted by $\mathbf{A}^{[r]}$. Hence

$$\mathbf{A}^{[r]} = \underbrace{\mathbf{A} \odot \mathbf{A} \odot \mathbf{A} \odot \cdots \odot \mathbf{A}}_{r \text{ times}}.$$

(This is well defined because the Boolean product of matrices is associative.) We also define $\mathbf{A}^{[0]}$ to be \mathbf{I}_n .

EXAMPLE 9 Let $\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$. Find $\mathbf{A}^{[n]}$ for all positive integers n .

Solution: We find that

$$\mathbf{A}^{[2]} = \mathbf{A} \odot \mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

We also find that

$$\mathbf{A}^{[3]} = \mathbf{A}^{[2]} \odot \mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{A}^{[4]} = \mathbf{A}^{[3]} \odot \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Additional computation shows that

$$\mathbf{A}^{[5]} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The reader can now see that $\mathbf{A}^{[n]} = \mathbf{A}^{[5]}$ for all positive integers n with $n \geq 5$. 

Exercises

1. Let $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 3 \\ 2 & 0 & 4 & 6 \\ 1 & 1 & 3 & 7 \end{bmatrix}$.

- a) What size is \mathbf{A} ?
- b) What is the third column of \mathbf{A} ?
- c) What is the second row of \mathbf{A} ?
- d) What is the element of \mathbf{A} in the (3, 2)th position?
- e) What is \mathbf{A}^t ?

2. Find $\mathbf{A} + \mathbf{B}$, where

a) $\mathbf{A} = \begin{bmatrix} 1 & 0 & 4 \\ -1 & 2 & 2 \\ 0 & -2 & -3 \end{bmatrix}$,

$\mathbf{B} = \begin{bmatrix} -1 & 3 & 5 \\ 2 & 2 & -3 \\ 2 & -3 & 0 \end{bmatrix}$.

b) $\mathbf{A} = \begin{bmatrix} -1 & 0 & 5 & 6 \\ -4 & -3 & 5 & -2 \end{bmatrix}$,

$\mathbf{B} = \begin{bmatrix} -3 & 9 & -3 & 4 \\ 0 & -2 & -1 & 2 \end{bmatrix}$.

3. Find \mathbf{AB} if

a) $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 & 4 \\ 1 & 3 \end{bmatrix}$.

b) $\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 2 & 3 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 3 & -2 & -1 \\ 1 & 0 & 2 \end{bmatrix}$.

c) $\mathbf{A} = \begin{bmatrix} 4 & -3 \\ 3 & -1 \\ 0 & -2 \\ -1 & 5 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} -1 & 3 & 2 & -2 \\ 0 & -1 & 4 & -3 \end{bmatrix}$.

4. Find the product \mathbf{AB} , where

a) $\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & -1 \\ -1 & 1 & 0 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 & 1 & -1 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$.

b) $\mathbf{A} = \begin{bmatrix} 1 & -3 & 0 \\ 1 & 2 & 2 \\ 2 & 1 & -1 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 & -1 & 2 & 3 \\ -1 & 0 & 3 & -1 \\ -3 & -2 & 0 & 2 \end{bmatrix}$.

c) $\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 7 & 2 \\ -4 & -3 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 4 & -1 & 2 & 3 & 0 \\ -2 & 0 & 3 & 4 & 1 \end{bmatrix}$.

5. Find a matrix \mathbf{A} such that

$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \mathbf{A} = \begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}.$$

[Hint: Finding \mathbf{A} requires that you solve systems of linear equations.]

6. Find a matrix \mathbf{A} such that

$$\begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 1 \\ 4 & 0 & 3 \end{bmatrix} \mathbf{A} = \begin{bmatrix} 7 & 1 & 3 \\ 1 & 0 & 3 \\ -1 & -3 & 7 \end{bmatrix}.$$

7. Let \mathbf{A} be an $m \times n$ matrix and let $\mathbf{0}$ be the $m \times n$ matrix that has all entries equal to zero. Show that $\mathbf{A} + \mathbf{0} = \mathbf{A}$.

8. Show that matrix addition is commutative; that is, show that if \mathbf{A} and \mathbf{B} are both $m \times n$ matrices, then $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.

9. Show that matrix addition is associative; that is, show that if \mathbf{A} , \mathbf{B} , and \mathbf{C} are all $m \times n$ matrices, then $\mathbf{A} + (\mathbf{B} + \mathbf{C}) = (\mathbf{A} + \mathbf{B}) + \mathbf{C}$.

10. Let \mathbf{A} be a 3×4 matrix, \mathbf{B} be a 4×5 matrix, and \mathbf{C} be a 4×4 matrix. Determine which of the following products are defined and find the size of those that are defined.

- a) \mathbf{AB} b) \mathbf{BA} c) \mathbf{AC}
 d) \mathbf{CA} e) \mathbf{BC} f) \mathbf{CB}

11. What do we know about the sizes of the matrices \mathbf{A} and \mathbf{B} if both of the products \mathbf{AB} and \mathbf{BA} are defined?

12. In this exercise we show that matrix multiplication is distributive over matrix addition.

- a) Suppose that \mathbf{A} and \mathbf{B} are $m \times k$ matrices and that \mathbf{C} is a $k \times n$ matrix. Show that $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$.
 b) Suppose that \mathbf{C} is an $m \times k$ matrix and that \mathbf{A} and \mathbf{B} are $k \times n$ matrices. Show that $\mathbf{C}(\mathbf{A} + \mathbf{B}) = \mathbf{CA} + \mathbf{CB}$.

13. In this exercise we show that matrix multiplication is associative. Suppose that \mathbf{A} is an $m \times p$ matrix, \mathbf{B} is a $p \times k$ matrix, and \mathbf{C} is a $k \times n$ matrix. Show that $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$.

14. The $n \times n$ matrix $\mathbf{A} = [a_{ij}]$ is called a **diagonal matrix** if $a_{ij} = 0$ when $i \neq j$. Show that the product of two $n \times n$ diagonal matrices is again a diagonal matrix. Give a simple rule for determining this product.

15. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Find a formula for \mathbf{A}^n , whenever n is a positive integer.

16. Show that $(\mathbf{A}^t)^t = \mathbf{A}$.

17. Let \mathbf{A} and \mathbf{B} be two $n \times n$ matrices. Show that

- a) $(\mathbf{A} + \mathbf{B})^t = \mathbf{A}^t + \mathbf{B}^t$.
 b) $(\mathbf{AB})^t = \mathbf{B}^t \mathbf{A}^t$.

If \mathbf{A} and \mathbf{B} are $n \times n$ matrices with $\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$, then \mathbf{B} is called the **inverse** of \mathbf{A} (this terminology is appropriate because such a matrix \mathbf{B} is unique) and \mathbf{A} is said to be **invertible**. The notation $\mathbf{B} = \mathbf{A}^{-1}$ denotes that \mathbf{B} is the inverse of \mathbf{A} .

18. Show that

$$\begin{bmatrix} 2 & 3 & -1 \\ 1 & 2 & 1 \\ -1 & -1 & 3 \end{bmatrix}$$

is the inverse of

$$\begin{bmatrix} 7 & -8 & 5 \\ -4 & 5 & -3 \\ 1 & -1 & 1 \end{bmatrix}.$$

19. Let \mathbf{A} be the 2×2 matrix

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Show that if $ad - bc \neq 0$, then

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{d}{ad - bc} & \frac{-b}{ad - bc} \\ \frac{-c}{ad - bc} & \frac{a}{ad - bc} \end{bmatrix}.$$

20. Let

$$\mathbf{A} = \begin{bmatrix} -1 & 2 \\ 1 & 3 \end{bmatrix}.$$

- a) Find \mathbf{A}^{-1} . [Hint: Use Exercise 19.]
 b) Find \mathbf{A}^3 .
 c) Find $(\mathbf{A}^{-1})^3$.
 d) Use your answers to (b) and (c) to show that $(\mathbf{A}^{-1})^3$ is the inverse of \mathbf{A}^3 .
 21. Let \mathbf{A} be an invertible matrix. Show that $(\mathbf{A}^n)^{-1} = (\mathbf{A}^{-1})^n$ whenever n is a positive integer.
 22. Let \mathbf{A} be a matrix. Show that the matrix \mathbf{AA}^t is symmetric. [Hint: Show that this matrix equals its transpose with the help of Exercise 17b.]
 23. Suppose that \mathbf{A} is an $n \times n$ matrix where n is a positive integer. Show that $\mathbf{A} + \mathbf{A}^t$ is symmetric.

24. a) Show that the system of simultaneous linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n. \end{aligned}$$

in the variables x_1, x_2, \dots, x_n can be expressed as $\mathbf{AX} = \mathbf{B}$, where $\mathbf{A} = [a_{ij}]$, \mathbf{X} is an $n \times 1$ matrix with x_i the entry in its i th row, and \mathbf{B} is an $n \times 1$ matrix with b_i the entry in its i th row.

b) Show that if the matrix $\mathbf{A} = [a_{ij}]$ is invertible (as defined in the preamble to Exercise 18), then the solution of the system in part (a) can be found using the equation $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$.

25. Use Exercises 18 and 24 to solve the system

$$\begin{aligned} 7x_1 - 8x_2 + 5x_3 &= 5 \\ -4x_1 + 5x_2 - 3x_3 &= -3 \\ x_1 - x_2 + x_3 &= 0 \end{aligned}$$

26. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Find

a) $\mathbf{A} \vee \mathbf{B}$. b) $\mathbf{A} \wedge \mathbf{B}$. c) $\mathbf{A} \odot \mathbf{B}$.

27. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

Find

a) $\mathbf{A} \vee \mathbf{B}$. b) $\mathbf{A} \wedge \mathbf{B}$. c) $\mathbf{A} \odot \mathbf{B}$.

Key Terms and Results

TERMS

set: a collection of distinct objects

axiom: a basic assumption of a theory

paradox: a logical inconsistency

element, member of a set: an object in a set

roster method: a method that describes a set by listing its elements

set builder notation: the notation that describes a set by stating a property an element must have to be a member

\emptyset (empty set, null set): the set with no members

universal set: the set containing all objects under consideration

Venn diagram: a graphical representation of a set or sets

$S = T$ (set equality): S and T have the same elements

28. Find the Boolean product of \mathbf{A} and \mathbf{B} , where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

29. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Find

a) $\mathbf{A}^{[2]}$. b) $\mathbf{A}^{[3]}$.
c) $\mathbf{A} \vee \mathbf{A}^{[2]} \vee \mathbf{A}^{[3]}$.

30. Let \mathbf{A} be a zero-one matrix. Show that

a) $\mathbf{A} \vee \mathbf{A} = \mathbf{A}$. b) $\mathbf{A} \wedge \mathbf{A} = \mathbf{A}$.

31. In this exercise we show that the meet and join operations are commutative. Let \mathbf{A} and \mathbf{B} be $m \times n$ zero-one matrices. Show that

a) $\mathbf{A} \vee \mathbf{B} = \mathbf{B} \vee \mathbf{A}$. b) $\mathbf{B} \wedge \mathbf{A} = \mathbf{A} \wedge \mathbf{B}$.

32. In this exercise we show that the meet and join operations are associative. Let \mathbf{A} , \mathbf{B} , and \mathbf{C} be $m \times n$ zero-one matrices. Show that

a) $(\mathbf{A} \vee \mathbf{B}) \vee \mathbf{C} = \mathbf{A} \vee (\mathbf{B} \vee \mathbf{C})$.
b) $(\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C} = \mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C})$.

33. We will establish distributive laws of the meet over the join operation in this exercise. Let \mathbf{A} , \mathbf{B} , and \mathbf{C} be $m \times n$ zero-one matrices. Show that

a) $\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C})$.
b) $\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})$.

34. Let \mathbf{A} be an $n \times n$ zero-one matrix. Let \mathbf{I} be the $n \times n$ identity matrix. Show that $\mathbf{A} \odot \mathbf{I} = \mathbf{I} \odot \mathbf{A} = \mathbf{A}$.

35. In this exercise we will show that the Boolean product of zero-one matrices is associative. Assume that \mathbf{A} is an $m \times p$ zero-one matrix, \mathbf{B} is a $p \times k$ zero-one matrix, and \mathbf{C} is a $k \times n$ zero-one matrix. Show that $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$.

$S \subseteq T$ (S is a subset of T): every element of S is also an element of T

$S \subset T$ (S is a proper subset of T): S is a subset of T and $S \neq T$

finite set: a set with n elements, where n is a nonnegative integer

infinite set: a set that is not finite

$|S|$ (the cardinality of S): the number of elements in S

$P(S)$ (the power set of S): the set of all subsets of S

$A \cup B$ (the union of A and B): the set containing those elements that are in at least one of A and B

$A \cap B$ (the intersection of A and B): the set containing those elements that are in both A and B .

$A - B$ (the difference of A and B): the set containing those elements that are in A but not in B

\bar{A} (the complement of A): the set of elements in the universal set that are not in A

$A \oplus B$ (the symmetric difference of A and B): the set containing those elements in exactly one of A and B

membership table: a table displaying the membership of elements in sets

function from A to B : an assignment of exactly one element of B to each element of A

domain of f : the set A , where f is a function from A to B

codomain of f : the set B , where f is a function from A to B

b is the image of a under f : $b = f(a)$

a is a pre-image of b under f : $f(a) = b$

range of f : the set of images of f

onto function, surjection: a function from A to B such that every element of B is the image of some element in A

one-to-one function, injection: a function such that the images of elements in its domain are distinct

one-to-one correspondence, bijection: a function that is both one-to-one and onto

inverse of f : the function that reverses the correspondence given by f (when f is a bijection)

$f \circ g$ (composition of f and g): the function that assigns $f(g(x))$ to x

$\lfloor x \rfloor$ (floor function): the largest integer not exceeding x

$\lceil x \rceil$ (ceiling function): the smallest integer greater than or equal to x

partial function: an assignment to each element in a subset of the domain a unique element in the codomain

sequence: a function with domain that is a subset of the set of integers

geometric progression: a sequence of the form a, ar, ar^2, \dots , where a and r are real numbers

arithmetic progression: a sequence of the form $a, a + d, a + 2d, \dots$, where a and d are real numbers

string: a finite sequence

empty string: a string of length zero

recurrence relation: a equation that expresses the n th term a_n of a sequence in terms of one or more of the previous terms of the sequence for all integers n greater than a particular integer

$\sum_{i=1}^n a_i$: the sum $a_1 + a_2 + \dots + a_n$

$\prod_{i=1}^n a_i$: the product $a_1 a_2 \dots a_n$

cardinality: two sets A and B have the same cardinality if there is a one-to-one correspondence from A to B

countable set: a set that either is finite or can be placed in one-to-one correspondence with the set of positive integers

uncountable set: a set that is not countable

\aleph_0 (aleph null): the cardinality of a countable set

c : the cardinality of the set of real numbers

Cantor diagonalization argument: a proof technique used to show that the set of real numbers is uncountable

computable function: a function for which there is a computer program in some programming language that finds its values

uncomputable function: a function for which no computer program in a programming language exists that finds its values

continuum hypothesis: the statement there no set A exists such that $\aleph_0 < |A| < c$

matrix: a rectangular array of numbers

matrix addition: see page 178

matrix multiplication: see page 179

I_n (identity matrix of order n): the $n \times n$ matrix that has entries equal to 1 on its diagonal and 0s elsewhere

A^t (transpose of A): the matrix obtained from A by interchanging the rows and columns

symmetric matrix: a matrix is symmetric if it equals its transpose

zero-one matrix: a matrix with each entry equal to either 0 or 1

$A \vee B$ (the join of A and B): see page 181

$A \wedge B$ (the meet of A and B): see page 181

$A \odot B$ (the Boolean product of A and B): see page 182

RESULTS

The set identities given in Table 1 in Section 2.2

The summation formulae in Table 2 in Section 2.4

The set of rational numbers is countable.

The set of real numbers is uncountable.

Review Questions

1. Explain what it means for one set to be a subset of another set. How do you prove that one set is a subset of another set?
2. What is the empty set? Show that the empty set is a subset of every set.
3. a) Define $|S|$, the cardinality of the set S .
b) Give a formula for $|A \cup B|$, where A and B are sets.
4. a) Define the power set of a set S .
b) When is the empty set in the power set of a set S ?
c) How many elements does the power set of a set S with n elements have?
5. a) Define the union, intersection, difference, and symmetric difference of two sets.
b) What are the union, intersection, difference, and symmetric difference of the set of positive integers and the set of odd integers?
6. a) Explain what it means for two sets to be equal.
b) Describe as many of the ways as you can to show that two sets are equal.
c) Show in at least two different ways that the sets $A - (B \cap C)$ and $(A - B) \cup (A - C)$ are equal.

7. Explain the relationship between logical equivalences and set identities.
8.
 - a) Define the domain, codomain, and range of a function.
 - b) Let $f(n)$ be the function from the set of integers to the set of integers such that $f(n) = n^2 + 1$. What are the domain, codomain, and range of this function?
9.
 - a) Define what it means for a function from the set of positive integers to the set of positive integers to be one-to-one.
 - b) Define what it means for a function from the set of positive integers to the set of positive integers to be onto.
 - c) Give an example of a function from the set of positive integers to the set of positive integers that is both one-to-one and onto.
 - d) Give an example of a function from the set of positive integers to the set of positive integers that is one-to-one but not onto.
 - e) Give an example of a function from the set of positive integers to the set of positive integers that is not one-to-one but is onto.
 - f) Give an example of a function from the set of positive integers to the set of positive integers that is neither one-to-one nor onto.
10.
 - a) Define the inverse of a function.
 - b) When does a function have an inverse?
 - c) Does the function $f(n) = 10 - n$ from the set of integers to the set of integers have an inverse? If so, what is it?
11.
 - a) Define the floor and ceiling functions from the set of real numbers to the set of integers.
 - b) For which real numbers x is it true that $\lfloor x \rfloor = \lceil x \rceil$?
12. Conjecture a formula for the terms of the sequence that begins 8, 14, 32, 86, 248 and find the next three terms of your sequence.
13. Suppose that $a_n = a_{n-1} - 5$ for $n = 1, 2, \dots$. Find a formula for a_n .
14. What is the sum of the terms of the geometric progression $a + ar + \dots + ar^n$ when $r \neq 1$?
15. Show that the set of odd integers is countable.
16. Give an example of an uncountable set.
17. Define the product of two matrices **A** and **B**. When is this product defined?
18. Show that matrix multiplication is not commutative.

Supplementary Exercises

1. Let A be the set of English words that contain the letter x , and let B be the set of English words that contain the letter q . Express each of these sets as a combination of A and B .
 - a) The set of English words that do not contain the letter x .
 - b) The set of English words that contain both an x and a q .
 - c) The set of English words that contain an x but not a q .
 - d) The set of English words that do not contain either an x or a q .
 - e) The set of English words that contain an x or a q , but not both.
2. Show that if A is a subset of B , then the power set of A is a subset of the power set of B .
3. Suppose that A and B are sets such that the power set of A is a subset of the power set of B . Does it follow that A is a subset of B ?
4. Let **E** denote the set of even integers and **O** denote the set of odd integers. As usual, let **Z** denote the set of all integers. Determine each of these sets.
 - a) $\mathbf{E} \cup \mathbf{O}$
 - b) $\mathbf{E} \cap \mathbf{O}$
 - c) $\mathbf{Z} - \mathbf{E}$
 - d) $\mathbf{Z} - \mathbf{O}$
5. Show that if A and B are sets, then $A - (A - B) = A \cap B$.
6. Let A and B be sets. Show that $A \subseteq B$ if and only if $A \cap B = A$.
7. Let A , B , and C be sets. Show that $(A - B) - C$ is not necessarily equal to $A - (B - C)$.
8. Suppose that A , B , and C are sets. Prove or disprove that $(A - B) - C = (A - C) - B$.
9. Suppose that A , B , C , and D are sets. Prove or disprove that $(A - B) - (C - D) = (A - C) - (B - D)$.
10. Show that if A and B are finite sets, then $|A \cap B| \leq |A \cup B|$. Determine when this relationship is an equality.
11. Let A and B be sets in a finite universal set U . List the following in order of increasing size.
 - a) $|A|, |A \cup B|, |A \cap B|, |U|, |\emptyset|$
 - b) $|A - B|, |A \oplus B|, |A| + |B|, |A \cup B|, |\emptyset|$
12. Let A and B be subsets of the finite universal set U . Show that $|\overline{A} \cap \overline{B}| = |U| - |A| - |B| + |A \cap B|$.
13. Let f and g be functions from $\{1, 2, 3, 4\}$ to $\{a, b, c, d\}$ and from $\{a, b, c, d\}$ to $\{1, 2, 3, 4\}$, respectively, with $f(1) = d$, $f(2) = c$, $f(3) = a$, and $f(4) = b$, and $g(a) = 2$, $g(b) = 1$, $g(c) = 3$, and $g(d) = 2$.
 - a) Is f one-to-one? Is g one-to-one?
 - b) Is f onto? Is g onto?
 - c) Does either f or g have an inverse? If so, find this inverse.
14. Suppose that f is a function from A to B where A and B are finite sets. Explain why $|f(S)| \leq |S|$ for all subsets S of A .

- 15.** Suppose that f is a function from A to B where A and B are finite sets. Explain why $|f(S)| = |S|$ for all subsets S of A if and only if f is one-to-one.

Suppose that f is a function from A to B . We define the function S_f from $\mathcal{P}(A)$ to $\mathcal{P}(B)$ by the rule $S_f(X) = f(X)$ for each subset X of A . Similarly, we define the function $S_{f^{-1}}$ from $\mathcal{P}(B)$ to $\mathcal{P}(A)$ by the rule $S_{f^{-1}}(Y) = f^{-1}(Y)$ for each subset Y of B . Here, we are using Definition 4, and the definition of the inverse image of a set found in the preamble to Exercise 42, both in Section 2.3.

- *16.** Suppose that f is a function from the set A to the set B . Prove that

- a) if f is one-to-one, then S_f is a one-to-one function from $\mathcal{P}(A)$ to $\mathcal{P}(B)$.
- b) if f is onto function, then S_f is an onto function from $\mathcal{P}(A)$ to $\mathcal{P}(B)$.
- c) if f is onto function, then $S_{f^{-1}}$ is a one-to-one function from $\mathcal{P}(B)$ to $\mathcal{P}(A)$.
- d) if f is one-to-one, then $S_{f^{-1}}$ is an onto function from $\mathcal{P}(B)$ to $\mathcal{P}(A)$.
- e) if f is a one-to-one correspondence, then S_f is a one-to-one correspondence from $\mathcal{P}(A)$ to $\mathcal{P}(B)$ and $S_{f^{-1}}$ is a one-to-one correspondence from $\mathcal{P}(B)$ to $\mathcal{P}(A)$.

[Hint: Use parts (a)-(d).]

- 17.** Prove that if f and g are functions from A to B and $S_f = S_g$ (using the definition in the preamble to Exercise 16), then $f(x) = g(x)$ for all $x \in A$.

- 18.** Show that if n is an integer, then $n = \lceil n/2 \rceil + \lfloor n/2 \rfloor$.

- 19.** For which real numbers x and y is it true that $\lfloor x+y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$?

- 20.** For which real numbers x and y is it true that $\lceil x+y \rceil = \lceil x \rceil + \lceil y \rceil$?

- 21.** For which real numbers x and y is it true that $\lceil x+y \rceil = \lceil x \rceil + \lfloor y \rfloor$?

- 22.** Prove that $\lfloor n/2 \rfloor \lceil n/2 \rceil = \lfloor n^2/4 \rfloor$ for all integers n .

- 23.** Prove that if m is an integer, then $\lfloor x \rfloor + \lfloor m-x \rfloor = m-1$, unless x is an integer, in which case, it equals m .

- 24.** Prove that if x is a real number, then $\lfloor \lfloor x/2 \rfloor/2 \rfloor = \lfloor x/4 \rfloor$.

- 25.** Prove that if n is an odd integer, then $\lceil n^2/4 \rceil = (n^2+3)/4$.

- 26.** Prove that if m and n are positive integers and x is a real number, then

$$\left\lfloor \frac{\lfloor x \rfloor + n}{m} \right\rfloor = \left\lfloor \frac{x+n}{m} \right\rfloor.$$

- *27.** Prove that if m is a positive integer and x is a real number, then

$$\begin{aligned} \lfloor mx \rfloor &= \lfloor x \rfloor + \left\lfloor x + \frac{1}{m} \right\rfloor + \left\lfloor x + \frac{2}{m} \right\rfloor + \dots \\ &\quad + \left\lfloor x + \frac{m-1}{m} \right\rfloor. \end{aligned}$$

- *28.** We define the **Ulam numbers** by setting $u_1 = 1$ and $u_2 = 2$. Furthermore, after determining whether the integers less than n are Ulam numbers, we set n equal to the next Ulam number if it can be written uniquely as the sum of two different Ulam numbers. Note that $u_3 = 3$, $u_4 = 4$, $u_5 = 6$, and $u_6 = 8$.

- a) Find the first 20 Ulam numbers.

- b) Prove that there are infinitely many Ulam numbers.

- 29.** Determine the value of $\prod_{k=1}^{100} \frac{k+1}{k}$. (The notation used here for products is defined in the preamble to Exercise 43 in Section 2.4.)

- *30.** Determine a rule for generating the terms of the sequence that begins $1, 3, 4, 8, 15, 27, 50, 92, \dots$, and find the next four terms of the sequence.

- *31.** Determine a rule for generating the terms of the sequence that begins $2, 3, 3, 5, 10, 13, 39, 43, 172, 177, 885, 891, \dots$, and find the next four terms of the sequence.

- 32.** Show that the set of irrational numbers is an uncountable set.

- 33.** Show that the set S is a countable set if there is a function f from S to the positive integers such that $f^{-1}(j)$ is countable whenever j is a positive integer.

- 34.** Show that the set of all finite subsets of the set of positive integers is a countable set.

- **35.** Show that $|\mathbf{R} \times \mathbf{R}| = |\mathbf{R}|$. [Hint: Use the Schröder-Bernstein theorem to show that $|(0, 1) \times (0, 1)| = |(0, 1)|$. To construct an injection from $(0, 1) \times (0, 1)$ to $(0, 1)$, suppose that $(x, y) \in (0, 1) \times (0, 1)$. Map (x, y) to the number with decimal expansion formed by alternating between the digits in the decimal expansions of x and y , which do not end with an infinite string of 9s.]

- **36.** Show that \mathbf{C} , the set of complex numbers has the same cardinality as \mathbf{R} , the set of real numbers.

- 37.** Find \mathbf{A}^n if \mathbf{A} is

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

- 38.** Show that if $\mathbf{A} = c\mathbf{I}$, where c is a real number and \mathbf{I} is the $n \times n$ identity matrix, then $\mathbf{AB} = \mathbf{BA}$ whenever \mathbf{B} is an $n \times n$ matrix.

- 39.** Show that if \mathbf{A} is a 2×2 matrix such that $\mathbf{AB} = \mathbf{BA}$ whenever \mathbf{B} is a 2×2 matrix, then $\mathbf{A} = c\mathbf{I}$, where c is a real number and \mathbf{I} is the 2×2 identity matrix.

- 40.** Show that if \mathbf{A} and \mathbf{B} are invertible matrices and \mathbf{AB} exists, then $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

- 41.** Let \mathbf{A} be an $n \times n$ matrix and let $\mathbf{0}$ be the $n \times n$ matrix all of whose entries are zero. Show that the following are true.

- a) $\mathbf{A} \odot \mathbf{0} = \mathbf{0} \odot \mathbf{A} = \mathbf{0}$

- b) $\mathbf{A} \vee \mathbf{0} = \mathbf{0} \vee \mathbf{A} = \mathbf{A}$

- c) $\mathbf{A} \wedge \mathbf{0} = \mathbf{0} \wedge \mathbf{A} = \mathbf{0}$

Computer Projects

Write programs with the specified input and output.

1. Given subsets A and B of a set with n elements, use bit strings to find \bar{A} , $A \cup B$, $A \cap B$, $A - B$, and $A \oplus B$.
2. Given multisets A and B from the same universal set, find $A \cup B$, $A \cap B$, $A - B$, and $A + B$ (see preamble to Exercise 61 of Section 2.2).
3. Given fuzzy sets A and B , find \bar{A} , $A \cup B$, and $A \cap B$ (see preamble to Exercise 63 of Section 2.2).
4. Given a function f from $\{1, 2, \dots, n\}$ to the set of integers, determine whether f is one-to-one.
5. Given a function f from $\{1, 2, \dots, n\}$ to itself, determine whether f is onto.
6. Given a bijection f from the set $\{1, 2, \dots, n\}$ to itself, find f^{-1} .
7. Given an $m \times k$ matrix \mathbf{A} and a $k \times n$ matrix \mathbf{B} , find \mathbf{AB} .
8. Given a square matrix \mathbf{A} and a positive integer n , find \mathbf{A}^n .
9. Given a square matrix, determine whether it is symmetric.
10. Given two $m \times n$ Boolean matrices, find their meet and join.
11. Given an $m \times k$ Boolean matrix \mathbf{A} and a $k \times n$ Boolean matrix \mathbf{B} , find the Boolean product of \mathbf{A} and \mathbf{B} .
12. Given a square Boolean matrix \mathbf{A} and a positive integer n , find $\mathbf{A}^{[n]}$.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Given two finite sets, list all elements in the Cartesian product of these two sets.
2. Given a finite set, list all elements of its power set.
3. Calculate the number of one-to-one functions from a set S to a set T , where S and T are finite sets of various sizes. Can you determine a formula for the number of such functions? (We will find such a formula in Chapter 6.)
4. Calculate the number of onto functions from a set S to a set T , where S and T are finite sets of various sizes. Can you determine a formula for the number of such functions? (We will find such a formula in Chapter 8.)
- *5. Develop a collection of different rules for generating the terms of a sequence and a program for randomly selecting one of these rules and the particular sequence generated using these rules. Make this part of an interactive program that prompts for the next term of the sequence and determines whether the response is the intended next term.

Writing Projects

Respond to these with essays using outside sources.

1. Discuss how an axiomatic set theory can be developed to avoid Russell's paradox. (See Exercise 46 of Section 2.1.)
2. Research where the concept of a function first arose, and describe how this concept was first used.
3. Explain the different ways in which the *Encyclopedia of Integer Sequences* has been found useful. Also, describe a few of the more unusual sequences in this encyclopedia and how they arise.
4. Define the recently invented EKG sequence and describe some of its properties and open questions about it.
5. Look up the definition of a transcendental number. Explain how to show that such numbers exist and how such numbers can be constructed. Which famous numbers can be shown to be transcendental and for which famous numbers is it still unknown whether they are transcendental?
6. Expand the discussion of the continuum hypothesis in the text.

3

Algorithms

3.1 Algorithms**3.2** The Growth of Functions**3.3** Complexity of Algorithms

Many problems can be solved by considering them as special cases of general problems. For instance, consider the problem of locating the largest integer in the sequence 101, 12, 144, 212, 98. This is a specific case of the problem of locating the largest integer in a sequence of integers. To solve this general problem we must give an algorithm, which specifies a sequence of steps used to solve this general problem. We will study algorithms for solving many different types of problems in this book. For example, in this chapter we will introduce algorithms for two of the most important problems in computer science, searching for an element in a list and sorting a list so its elements are in some prescribed order, such as increasing, decreasing, or alphabetic. Later in the book we will develop algorithms that find the greatest common divisor of two integers, that generate all the orderings of a finite set, that find the shortest path between nodes in a network, and for solving many other problems.

We will also introduce the notion of an algorithmic paradigm, which provides a general method for designing algorithms. In particular we will discuss brute-force algorithms, which find solutions using a straightforward approach without introducing any cleverness. We will also discuss greedy algorithms, a class of algorithms used to solve optimization problems. Proofs are important in the study of algorithms. In this chapter we illustrate this by proving that a particular greedy algorithm always finds an optimal solution.

One important consideration concerning an algorithm is its computational complexity, which measures the processing time and computer memory required by the algorithm to solve problems of a particular size. To measure the complexity of algorithms we use big-*O* and big-Theta notation, which we develop in this chapter. We will illustrate the analysis of the complexity of algorithms in this chapter, focusing on the time an algorithm takes to solve a problem. Furthermore, we will discuss what the time complexity of an algorithm means in practical and theoretical terms.

3.1 Algorithms

Introduction

There are many general classes of problems that arise in discrete mathematics. For instance: given a sequence of integers, find the largest one; given a set, list all its subsets; given a set of integers, put them in increasing order; given a network, find the shortest path between two vertices. When presented with such a problem, the first thing to do is to construct a model that translates the problem into a mathematical context. Discrete structures used in such models include sets, sequences, and functions—structures discussed in Chapter 2—as well as such other structures as permutations, relations, graphs, trees, networks, and finite state machines—concepts that will be discussed in later chapters.

Setting up the appropriate mathematical model is only part of the solution. To complete the solution, a method is needed that will solve the general problem using the model. Ideally, what is required is a procedure that follows a sequence of steps that leads to the desired answer. Such a sequence of steps is called an **algorithm**.

DEFINITION 1

An *algorithm* is a finite sequence of precise instructions for performing a computation or for solving a problem.

The term *algorithm* is a corruption of the name *al-Khowarizmi*, a mathematician of the ninth century, whose book on Hindu numerals is the basis of modern decimal notation. Originally, the word *algorism* was used for the rules for performing arithmetic using decimal notation. *Algorism* evolved into the word *algorithm* by the eighteenth century. With the growing interest in computing machines, the concept of an algorithm was given a more general meaning, to include all definite procedures for solving problems, not just the procedures for performing arithmetic. (We will discuss algorithms for performing arithmetic with integers in Chapter 4.)

In this book, we will discuss algorithms that solve a wide variety of problems. In this section we will use the problem of finding the largest integer in a finite sequence of integers to illustrate the concept of an algorithm and the properties algorithms have. Also, we will describe algorithms for locating a particular element in a finite set. In subsequent sections, procedures for finding the greatest common divisor of two integers, for finding the shortest path between two points in a network, for multiplying matrices, and so on, will be discussed.

EXAMPLE 1 Describe an algorithm for finding the maximum (largest) value in a finite sequence of integers.

Extra Examples

Even though the problem of finding the maximum element in a sequence is relatively trivial, it provides a good illustration of the concept of an algorithm. Also, there are many instances where the largest integer in a finite sequence of integers is required. For instance, a university may need to find the highest score on a competitive exam taken by thousands of students. Or a sports organization may want to identify the member with the highest rating each month. We want to develop an algorithm that can be used whenever the problem of finding the largest element in a finite sequence of integers arises.

We can specify a procedure for solving this problem in several ways. One method is simply to use the English language to describe the sequence of steps used. We now provide such a solution.

Solution of Example 1: We perform the following steps.

1. Set the temporary maximum equal to the first integer in the sequence. (The temporary maximum will be the largest integer examined at any stage of the procedure.)
2. Compare the next integer in the sequence to the temporary maximum, and if it is larger than the temporary maximum, set the temporary maximum equal to this integer.
3. Repeat the previous step if there are more integers in the sequence.
4. Stop when there are no integers left in the sequence. The temporary maximum at this point is the largest integer in the sequence.



An algorithm can also be described using a computer language. However, when that is done, only those instructions permitted in the language can be used. This often leads to a description of the algorithm that is complicated and difficult to understand. Furthermore, because many programming languages are in common use, it would be undesirable to choose one particular language. So, instead of using a particular computer language to specify algorithms, a form of **pseudocode**, described in Appendix 3, will be used in this book. (We will also describe algorithms using the English language.) Pseudocode provides an intermediate step between

Links



ABU JA'FAR MOHAMMED IBN MUSA AL-KHOWARIZMI (C. 780–C. 850) al-Khowarizmi, an astronomer and mathematician, was a member of the House of Wisdom, an academy of scientists in Baghdad. The name al-Khowarizmi means “from the town of Kowarizm,” which was then part of Persia, but is now called *Khiva* and is part of Uzbekistan. al-Khowarizmi wrote books on mathematics, astronomy, and geography. Western Europeans first learned about algebra from his works. The word *algebra* comes from al-jabr, part of the title of his book *Kitab al-jabr w'al muqabala*. This book was translated into Latin and was a widely used textbook. His book on the use of Hindu numerals describes procedures for arithmetic operations using these numerals. European authors used a Latin corruption of his name, which later evolved to the word *algorithm*, to describe the subject of arithmetic with Hindu numerals.

an English language description of an algorithm and an implementation of this algorithm in a programming language. The steps of the algorithm are specified using instructions resembling those used in programming languages. However, in pseudocode, the instructions used can include any well-defined operations or statements. A computer program can be produced in any computer language using the pseudocode description as a starting point.

The pseudocode used in this book is designed to be easily understood. It can serve as an intermediate step in the construction of programs implementing algorithms in one of a variety of different programming languages. Although this pseudocode does not follow the syntax of Java, C, C++, or any other programming language, students familiar with a modern programming language will find it easy to follow. A key difference between this pseudocode and code in a programming language is that we can use any well-defined instruction even if it would take many lines of code to implement this instruction. The details of the pseudocode used in the text are given in Appendix 3. The reader should refer to this appendix whenever the need arises.

A pseudocode description of the algorithm for finding the maximum element in a finite sequence follows.

ALGORITHM 1 Finding the Maximum Element in a Finite Sequence.

```

procedure max( $a_1, a_2, \dots, a_n$ : integers)
   $max := a_1$ 
  for  $i := 2$  to  $n$ 
    if  $max < a_i$  then  $max := a_i$ 
  return  $max$  { $max$  is the largest element}

```

This algorithm first assigns the initial term of the sequence, a_1 , to the variable max . The “for” loop is used to successively examine terms of the sequence. If a term is greater than the current value of max , it is assigned to be the new value of max .

PROPERTIES OF ALGORITHMS There are several properties that algorithms generally share. They are useful to keep in mind when algorithms are described. These properties are:

- ***Input.*** An algorithm has input values from a specified set.
- ***Output.*** From each set of input values an algorithm produces output values from a specified set. The output values are the solution to the problem.
- ***Definiteness.*** The steps of an algorithm must be defined precisely.
- ***Correctness.*** An algorithm should produce the correct output values for each set of input values.
- ***Finiteness.*** An algorithm should produce the desired output after a finite (but perhaps large) number of steps for any input in the set.
- ***Effectiveness.*** It must be possible to perform each step of an algorithm exactly and in a finite amount of time.
- ***Generality.*** The procedure should be applicable for all problems of the desired form, not just for a particular set of input values.

EXAMPLE 2 Show that Algorithm 1 for finding the maximum element in a finite sequence of integers has all the properties listed.

Solution: The input to Algorithm 1 is a sequence of integers. The output is the largest integer in the sequence. Each step of the algorithm is precisely defined, because only assignments, a finite loop, and conditional statements occur. To show that the algorithm is correct, we must show that when the algorithm terminates, the value of the variable max equals the maximum

of the terms of the sequence. To see this, note that the initial value of *max* is the first term of the sequence; as successive terms of the sequence are examined, *max* is updated to the value of a term if the term exceeds the maximum of the terms previously examined. This (informal) argument shows that when all the terms have been examined, *max* equals the value of the largest term. (A rigorous proof of this requires techniques developed in Section 5.1.) The algorithm uses a finite number of steps, because it terminates after all the integers in the sequence have been examined. The algorithm can be carried out in a finite amount of time because each step is either a comparison or an assignment, there are a finite number of these steps, and each of these two operations takes a finite amount of time. Finally, Algorithm 1 is general, because it can be used to find the maximum of any finite sequence of integers. 

Searching Algorithms

The problem of locating an element in an ordered list occurs in many contexts. For instance, a program that checks the spelling of words searches for them in a dictionary, which is just an ordered list of words. Problems of this kind are called **searching problems**. We will discuss several algorithms for searching in this section. We will study the number of steps used by each of these algorithms in Section 3.3.

The general searching problem can be described as follows: Locate an element x in a list of distinct elements a_1, a_2, \dots, a_n , or determine that it is not in the list. The solution to this search problem is the location of the term in the list that equals x (that is, i is the solution if $x = a_i$) and is 0 if x is not in the list.



THE LINEAR SEARCH The first algorithm that we will present is called the **linear search**, or **sequential search**, algorithm. The linear search algorithm begins by comparing x and a_1 . When $x = a_1$, the solution is the location of a_1 , namely, 1. When $x \neq a_1$, compare x with a_2 . If $x = a_2$, the solution is the location of a_2 , namely, 2. When $x \neq a_2$, compare x with a_3 . Continue this process, comparing x successively with each term of the list until a match is found, where the solution is the location of that term, unless no match occurs. If the entire list has been searched without locating x , the solution is 0. The pseudocode for the linear search algorithm is displayed as Algorithm 2.

ALGORITHM 2 The Linear Search Algorithm.

```

procedure linear search( $x$ : integer,  $a_1, a_2, \dots, a_n$ : distinct integers)
   $i := 1$ 
  while ( $i \leq n$  and  $x \neq a_i$ )
     $i := i + 1$ 
  if  $i \leq n$  then  $location := i$ 
  else  $location := 0$ 
  return  $location$  { $location$  is the subscript of the term that equals  $x$ , or is 0 if  $x$  is not found}

```



THE BINARY SEARCH We will now consider another searching algorithm. This algorithm can be used when the list has terms occurring in order of increasing size (for instance: if the terms are numbers, they are listed from smallest to largest; if they are words, they are listed in lexicographic, or alphabetic, order). This second searching algorithm is called the **binary search algorithm**. It proceeds by comparing the element to be located to the middle term of the list. The list is then split into two smaller sublists of the same size, or where one of these smaller lists has one fewer term than the other. The search continues by restricting the search to the appropriate sublist based on the comparison of the element to be located and the middle term. In Section 3.3, it will be shown that the binary search algorithm is much more efficient than the linear search algorithm. Example 3 demonstrates how a binary search works.

EXAMPLE 3 To search for 19 in the list

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22,

first split this list, which has 16 terms, into two smaller lists with eight terms each, namely,

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22.

Then, compare 19 and the largest term in the first list. Because $10 < 19$, the search for 19 can be restricted to the list containing the 9th through the 16th terms of the original list. Next, split this list, which has eight terms, into the two smaller lists of four terms each, namely,

12 13 15 16 18 19 20 22.

Because $16 < 19$ (comparing 19 with the largest term of the first list) the search is restricted to the second of these lists, which contains the 13th through the 16th terms of the original list. The list 18 19 20 22 is split into two lists, namely,

18 19 20 22.

Because 19 is not greater than the largest term of the first of these two lists, which is also 19, the search is restricted to the first list: 18 19, which contains the 13th and 14th terms of the original list. Next, this list of two terms is split into two lists of one term each: 18 and 19. Because $18 < 19$, the search is restricted to the second list: the list containing the 14th term of the list, which is 19. Now that the search has been narrowed down to one term, a comparison is made, and 19 is located as the 14th term in the original list. 

We now specify the steps of the binary search algorithm. To search for the integer x in the list a_1, a_2, \dots, a_n , where $a_1 < a_2 < \dots < a_n$, begin by comparing x with the middle term a_m of the list, where $m = \lfloor (n+1)/2 \rfloor$. (Recall that $\lfloor x \rfloor$ is the greatest integer not exceeding x .) If $x > a_m$, the search for x is restricted to the second half of the list, which is $a_{m+1}, a_{m+2}, \dots, a_n$. If x is not greater than a_m , the search for x is restricted to the first half of the list, which is a_1, a_2, \dots, a_m .

The search has now been restricted to a list with no more than $\lceil n/2 \rceil$ elements. (Recall that $\lceil x \rceil$ is the smallest integer greater than or equal to x .) Using the same procedure, compare x to the middle term of the restricted list. Then restrict the search to the first or second half of the list. Repeat this process until a list with one term is obtained. Then determine whether this term is x . Pseudocode for the binary search algorithm is displayed as Algorithm 3.

ALGORITHM 3 The Binary Search Algorithm.

```

procedure binary search ( $x$ : integer,  $a_1, a_2, \dots, a_n$ : increasing integers)
i := 1 {i is left endpoint of search interval}
j :=  $n$  {j is right endpoint of search interval}
while i < j
    m :=  $\lfloor (i + j)/2 \rfloor$ 
    if  $x > a_m$  then i := m + 1
    else j := m
if  $x = a_i$  then location := i
else location := 0
return location {location is the subscript i of the term  $a_i$  equal to  $x$ , or 0 if  $x$  is not found}

```

Algorithm 3 proceeds by successively narrowing down the part of the sequence being searched. At any given stage only the terms from a_i to a_j are under consideration. In other words, i and j are the smallest and largest subscripts of the remaining terms, respectively. Algorithm 3 continues narrowing the part of the sequence being searched until only one term of the sequence remains. When this is done, a comparison is made to see whether this term equals x .

Sorting



Ordering the elements of a list is a problem that occurs in many contexts. For example, to produce a telephone directory it is necessary to alphabetize the names of subscribers. Similarly, producing a directory of songs available for downloading requires that their titles be put in alphabetic order. Putting addresses in order in an e-mail mailing list can determine whether there are duplicated addresses. Creating a useful dictionary requires that words be put in alphabetical order. Similarly, generating a parts list requires that we order them according to increasing part number.

Suppose that we have a list of elements of a set. Furthermore, suppose that we have a way to order elements of the set. (The notion of ordering elements of sets will be discussed in detail in Section 9.6.) **Sorting** is putting these elements into a list in which the elements are in increasing order. For instance, sorting the list 7, 2, 1, 4, 5, 9 produces the list 1, 2, 4, 5, 7, 9. Sorting the list d, h, c, a, f (using alphabetical order) produces the list a, c, d, f, h .

An amazingly large percentage of computing resources is devoted to sorting one thing or another. Hence, much effort has been devoted to the development of sorting algorithms. A surprisingly large number of sorting algorithms have been devised using distinct strategies, with new ones introduced regularly. In his fundamental work, *The Art of Computer Programming*, Donald Knuth devotes close to 400 pages to sorting, covering around 15 different sorting algorithms in depth! More than 100 sorting algorithms have been devised, and it is surprising how often new sorting algorithms are developed. Among the newest sorting algorithms that have caught on is the library sort, also known as the gapped insertion sort, invented as recently as 2006. There are many reasons why sorting algorithms interest computer scientists and mathematicians. Among these reasons are that some algorithms are easier to implement, some algorithms are more efficient (either in general, or when given input with certain characteristics, such as lists slightly out of order), some algorithms take advantage of particular computer architectures, and some algorithms are particularly clever. In this section we will introduce two sorting algorithms, the bubble sort and the insertion sort. Two other sorting algorithms, the selection sort and the binary insertion sort, are introduced in the exercises, and the shaker sort is introduced in the Supplementary Exercises. In Section 5.4 we will discuss the merge sort and introduce the quick sort in the exercises in that section; the tournament sort is introduced in the exercise set in Section 11.2. We cover sorting algorithms both because sorting is an important problem and because these algorithms can serve as examples for many important concepts.

Sorting is thought to hold the record as the problem solved by the most fundamentally different algorithms!



THE BUBBLE SORT The **bubble sort** is one of the simplest sorting algorithms, but not one of the most efficient. It puts a list into increasing order by successively comparing adjacent elements, interchanging them if they are in the wrong order. To carry out the bubble sort, we perform the basic operation, that is, interchanging a larger element with a smaller one following it, starting at the beginning of the list, for a full pass. We iterate this procedure until the sort is complete. Pseudocode for the bubble sort is given as Algorithm 4. We can imagine the elements in the list placed in a column. In the bubble sort, the smaller elements “bubble” to the top as they are interchanged with larger elements. The larger elements “sink” to the bottom. This is illustrated in Example 4.

EXAMPLE 4 Use the bubble sort to put 3, 2, 4, 1, 5 into increasing order.

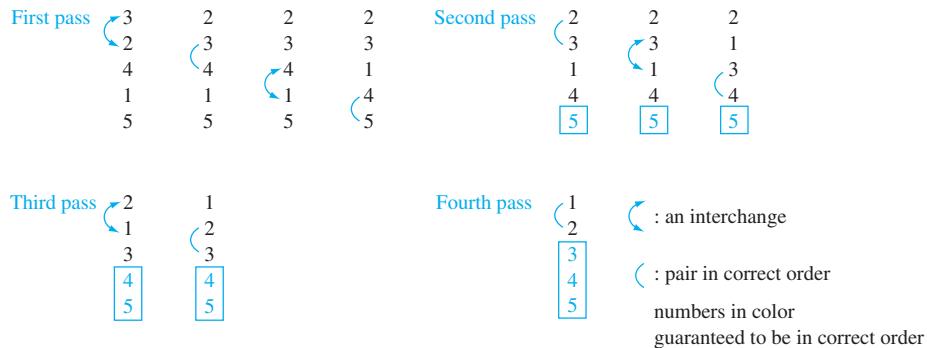


FIGURE 1 The Steps of a Bubble Sort.

Solution: The steps of this algorithm are illustrated in Figure 1. Begin by comparing the first two elements, 3 and 2. Because $3 > 2$, interchange 3 and 2, producing the list 2, 3, 4, 1, 5. Because $3 < 4$, continue by comparing 4 and 1. Because $4 > 1$, interchange 1 and 4, producing the list 2, 3, 1, 4, 5. Because $4 < 5$, the first pass is complete. The first pass guarantees that the largest element, 5, is in the correct position.

The second pass begins by comparing 2 and 3. Because these are in the correct order, 3 and 1 are compared. Because $3 > 1$, these numbers are interchanged, producing 2, 1, 3, 4, 5. Because $3 < 4$, these numbers are in the correct order. It is not necessary to do any more comparisons for this pass because 5 is already in the correct position. The second pass guarantees that the two largest elements, 4 and 5, are in their correct positions.

The third pass begins by comparing 2 and 1. These are interchanged because $2 > 1$, producing 1, 2, 3, 4, 5. Because $2 < 3$, these two elements are in the correct order. It is not necessary to do any more comparisons for this pass because 4 and 5 are already in the correct positions. The third pass guarantees that the three largest elements, 3, 4, and 5, are in their correct positions.

The fourth pass consists of one comparison, namely, the comparison of 1 and 2. Because $1 < 2$, these elements are in the correct order. This completes the bubble sort. 

ALGORITHM 4 The Bubble Sort.

```

procedure bubblesort( $a_1, \dots, a_n$  : real numbers with  $n \geq 2$ )
for  $i := 1$  to  $n - 1$ 
    for  $j := 1$  to  $n - i$ 
        if  $a_j > a_{j+1}$  then interchange  $a_j$  and  $a_{j+1}$ 
    { $a_1, \dots, a_n$  is in increasing order}

```



THE INSERTION SORT The **insertion sort** is a simple sorting algorithm, but it is usually not the most efficient. To sort a list with n elements, the insertion sort begins with the second element. The insertion sort compares this second element with the first element and inserts it before the first element if it does not exceed the first element and after the first element if it exceeds the first element. At this point, the first two elements are in the correct order. The third element is then compared with the first element, and if it is larger than the first element, it is compared with the second element; it is inserted into the correct position among the first three elements.

In general, in the j th step of the insertion sort, the j th element of the list is inserted into the correct position in the list of the previously sorted $j - 1$ elements. To insert the j th element in the list, a linear search technique is used (see Exercise 43); the j th element is successively compared with the already sorted $j - 1$ elements at the start of the list until the first element that

is not less than this element is found or until it has been compared with all $j - 1$ elements; the j th element is inserted in the correct position so that the first j elements are sorted. The algorithm continues until the last element is placed in the correct position relative to the already sorted list of the first $n - 1$ elements. The insertion sort is described in pseudocode in Algorithm 5.

EXAMPLE 5 Use the insertion sort to put the elements of the list 3, 2, 4, 1, 5 in increasing order.

Solution: The insertion sort first compares 2 and 3. Because $3 > 2$, it places 2 in the first position, producing the list 2, 3, 4, 1, 5 (the sorted part of the list is shown in color). At this point, 2 and 3 are in the correct order. Next, it inserts the third element, 4, into the already sorted part of the list by making the comparisons $4 > 2$ and $4 > 3$. Because $4 > 3$, 4 remains in the third position. At this point, the list is 2, 3, 4, 1, 5 and we know that the ordering of the first three elements is correct. Next, we find the correct place for the fourth element, 1, among the already sorted elements, 2, 3, 4. Because $1 < 2$, we obtain the list 1, 2, 3, 4, 5. Finally, we insert 5 into the correct position by successively comparing it to 1, 2, 3, and 4. Because $5 > 4$, it stays at the end of the list, producing the correct order for the entire list. 

ALGORITHM 5 The Insertion Sort.

```

procedure insertion sort( $a_1, a_2, \dots, a_n$ : real numbers with  $n \geq 2$ )
for  $j := 2$  to  $n$ 
     $i := 1$ 
    while  $a_j > a_i$ 
         $i := i + 1$ 
     $m := a_j$ 
    for  $k := 0$  to  $j - i - 1$ 
         $a_{j-k} := a_{j-k-1}$ 
     $a_i := m$ 
{ $a_1, \dots, a_n$  is in increasing order}

```

Greedy Algorithms

“Greed is good ... Greed is right, greed works.
Greed clarifies ...” –
spoken by the character
Gordon Gecko in the film
Wall Street.



You have to prove that a greedy algorithm always finds an optimal solution.

Many algorithms we will study in this book are designed to solve **optimization problems**. The goal of such problems is to find a solution to the given problem that either minimizes or maximizes the value of some parameter. Optimization problems studied later in this text include finding a route between two cities with smallest total mileage, determining a way to encode messages using the fewest bits possible, and finding a set of fiber links between network nodes using the least amount of fiber.

Surprisingly, one of the simplest approaches often leads to a solution of an optimization problem. This approach selects the best choice at each step, instead of considering all sequences of steps that may lead to an optimal solution. Algorithms that make what seems to be the “best” choice at each step are called **greedy algorithms**. Once we know that a greedy algorithm finds a feasible solution, we need to determine whether it has found an optimal solution. (Note that we call the algorithm “greedy” whether or not it finds an optimal solution.) To do this, we either prove that the solution is optimal or we show that there is a counterexample where the algorithm yields a nonoptimal solution. To make these concepts more concrete, we will consider an algorithm that makes change using coins.

EXAMPLE 6 Consider the problem of making n cents change with quarters, dimes, nickels, and pennies, and using the least total number of coins. We can devise a greedy algorithm for making change for n cents by making a locally optimal choice at each step; that is, at each step we choose the coin of the largest denomination possible to add to the pile of change without exceeding n cents. For example, to make change for 67 cents, we first select a quarter (leaving 42 cents). We next select a second quarter (leaving 17 cents), followed by a dime (leaving 7 cents), followed by a nickel (leaving 2 cents), followed by a penny (leaving 1 cent), followed by a penny. \blacktriangleleft



We display a greedy change-making algorithm for n cents, using any set of denominations of coins, as Algorithm 6.

ALGORITHM 6 Greedy Change-Making Algorithm.

```

procedure change( $c_1, c_2, \dots, c_r$ : values of denominations of coins, where
 $c_1 > c_2 > \dots > c_r$ ;  $n$ : a positive integer)
for  $i := 1$  to  $r$ 
     $d_i := 0$  { $d_i$  counts the coins of denomination  $c_i$  used}
    while  $n \geq c_i$ 
         $d_i := d_i + 1$  {add a coin of denomination  $c_i$ }
         $n := n - c_i$ 
{ $d_i$  is the number of coins of denomination  $c_i$  in the change for  $i = 1, 2, \dots, r$ }
```

We have described a greedy algorithm for making change using any finite set of coins with denominations c_1, c_2, \dots, c_r . In the particular case where the four denominations are quarters, dimes, nickels, and pennies, we have $c_1 = 25$, $c_2 = 10$, $c_3 = 5$, and $c_4 = 1$. For this case, we will show that this algorithm leads to an optimal solution in the sense that it uses the fewest coins possible. Before we embark on our proof, we show that there are sets of coins for which the greedy algorithm (Algorithm 6) does not necessarily produce change using the fewest coins possible. For example, if we have only quarters, dimes, and pennies (and no nickels) to use, the greedy algorithm would make change for 30 cents using six coins—a quarter and five pennies—whereas we could have used three coins, namely, three dimes.

LEMMA 1

If n is a positive integer, then n cents in change using quarters, dimes, nickels, and pennies using the fewest coins possible has at most two dimes, at most one nickel, at most four pennies, and cannot have two dimes and a nickel. The amount of change in dimes, nickels, and pennies cannot exceed 24 cents.

Proof: We use a proof by contradiction. We will show that if we had more than the specified number of coins of each type, we could replace them using fewer coins that have the same value. We note that if we had three dimes we could replace them with a quarter and a nickel, if we had two nickels we could replace them with a dime, if we had five pennies we could replace them with a nickel, and if we had two dimes and a nickel we could replace them with a quarter. Because we can have at most two dimes, one nickel, and four pennies, but we cannot have two dimes and a nickel, it follows that 24 cents is the most money we can have in dimes, nickels, and pennies when we make change using the fewest number of coins for n cents. \blacktriangleleft

THEOREM 1

The greedy algorithm (Algorithm 6) produces change using the fewest coins possible.

Proof: We will use a proof by contradiction. Suppose that there is a positive integer n such that there is a way to make change for n cents using quarters, dimes, nickels, and pennies that uses fewer coins than the greedy algorithm finds. We first note that q' , the number of quarters used in this optimal way to make change for n cents, must be the same as q , the number of quarters used by the greedy algorithm. To show this, first note that the greedy algorithm uses the most quarters possible, so $q' \leq q$. However, it is also the case that q' cannot be less than q . If it were, we would need to make up at least 25 cents from dimes, nickels, and pennies in this optimal way to make change. But this is impossible by Lemma 1.

Because there must be the same number of quarters in the two ways to make change, the value of the dimes, nickels, and pennies in these two ways must be the same, and these coins are worth no more than 24 cents. There must be the same number of dimes, because the greedy algorithm used the most dimes possible and by Lemma 1, when change is made using the fewest coins possible, at most one nickel and at most four pennies are used, so that the most dimes possible are also used in the optimal way to make change. Similarly, we have the same number of nickels and, finally, the same number of pennies. \triangleleft

A greedy algorithm makes the best choice at each step according to a specified criterion. The next example shows that it can be difficult to determine which of many possible criteria to choose.

EXAMPLE 7

Suppose we have a group of proposed talks with preset start and end times. Devise a greedy algorithm to schedule as many of these talks as possible in a lecture hall, under the assumptions that once a talk starts, it continues until it ends, no two talks can proceed at the same time, and a talk can begin at the same time another one ends. Assume that talk j begins at time s_j (where s stands for *start*) and ends at time e_j (where e stands for *end*).

Solution: To use a greedy algorithm to schedule the most talks, that is, an optimal schedule, we need to decide how to choose which talk to add at each step. There are many criteria we could use to select a talk at each step, where we chose from the talks that do not overlap talks already selected. For example, we could add talks in order of earliest start time, we could add talks in order of shortest time, we could add talks in order of earliest finish time, or we could use some other criterion.

We now consider these possible criteria. Suppose we add the talk that starts earliest among the talks compatible with those already selected. We can construct a counterexample to see that the resulting algorithm does not always produce an optimal schedule. For instance, suppose that we have three talks: Talk 1 starts at 8 A.M. and ends at 12 noon, Talk 2 starts at 9 A.M. and ends at 10 A.M., and Talk 3 starts at 11 A.M. and ends at 12 noon. We first select the Talk 1 because it starts earliest. But once we have selected Talk 1 we cannot select either Talk 2 or Talk 3 because both overlap Talk 1. Hence, this greedy algorithm selects only one talk. This is not optimal because we could schedule Talk 2 and Talk 3, which do not overlap.

Now suppose we add the talk that is shortest among the talks that do not overlap any of those already selected. Again we can construct a counterexample to show that this greedy algorithm does not always produce an optimal schedule. So, suppose that we have three talks: Talk 1 starts at 8 A.M. and ends at 9:15 A.M., Talk 2 starts at 9 A.M. and ends at 10 A.M., and Talk 3 starts at 9:45 A.M. and ends at 11 A.M. We select Talk 2 because it is shortest, requiring one hour. Once we select Talk 2, we cannot select either Talk 1 or Talk 3 because neither is compatible with Talk 2. Hence, this greedy algorithm selects only one talk. However, it is possible to select two talks, Talk 1 and Talk 3, which are compatible.

However, it can be shown that we schedule the most talks possible if in each step we select the talk with the earliest ending time among the talks compatible with those already selected. We will prove this in Chapter 5 using the method of mathematical induction. The first step we will make is to sort the talks according to increasing finish time. After this sorting, we relabel the talks so that $e_1 \leq e_2 \leq \dots \leq e_n$. The resulting greedy algorithm is given as Algorithm 7. \triangleleft

ALGORITHM 7 Greedy Algorithm for Scheduling Talks.

```

procedure schedule( $s_1 \leq s_2 \leq \dots \leq s_n$ : start times of talks,
 $e_1 \leq e_2 \leq \dots \leq e_n$ : ending times of talks)
sort talks by finish time and reorder so that  $e_1 \leq e_2 \leq \dots \leq e_n$ 
 $S := \emptyset$ 
for  $j := 1$  to  $n$ 
    if talk  $j$  is compatible with  $S$  then
         $S := S \cup \{\text{talk } j\}$ 
return  $S\{S$  is the set of talks scheduled}

```

The Halting Problem



We will now describe a proof of one of the most famous theorems in computer science. We will show that there is a problem that cannot be solved using any procedure. That is, we will show there are unsolvable problems. The problem we will study is the **halting problem**. It asks whether there is a procedure that does this: It takes as input a computer program and input to the program and determines whether the program will eventually stop when run with this input. It would be convenient to have such a procedure, if it existed. Certainly being able to test whether a program entered into an infinite loop would be helpful when writing and debugging programs. However, in 1936 Alan Turing showed that no such procedure exists (see his biography in Section 13.4).

Before we present a proof that the halting problem is unsolvable, first note that we cannot simply run a program and observe what it does to determine whether it terminates when run with the given input. If the program halts, we have our answer, but if it is still running after any fixed length of time has elapsed, we do not know whether it will never halt or we just did not wait long enough for it to terminate. After all, it is not hard to design a program that will stop only after more than a billion years has elapsed.

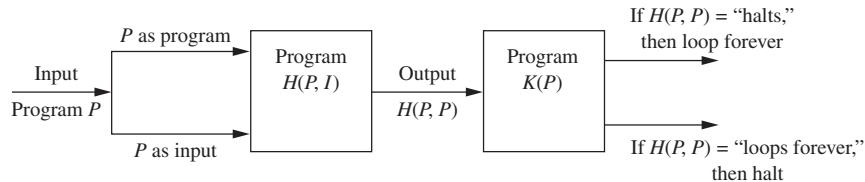
We will describe Turing's proof that the halting problem is unsolvable; it is a proof by contradiction. (The reader should note that our proof is not completely rigorous, because we have not explicitly defined what a procedure is. To remedy this, the concept of a Turing machine is needed. This concept is introduced in Section 13.5.)

Proof: Assume there is a solution to the halting problem, a procedure called $H(P, I)$. The procedure $H(P, I)$ takes two inputs, one a program P and the other I , an input to the program P . $H(P, I)$ generates the string "halt" as output if H determines that P stops when given I as input. Otherwise, $H(P, I)$ generates the string "loops forever" as output. We will now derive a contradiction.

When a procedure is coded, it is expressed as a string of characters; this string can be interpreted as a sequence of bits. This means that a program itself can be used as data. Therefore a program can be thought of as input to another program, or even itself. Hence, H can take a program P as both of its inputs, which are a program and input to this program. H should be able to determine whether P will halt when it is given a copy of itself as input.

To show that no procedure H exists that solves the halting problem, we construct a simple procedure $K(P)$, which works as follows, making use of the output $H(P, P)$. If the output of $H(P, P)$ is "loops forever," which means that P loops forever when given a copy of itself as input, then $K(P)$ halts. If the output of $H(P, P)$ is "halt," which means that P halts when given a copy of itself as input, then $K(P)$ loops forever. That is, $K(P)$ does the opposite of what the output of $H(P, P)$ specifies. (See Figure 2.)

Now suppose we provide K as input to K . We note that if the output of $H(K, K)$ is "loops forever," then by the definition of K we see that $K(K)$ halts. Otherwise, if the output of $H(K, K)$

**FIGURE 2** Showing that the Halting Problem is Unsolvable.

is “halt,” then by the definition of K we see that $K(K)$ loops forever, in violation of what H tells us. In both cases, we have a contradiction.

Thus, H cannot always give the correct answers. Consequently, there is no procedure that solves the halting problem. \triangleleft

Exercises

1. List all the steps used by Algorithm 1 to find the maximum of the list 1, 8, 12, 9, 11, 2, 14, 5, 10, 4.
2. Determine which characteristics of an algorithm described in the text (after Algorithm 1) the following procedures have and which they lack.
 - a) **procedure** *double*(*n*: positive integer)
while *n* > 0
n := 2*n*
 - b) **procedure** *divide*(*n*: positive integer)
while *n* ≥ 0
m := 1/*n*
n := *n* – 1
 - c) **procedure** *sum*(*n*: positive integer)
sum := 0
while *i* < 10
sum := *sum* + *i*
 - d) **procedure** *choose*(*a*, *b*: integers)
x := either *a* or *b*
3. Devise an algorithm that finds the sum of all the integers in a list.
4. Devise an algorithm that takes as input a list of n integers and produces as output the largest difference obtained by subtracting an integer in the list from the one following it.
5. Devise an algorithm that takes as input a list of n integers in nondecreasing order and produces the list of all values that occur more than once. (Recall that a list of integers is **nondecreasing** if each integer in the list is at least as large as the previous integer in the list.)
6. Devise an algorithm that takes as input a list of n integers and finds the number of negative integers in the list.
7. Devise an algorithm that takes as input a list of n integers and finds the location of the last even integer in the list or returns 0 if there are no even integers in the list.
8. Describe an algorithm that takes as input a list of n distinct integers and finds the location of the largest even integer in the list or returns 0 if there are no even integers in the list.
9. A **palindrome** is a string that reads the same forward and backward. Describe an algorithm for determining whether a string of n characters is a palindrome.
10. Devise an algorithm to compute x^n , where x is a real number and n is an integer. [Hint: First give a procedure for computing x^n when n is nonnegative by successive multiplication by x , starting with 1. Then extend this procedure, and use the fact that $x^{-n} = 1/x^n$ to compute x^n when n is negative.]
11. Describe an algorithm that interchanges the values of the variables x and y , using only assignments. What is the minimum number of assignment statements needed to do this?
12. Describe an algorithm that uses only assignment statements that replaces the triple (x, y, z) with (y, z, x) . What is the minimum number of assignment statements needed?
13. List all the steps used to search for 9 in the sequence 1, 3, 4, 5, 6, 8, 9, 11 using
 - a) a linear search.
 - b) a binary search.
14. List all the steps used to search for 7 in the sequence given in Exercise 13 for both a linear search and a binary search.
15. Describe an algorithm that inserts an integer x in the appropriate position into the list a_1, a_2, \dots, a_n of integers that are in increasing order.
16. Describe an algorithm for finding the smallest integer in a finite sequence of natural numbers.
17. Describe an algorithm that locates the first occurrence of the largest element in a finite list of integers, where the integers in the list are not necessarily distinct.
18. Describe an algorithm that locates the last occurrence of the smallest element in a finite list of integers, where the integers in the list are not necessarily distinct.

- 19.** Describe an algorithm that produces the maximum, median, mean, and minimum of a set of three integers. (The **median** of a set of integers is the middle element in the list when these integers are listed in order of increasing size. The **mean** of a set of integers is the sum of the integers divided by the number of integers in the set.)
- 20.** Describe an algorithm for finding both the largest and the smallest integers in a finite sequence of integers.
- 21.** Describe an algorithm that puts the first three terms of a sequence of integers of arbitrary length in increasing order.
- 22.** Describe an algorithm to find the longest word in an English sentence (where a sentence is a sequence of symbols, either a letter or a blank, which can then be broken into alternating words and blanks).
- 23.** Describe an algorithm that determines whether a function from a finite set of integers to another finite set of integers is onto.
- 24.** Describe an algorithm that determines whether a function from a finite set to another finite set is one-to-one.
- 25.** Describe an algorithm that will count the number of 1s in a bit string by examining each bit of the string to determine whether it is a 1 bit.
- 26.** Change Algorithm 3 so that the binary search procedure compares x to a_m at each stage of the algorithm, with the algorithm terminating if $x = a_m$. What advantage does this version of the algorithm have?
- 27.** The **ternary search algorithm** locates an element in a list of increasing integers by successively splitting the list into three sublists of equal (or as close to equal as possible) size, and restricting the search to the appropriate piece. Specify the steps of this algorithm.
- 28.** Specify the steps of an algorithm that locates an element in a list of increasing integers by successively splitting the list into four sublists of equal (or as close to equal as possible) size, and restricting the search to the appropriate piece.
- In a list of elements the same element may appear several times. A **mode** of such a list is an element that occurs at least as often as each of the other elements; a list has more than one mode when more than one element appears the maximum number of times.
- 29.** Devise an algorithm that finds a mode in a list of nondecreasing integers. (Recall that a list of integers is nondecreasing if each term is at least as large as the preceding term.)
- 30.** Devise an algorithm that finds all modes. (Recall that a list of integers is nondecreasing if each term of the list is at least as large as the preceding term.)
- 31.** Devise an algorithm that finds the first term of a sequence of integers that equals some previous term in the sequence.
- 32.** Devise an algorithm that finds all terms of a finite sequence of integers that are greater than the sum of all previous terms of the sequence.
- 33.** Devise an algorithm that finds the first term of a sequence of positive integers that is less than the immediately preceding term of the sequence.
- 34.** Use the bubble sort to sort 6, 2, 3, 1, 5, 4, showing the lists obtained at each step.
- 35.** Use the bubble sort to sort 3, 1, 5, 7, 4, showing the lists obtained at each step.
- 36.** Use the bubble sort to sort d, f, k, m, a, b , showing the lists obtained at each step.
- *37.** Adapt the bubble sort algorithm so that it stops when no interchanges are required. Express this more efficient version of the algorithm in pseudocode.
- 38.** Use the insertion sort to sort the list in Exercise 34, showing the lists obtained at each step.
- 39.** Use the insertion sort to sort the list in Exercise 35, showing the lists obtained at each step.
- 40.** Use the insertion sort to sort the list in Exercise 36, showing the lists obtained at each step.
- The **selection sort** begins by finding the least element in the list. This element is moved to the front. Then the least element among the remaining elements is found and put into the second position. This procedure is repeated until the entire list has been sorted.
- 41.** Sort these lists using the selection sort.
- a)** 3, 5, 4, 1, 2 **b)** 5, 4, 3, 2, 1
c) 1, 2, 3, 4, 5
- 42.** Write the selection sort algorithm in pseudocode.
-  **43.** Describe an algorithm based on the linear search for determining the correct position in which to insert a new element in an already sorted list.
- 44.** Describe an algorithm based on the binary search for determining the correct position in which to insert a new element in an already sorted list.
- 45.** How many comparisons does the insertion sort use to sort the list $1, 2, \dots, n$?
- 46.** How many comparisons does the insertion sort use to sort the list $n, n - 1, \dots, 2, 1$?
- The **binary insertion sort** is a variation of the insertion sort that uses a binary search technique (see Exercise 44) rather than a linear search technique to insert the i th element in the correct place among the previously sorted elements.
- 47.** Show all the steps used by the binary insertion sort to sort the list 3, 2, 4, 5, 1, 6.
- 48.** Compare the number of comparisons used by the insertion sort and the binary insertion sort to sort the list 7, 4, 3, 8, 1, 5, 4, 2.
- *49.** Express the binary insertion sort in pseudocode.
- 50.** **a)** Devise a variation of the insertion sort that uses a linear search technique that inserts the j th element in the correct place by first comparing it with the $(j - 1)$ st element, then the $(j - 2)$ th element if necessary, and so on.
b) Use your algorithm to sort 3, 2, 4, 5, 1, 6.
c) Answer Exercise 45 using this algorithm.
d) Answer Exercise 46 using this algorithm.

51. When a list of elements is in close to the correct order, would it be better to use an insertion sort or its variation described in Exercise 50?
52. Use the greedy algorithm to make change using quarters, dimes, nickels, and pennies for
- 87 cents.
 - 49 cents.
 - 99 cents.
 - 33 cents.
53. Use the greedy algorithm to make change using quarters, dimes, nickels, and pennies for
- 51 cents.
 - 69 cents.
 - 76 cents.
 - 60 cents.
54. Use the greedy algorithm to make change using quarters, dimes, and pennies (but no nickels) for each of the amounts given in Exercise 52. For which of these amounts does the greedy algorithm use the fewest coins of these denominations possible?
55. Use the greedy algorithm to make change using quarters, dimes, and pennies (but no nickels) for each of the amounts given in Exercise 53. For which of these amounts does the greedy algorithm use the fewest coins of these denominations possible?
56. Show that if there were a coin worth 12 cents, the greedy algorithm using quarters, 12-cent coins, dimes, nickels, and pennies would not always produce change using the fewest coins possible.
57. Use Algorithm 7 to schedule the largest number of talks in a lecture hall from a proposed set of talks, if the starting and ending times of the talks are 9:00 A.M. and 9:45 A.M.; 9:30 A.M. and 10:00 A.M.; 9:50 A.M. and 10:15 A.M.; 10:00 A.M. and 10:30 A.M.; 10:10 A.M. and 10:25 A.M.; 10:30 A.M. and 10:55 A.M.; 10:15 A.M. and 10:45 A.M.; 10:30 A.M. and 11:00 A.M.; 10:45 A.M. and 11:30 A.M.; 10:55 A.M. and 11:25 A.M.; 11:00 A.M. and 11:15 A.M.
58. Show that a greedy algorithm that schedules talks in a lecture hall, as described in Example 7, by selecting at each step the talk that overlaps the fewest other talks, does not always produce an optimal schedule.
- *59. a) Devise a greedy algorithm that determines the fewest lecture halls needed to accommodate n talks given the starting and ending time for each talk.
b) Prove that your algorithm is optimal.

Suppose we have s men m_1, m_2, \dots, m_s and s women w_1, w_2, \dots, w_s . We wish to match each person with a member



of the opposite gender. Furthermore, suppose that each person ranks, in order of preference, with no ties, the people of the opposite gender. We say that a matching of people of opposite genders to form couples is **stable** if we cannot find a man m and a woman w who are not assigned to each other such that m prefers w over his assigned partner and w prefers m to her assigned partner.

60. Suppose we have three men m_1, m_2 , and m_3 and three women w_1, w_2 , and w_3 . Furthermore, suppose that the preference rankings of the men for the three women, from highest to lowest, are $m_1: w_3, w_1, w_2; m_2: w_1, w_2, w_3; m_3: w_2, w_3, w_1$; and the preference rankings of the women for the three men, from highest to lowest, are $w_1: m_1, m_2, m_3; w_2: m_2, m_1, m_3; w_3: m_3, m_2, m_1$. For each of the six possible matchings of men and women to form three couples, determine whether this matching is stable.

The **deferred acceptance algorithm**, also known as the **Gale-Shapley algorithm**, can be used to construct a stable matching of men and women. In this algorithm, members of one gender are the **suitors** and members of the other gender the **suitees**. The algorithm uses a sequence of rounds; in each round every suitor whose proposal was rejected in the previous round proposes to his or her highest ranking suitee who has not already rejected a proposal from this suitor. A suitee rejects all proposals except that from the suitor that this suitee ranks highest among all the suitors who have proposed to this suitee in this round or previous rounds. The proposal of this highest ranking suitor remains pending and is rejected in a later round if a more appealing suitor proposes in that round. The series of rounds ends when every suitor has exactly one pending proposal. All pending proposals are then accepted.

61. Write the deferred acceptance algorithm in pseudocode.
62. Show that the deferred acceptance algorithm terminates.
- *63. Show that the deferred acceptance always terminates with a stable assignment.
64. Show that the problem of determining whether a program with a given input ever prints the digit 1 is unsolvable.
65. Show that the following problem is solvable. Given two programs with their inputs and the knowledge that exactly one of them halts, determine which halts.
66. Show that the problem of deciding whether a specific program with a specific input halts is solvable.

3.2 The Growth of Functions

Introduction

In Section 3.1 we discussed the concept of an algorithm. We introduced algorithms that solve a variety of problems, including searching for an element in a list and sorting a list. In Section 3.3 we will study the number of operations used by these algorithms. In particular, we will estimate the number of comparisons used by the linear and binary search algorithms to find an element in a sequence of n elements. We will also estimate the number of comparisons used by the

bubble sort and by the insertion sort to sort a list of n elements. The time required to solve a problem depends on more than only the number of operations it uses. The time also depends on the hardware and software used to run the program that implements the algorithm. However, when we change the hardware and software used to implement an algorithm, we can closely approximate the time required to solve a problem of size n by multiplying the previous time required by a constant. For example, on a supercomputer we might be able to solve a problem of size n a million times faster than we can on a PC. However, this factor of one million will not depend on n (except perhaps in some minor ways). One of the advantages of using **big- O notation**, which we introduce in this section, is that we can estimate the growth of a function without worrying about constant multipliers or smaller order terms. This means that, using big- O notation, we do not have to worry about the hardware and software used to implement an algorithm. Furthermore, using big- O notation, we can assume that the different operations used in an algorithm take the same time, which simplifies the analysis considerably.

Big- O notation is used extensively to estimate the number of operations an algorithm uses as its input grows. With the help of this notation, we can determine whether it is practical to use a particular algorithm to solve a problem as the size of the input increases. Furthermore, using big- O notation, we can compare two algorithms to determine which is more efficient as the size of the input grows. For instance, if we have two algorithms for solving a problem, one using $100n^2 + 17n + 4$ operations and the other using n^3 operations, big- O notation can help us see that the first algorithm uses far fewer operations when n is large, even though it uses more operations for small values of n , such as $n = 10$.

This section introduces big- O notation and the related big-Omega and big-Theta notations. We will explain how big- O , big-Omega, and big-Theta estimates are constructed and establish estimates for some important functions that are used in the analysis of algorithms.

Big- O Notation

The growth of functions is often described using a special notation. Definition 1 describes this notation.

DEFINITION 1

Let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $O(g(x))$ if there are constants C and k such that

$$|f(x)| \leq C|g(x)|$$

whenever $x > k$. [This is read as “ $f(x)$ is big-oh of $g(x)$.”]

Remark: Intuitively, the definition that $f(x)$ is $O(g(x))$ says that $f(x)$ grows slower than some fixed multiple of $g(x)$ as x grows without bound.

The constants C and k in the definition of big- O notation are called **witnesses** to the relationship $f(x)$ is $O(g(x))$. To establish that $f(x)$ is $O(g(x))$ we need only one pair of witnesses to this relationship. That is, to show that $f(x)$ is $O(g(x))$, we need find only *one* pair of constants C and k , the witnesses, such that $|f(x)| \leq C|g(x)|$ whenever $x > k$.



Note that when there is one pair of witnesses to the relationship $f(x)$ is $O(g(x))$, there are *infinitely many* pairs of witnesses. To see this, note that if C and k are one pair of witnesses, then any pair C' and k' , where $C < C'$ and $k < k'$, is also a pair of witnesses, because $|f(x)| \leq C|g(x)| \leq C'|g(x)|$ whenever $x > k' > k$.



THE HISTORY OF BIG-O NOTATION Big- O notation has been used in mathematics for more than a century. In computer science it is widely used in the analysis of algorithms, as will be seen in Section 3.3. The German mathematician Paul Bachmann first introduced big- O notation in 1892 in an important book on number theory. The big- O symbol is sometimes called a **Landau symbol** after the German mathematician Edmund Landau, who used this notation throughout his work. The use of big- O notation in computer science was popularized by Donald Knuth, who also introduced the big- Ω and big- Θ notations defined later in this section.

WORKING WITH THE DEFINITION OF BIG-O NOTATION A useful approach for finding a pair of witnesses is to first select a value of k for which the size of $|f(x)|$ can be readily estimated when $x > k$ and to see whether we can use this estimate to find a value of C for which $|f(x)| \leq C|g(x)|$ for $x > k$. This approach is illustrated in Example 1.

EXAMPLE 1 Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$.



Solution: We observe that we can readily estimate the size of $f(x)$ when $x > 1$ because $x < x^2$ and $1 < x^2$ when $x > 1$. It follows that

$$0 \leq x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2$$

whenever $x > 1$, as shown in Figure 1. Consequently, we can take $C = 4$ and $k = 1$ as witnesses to show that $f(x)$ is $O(x^2)$. That is, $f(x) = x^2 + 2x + 1 < 4x^2$ whenever $x > 1$. (Note that it is not necessary to use absolute values here because all functions in these equalities are positive when x is positive.)

Alternatively, we can estimate the size of $f(x)$ when $x > 2$. When $x > 2$, we have $2x \leq x^2$ and $1 \leq x^2$. Consequently, if $x > 2$, we have

$$0 \leq x^2 + 2x + 1 \leq x^2 + x^2 + x^2 = 3x^2.$$

It follows that $C = 3$ and $k = 2$ are also witnesses to the relation $f(x)$ is $O(x^2)$.

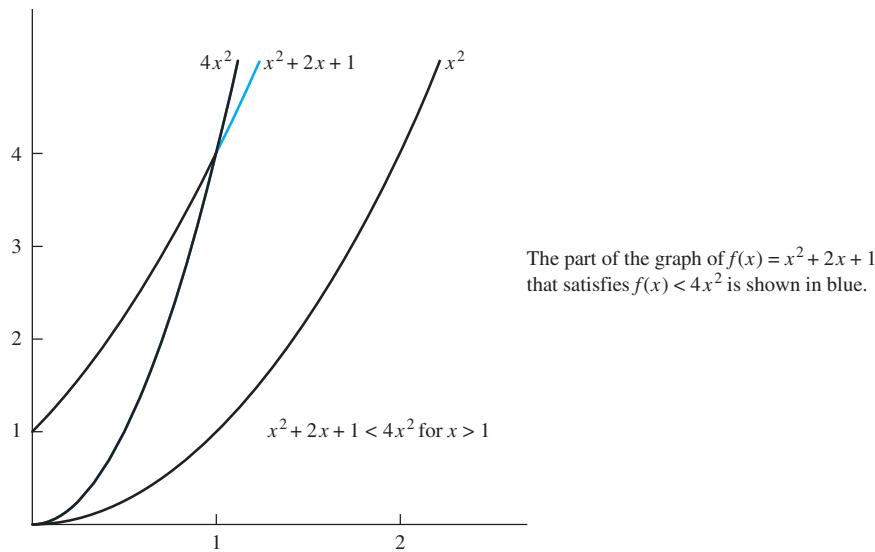


FIGURE 1 The Function $x^2 + 2x + 1$ is $O(x^2)$.

Observe that in the relationship “ $f(x)$ is $O(x^2)$,” x^2 can be replaced by any function with larger values than x^2 . For example, $f(x)$ is $O(x^3)$, $f(x)$ is $O(x^2 + x + 7)$, and so on.

It is also true that x^2 is $O(x^2 + 2x + 1)$, because $x^2 < x^2 + 2x + 1$ whenever $x > 1$. This means that $C = 1$ and $k = 1$ are witnesses to the relationship x^2 is $O(x^2 + 2x + 1)$. \blacktriangleleft

Note that in Example 1 we have two functions, $f(x) = x^2 + 2x + 1$ and $g(x) = x^2$, such that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$ —the latter fact following from the inequality $x^2 \leq x^2 + 2x + 1$, which holds for all nonnegative real numbers x . We say that two functions $f(x)$ and $g(x)$ that satisfy both of these big- O relationships are of the **same order**. We will return to this notion later in this section.

Remark: The fact that $f(x)$ is $O(g(x))$ is sometimes written $f(x) = O(g(x))$. However, the equals sign in this notation does *not* represent a genuine equality. Rather, this notation tells us that an inequality holds relating the values of the functions f and g for sufficiently large numbers in the domains of these functions. However, it is acceptable to write $f(x) \in O(g(x))$ because $O(g(x))$ represents the set of functions that are $O(g(x))$.

When $f(x)$ is $O(g(x))$, and $h(x)$ is a function that has larger absolute values than $g(x)$ does for sufficiently large values of x , it follows that $f(x)$ is $O(h(x))$. In other words, the function $g(x)$ in the relationship $f(x)$ is $O(g(x))$ can be replaced by a function with larger absolute values. To see this, note that if

$$|f(x)| \leq C|g(x)| \quad \text{if } x > k,$$

and if $|h(x)| > |g(x)|$ for all $x > k$, then

$$|f(x)| \leq C|h(x)| \quad \text{if } x > k.$$

Hence, $f(x)$ is $O(h(x))$.

When big- O notation is used, the function g in the relationship $f(x)$ is $O(g(x))$ is chosen to be as small as possible (sometimes from a set of reference functions, such as functions of the form x^n , where n is a positive integer).

Links



PAUL GUSTAV HEINRICH BACHMANN (1837–1920) Paul Bachmann, the son of a Lutheran pastor, shared his father’s pious lifestyle and love of music. His mathematical talent was discovered by one of his teachers, even though he had difficulties with some of his early mathematical studies. After recuperating from tuberculosis in Switzerland, Bachmann studied mathematics, first at the University of Berlin and later at Göttingen, where he attended lectures presented by the famous number theorist Dirichlet. He received his doctorate under the German number theorist Kummer in 1862; his thesis was on group theory. Bachmann was a professor at Breslau and later at Münster. After he retired from his professorship, he continued his mathematical writing, played the piano, and served as a music critic for newspapers. Bachmann’s mathematical writings include a five-volume survey of results and methods in number theory, a two-volume work on elementary number theory, a book on irrational numbers, and a book on the famous conjecture known as Fermat’s Last Theorem. He introduced big- O notation in his 1892 book *Analytische Zahlentheorie*.

Links



EDMUND LANDAU (1877–1938) Edmund Landau, the son of a Berlin gynecologist, attended high school and university in Berlin. He received his doctorate in 1899, under the direction of Frobenius. Landau first taught at the University of Berlin and then moved to Göttingen, where he was a full professor until the Nazis forced him to stop teaching. Landau’s main contributions to mathematics were in the field of analytic number theory. In particular, he established several important results concerning the distribution of primes. He authored a three-volume exposition on number theory as well as other books on number theory and mathematical analysis.

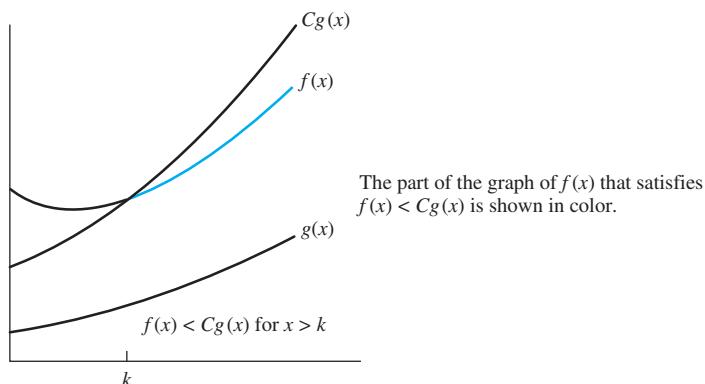


FIGURE 2 The Function $f(x)$ is $O(g(x))$.

In subsequent discussions, we will almost always deal with functions that take on only positive values. All references to absolute values can be dropped when working with big- O estimates for such functions. Figure 2 illustrates the relationship $f(x)$ is $O(g(x))$.

Example 2 illustrates how big- O notation is used to estimate the growth of functions.

EXAMPLE 2 Show that $7x^2$ is $O(x^3)$.

Solution: Note that when $x > 7$, we have $7x^2 < x^3$. (We can obtain this inequality by multiplying both sides of $x > 7$ by x^2 .) Consequently, we can take $C = 1$ and $k = 7$ as witnesses to establish



DONALD E. KNUTH (BORN 1938) Knuth grew up in Milwaukee, where his father taught bookkeeping at a Lutheran high school and owned a small printing business. He was an excellent student, earning academic achievement awards. He applied his intelligence in unconventional ways, winning a contest when he was in the eighth grade by finding over 4500 words that could be formed from the letters in “Ziegler’s Giant Bar.” This won a television set for his school and a candy bar for everyone in his class.

Knuth had a difficult time choosing physics over music as his major at the Case Institute of Technology. He then switched from physics to mathematics, and in 1960 he received his bachelor of science degree, simultaneously receiving a master of science degree by a special award of the faculty who considered his work outstanding. At Case, he managed the basketball team and applied his talents by constructing a formula for the value of each player. This novel approach was covered by *Newsweek* and by Walter Cronkite on the CBS television network. Knuth began graduate work at the California Institute of Technology in 1960 and received his Ph.D. there in 1963. During this time he worked as a consultant, writing compilers for different computers.

Knuth joined the staff of the California Institute of Technology in 1963, where he remained until 1968, when he took a job as a full professor at Stanford University. He retired as Professor Emeritus in 1992 to concentrate on writing. He is especially interested in updating and completing new volumes of his series *The Art of Computer Programming*, a work that has had a profound influence on the development of computer science, which he began writing as a graduate student in 1962, focusing on compilers. In common jargon, “Knuth,” referring to *The Art of Computer Programming*, has come to mean the reference that answers all questions about such topics as data structures and algorithms.

Knuth is the founder of the modern study of computational complexity. He has made fundamental contributions to the subject of compilers. His dissatisfaction with mathematics typography sparked him to invent the now widely used TeX and Metafont systems. TeX has become a standard language for computer typography. Two of the many awards Knuth has received are the 1974 Turing Award and the 1979 National Medal of Technology, awarded to him by President Carter.

Knuth has written for a wide range of professional journals in computer science and in mathematics. However, his first publication, in 1957, when he was a college freshman, was a parody of the metric system called “The Potrzebie Systems of Weights and Measures,” which appeared in *MAD Magazine* and has been in reprint several times. He is a church organist, as his father was. He is also a composer of music for the organ. Knuth believes that writing computer programs can be an aesthetic experience, much like writing poetry or composing music.

Knuth pays \$2.56 for the first person to find each error in his books and \$0.32 for significant suggestions. If you send him a letter with an error (you will need to use regular mail, because he has given up reading e-mail), he will eventually inform you whether you were the first person to tell him about this error. Be prepared for a long wait, because he receives an overwhelming amount of mail. (The author received a letter years after sending an error report to Knuth, noting that this report arrived several months after the first report of this error.)

the relationship $7x^2$ is $O(x^3)$. Alternatively, when $x > 1$, we have $7x^2 < 7x^3$, so that $C = 7$ and $k = 1$ are also witnesses to the relationship $7x^2$ is $O(x^3)$. 

Example 3 illustrates how to show that a big- O relationship does not hold.

EXAMPLE 3 Show that n^2 is not $O(n)$.

Solution: To show that n^2 is not $O(n)$, we must show that no pair of witnesses C and k exist such that $n^2 \leq Cn$ whenever $n > k$. We will use a proof by contradiction to show this.

Suppose that there are constants C and k for which $n^2 \leq Cn$ whenever $n > k$. Observe that when $n > 0$ we can divide both sides of the inequality $n^2 \leq Cn$ by n to obtain the equivalent inequality $n \leq C$. However, no matter what C and k are, the inequality $n \leq C$ cannot hold for all n with $n > k$. In particular, once we set a value of k , we see that when n is larger than the maximum of k and C , it is not true that $n \leq C$ even though $n > k$. This contradiction shows that n^2 is not $O(n)$. 

EXAMPLE 4 Example 2 shows that $7x^2$ is $O(x^3)$. Is it also true that x^3 is $O(7x^2)$?

Solution: To determine whether x^3 is $O(7x^2)$, we need to determine whether witnesses C and k exist, so that $x^3 \leq C(7x^2)$ whenever $x > k$. We will show that no such witnesses exist using a proof by contradiction.

If C and k are witnesses, the inequality $x^3 \leq C(7x^2)$ holds for all $x > k$. Observe that the inequality $x^3 \leq C(7x^2)$ is equivalent to the inequality $x \leq 7C$, which follows by dividing both sides by the positive quantity x^2 . However, no matter what C is, it is not the case that $x \leq 7C$ for all $x > k$ no matter what k is, because x can be made arbitrarily large. It follows that no witnesses C and k exist for this proposed big- O relationship. Hence, x^3 is not $O(7x^2)$. 

Big- O Estimates for Some Important Functions

Polynomials can often be used to estimate the growth of functions. Instead of analyzing the growth of polynomials each time they occur, we would like a result that can always be used to estimate the growth of a polynomial. Theorem 1 does this. It shows that the leading term of a polynomial dominates its growth by asserting that a polynomial of degree n or less is $O(x^n)$.

THEOREM 1

Let $f(x) = a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$, where $a_0, a_1, \dots, a_{n-1}, a_n$ are real numbers. Then $f(x)$ is $O(x^n)$.

Proof: Using the triangle inequality (see Exercise 7 in Section 1.8), if $x > 1$ we have

$$\begin{aligned}|f(x)| &= |a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0| \\ &\leq |a_n|x^n| + |a_{n-1}|x^{n-1}| + \cdots + |a_1||x| + |a_0| \\ &= x^n(|a_n| + |a_{n-1}|/x + \cdots + |a_1|/x^{n-1} + |a_0|/x^n) \\ &\leq x^n(|a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0|).\end{aligned}$$

This shows that

$$|f(x)| \leq Cx^n,$$

where $C = |a_n| + |a_{n-1}| + \dots + |a_0|$ whenever $x > 1$. Hence, the witnesses $C = |a_n| + |a_{n-1}| + \dots + |a_0|$ and $k = 1$ show that $f(x)$ is $O(x^n)$. 

We now give some examples involving functions that have the set of positive integers as their domains.

EXAMPLE 5 How can big- O notation be used to estimate the sum of the first n positive integers?

Solution: Because each of the integers in the sum of the first n positive integers does not exceed n , it follows that

$$1 + 2 + \dots + n \leq n + n + \dots + n = n^2.$$

From this inequality it follows that $1 + 2 + 3 + \dots + n$ is $O(n^2)$, taking $C = 1$ and $k = 1$ as witnesses. (In this example the domains of the functions in the big- O relationship are the set of positive integers.) 

In Example 6 big- O estimates will be developed for the factorial function and its logarithm. These estimates will be important in the analysis of the number of steps used in sorting procedures.

EXAMPLE 6 Give big- O estimates for the factorial function and the logarithm of the factorial function, where the factorial function $f(n) = n!$ is defined by

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

whenever n is a positive integer, and $0! = 1$. For example,

$$1! = 1, \quad 2! = 1 \cdot 2 = 2, \quad 3! = 1 \cdot 2 \cdot 3 = 6, \quad 4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24.$$

Note that the function $n!$ grows rapidly. For instance,

$$20! = 2,432,902,008,176,640,000.$$

Solution: A big- O estimate for $n!$ can be obtained by noting that each term in the product does not exceed n . Hence,

$$\begin{aligned} n! &= 1 \cdot 2 \cdot 3 \cdot \dots \cdot n \\ &\leq n \cdot n \cdot n \cdot \dots \cdot n \\ &= n^n. \end{aligned}$$

This inequality shows that $n!$ is $O(n^n)$, taking $C = 1$ and $k = 1$ as witnesses. Taking logarithms of both sides of the inequality established for $n!$, we obtain

$$\log n! \leq \log n^n = n \log n.$$

This implies that $\log n!$ is $O(n \log n)$, again taking $C = 1$ and $k = 1$ as witnesses. 

EXAMPLE 7 In Section 4.1, we will show that $n < 2^n$ whenever n is a positive integer. Show that this inequality implies that n is $O(2^n)$, and use this inequality to show that $\log n$ is $O(n)$.

Solution: Using the inequality $n < 2^n$, we quickly can conclude that n is $O(2^n)$ by taking $k = C = 1$ as witnesses. Note that because the logarithm function is increasing, taking logarithms (base 2) of both sides of this inequality shows that

$$\log n < n.$$

It follows that

$$\log n \text{ is } O(n).$$

(Again we take $C = k = 1$ as witnesses.)

If we have logarithms to a base b , where b is different from 2, we still have $\log_b n$ is $O(n)$ because

$$\log_b n = \frac{\log n}{\log b} < \frac{n}{\log b}$$

whenever n is a positive integer. We take $C = 1/\log b$ and $k = 1$ as witnesses. (We have used Theorem 3 in Appendix 2 to see that $\log_b n = \log n / \log b$.) ◀

As mentioned before, big- O notation is used to estimate the number of operations needed to solve a problem using a specified procedure or algorithm. The functions used in these estimates often include the following:

$$1, \log n, n, n \log n, n^2, 2^n, n!$$

Using calculus it can be shown that each function in the list is smaller than the succeeding function, in the sense that the ratio of a function and the succeeding function tends to zero as n grows without bound. Figure 3 displays the graphs of these functions, using a scale for the values of the functions that doubles for each successive marking on the graph. That is, the vertical scale in this graph is logarithmic.

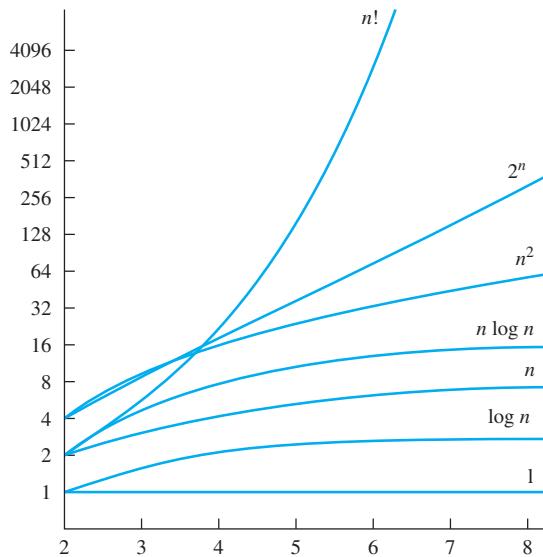


FIGURE 3 A Display of the Growth of Functions Commonly Used in Big- O Estimates.

USEFUL BIG- O ESTIMATES INVOLVING LOGARITHMS, POWERS, AND EXPONENTIAL FUNCTIONS We now give some useful facts that help us determine whether big- O relationships hold between pairs of functions when each of the functions is a power of a logarithm, a power, or an exponential function of the form b^n where $b > 1$. Their proofs are left as Exercises 57–60 for readers skilled with calculus.

Theorem 1 shows that if $f(n)$ is a polynomial of degree d , then $f(n)$ is $O(n^d)$. Applying this theorem, we see that if $d > c > 1$, then n^c is $O(n^d)$. We leave it to the reader to show that the reverse of this relationship does not hold. Putting these facts together, we see that if $d > c > 1$, then

$$n^c \text{ is } O(n^d), \text{ but } n^d \text{ is not } O(n^c).$$

In Example 7 we showed that $\log_b n$ is $O(n)$ whenever $b > 1$. More generally, whenever $b > 1$ and c and d are positive, we have

$$(\log_b n)^c \text{ is } O(n^d), \text{ but } n^d \text{ is not } O((\log_b n)^c).$$

This tells us that every positive power of the logarithm of n to the base b , where $b > 1$, is big- O of every positive power of n , but the reverse relationship never holds.

In Example 7, we also showed that n is $O(2^n)$. More generally, whenever d is positive and $b > 1$, we have

$$n^d \text{ is } O(b^n), \text{ but } b^n \text{ is not } O(n^d).$$

This tells us that every power of n is big- O of every exponential function of n with a base that is greater than one, but the reverse relationship never holds. Furthermore, we have when $c > b > 1$,

$$b^n \text{ is } O(c^n) \text{ but } c^n \text{ is not } O(b^n).$$

This tells us that if we have two exponential functions with different bases greater than one, one of these functions is big- O of the other if and only if its base is smaller or equal.

The Growth of Combinations of Functions

Many algorithms are made up of two or more separate subprocedures. The number of steps used by a computer to solve a problem with input of a specified size using such an algorithm is the sum of the number of steps used by these subprocedures. To give a big- O estimate for the number of steps needed, it is necessary to find big- O estimates for the number of steps used by each subprocedure and then combine these estimates.

Big- O estimates of combinations of functions can be provided if care is taken when different big- O estimates are combined. In particular, it is often necessary to estimate the growth of the sum and the product of two functions. What can be said if big- O estimates for each of two functions are known? To see what sort of estimates hold for the sum and the product of two functions, suppose that $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$.

From the definition of big- O notation, there are constants C_1, C_2, k_1 , and k_2 such that

$$|f_1(x)| \leq C_1|g_1(x)|$$

when $x > k_1$, and

$$|f_2(x)| \leq C_2|g_2(x)|$$

when $x > k_2$. To estimate the sum of $f_1(x)$ and $f_2(x)$, note that

$$\begin{aligned} |(f_1 + f_2)(x)| &= |f_1(x) + f_2(x)| \\ &\leq |f_1(x)| + |f_2(x)| \quad \text{using the triangle inequality } |a + b| \leq |a| + |b|. \end{aligned}$$

When x is greater than both k_1 and k_2 , it follows from the inequalities for $|f_1(x)|$ and $|f_2(x)|$ that

$$\begin{aligned} |f_1(x)| + |f_2(x)| &\leq C_1|g_1(x)| + C_2|g_2(x)| \\ &\leq C_1|g(x)| + C_2|g(x)| \\ &= (C_1 + C_2)|g(x)| \\ &= C|g(x)|, \end{aligned}$$

where $C = C_1 + C_2$ and $g(x) = \max(|g_1(x)|, |g_2(x)|)$. [Here $\max(a, b)$ denotes the maximum, or larger, of a and b .]

This inequality shows that $|(f_1 + f_2)(x)| \leq C|g(x)|$ whenever $x > k$, where $k = \max(k_1, k_2)$. We state this useful result as Theorem 2.

THEOREM 2

Suppose that $f_1(x)$ is $O(g_1(x))$ and that $f_2(x)$ is $O(g_2(x))$. Then $(f_1 + f_2)(x)$ is $O(\max(|g_1(x)|, |g_2(x)|))$.

We often have big- O estimates for f_1 and f_2 in terms of the same function g . In this situation, Theorem 2 can be used to show that $(f_1 + f_2)(x)$ is also $O(g(x))$, because $\max(g(x), g(x)) = g(x)$. This result is stated in Corollary 1.

COROLLARY 1

Suppose that $f_1(x)$ and $f_2(x)$ are both $O(g(x))$. Then $(f_1 + f_2)(x)$ is $O(g(x))$.

In a similar way big- O estimates can be derived for the product of the functions f_1 and f_2 . When x is greater than $\max(k_1, k_2)$ it follows that

$$\begin{aligned} |(f_1 f_2)(x)| &= |f_1(x)||f_2(x)| \\ &\leq C_1|g_1(x)|C_2|g_2(x)| \\ &\leq C_1 C_2 |(g_1 g_2)(x)| \\ &\leq C |(g_1 g_2)(x)|, \end{aligned}$$

where $C = C_1 C_2$. From this inequality, it follows that $f_1(x)f_2(x)$ is $O(g_1 g_2(x))$, because there are constants C and k , namely, $C = C_1 C_2$ and $k = \max(k_1, k_2)$, such that $|(f_1 f_2)(x)| \leq C|g_1(x)g_2(x)|$ whenever $x > k$. This result is stated in Theorem 3.

THEOREM 3

Suppose that $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$. Then $(f_1 f_2)(x)$ is $O(g_1(x)g_2(x))$.

The goal in using big- O notation to estimate functions is to choose a function $g(x)$ as simple as possible, that grows relatively slowly so that $f(x)$ is $O(g(x))$. Examples 8 and 9 illustrate how to use Theorems 2 and 3 to do this. The type of analysis given in these examples is often used in the analysis of the time used to solve problems using computer programs.

EXAMPLE 8 Give a big- O estimate for $f(n) = 3n \log(n!) + (n^2 + 3) \log n$, where n is a positive integer.

Solution: First, the product $3n \log(n!)$ will be estimated. From Example 6 we know that $\log(n!)$ is $O(n \log n)$. Using this estimate and the fact that $3n$ is $O(n)$, Theorem 3 gives the estimate that $3n \log(n!)$ is $O(n^2 \log n)$.

Next, the product $(n^2 + 3) \log n$ will be estimated. Because $(n^2 + 3) < 2n^2$ when $n > 2$, it follows that $n^2 + 3$ is $O(n^2)$. Thus, from Theorem 3 it follows that $(n^2 + 3) \log n$ is $O(n^2 \log n)$. Using Theorem 2 to combine the two big- O estimates for the products shows that $f(n) = 3n \log(n!) + (n^2 + 3) \log n$ is $O(n^2 \log n)$. \blacktriangleleft

EXAMPLE 9 Give a big- O estimate for $f(x) = (x + 1) \log(x^2 + 1) + 3x^2$.

Solution: First, a big- O estimate for $(x + 1) \log(x^2 + 1)$ will be found. Note that $(x + 1)$ is $O(x)$. Furthermore, $x^2 + 1 \leq 2x^2$ when $x > 1$. Hence,

$$\log(x^2 + 1) \leq \log(2x^2) = \log 2 + \log x^2 = \log 2 + 2 \log x \leq 3 \log x,$$

if $x > 2$. This shows that $\log(x^2 + 1)$ is $O(\log x)$.

From Theorem 3 it follows that $(x + 1) \log(x^2 + 1)$ is $O(x \log x)$. Because $3x^2$ is $O(x^2)$, Theorem 2 tells us that $f(x)$ is $O(\max(x \log x, x^2))$. Because $x \log x \leq x^2$, for $x > 1$, it follows that $f(x)$ is $O(x^2)$. \blacktriangleleft

Big-Omega and Big-Theta Notation

Ω and Θ are the Greek uppercase letters omega and theta, respectively.

Big- O notation is used extensively to describe the growth of functions, but it has limitations. In particular, when $f(x)$ is $O(g(x))$, we have an upper bound, in terms of $g(x)$, for the size of $f(x)$ for large values of x . However, big- O notation does not provide a lower bound for the size of $f(x)$ for large x . For this, we use **big-Omega (big- Ω) notation**. When we want to give both an upper and a lower bound on the size of a function $f(x)$, relative to a reference function $g(x)$, we use **big-Theta (big- Θ) notation**. Both big-Omega and big-Theta notation were introduced by Donald Knuth in the 1970s. His motivation for introducing these notations was the common misuse of big- O notation when both an upper and a lower bound on the size of a function are needed.

We now define big-Omega notation and illustrate its use. After doing so, we will do the same for big-Theta notation.

DEFINITION 2

Let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $\Omega(g(x))$ if there are positive constants C and k such that

$$|f(x)| \geq C|g(x)|$$

whenever $x > k$. [This is read as “ $f(x)$ is big-Omega of $g(x)$.”]

There is a strong connection between big- O and big-Omega notation. In particular, $f(x)$ is $\Omega(g(x))$ if and only if $g(x)$ is $O(f(x))$. We leave the verification of this fact as a straightforward exercise for the reader.

EXAMPLE 10 The function $f(x) = 8x^3 + 5x^2 + 7$ is $\Omega(g(x))$, where $g(x)$ is the function $g(x) = x^3$. This is easy to see because $f(x) = 8x^3 + 5x^2 + 7 \geq 8x^3$ for all positive real numbers x . This is equivalent to saying that $g(x) = x^3$ is $O(8x^3 + 5x^2 + 7)$, which can be established directly by turning the inequality around. \blacktriangleleft

Often, it is important to know the order of growth of a function in terms of some relatively simple reference function such as x^n when n is a positive integer or c^x , where $c > 1$. Knowing the order of growth requires that we have both an upper bound and a lower bound for the size of the function. That is, given a function $f(x)$, we want a reference function $g(x)$ such that $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$. Big-Theta notation, defined as follows, is used to express both of these relationships, providing both an upper and a lower bound on the size of a function.

DEFINITION 3

Let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$. When $f(x)$ is $\Theta(g(x))$ we say that f is big-Theta of $g(x)$, that $f(x)$ is of *order* $g(x)$, and that $f(x)$ and $g(x)$ are of the *same order*.

When $f(x)$ is $\Theta(g(x))$, it is also the case that $g(x)$ is $\Theta(f(x))$. Also note that $f(x)$ is $\Theta(g(x))$ if and only if $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$ (see Exercise 31). Furthermore, note that $f(x)$ is $\Theta(g(x))$ if and only if there are real numbers C_1 and C_2 and a positive real number k such that

$$C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$$

whenever $x > k$. The existence of the constants C_1 , C_2 , and k tells us that $f(x)$ is $\Omega(g(x))$ and that $f(x)$ is $O(g(x))$, respectively.

Usually, when big-Theta notation is used, the function $g(x)$ in $\Theta(g(x))$ is a relatively simple reference function, such as x^n , c^x , $\log x$, and so on, while $f(x)$ can be relatively complicated.

EXAMPLE 11

We showed (in Example 5) that the sum of the first n positive integers is $O(n^2)$. Is this sum of order n^2 ?



Solution: Let $f(n) = 1 + 2 + 3 + \dots + n$. Because we already know that $f(n)$ is $O(n^2)$, to show that $f(n)$ is of order n^2 we need to find a positive constant C such that $f(n) > Cn^2$ for sufficiently large integers n . To obtain a lower bound for this sum, we can ignore the first half of the terms. Summing only the terms greater than $\lceil n/2 \rceil$, we find that

$$\begin{aligned} 1 + 2 + \dots + n &\geq \lceil n/2 \rceil + (\lceil n/2 \rceil + 1) + \dots + n \\ &\geq \lceil n/2 \rceil + \lceil n/2 \rceil + \dots + \lceil n/2 \rceil \\ &= (n - \lceil n/2 \rceil + 1) \lceil n/2 \rceil \\ &\geq (n/2)(n/2) \\ &= n^2/4. \end{aligned}$$

This shows that $f(n)$ is $\Omega(n^2)$. We conclude that $f(n)$ is of order n^2 , or in symbols, $f(n)$ is $\Theta(n^2)$.

EXAMPLE 12

Show that $3x^2 + 8x \log x$ is $\Theta(x^2)$.



Solution: Because $0 \leq 8x \log x \leq 8x^2$, it follows that $3x^2 + 8x \log x \leq 11x^2$ for $x > 1$. Consequently, $3x^2 + 8x \log x$ is $O(x^2)$. Clearly, x^2 is $O(3x^2 + 8x \log x)$. Consequently, $3x^2 + 8x \log x$ is $\Theta(x^2)$.

One useful fact is that the leading term of a polynomial determines its order. For example, if $f(x) = 3x^5 + x^4 + 17x^3 + 2$, then $f(x)$ is of order x^5 . This is stated in Theorem 4, whose proof is left as Exercise 50.

THEOREM 4

Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$, where a_0, a_1, \dots, a_n are real numbers with $a_n \neq 0$. Then $f(x)$ is of order x^n .

EXAMPLE 13

The polynomials $3x^8 + 10x^7 + 221x^2 + 1444$, $x^{19} - 18x^4 - 10,112$, and $-x^{99} + 40,001x^{98} + 100,003x$ are of orders x^8 , x^{19} , and x^{99} , respectively. 

Unfortunately, as Knuth observed, big- O notation is often used by careless writers and speakers as if it had the same meaning as big-Theta notation. Keep this in mind when you see big- O notation used. The recent trend has been to use big-Theta notation whenever both upper and lower bounds on the size of a function are needed.

Exercises

In Exercises 1–14, to establish a big- O relationship, find witnesses C and k such that $|f(x)| \leq C|g(x)|$ whenever $x > k$.

1. Determine whether each of these functions is $O(x)$.
 - a) $f(x) = 10$
 - b) $f(x) = 3x + 7$
 - c) $f(x) = x^2 + x + 1$
 - d) $f(x) = 5 \log x$
 - e) $f(x) = \lfloor x \rfloor$
 - f) $f(x) = \lceil x/2 \rceil$
2. Determine whether each of these functions is $O(x^2)$.
 - a) $f(x) = 17x + 11$
 - b) $f(x) = x^2 + 1000$
 - c) $f(x) = x \log x$
 - d) $f(x) = x^4/2$
 - e) $f(x) = 2^x$
 - f) $f(x) = \lfloor x \rfloor \cdot \lceil x \rceil$
3. Use the definition of “ $f(x)$ is $O(g(x))$ ” to show that $x^4 + 9x^3 + 4x + 7$ is $O(x^4)$.
4. Use the definition of “ $f(x)$ is $O(g(x))$ ” to show that $2^x + 17$ is $O(3^x)$.
5. Show that $(x^2 + 1)/(x + 1)$ is $O(x)$.
6. Show that $(x^3 + 2x)/(2x + 1)$ is $O(x^2)$.
7. Find the least integer n such that $f(x)$ is $O(x^n)$ for each of these functions.
 - a) $f(x) = 2x^3 + x^2 \log x$
 - b) $f(x) = 3x^3 + (\log x)^4$
 - c) $f(x) = (x^4 + x^2 + 1)/(x^3 + 1)$
 - d) $f(x) = (x^4 + 5 \log x)/(x^4 + 1)$
8. Find the least integer n such that $f(x)$ is $O(x^n)$ for each of these functions.
 - a) $f(x) = 2x^2 + x^3 \log x$
 - b) $f(x) = 3x^5 + (\log x)^4$
 - c) $f(x) = (x^4 + x^2 + 1)/(x^4 + 1)$
 - d) $f(x) = (x^3 + 5 \log x)/(x^4 + 1)$
9. Show that $x^2 + 4x + 17$ is $O(x^3)$ but that x^3 is not $O(x^2 + 4x + 17)$.
10. Show that x^3 is $O(x^4)$ but that x^4 is not $O(x^3)$.
11. Show that $3x^4 + 1$ is $O(x^4/2)$ and $x^4/2$ is $O(3x^4 + 1)$.

12. Show that $x \log x$ is $O(x^2)$ but that x^2 is not $O(x \log x)$.
13. Show that 2^n is $O(3^n)$ but that 3^n is not $O(2^n)$. (Note that this is a special case of Exercise 60.)
14. Determine whether x^3 is $O(g(x))$ for each of these functions $g(x)$.
 - a) $g(x) = x^2$
 - b) $g(x) = x^3$
 - c) $g(x) = x^2 + x^3$
 - d) $g(x) = x^2 + x^4$
 - e) $g(x) = 3^x$
 - f) $g(x) = x^3/2$
15. Explain what it means for a function to be $O(1)$.
16. Show that if $f(x)$ is $O(x)$, then $f(x)$ is $O(x^2)$.
17. Suppose that $f(x)$, $g(x)$, and $h(x)$ are functions such that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(h(x))$. Show that $f(x)$ is $O(h(x))$.
18. Let k be a positive integer. Show that $1^k + 2^k + \dots + n^k$ is $O(n^{k+1})$.
19. Determine whether each of the functions 2^{n+1} and 2^{2n} is $O(2^n)$.
20. Determine whether each of the functions $\log(n+1)$ and $\log(n^2+1)$ is $O(\log n)$.
21. Arrange the functions \sqrt{n} , $1000 \log n$, $n \log n$, $2n!$, 2^n , 3^n , and $n^2/1,000,000$ in a list so that each function is big- O of the next function.
22. Arrange the function $(1.5)^n$, n^{100} , $(\log n)^3$, $\sqrt{n} \log n$, 10^n , $(n!)^2$, and $n^{99} + n^{98}$ in a list so that each function is big- O of the next function.
23. Suppose that you have two different algorithms for solving a problem. To solve a problem of size n , the first algorithm uses exactly $n(\log n)$ operations and the second algorithm uses exactly $n^{3/2}$ operations. As n grows, which algorithm uses fewer operations?
24. Suppose that you have two different algorithms for solving a problem. To solve a problem of size n , the first algorithm uses exactly $n^2 2^n$ operations and the second algorithm uses exactly $n!$ operations. As n grows, which algorithm uses fewer operations?

- 25.** Give as good a big- O estimate as possible for each of these functions.
- $(n^2 + 8)(n + 1)$
 - $(n \log n + n^2)(n^3 + 2)$
 - $(n! + 2^n)(n^3 + \log(n^2 + 1))$
- 26.** Give a big- O estimate for each of these functions. For the function g in your estimate $f(x)$ is $O(g(x))$, use a simple function g of smallest order.
- $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$
 - $(2^n + n^2)(n^3 + 3^n)$
 - $(n^n + n2^n + 5^n)(n! + 5^n)$
- 27.** Give a big- O estimate for each of these functions. For the function g in your estimate that $f(x)$ is $O(g(x))$, use a simple function g of the smallest order.
- $n \log(n^2 + 1) + n^2 \log n$
 - $(n \log n + 1)^2 + (\log n + 1)(n^2 + 1)$
 - $n^{2^n} + n^{n^2}$
- 28.** For each function in Exercise 1, determine whether that function is $\Omega(x)$ and whether it is $\Theta(x)$.
- 29.** For each function in Exercise 2, determine whether that function is $\Omega(x^2)$ and whether it is $\Theta(x^2)$.
- 30.** Show that each of these pairs of functions are of the same order.
- $3x + 7, x$
 - $2x^2 + x - 7, x^2$
 - $\lfloor x + 1/2 \rfloor, x$
 - $\log(x^2 + 1), \log_2 x$
 - $\log_{10} x, \log_2 x$
- 31.** Show that $f(x)$ is $\Theta(g(x))$ if and only if $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$.
- 32.** Show that if $f(x)$ and $g(x)$ are functions from the set of real numbers to the set of real numbers, then $f(x)$ is $O(g(x))$ if and only if $g(x)$ is $\Omega(f(x))$.
- 33.** Show that if $f(x)$ and $g(x)$ are functions from the set of real numbers to the set of real numbers, then $f(x)$ is $\Theta(g(x))$ if and only if there are positive constants k , C_1 , and C_2 such that $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$ whenever $x > k$.
- 34.** **a)** Show that $3x^2 + x + 1$ is $\Theta(3x^2)$ by directly finding the constants k , C_1 , and C_2 in Exercise 33.
b) Express the relationship in part (a) using a picture showing the functions $3x^2 + x + 1$, $C_1 \cdot 3x^2$, and $C_2 \cdot 3x^2$, and the constant k on the x -axis, where C_1 , C_2 , and k are the constants you found in part (a) to show that $3x^2 + x + 1$ is $\Theta(3x^2)$.
- 35.** Express the relationship $f(x)$ is $\Theta(g(x))$ using a picture. Show the graphs of the functions $f(x)$, $C_1|g(x)|$, and $C_2|g(x)|$, as well as the constant k on the x -axis.
- 36.** Explain what it means for a function to be $\Omega(1)$.
- 37.** Explain what it means for a function to be $\Theta(1)$.
- 38.** Give a big- O estimate of the product of the first n odd positive integers.
- 39.** Show that if f and g are real-valued functions such that $f(x)$ is $O(g(x))$, then for every positive integer n , $f^n(x)$ is $O(g^n(x))$. [Note that $f^n(x) = f(x)^n$.]
- 40.** Show that for all real numbers a and b with $a > 1$ and $b > 1$, if $f(x)$ is $O(\log_b x)$, then $f(x)$ is $O(\log_a x)$.
- 41.** Suppose that $f(x)$ is $O(g(x))$ where f and g are increasing and unbounded functions. Show that $\log |f(x)|$ is $O(\log |g(x)|)$.
- 42.** Suppose that $f(x)$ is $O(g(x))$. Does it follow that $2^{f(x)}$ is $O(2^{g(x)})$?
- 43.** Let $f_1(x)$ and $f_2(x)$ be functions from the set of real numbers to the set of positive real numbers. Show that if $f_1(x)$ and $f_2(x)$ are both $\Theta(g(x))$, where $g(x)$ is a function from the set of real numbers to the set of positive real numbers, then $f_1(x) + f_2(x)$ is $\Theta(g(x))$. Is this still true if $f_1(x)$ and $f_2(x)$ can take negative values?
- 44.** Suppose that $f(x)$, $g(x)$, and $h(x)$ are functions such that $f(x)$ is $\Theta(g(x))$ and $g(x)$ is $\Theta(h(x))$. Show that $f(x)$ is $\Theta(h(x))$.
- 45.** If $f_1(x)$ and $f_2(x)$ are functions from the set of positive integers to the set of positive real numbers and $f_1(x)$ and $f_2(x)$ are both $\Theta(g(x))$, is $(f_1 - f_2)(x)$ also $\Theta(g(x))$? Either prove that it is or give a counterexample.
- 46.** Show that if $f_1(x)$ and $f_2(x)$ are functions from the set of positive integers to the set of real numbers and $f_1(x)$ is $\Theta(g_1(x))$ and $f_2(x)$ is $\Theta(g_2(x))$, then $(f_1 f_2)(x)$ is $\Theta((g_1 g_2)(x))$.
- 47.** Find functions f and g from the set of positive integers to the set of real numbers such that $f(n)$ is not $O(g(n))$ and $g(n)$ is not $O(f(n))$.
- 48.** Express the relationship $f(x)$ is $\Omega(g(x))$ using a picture. Show the graphs of the functions $f(x)$ and $Cg(x)$, as well as the constant k on the real axis.
- 49.** Show that if $f_1(x)$ is $\Theta(g_1(x))$, $f_2(x)$ is $\Theta(g_2(x))$, and $f_2(x) \neq 0$ and $g_2(x) \neq 0$ for all real numbers $x > 0$, then $(f_1/f_2)(x)$ is $\Theta((g_1/g_2)(x))$.
- 50.** Show that if $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$, where a_0, a_1, \dots, a_{n-1} , and a_n are real numbers and $a_n \neq 0$, then $f(x)$ is $\Theta(x^n)$.
- Big- O , big- Θ , and big- Ω notation can be extended to functions in more than one variable. For example, the statement $f(x, y)$ is $O(g(x, y))$ means that there exist constants C , k_1 , and k_2 such that $|f(x, y)| \leq C|g(x, y)|$ whenever $x > k_1$ and $y > k_2$.
- 51.** Define the statement $f(x, y)$ is $\Theta(g(x, y))$.
- 52.** Define the statement $f(x, y)$ is $\Omega(g(x, y))$.
- 53.** Show that $(x^2 + xy + x \log y)^3$ is $O(x^6 y^3)$.
- 54.** Show that $x^5 y^3 + x^4 y^4 + x^3 y^5$ is $\Omega(x^3 y^3)$.
- 55.** Show that $\lfloor xy \rfloor$ is $O(xy)$.
- 56.** Show that $\lceil xy \rceil$ is $\Omega(xy)$.
- 57.** (Requires calculus) Show that if $c > d > 0$, then n^d is $O(n^c)$, but n^c is not $O(n^d)$.
- 58.** (Requires calculus) Show that if $b > 1$ and c and d are positive, then $(\log_b n)^c$ is $O(n^d)$, but n^d is not $O((\log_b n)^c)$.

- 59.** (*Requires calculus*) Show that if d is positive and $b > 1$, then n^d is $O(b^n)$ but b^n is not $O(n^d)$.
- 60.** (*Requires calculus*) Show that if $c > b > 1$, then b^n is $O(c^n)$ but c^n is not $O(b^n)$.

The following problems deal with another type of asymptotic notation, called **little- o** notation. Because little- o notation is based on the concept of limits, a knowledge of calculus is needed for these problems. We say that $f(x)$ is $o(g(x))$ [read $f(x)$ is “little-oh” of $g(x)$], when

$$\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0.$$

- 61.** (*Requires calculus*) Show that

- a) x^2 is $o(x^3)$. b) $x \log x$ is $o(x^2)$.
 c) x^2 is $o(2^x)$. d) $x^2 + x + 1$ is not $o(x^2)$.

- 62.** (*Requires calculus*)

- a) Show that if $f(x)$ and $g(x)$ are functions such that $f(x)$ is $o(g(x))$ and c is a constant, then $cf(x)$ is $o(g(x))$, where $(cf)(x) = cf(x)$.
 b) Show that if $f_1(x)$, $f_2(x)$, and $g(x)$ are functions such that $f_1(x)$ is $o(g(x))$ and $f_2(x)$ is $o(g(x))$, then $(f_1 + f_2)(x)$ is $o(g(x))$, where $(f_1 + f_2)(x) = f_1(x) + f_2(x)$.

- 63.** (*Requires calculus*) Represent pictorially that $x \log x$ is $o(x^2)$ by graphing $x \log x$, x^2 , and $x \log x/x^2$. Explain how this picture shows that $x \log x$ is $o(x^2)$.

- 64.** (*Requires calculus*) Express the relationship $f(x)$ is $o(g(x))$ using a picture. Show the graphs of $f(x)$, $g(x)$, and $f(x)/g(x)$.

- *65.** (*Requires calculus*) Suppose that $f(x)$ is $o(g(x))$. Does it follow that $2^{f(x)}$ is $o(2^{g(x)})$?

- *66.** (*Requires calculus*) Suppose that $f(x)$ is $o(g(x))$. Does it follow that $\log |f(x)|$ is $o(\log |g(x)|)$?

- 67.** (*Requires calculus*) The two parts of this exercise describe the relationship between little- o and big- O notation.

- a) Show that if $f(x)$ and $g(x)$ are functions such that $f(x)$ is $o(g(x))$, then $f(x)$ is $O(g(x))$.
 b) Show that if $f(x)$ and $g(x)$ are functions such that $f(x)$ is $O(g(x))$, then it does not necessarily follow that $f(x)$ is $o(g(x))$.

- 68.** (*Requires calculus*) Show that if $f(x)$ is a polynomial of degree n and $g(x)$ is a polynomial of degree m where $m > n$, then $f(x)$ is $o(g(x))$.

- 69.** (*Requires calculus*) Show that if $f_1(x)$ is $O(g(x))$ and $f_2(x)$ is $o(g(x))$, then $f_1(x) + f_2(x)$ is $O(g(x))$.

- 70.** (*Requires calculus*) Let H_n be the n th **harmonic number**

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

Show that H_n is $O(\log n)$. [*Hint:* First establish the inequality

$$\sum_{j=2}^n \frac{1}{j} < \int_1^n \frac{1}{x} dx$$

by showing that the sum of the areas of the rectangles of height $1/j$ with base from $j - 1$ to j , for $j = 2, 3, \dots, n$, is less than the area under the curve $y = 1/x$ from 2 to n .]

- *71.** Show that $n \log n$ is $O(\log n!)$.

- 72.** Determine whether $\log n!$ is $\Theta(n \log n)$. Justify your answer.

- *73.** Show that $\log n!$ is greater than $(n \log n)/4$ for $n > 4$. [*Hint:* Begin with the inequality $n! > n(n - 1)(n - 2) \cdots [n/2]$.]

Let $f(x)$ and $g(x)$ be functions from the set of real numbers to the set of real numbers. We say that the functions f and g are **asymptotic** and write $f(x) \sim g(x)$ if $\lim_{x \rightarrow \infty} f(x)/g(x) = 1$.

- 74.** (*Requires calculus*) For each of these pairs of functions, determine whether f and g are asymptotic.

- a) $f(x) = x^2 + 3x + 7$, $g(x) = x^2 + 10$
 b) $f(x) = x^2 \log x$, $g(x) = x^3$
 c) $f(x) = x^4 + \log(3x^8 + 7)$,
 $g(x) = (x^2 + 17x + 3)^2$
 d) $f(x) = (x^3 + x^2 + x + 1)^4$,
 $g(x) = (x^4 + x^3 + x^2 + x + 1)^3$.

- 75.** (*Requires calculus*) For each of these pairs of functions, determine whether f and g are asymptotic.

- a) $f(x) = \log(x^2 + 1)$, $g(x) = \log x$
 b) $f(x) = 2^{x+3}$, $g(x) = 2^{x+7}$
 c) $f(x) = 2^{2^x}$, $g(x) = 2^{x^2}$
 d) $f(x) = 2^{x^2+x+1}$, $g(x) = 2^{x^2+2x}$

3.3 Complexity of Algorithms

Introduction

When does an algorithm provide a satisfactory solution to a problem? First, it must always produce the correct answer. How this can be demonstrated will be discussed in Chapter 5. Second, it should be efficient. The efficiency of algorithms will be discussed in this section.

How can the efficiency of an algorithm be analyzed? One measure of efficiency is the time used by a computer to solve a problem using the algorithm, when input values are of a specified

size. A second measure is the amount of computer memory required to implement the algorithm when input values are of a specified size.

Questions such as these involve the **computational complexity** of the algorithm. An analysis of the time required to solve a problem of a particular size involves the **time complexity** of the algorithm. An analysis of the computer memory required involves the **space complexity** of the algorithm. Considerations of the time and space complexity of an algorithm are essential when algorithms are implemented. It is obviously important to know whether an algorithm will produce an answer in a microsecond, a minute, or a billion years. Likewise, the required memory must be available to solve a problem, so that space complexity must be taken into account.

Considerations of space complexity are tied in with the particular data structures used to implement the algorithm. Because data structures are not dealt with in detail in this book, space complexity will not be considered. We will restrict our attention to time complexity.

Time Complexity

The time complexity of an algorithm can be expressed in terms of the number of operations used by the algorithm when the input has a particular size. The operations used to measure time complexity can be the comparison of integers, the addition of integers, the multiplication of integers, the division of integers, or any other basic operation.

Time complexity is described in terms of the number of operations required instead of actual computer time because of the difference in time needed for different computers to perform basic operations. Moreover, it is quite complicated to break all operations down to the basic bit operations that a computer uses. Furthermore, the fastest computers in existence can perform basic bit operations (for instance, adding, multiplying, comparing, or exchanging two bits) in 10^{-11} second (10 picoseconds), but personal computers may require 10^{-8} second (10 nanoseconds), which is 1000 times as long, to do the same operations.

We illustrate how to analyze the time complexity of an algorithm by considering Algorithm 1 of Section 3.1, which finds the maximum of a finite set of integers.

EXAMPLE 1 Describe the time complexity of Algorithm 1 of Section 3.1 for finding the maximum element in a finite set of integers.



Solution: The number of comparisons will be used as the measure of the time complexity of the algorithm, because comparisons are the basic operations used.

To find the maximum element of a set with n elements, listed in an arbitrary order, the temporary maximum is first set equal to the initial term in the list. Then, after a comparison $i \leq n$ has been done to determine that the end of the list has not yet been reached, the temporary maximum and second term are compared, updating the temporary maximum to the value of the second term if it is larger. This procedure is continued, using two additional comparisons for each term of the list—one $i \leq n$, to determine that the end of the list has not been reached and another $\max < a_i$, to determine whether to update the temporary maximum. Because two comparisons are used for each of the second through the n th elements and one more comparison is used to exit the loop when $i = n + 1$, exactly $2(n - 1) + 1 = 2n - 1$ comparisons are used whenever this algorithm is applied. Hence, the algorithm for finding the maximum of a set of n elements has time complexity $\Theta(n)$, measured in terms of the number of comparisons used. Note that for this algorithm the number of comparisons is independent of particular input of n numbers.

Next, we will analyze the time complexity of searching algorithms.

EXAMPLE 2 Describe the time complexity of the linear search algorithm (specified as Algorithm 2 in Section 3.1).

Solution: The number of comparisons used by Algorithm 2 in Section 3.1 will be taken as the measure of the time complexity. At each step of the loop in the algorithm, two comparisons are performed—one $i \leq n$, to see whether the end of the list has been reached and one $x \leq a_i$, to compare the element x with a term of the list. Finally, one more comparison $i \leq n$ is made outside the loop. Consequently, if $x = a_i$, $2i + 1$ comparisons are used. The most comparisons, $2n + 2$, are required when the element is not in the list. In this case, $2n$ comparisons are used to determine that x is not a_i , for $i = 1, 2, \dots, n$, an additional comparison is used to exit the loop, and one comparison is made outside the loop. So when x is not in the list, a total of $2n + 2$ comparisons are used. Hence, a linear search requires $\Theta(n)$ comparisons in the worst case, because $2n + 2$ is $\Theta(n)$. 

WORST-CASE COMPLEXITY The type of complexity analysis done in Example 2 is a **worst-case** analysis. By the worst-case performance of an algorithm, we mean the largest number of operations needed to solve the given problem using this algorithm on input of specified size. Worst-case analysis tells us how many operations an algorithm requires to guarantee that it will produce a solution.

EXAMPLE 3 Describe the time complexity of the binary search algorithm (specified as Algorithm 3 in Section 3.1) in terms of the number of comparisons used (and ignoring the time required to compute $m = \lfloor (i + j)/2 \rfloor$ in each iteration of the loop in the algorithm).

Solution: For simplicity, assume there are $n = 2^k$ elements in the list a_1, a_2, \dots, a_n , where k is a nonnegative integer. Note that $k = \log n$. (If n , the number of elements in the list, is not a power of 2, the list can be considered part of a larger list with 2^{k+1} elements, where $2^k < n < 2^{k+1}$. Here 2^{k+1} is the smallest power of 2 larger than n .)

At each stage of the algorithm, i and j , the locations of the first term and the last term of the restricted list at that stage, are compared to see whether the restricted list has more than one term. If $i < j$, a comparison is done to determine whether x is greater than the middle term of the restricted list.

At the first stage the search is restricted to a list with 2^{k-1} terms. So far, two comparisons have been used. This procedure is continued, using two comparisons at each stage to restrict the search to a list with half as many terms. In other words, two comparisons are used at the first stage of the algorithm when the list has 2^k elements, two more when the search has been reduced to a list with 2^{k-1} elements, two more when the search has been reduced to a list with 2^{k-2} elements, and so on, until two comparisons are used when the search has been reduced to a list with $2^1 = 2$ elements. Finally, when one term is left in the list, one comparison tells us that there are no additional terms left, and one more comparison is used to determine if this term is x .

Hence, at most $2k + 2 = 2\log n + 2$ comparisons are required to perform a binary search when the list being searched has 2^k elements. (If n is not a power of 2, the original list is expanded to a list with 2^{k+1} terms, where $k = \lfloor \log n \rfloor$, and the search requires at most $2 \lceil \log n \rceil + 2$ comparisons.) It follows that in the worst case, binary search requires $O(\log n)$ comparisons. Note that in the worst case, $2\log n + 2$ comparisons are used by the binary search. Hence, the binary search uses $\Theta(\log n)$ comparisons in the worst case, because $2\log n + 2 = \Theta(\log n)$. From this analysis it follows that in the worst case, the binary search algorithm is more efficient than the linear search algorithm, because we know by Example 2 that the linear search algorithm has $\Theta(n)$ worst-case time complexity. 

AVERAGE-CASE COMPLEXITY Another important type of complexity analysis, besides worst-case analysis, is called **average-case** analysis. The average number of operations used to solve the problem over all possible inputs of a given size is found in this type of analysis. Average-case time complexity analysis is usually much more complicated than worst-case analysis.

However, the average-case analysis for the linear search algorithm can be done without difficulty, as shown in Example 4.

EXAMPLE 4 Describe the average-case performance of the linear search algorithm in terms of the average number of comparisons used, assuming that the integer x is in the list and it is equally likely that x is in any position.

Solution: By hypothesis, the integer x is one of the integers a_1, a_2, \dots, a_n in the list. If x is the first term a_1 of the list, three comparisons are needed, one $i \leq n$ to determine whether the end of the list has been reached, one $x \neq a_i$ to compare x and the first term, and one $i \leq n$ outside the loop. If x is the second term a_2 of the list, two more comparisons are needed, so that a total of five comparisons are used. In general, if x is the i th term of the list a_i , two comparisons will be used at each of the i steps of the loop, and one outside the loop, so that a total of $2i + 1$ comparisons are needed. Hence, the average number of comparisons used equals

$$\frac{3 + 5 + 7 + \dots + (2n + 1)}{n} = \frac{2(1 + 2 + 3 + \dots + n) + n}{n}.$$

Using the formula from line 2 of Table 2 in Section 2.4 (and see Exercise 37(b) of Section 2.4),

$$1 + 2 + 3 + \dots + n = \frac{n(n + 1)}{2}.$$

Hence, the average number of comparisons used by the linear search algorithm (when x is known to be in the list) is

$$\frac{2[n(n + 1)/2]}{n} + 1 = n + 2,$$

which is $\Theta(n)$. 

Remark: In the analysis in Example 4 we assumed that x is in the list being searched. It is also possible to do an average-case analysis of this algorithm when x may not be in the list (see Exercise 23).

Remark: Although we have counted the comparisons needed to determine whether we have reached the end of a loop, these comparisons are often not counted. From this point on we will ignore such comparisons.

WORST-CASE COMPLEXITY OF TWO SORTING ALGORITHMS We analyze the worst-case complexity of the bubble sort and the insertion sort in Examples 5 and 6.

EXAMPLE 5 What is the worst-case complexity of the bubble sort in terms of the number of comparisons made?

Solution: The bubble sort described before Example 4 in Section 3.1 sorts a list by performing a sequence of passes through the list. During each pass the bubble sort successively compares adjacent elements, interchanging them if necessary. When the i th pass begins, the $i - 1$ largest elements are guaranteed to be in the correct positions. During this pass, $n - i$ comparisons are used. Consequently, the total number of comparisons used by the bubble sort to order a list of n elements is

$$(n - 1) + (n - 2) + \dots + 2 + 1 = \frac{(n - 1)n}{2}$$

using a summation formula from line 2 in Table 2 in Section 2.4 (and Exercise 37(b) in Section 2.4). Note that the bubble sort always uses this many comparisons, because it continues even if the list becomes completely sorted at some intermediate step. Consequently, the bubble sort uses $(n - 1)n/2$ comparisons, so it has $\Theta(n^2)$ worst-case complexity in terms of the number of comparisons used. 

EXAMPLE 6 What is the worst-case complexity of the insertion sort in terms of the number of comparisons made?

Solution: The insertion sort (described in Section 3.1) inserts the j th element into the correct position among the first $j - 1$ elements that have already been put into the correct order. It does this by using a linear search technique, successively comparing the j th element with successive terms until a term that is greater than or equal to it is found or it compares a_j with itself and stops because a_j is not less than itself. Consequently, in the worst case, j comparisons are required to insert the j th element into the correct position. Therefore, the total number of comparisons used by the insertion sort to sort a list of n elements is

$$2 + 3 + \dots + n = \frac{n(n + 1)}{2} - 1,$$

using the summation formula for the sum of consecutive integers in line 2 of Table 2 of Section 2.4 (and see Exercise 37(b) of Section 2.4), and noting that the first term, 1, is missing in this sum. Note that the insertion sort may use considerably fewer comparisons if the smaller elements started out at the end of the list. We conclude that the insertion sort has worst-case complexity $\Theta(n^2)$. 

In Examples 5 and 6 we showed that both the bubble sort and the insertion sort have worst-case time complexity $\Theta(n^2)$. However, the most efficient sorting algorithms can sort n items in $O(n \log n)$ time, as we will show in Sections 8.3 and 11.1 using techniques we develop in those sections. From this point on, we will assume that sorting n items can be done in $O(n \log n)$ time.

Complexity of Matrix Multiplication

The definition of the product of two matrices can be expressed as an algorithm for computing the product of two matrices. Suppose that $\mathbf{C} = [c_{ij}]$ is the $m \times n$ matrix that is the product of the $m \times k$ matrix $\mathbf{A} = [a_{ij}]$ and the $k \times n$ matrix $\mathbf{B} = [b_{ij}]$. The algorithm based on the definition of the matrix product is expressed in pseudocode in Algorithm 1.

ALGORITHM 1 Matrix Multiplication.

```

procedure matrix multiplication(A, B: matrices)
for i := 1 to m
    for j := 1 to n
        cij := 0
        for q := 1 to k
            cij := cij + aiqbqj
return C {C = [cij] is the product of A and B}

```

We can determine the complexity of this algorithm in terms of the number of additions and multiplications used.

EXAMPLE 7 How many additions of integers and multiplications of integers are used by Algorithm 1 to multiply two $n \times n$ matrices with integer entries?

Solution: There are n^2 entries in the product of **A** and **B**. To find each entry requires a total of n multiplications and $n - 1$ additions. Hence, a total of n^3 multiplications and $n^2(n - 1)$ additions are used. \blacktriangleleft

Surprisingly, there are more efficient algorithms for matrix multiplication than that given in Algorithm 1. As Example 7 shows, multiplying two $n \times n$ matrices directly from the definition requires $O(n^3)$ multiplications and additions. Using other algorithms, two $n \times n$ matrices can be multiplied using $O(n^{\sqrt{7}})$ multiplications and additions. (Details of such algorithms can be found in [CoLeRiSt09].)

We can also analyze the complexity of the algorithm we described in Chapter 2 for computing the Boolean product of two matrices, which we display as Algorithm 2.

ALGORITHM 2 *The Boolean Product of Zero-One Matrices.*

```

procedure Boolean product of Zero-One Matrices (A, B: zero–one matrices)
for  $i := 1$  to  $m$ 
    for  $j := 1$  to  $n$ 
         $c_{ij} := 0$ 
        for  $q := 1$  to  $k$ 
             $c_{ij} := c_{ij} \vee (a_{iq} \wedge b_{qj})$ 
return C {C =  $[c_{ij}]$  is the Boolean product of A and B}
```

The number of bit operations used to find the Boolean product of two $n \times n$ matrices can be easily determined.

EXAMPLE 8 How many bit operations are used to find $\mathbf{A} \odot \mathbf{B}$, where **A** and **B** are $n \times n$ zero–one matrices?

Solution: There are n^2 entries in $\mathbf{A} \odot \mathbf{B}$. Using Algorithm 2, a total of n *ORs* and n *ANDs* are used to find an entry of $\mathbf{A} \odot \mathbf{B}$. Hence, $2n$ bit operations are used to find each entry. Therefore, $2n^3$ bit operations are required to compute $\mathbf{A} \odot \mathbf{B}$ using Algorithm 2. \blacktriangleleft



MATRIX-CHAIN MULTIPLICATION There is another important problem involving the complexity of the multiplication of matrices. How should the **matrix-chain** $\mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_n$ be computed using the fewest multiplications of integers, where $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ are $m_1 \times m_2, m_2 \times m_3, \dots, m_n \times m_{n+1}$ matrices, respectively, and each has integers as entries? (Because matrix multiplication is associative, as shown in Exercise 13 in Section 2.6, the order of the multiplication used does not change the product.) Note that $m_1 m_2 m_3$ multiplications of integers are performed to multiply an $m_1 \times m_2$ matrix and an $m_2 \times m_3$ matrix using Algorithm 1. Example 9 illustrates this problem.

EXAMPLE 9 In which order should the matrices **A**₁, **A**₂, and **A**₃—where **A**₁ is 30×20 , **A**₂ is 20×40 , and **A**₃ is 40×10 , all with integer entries—be multiplied to use the least number of multiplications of integers?

Solution: There are two possible ways to compute $\mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3$. These are $\mathbf{A}_1 (\mathbf{A}_2 \mathbf{A}_3)$ and $(\mathbf{A}_1 \mathbf{A}_2) \mathbf{A}_3$.

If **A**₂ and **A**₃ are first multiplied, a total of $20 \cdot 40 \cdot 10 = 8000$ multiplications of integers are used to obtain the 20×10 matrix **A**₂**A**₃. Then, to multiply **A**₁ and **A**₂**A**₃ requires $30 \cdot 20 \cdot 10 = 6000$ multiplications. Hence, a total of

$$8000 + 6000 = 14,000$$

multiplications are used. On the other hand, if \mathbf{A}_1 and \mathbf{A}_2 are first multiplied, then $30 \cdot 20 \cdot 40 = 24,000$ multiplications are used to obtain the 30×40 matrix $\mathbf{A}_1\mathbf{A}_2$. Then, to multiply $\mathbf{A}_1\mathbf{A}_2$ and \mathbf{A}_3 requires $30 \cdot 40 \cdot 10 = 12,000$ multiplications. Hence, a total of

$$24,000 + 12,000 = 36,000$$

multiplications are used.

Clearly, the first method is more efficient. 

We will return to this problem in Exercise 57 in Section 8.1. Algorithms for determining the most efficient way to carry out matrix-chain multiplication are discussed in [CoLeRiSt09].

Algorithmic Paradigms

In Section 3.1 we introduced the basic notion of an algorithm. We provided examples of many different algorithms, including searching and sorting algorithms. We also introduced the concept of a greedy algorithm, giving examples of several problems that can be solved by greedy algorithms. Greedy algorithms provide an example of an **algorithmic paradigm**, that is, a general approach based on a particular concept that can be used to construct algorithms for solving a variety of problems.

In this book we will construct algorithms for solving many different problems based on a variety of algorithmic paradigms, including the most widely used algorithmic paradigms. These paradigms can serve as the basis for constructing efficient algorithms for solving a wide range of problems.

Some of the algorithms we have already studied are based on an algorithmic paradigm known as brute force, which we will describe in this section. Algorithmic paradigms, studied later in this book, include divide-and-conquer algorithms studied in Chapter 8, dynamic programming, also studied in Chapter 8, backtracking, studied in Chapter 10, and probabilistic algorithms, studied in Chapter 7. There are many important algorithmic paradigms besides those described in this book. Consult books on algorithm design such as [KITA06] to learn more about them.

BRUTE-FORCE ALGORITHMS Brute force is an important, and basic, algorithmic paradigm. In a **brute-force algorithm**, a problem is solved in the most straightforward manner based on the statement of the problem and the definitions of terms. Brute-force algorithms are designed to solve problems without regard to the computing resources required. For example, in some brute-force algorithms the solution to a problem is found by examining every possible solution, looking for the best possible. In general, brute-force algorithms are naive approaches for solving problems that do not take advantage of any special structure of the problem or clever ideas.

Note that Algorithm 1 in Section 3.1 for finding the maximum number in a sequence is a brute-force algorithm because it examines each of the n numbers in a sequence to find the maximum term. The algorithm for finding the sum of n numbers by adding one additional number at a time is also a brute-force algorithm, as is the algorithm for matrix multiplication based on its definition (Algorithm 1). The bubble, insertion, and selection sorts (described in Section 3.1 in Algorithms 4 and 5 and in Exercise 42, respectively) are also considered to be brute-force algorithms; all three of these sorting algorithms are straightforward approaches much less efficient than other sorting algorithms such as the merge sort and the quick sort discussed in Chapters 5 and 8.

Although brute-force algorithms are often inefficient, they are often quite useful. A brute-force algorithm may be able to solve practical instances of problems, particularly when the input

is not too large, even if it is impractical to use this algorithm for larger inputs. Furthermore, when designing new algorithms to solve a problem, the goal is often to find a new algorithm that is more efficient than a brute-force algorithm. One such problem of this type is described in Example 10.

EXAMPLE 10 Construct a brute-force algorithm for finding the closest pair of points in a set of n points in the plane and provide a worst-case big- O estimate for the number of bit operations used by the algorithm.

Solution: Suppose that we are given as input the points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Recall that the distance between (x_i, y_i) and (x_j, y_j) is $\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$. A brute-force algorithm can find the closest pair of these points by computing the distances between all pairs of the n points and determining the smallest distance. (We can make one small simplification to make the computation easier; we can compute the square of the distance between pairs of points to find the closest pair, rather than the distance between these points. We can do this because the square of the distance between a pair of points is smallest when the distance between these points is smallest.)

ALGORITHM 3 Brute-Force Algorithm for Closest Pair of Points.

```

procedure closest-pair( $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ : pairs of real numbers)
   $min = \infty$ 
  for  $i := 2$  to  $n$ 
    for  $j := 1$  to  $i - 1$ 
      if  $(x_j - x_i)^2 + (y_j - y_i)^2 < min$  then
         $min := (x_j - x_i)^2 + (y_j - y_i)^2$ 
         $closest\ pair := ((x_i, y_i), (x_j, y_j))$ 
  return  $closest\ pair$ 

```

To estimate the number of operations used by the algorithm, first note that there are $n(n - 1)/2$ pairs of points $((x_i, y_i), (x_j, y_j))$ that we loop through (as the reader should verify). For each such pair we compute $(x_j - x_i)^2 + (y_j - y_i)^2$, compare it with the current value of min , and if it is smaller than min replace the current value of min by this new value. It follows that this algorithm uses $\Theta(n^2)$ operations, in terms of arithmetic operations and comparisons.

In Chapter 8 we will devise an algorithm that determines the closest pair of points when given n points in the plane as input that has $O(n \log n)$ worst-case complexity. The original discovery of such an algorithm, much more efficient than the brute-force approach, was considered quite surprising. 

Understanding the Complexity of Algorithms

Table 1 displays some common terminology used to describe the time complexity of algorithms. For example, an algorithm that finds the largest of the first 100 terms of a list of n elements by applying Algorithm 1 to the sequence of the first 100 terms, where n is an integer with $n \geq 100$, has **constant complexity** because it uses 99 comparisons no matter what n is (as the reader can verify). The linear search algorithm has **linear** (worst-case or average-case) **complexity** and the binary search algorithm has **logarithmic** (worst-case) **complexity**. Many important algorithms have $n \log n$, or **linearithmic** (worst-case) **complexity**, such as the merge sort, which we will introduce in Chapter 4. (The word *linearithmic* is a combination of the words *linear* and *logarithmic*.)

TABLE 1 Commonly Used Terminology for the Complexity of Algorithms.

Complexity	Terminology
$\Theta(1)$	Constant complexity
$\Theta(\log n)$	Logarithmic complexity
$\Theta(n)$	Linear complexity
$\Theta(n \log n)$	Linearithmic complexity
$\Theta(n^b)$	Polynomial complexity
$\Theta(b^n)$, where $b > 1$	Exponential complexity
$\Theta(n!)$	Factorial complexity

An algorithm has **polynomial complexity** if it has complexity $\Theta(n^b)$, where b is an integer with $b \geq 1$. For example, the bubble sort algorithm is a polynomial-time algorithm because it uses $\Theta(n^2)$ comparisons in the worst case. An algorithm has **exponential complexity** if it has time complexity $\Theta(b^n)$, where $b > 1$. The algorithm that determines whether a compound proposition in n variables is satisfiable by checking all possible assignments of truth variables is an algorithm with exponential complexity, because it uses $\Theta(2^n)$ operations. Finally, an algorithm has **factorial complexity** if it has $\Theta(n!)$ time complexity. The algorithm that finds all orders that a traveling salesperson could use to visit n cities has factorial complexity; we will discuss this algorithm in Chapter 9.

TRACTABILITY A problem that is solvable using an algorithm with polynomial worst-case complexity is called **tractable**, because the expectation is that the algorithm will produce the solution to the problem for reasonably sized input in a relatively short time. However, if the polynomial in the big- Θ estimate has high degree (such as degree 100) or if the coefficients are extremely large, the algorithm may take an extremely long time to solve the problem. Consequently, that a problem can be solved using an algorithm with polynomial worst-case time complexity is no guarantee that the problem can be solved in a reasonable amount of time for even relatively small input values. Fortunately, in practice, the degree and coefficients of polynomials in such estimates are often small.

The situation is much worse for problems that cannot be solved using an algorithm with worst-case polynomial time complexity. Such problems are called **intractable**. Usually, but not always, an extremely large amount of time is required to solve the problem for the worst cases of even small input values. In practice, however, there are situations where an algorithm with a certain worst-case time complexity may be able to solve a problem much more quickly for most cases than for its worst case. When we are willing to allow that some, perhaps small, number of cases may not be solved in a reasonable amount of time, the average-case time complexity is a better measure of how long an algorithm takes to solve a problem. Many problems important in industry are thought to be intractable but can be practically solved for essentially all sets of input that arise in daily life. Another way that intractable problems are handled when they arise in practical applications is that instead of looking for exact solutions of a problem, approximate solutions are sought. It may be the case that fast algorithms exist for finding such approximate solutions, perhaps even with a guarantee that they do not differ by very much from an exact solution.

Some problems even exist for which it can be shown that no algorithm exists for solving them. Such problems are called **unsolvable** (as opposed to **solvable** problems that can be solved using an algorithm). The first proof that there are unsolvable problems was provided by the great English mathematician and computer scientist Alan Turing when he showed that the halting problem is unsolvable. Recall that we proved that the halting problem is unsolvable in Section 3.1. (A biography of Alan Turing and a description of some of his other work can be found in Chapter 13.)

P VERSUS NP The study of the complexity of algorithms goes far beyond what we can describe here. Note, however, that many solvable problems are believed to have the property that no algorithm with polynomial worst-case time complexity solves them, but that a solution, if known, can be checked in polynomial time. Problems for which a solution can be checked in polynomial time are said to belong to the **class NP** (tractable problems are said to belong to **class P**). The abbreviation NP stands for *nondeterministic polynomial* time. The satisfiability problem, discussed in Section 1.3, is an example of an NP problem—we can quickly verify that an assignment of truth values to the variables of a compound proposition makes it true, but no polynomial time algorithm has been discovered for finding such an assignment of truth values. (For example, an exhaustive search of all possible truth values requires $\Omega(2^n)$ bit operations where n is the number of variables in the compound proposition.)

There is also an important class of problems, called **NP-complete problems**, with the property that if any of these problems can be solved by a polynomial worst-case time algorithm, then all problems in the class NP can be solved by polynomial worst-case time algorithms.

The satisfiability problem, is also an example of an NP-complete problem. It is an NP problem and if a polynomial time algorithm for solving it were known, there would be polynomial time algorithms for all problems known to be in this class of problems (and there are many important problems in this class). This last statement follows from the fact that every problem in NP can be reduced in polynomial time to the satisfiability problem. Although more than 3000 NP-complete problems are now known, the satisfiability problem was the first problem shown to be NP-complete. The theorem that asserts this is known as the **Cook-Levin theorem** after Stephen Cook and Leonid Levin, who independently proved it in the early 1970s.

The **P versus NP problem** asks whether NP, the class of problems for which it is possible to check solutions in polynomial time, equals P, the class of tractable problems. If $P \neq NP$, there would be some problems that cannot be solved in polynomial time, but whose solutions could be verified in polynomial time. The concept of NP-completeness is helpful in research aimed at solving the P versus NP problem, because NP-complete problems are the problems in NP considered most likely not to be in P, as every problem in NP can be reduced to an NP-complete problem in polynomial time. A large majority of theoretical computer scientists believe that $P \neq NP$, which would mean that no NP-complete problem can be solved in polynomial time. One reason for this belief is that despite extensive research, no one has succeeded in showing that $P = NP$. In particular, no one has been able to find an algorithm with worst-case polynomial time complexity that solves any NP-complete problem. The P versus NP problem is one of the most famous unsolved problems in the mathematical sciences (which include theoretical computer science). It is one of the seven famous Millennium Prize Problems, of which six remain unsolved. A prize of \$1,000,000 is offered by the Clay Mathematics Institute for its solution.



STEPHEN COOK (BORN 1939) Stephen Cook was born in Buffalo where his father worked as an industrial chemist and taught university courses. His mother taught English courses in a community college. While in high school Cook developed an interest in electronics through his work with a famous local inventor noted for inventing the first implantable cardiac pacemaker.

Cook was a mathematics major at the University of Michigan, graduating in 1961. He did graduate work at Harvard, receiving a master's degree in 1962 and a Ph.D. in 1966. Cook was appointed an assistant professor in the Mathematics Department at the University of California, Berkeley in 1966. He was not granted tenure there, possibly because the members of the Mathematics Department did not find his work on what is now considered to be one of the most important areas of theoretical computer science of sufficient interest. In 1970, he joined the University of Toronto as an assistant professor, holding a joint appointment in the Computer Science Department and the Mathematics Department. He has remained at the University of Toronto, where he was appointed a University Professor in 1985.



Cook is considered to be one of the founders of computational complexity theory. His 1971 paper “The Complexity of Theorem Proving Procedures” formalized the notions of NP-completeness and polynomial-time reduction, showed that NP-complete problems exist by showing that the satisfiability problem is such a problem, and introduced the notorious P versus NP problem.

Cook has received many awards, including the 1982 Turing Award. He is married and has two sons. Among his interests are playing the violin and racing sailboats.

For more information about the complexity of algorithms, consult the references, including [CoLeRiSt09], for this section listed at the end of this book. (Also, for a more formal discussion of computational complexity in terms of Turing machines, see Section 13.5.)

PRACTICAL CONSIDERATIONS Note that a big- Θ estimate of the time complexity of an algorithm expresses how the time required to solve the problem increases as the input grows in size. In practice, the best estimate (that is, with the smallest reference function) that can be shown is used. However, big- Θ estimates of time complexity cannot be directly translated into the actual amount of computer time used. One reason is that a big- Θ estimate $f(n)$ is $\Theta(g(n))$, where $f(n)$ is the time complexity of an algorithm and $g(n)$ is a reference function, means that $C_1 g(n) \leq f(n) \leq C_2 g(n)$ when $n > k$, where C_1 , C_2 , and k are constants. So without knowing the constants C_1 , C_2 , and k in the inequality, this estimate cannot be used to determine a lower bound and an upper bound on the number of operations used in the worst case. As remarked before, the time required for an operation depends on the type of operation and the computer being used. Often, instead of a big- Θ estimate on the worst-case time complexity of an algorithm, we have only a big- O estimate. Note that a big- O estimate on the time complexity of an algorithm provides an upper, but not a lower, bound on the worst-case time required for the algorithm as a function of the input size. Nevertheless, for simplicity, we will often use big- O estimates when describing the time complexity of algorithms, with the understanding that big- Θ estimates would provide more information.

Table 2 displays the time needed to solve problems of various sizes with an algorithm using the indicated number n of bit operations, assuming that each bit operation takes 10^{-11} seconds, a reasonable estimate of the time required for a bit operation using the fastest computers available today. Times of more than 10^{100} years are indicated with an asterisk. In the future, these times will decrease as faster computers are developed. We can use the times shown in Table 2 to see whether it is reasonable to expect a solution to a problem of a specified size using an algorithm with known worst-case time complexity when we run this algorithm on a modern computer. Note that we cannot determine the exact time a computer uses to solve a problem with input of a particular size because of a myriad of issues involving computer hardware and the particular software implementation of the algorithm.

It is important to have a reasonable estimate for how long it will take a computer to solve a problem. For instance, if an algorithm requires approximately 10 hours, it may be worthwhile to spend the computer time (and money) required to solve this problem. But, if an algorithm requires approximately 10 billion years to solve a problem, it would be unreasonable to use resources to implement this algorithm. One of the most interesting phenomena of modern technology is the tremendous increase in the speed and memory space of computers. Another important factor that decreases the time needed to solve problems on computers is **parallel processing**, which is the technique of performing sequences of operations simultaneously.

Efficient algorithms, including most algorithms with polynomial time complexity, benefit most from significant technology improvements. However, these technology improvements

TABLE 2 The Computer Time Used by Algorithms.

<i>Problem Size</i>	<i>Bit Operations Used</i>					
	$\log n$	n	$n \log n$	n^2	2^n	$n!$
10	3×10^{-11} s	10^{-10} s	3×10^{-10} s	10^{-9} s	10^{-8} s	3×10^{-7} s
10^2	7×10^{-11} s	10^{-9} s	7×10^{-9} s	10^{-7} s	4×10^{11} yr	*
10^3	1.0×10^{-10} s	10^{-8} s	1×10^{-7} s	10^{-5} s	*	*
10^4	1.3×10^{-10} s	10^{-7} s	1×10^{-6} s	10^{-3} s	*	*
10^5	1.7×10^{-10} s	10^{-6} s	2×10^{-5} s	0.1 s	*	*
10^6	2×10^{-10} s	10^{-5} s	2×10^{-4} s	0.17 min	*	*

offer little help in overcoming the complexity of algorithms of exponential or factorial time complexity. Because of the increased speed of computation, increases in computer memory, and the use of algorithms that take advantage of parallel processing, many problems that were considered impossible to solve five years ago are now routinely solved, and certainly five years from now this statement will still be true. This is even true when the algorithms used are intractable.

Exercises

1. Give a big- O estimate for the number of operations (where an operation is an addition or a multiplication) used in this segment of an algorithm.

```
t := 0
for i := 1 to 3
    for j := 1 to 4
        t := t + ij
```

2. Give a big- O estimate for the number additions used in this segment of an algorithm.

```
t := 0
for i := 1 to n
    for j := 1 to n
        t := t + i + j
```

3. Give a big- O estimate for the number of operations, where an operation is a comparison or a multiplication, used in this segment of an algorithm (ignoring comparisons used to test the conditions in the **for** loops, where a_1, a_2, \dots, a_n are positive real numbers).

```
m := 0
for i := 1 to n
    for j := i + 1 to n
        m := max(ai, aj, m)
```

4. Give a big- O estimate for the number of operations, where an operation is an addition or a multiplication, used in this segment of an algorithm (ignoring comparisons used to test the conditions in the **while** loop).

```
i := 1
t := 0
while i ≤ n
    t := t + i
    i := 2i
```

5. How many comparisons are used by the algorithm given in Exercise 16 of Section 3.1 to find the smallest natural number in a sequence of n natural numbers?

6. a) Use pseudocode to describe the algorithm that puts the first four terms of a list of real numbers of arbitrary length in increasing order using the insertion sort.

- b) Show that this algorithm has time complexity $O(1)$ in terms of the number of comparisons used.

7. Suppose that an element is known to be among the first four elements in a list of 32 elements. Would a linear search or a binary search locate this element more rapidly?

8. Given a real number x and a positive integer k , determine the number of multiplications used to find x^{2^k} starting

with x and successively squaring (to find x^2, x^4 , and so on). Is this a more efficient way to find x^{2^k} than by multiplying x by itself the appropriate number of times?

9. Give a big- O estimate for the number of comparisons used by the algorithm that determines the number of 1s in a bit string by examining each bit of the string to determine whether it is a 1 bit (see Exercise 25 of Section 3.1).

- *10. a) Show that this algorithm determines the number of 1 bits in the bit string S :

```
procedure bit count(S: bit string)
    count := 0
    while S ≠ 0
        count := count + 1
        S := S ∧ (S - 1)
    return count {count is the number of 1s in S}
```

Here $S - 1$ is the bit string obtained by changing the rightmost 1 bit of S to a 0 and all the 0 bits to the right of this to 1s. [Recall that $S \wedge (S - 1)$ is the bitwise AND of S and $S - 1$.]

- b) How many bitwise AND operations are needed to find the number of 1 bits in a string S using the algorithm in part (a)?

11. a) Suppose we have n subsets S_1, S_2, \dots, S_n of the set $\{1, 2, \dots, n\}$. Express a brute-force algorithm that determines whether there is a disjoint pair of these subsets. [Hint: The algorithm should loop through the subsets; for each subset S_i , it should then loop through all other subsets; and for each of these other subsets S_j , it should loop through all elements k in S_i to determine whether k also belongs to S_j .]

- b) Give a big- O estimate for the number of times the algorithm needs to determine whether an integer is in one of the subsets.

12. Consider the following algorithm, which takes as input a sequence of n integers a_1, a_2, \dots, a_n and produces as output a matrix $\mathbf{M} = \{m_{ij}\}$ where m_{ij} is the minimum term in the sequence of integers a_i, a_{i+1}, \dots, a_j for $j \geq i$ and $m_{ij} = 0$ otherwise.

```
initialize M so that mij = ai if j ≥ i and mij = 0
    otherwise
for i := 1 to n
    for j := i + 1 to n
        for k := i + 1 to j
            mij := min(mij, ak)
return M = {mij} {mij is the minimum term of
    ai, ai+1, ..., aj}
```

- a) Show that this algorithm uses $O(n^3)$ comparisons to compute the matrix \mathbf{M} .
- b) Show that this algorithm uses $\Omega(n^3)$ comparisons to compute the matrix \mathbf{M} . Using this fact and part (a), conclude that the algorithm uses $\Theta(n^3)$ comparisons. [Hint: Only consider the cases where $i \leq n/4$ and $j \geq 3n/4$ in the two outer loops in the algorithm.]
13. The conventional algorithm for evaluating a polynomial $a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ at $x = c$ can be expressed in pseudocode by

```
procedure polynomial( $c, a_0, a_1, \dots, a_n$ : real numbers)
  power := 1
  y :=  $a_0$ 
  for  $i := 1$  to  $n$ 
    power := power *  $c$ 
    y :=  $y + a_i * power$ 
  return  $y$  { $y = a_n c^n + a_{n-1} c^{n-1} + \dots + a_1 c + a_0$ }
```

where the final value of y is the value of the polynomial at $x = c$.

- a) Evaluate $3x^2 + x + 1$ at $x = 2$ by working through each step of the algorithm showing the values assigned at each assignment step.
- b) Exactly how many multiplications and additions are used to evaluate a polynomial of degree n at $x = c$? (Do not count additions used to increment the loop variable.)
14. There is a more efficient algorithm (in terms of the number of multiplications and additions used) for evaluating polynomials than the conventional algorithm described in the previous exercise. It is called **Horner's method**. This pseudocode shows how to use this method to find the value of $a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ at $x = c$.

```
procedure Horner( $c, a_0, a_1, a_2, \dots, a_n$ : real numbers)
  y :=  $a_n$ 
  for  $i := 1$  to  $n$ 
    y :=  $y * c + a_{n-i}$ 
  return  $y$  { $y = a_n c^n + a_{n-1} c^{n-1} + \dots + a_1 c + a_0$ }
```

- a) Evaluate $3x^2 + x + 1$ at $x = 2$ by working through each step of the algorithm showing the values assigned at each assignment step.
- b) Exactly how many multiplications and additions are used by this algorithm to evaluate a polynomial of degree n at $x = c$? (Do not count additions used to increment the loop variable.)
15. What is the largest n for which one can solve within one second a problem using an algorithm that requires $f(n)$ bit operations, where each bit operation is carried out in 10^{-9} seconds, with these functions $f(n)$?
- a) $\log n$ b) n c) $n \log n$
 d) n^2 e) 2^n f) $n!$
16. What is the largest n for which one can solve within a day using an algorithm that requires $f(n)$ bit operations, where each bit operation is carried out in 10^{-11} seconds, with these functions $f(n)$?

- a) $\log n$ b) $1000n$ c) n^2
 d) $1000n^2$ e) n^3 f) 2^n
 g) 2^{2n} h) 2^{2^n}

17. What is the largest n for which one can solve within a minute using an algorithm that requires $f(n)$ bit operations, where each bit operation is carried out in 10^{-12} seconds, with these functions $f(n)$?
- a) $\log \log n$ b) $\log n$ c) $(\log n)^2$
 d) $1000000n$ e) n^2 f) 2^n
 g) 2^{n^2}
18. How much time does an algorithm take to solve a problem of size n if this algorithm uses $2n^2 + 2^n$ operations, each requiring 10^{-9} seconds, with these values of n ?
- a) 10 b) 20 c) 50 d) 100
19. How much time does an algorithm using 2^{50} operations need if each operation takes these amounts of time?
- a) 10^{-6} s b) 10^{-9} s c) 10^{-12} s
20. What is the effect in the time required to solve a problem when you double the size of the input from n to $2n$, assuming that the number of milliseconds the algorithm uses to solve the problem with input size n is each of these function? [Express your answer in the simplest form possible, either as a ratio or a difference. Your answer may be a function of n or a constant.]
- a) $\log \log n$ b) $\log n$ c) $100n$
 d) $n \log n$ e) n^2 f) n^3
 g) 2^n
21. What is the effect in the time required to solve a problem when you increase the size of the input from n to $n + 1$, assuming that the number of milliseconds the algorithm uses to solve the problem with input size n is each of these function? [Express your answer in the simplest form possible, either as a ratio or a difference. Your answer may be a function of n or a constant.]
- a) $\log n$ b) $100n$ c) n^2
 d) n^3 e) 2^n f) 2^{n^2}
 g) $n!$
22. Determine the least number of comparisons, or best-case performance,
- a) required to find the maximum of a sequence of n integers, using Algorithm 1 of Section 3.1.
- b) used to locate an element in a list of n terms with a linear search.
- c) used to locate an element in a list of n terms using a binary search.
23. Analyze the average-case performance of the linear search algorithm, if exactly half the time the element x is not in the list and if x is in the list it is equally likely to be in any position.
24. An algorithm is called **optimal** for the solution of a problem with respect to a specified operation if there is no algorithm for solving this problem using fewer operations.

- a) Show that Algorithm 1 in Section 3.1 is an optimal algorithm with respect to the number of comparisons of integers. [Note: Comparisons used for bookkeeping in the loop are not of concern here.]

b) Is the linear search algorithm optimal with respect to the number of comparisons of integers (not including comparisons used for bookkeeping in the loop)?

25. Describe the worst-case time complexity, measured in terms of comparisons, of the ternary search algorithm described in Exercise 27 of Section 3.1.

26. Describe the worst-case time complexity, measured in terms of comparisons, of the search algorithm described in Exercise 28 of Section 3.1.

27. Analyze the worst-case time complexity of the algorithm you devised in Exercise 29 of Section 3.1 for locating a mode in a list of nondecreasing integers.

28. Analyze the worst-case time complexity of the algorithm you devised in Exercise 30 of Section 3.1 for locating all modes in a list of nondecreasing integers.

29. Analyze the worst-case time complexity of the algorithm you devised in Exercise 31 of Section 3.1 for finding the first term of a sequence of integers equal to some previous term.

30. Analyze the worst-case time complexity of the algorithm you devised in Exercise 32 of Section 3.1 for finding all terms of a sequence that are greater than the sum of all previous terms.

31. Analyze the worst-case time complexity of the algorithm you devised in Exercise 33 of Section 3.1 for finding the first term of a sequence less than the immediately preceding term.

32. Determine the worst-case complexity in terms of comparisons of the algorithm from Exercise 5 in Section 3.1 for determining all values that occur more than once in a sorted list of integers.

33. Determine the worst-case complexity in terms of comparisons of the algorithm from Exercise 9 in Section 3.1 for determining whether a string of n characters is a palindrome.

34. How many comparisons does the selection sort (see preamble to Exercise 41 in Section 3.1) use to sort n items? Use your answer to give a big- O estimate of the complexity of the selection sort in terms of number of comparisons for the selection sort.

35. Find a big- O estimate for the worst-case complexity in terms of number of comparisons used and the number of terms swapped by the binary insertion sort described in the preamble to Exercise 47 in Section 3.1.

36. Show that the greedy algorithm for making change for n cents using quarters, dimes, nickels, and pennies has $O(n)$ complexity measured in terms of comparisons needed.

Exercises 37 and 38 deal with the problem of scheduling the most talks possible given the start and end times of n talks.

37. Find the complexity of a brute-force algorithm for scheduling the talks by examining all possible subsets of the talks. [Hint: Use the fact that a set with n elements has 2^n subsets.]

38. Find the complexity of the greedy algorithm for scheduling the most talks by adding at each step the talk with the earliest end time compatible with those already scheduled (Algorithm 7 in Section 3.1). Assume that the talks are not already sorted by earliest end time and assume that the worst-case time complexity of sorting is $O(n \log n)$.

39. Describe how the number of comparisons used in the worst case changes when these algorithms are used to search for an element of a list when the size of the list doubles from n to $2n$, where n is a positive integer.

a) linear search b) binary search

40. Describe how the number of comparisons used in the worst case changes when the size of the list to be sorted doubles from n to $2n$, where n is a positive integer when these sorting algorithms are used.

a) bubble sort b) insertion sort

c) selection sort (described in the preamble to Exercise 41 in Section 3.1)

d) binary insertion sort (described in the preamble to Exercise 47 in Section 3.1)

An $n \times n$ matrix is called **upper triangular** if $a_{ij} = 0$ whenever $i > j$.

41. From the definition of the matrix product, describe an algorithm in English for computing the product of two upper triangular matrices that ignores those products in the computation that are automatically equal to zero.

42. Give a pseudocode description of the algorithm in Exercise 41 for multiplying two upper triangular matrices.

43. How many multiplications of entries are used by the algorithm found in Exercise 41 for multiplying two $n \times n$ upper triangular matrices?

In Exercises 44–45 assume that the number of multiplications of entries used to multiply a $p \times q$ matrix and a $q \times r$ matrix is pqr .

44. What is the best order to form the product **ABC** if **A**, **B**, and **C** are matrices with dimensions 3×9 , 9×4 , and 4×2 , respectively?

45. What is the best order to form the product **ABCD** if **A**, **B**, **C**, and **D** are matrices with dimensions 30×10 , 10×40 , 40×50 , and 50×30 , respectively?

*46. In this exercise we deal with the problem of **string matching**.

a) Explain how to use a brute-force algorithm to find the first occurrence of a given string of m characters, called the **target**, in a string of n characters, where $m \leq n$, called the **text**. [Hint: Think in terms of finding a match for the first character of the target and checking successive characters for a match, and if they do not all match, moving the start location one character to the right.]

b) Express your algorithm in pseudocode.

c) Give a big- O estimate for the worst-case time complexity of the brute-force algorithm you described.

Key Terms and Results

TERMS

- algorithm:** a finite sequence of precise instructions for performing a computation or solving a problem
- searching algorithm:** the problem of locating an element in a list
- linear search algorithm:** a procedure for searching a list element by element
- binary search algorithm:** a procedure for searching an ordered list by successively splitting the list in half
- sorting:** the reordering of the elements of a list into prescribed order
- $f(x)$ is $O(g(x))$:** the fact that $|f(x)| \leq C|g(x)|$ for all $x > k$ for some constants C and k
- witness to the relationship $f(x)$ is $O(g(x))$:** a pair C and k such that $|f(x)| \leq C|g(x)|$ whenever $x > k$
- $f(x)$ is $\Omega(g(x))$:** the fact that $|f(x)| \geq C|g(x)|$ for all $x > k$ for some positive constants C and k
- $f(x)$ is $\Theta(g(x))$:** the fact that $f(x)$ is both $O(g(x))$ and $\Omega(g(x))$
- time complexity:** the amount of time required for an algorithm to solve a problem
- space complexity:** the amount of space in computer memory required for an algorithm to solve a problem
- worst-case time complexity:** the greatest amount of time required for an algorithm to solve a problem of a given size
- average-case time complexity:** the average amount of time required for an algorithm to solve a problem of a given size
- algorithmic paradigm:** a general approach for constructing algorithms based on a particular concept
- brute force:** the algorithmic paradigm based on constructing algorithms for solving problems in a naive manner from the statement of the problem and definitions

Review Questions

1. a) Define the term *algorithm*.
b) What are the different ways to describe algorithms?
c) What is the difference between an algorithm for solving a problem and a computer program that solves this problem?
2. a) Describe, using English, an algorithm for finding the largest integer in a list of n integers.
b) Express this algorithm in pseudocode.
c) How many comparisons does the algorithm use?
3. a) State the definition of the fact that $f(n)$ is $O(g(n))$, where $f(n)$ and $g(n)$ are functions from the set of positive integers to the set of real numbers.
b) Use the definition of the fact that $f(n)$ is $O(g(n))$ directly to prove or disprove that $n^2 + 18n + 107$ is $O(n^3)$.
c) Use the definition of the fact that $f(n)$ is $O(g(n))$ directly to prove or disprove that n^3 is $O(n^2 + 18n + 107)$.
4. List these functions so that each function is big- O of the next function in the list: $(\log n)^3$, $n^3/1000000$, \sqrt{n} , $100n + 101$, 3^n , $n!$, $2^n n^2$.
5. a) How can you produce a big- O estimate for a function that is the sum of different terms where each term is the product of several functions?
b) Give a big- O estimate for the function $f(n) = (n! + 1)(2^n + 1) + (n^{n-2} + 8n^{n-3})(n^3 + 2^n)$. For the function g in your estimate $f(x)$ is $O(g(x))$ use a simple function of smallest possible order.
6. a) Define what the worst-case time complexity, average-case time complexity, and best-case time complexity (in terms of comparisons) mean for an algorithm that finds the smallest integer in a list of n integers.
b) What are the worst-case, average-case, and best-case time complexities, in terms of comparisons, of the algorithm that finds the smallest integer in a list of n integers by comparing each of the integers with the smallest integer found so far?

greedy algorithm: an algorithm that makes the best choice at each step according to some specified condition

tractable problem: a problem for which there is a worst-case polynomial-time algorithm that solves it

intractable problem: a problem for which no worst-case polynomial-time algorithm exists for solving it

solvable problem: a problem that can be solved by an algorithm

unsolvable problem: a problem that cannot be solved by an algorithm

RESULTS

linear and binary search algorithms: (given in Section 3.1)

bubble sort: a sorting that uses passes where successive items are interchanged if they are in the wrong order

insertion sort: a sorting that at the j th step inserts the j th element into the correct position in the list, when the first $j - 1$ elements of the list are already sorted

The linear search has $O(n)$ worst case time complexity.

The binary search has $O(\log n)$ worst case time complexity.

The bubble and insertion sorts have $O(n^2)$ worst case time complexity.

$\log n!$ is $O(n \log n)$.

If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1 + f_2)(x)$ is $O(\max(g_1(x), g_2(x)))$ and $(f_1 f_2)(x)$ is $O((g_1 g_2)(x))$.

If a_0, a_1, \dots, a_n are real numbers with $a_n \neq 0$, then $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ is $\Theta(x^n)$, and hence $O(n)$ and $\Omega(n)$.

7. a) Describe the linear search and binary search algorithm for finding an integer in a list of integers in increasing order.
 - b) Compare the worst-case time complexities of these two algorithms.
 - c) Is one of these algorithms always faster than the other (measured in terms of comparisons)?
 8. a) Describe the bubble sort algorithm.
 - b) Use the bubble sort algorithm to sort the list 5, 2, 4, 1, 3.
 - c) Give a big- O estimate for the number of comparisons used by the bubble sort.
 9. a) Describe the insertion sort algorithm.
- b) Use the insertion sort algorithm to sort the list 2, 5, 1, 4, 3.
 - c) Give a big- O estimate for the number of comparisons used by the insertion sort.
10. a) Explain the concept of a greedy algorithm.
 - b) Provide an example of a greedy algorithm that produces an optimal solution and explain why it produces an optimal solution.
 - c) Provide an example of a greedy algorithm that does not always produce an optimal solution and explain why it fails to do so.
11. Define what it means for a problem to be tractable and what it means for a problem to be solvable.

Supplementary Exercises

1. a) Describe an algorithm for locating the last occurrence of the largest number in a list of integers.
 - b) Estimate the number of comparisons used.
 2. a) Describe an algorithm for finding the first and second largest elements in a list of integers.
 - b) Estimate the number of comparisons used.
 3. a) Give an algorithm to determine whether a bit string contains a pair of consecutive zeros.
 - b) How many comparisons does the algorithm use?
 4. a) Suppose that a list contains integers that are in order of largest to smallest and an integer can appear repeatedly in this list. Devise an algorithm that locates all occurrences of an integer x in the list.
 - b) Estimate the number of comparisons used.
 5. a) Adapt Algorithm 1 in Section 3.1 to find the maximum and the minimum of a sequence of n elements by employing a temporary maximum and a temporary minimum that is updated as each successive element is examined.
 - b) Describe the algorithm from part (a) in pseudocode.
 - c) How many comparisons of elements in the sequence are carried out by this algorithm? (Do not count comparisons used to determine whether the end of the sequence has been reached.)
 6. a) Describe in detail (and in English) the steps of an algorithm that finds the maximum and minimum of a sequence of n elements by examining pairs of successive elements, keeping track of a temporary maximum and a temporary minimum. If n is odd, both the temporary maximum and temporary minimum should initially equal the first term, and if n is even, the temporary minimum and temporary maximum should be found by comparing the initial two elements. The temporary maximum and temporary minimum should be updated by comparing them with the maximum and minimum of the pair of elements being examined.
 - b) Express the algorithm described in part (a) in pseudocode.
- c) How many comparisons of elements of the sequence are carried out by this algorithm? (Do not count comparisons used to determine whether the end of the sequence has been reached.) How does this compare to the number of comparisons used by the algorithm in Exercise 5?
- *7. Show that the worst-case complexity in terms of comparisons of an algorithm that finds the maximum and minimum of n elements is at least $\lceil 3n/2 \rceil - 2$.
8. Devise an efficient algorithm for finding the second largest element in a sequence of n elements and determine the worst-case complexity of your algorithm.
 9. Devise an algorithm that finds all equal pairs of sums of two terms of a sequence of n numbers, and determine the worst-case complexity of your algorithm.
 10. Devise an algorithm that finds the closest pair of integers in a sequence of n integers, and determine the worst-case complexity of your algorithm. [Hint: Sort the sequence. Use the fact that sorting can be done with worst-case time complexity $O(n \log n)$.]
- The **shaker sort** (or **bidirectional bubble sort**) successively compares pairs of adjacent elements, exchanging them if they are out of order, and alternately passing through the list from the beginning to the end and then from the end to the beginning until no exchanges are needed.
11. Show the steps used by the shaker sort to sort the list 3, 5, 1, 4, 6, 2.
 12. Express the shaker sort in pseudocode.
 13. Show that the shaker sort has $O(n^2)$ complexity measured in terms of the number of comparisons it uses.
 14. Explain why the shaker sort is efficient for sorting lists that are already in close to the correct order.
 15. Show that $(n \log n + n^2)^3$ is $O(n^6)$.
 16. Show that $8x^3 + 12x + 100 \log x$ is $O(x^3)$.
 17. Give a big- O estimate for $(x^2 + x(\log x)^3) \cdot (2^x + x^3)$.
 18. Find a big- O estimate for $\sum_{j=1}^n j(j+1)$.
- *19. Show that $n!$ is not $O(2^n)$.
- *20. Show that n^n is not $O(n!)$.

- 21.** Find all pairs of functions of the same order in this list of functions: $n^2 + (\log n)^2$, $n^2 + n$, $n^2 + \log 2^n + 1$, $(n+1)^3 - (n-1)^3$, and $(n + \log n)^2$.
- 22.** Find all pairs of functions of the same order in this list of functions $n^2 + 2^n$, $n^2 + 2^{100}$, $n^2 + 2^{2n}$, $n^2 + n!$, $n^2 + 3^n$, and $(n^2 + 1)^2$.
- 23.** Find an integer n with $n > 2$ for which $n^{2^{100}} < 2^n$.
- 24.** Find an integer n with $n > 2$ for which $(\log n)^{2^{100}} < \sqrt{n}$.
- *25.** Arrange the functions n^n , $(\log n)^2$, $n^{1.0001}$, $(1.0001)^n$, $2\sqrt{\log_2 n}$, and $n(\log n)^{1001}$ in a list so that each function is big- O of the next function. [Hint: To determine the relative size of some of these functions, take logarithms.]
- *26.** Arrange the function 2^{100n} , 2^{n^2} , $2^{n!}$, 2^{2^n} , $n^{\log n}$, $n \log n \log \log n$, $n^{3/2}$, $n(\log n)^{3/2}$, and $n^{4/3}(\log n)^2$ in a list so that each function is big- O of the next function. [Hint: To determine the relative size of some of these functions, take logarithms.]
- *27.** Give an example of two increasing functions $f(n)$ and $g(n)$ from the set of positive integers to the set of positive integers such that neither $f(n)$ is $O(g(n))$ nor $g(n)$ is $O(f(n))$.
- 28.** Show that if the denominations of coins are c^0, c^1, \dots, c^k , where k is a positive integer and c is a positive integer, $c > 1$, the greedy algorithm always produces change using the fewest coins possible.
- 29.** **a)** Use pseudocode to specify a brute-force algorithm that determines when given as input a sequence of n positive integers whether there are two distinct terms of the sequence that have as sum a third term. The algorithm should loop through all triples of terms of the sequence, checking whether the sum of the first two terms equals the third.
b) Give a big- O estimate for the complexity of the brute-force algorithm from part (a).
- 30.** **a)** Devise a more efficient algorithm for solving the problem described in Exercise 29 that first sorts the input sequence and then checks for each pair of terms whether their difference is in the sequence.
b) Give a big- O estimate for the complexity of this algorithm. Is it more efficient than the brute-force algorithm from Exercise 29?
- Suppose we have s men and s women each with their preference lists for the members of the opposite gender, as described in the preamble to Exercise 60 in Section 3.1. We say that a woman w is a **valid partner** for a man m if there is some stable matching in which they are paired. Similarly, a man m is a **valid partner** for a woman w if there is some stable matching in which they are paired. A matching in which each man is assigned his valid partner ranking highest on his preference list is called **male optimal**, and a matching in which each woman is assigned her valid partner ranking lowest on her preference list is called **female pessimal**.
- 31.** Find all valid partners for each man and each woman if there are three men m_1, m_2 , and m_3 and three women w_1, w_2, w_3 with these preference rankings of the men for the women, from highest to lowest: $m_1: w_3, w_1, w_2$; $m_2: w_3, w_2, w_1$; $m_3: w_2, w_3, w_1$; and with these preference rankings of the women for the men, from highest to lowest: $w_1: m_3, m_2, m_1$; $w_2: m_1, m_3, m_2$; $w_3: m_3, m_2, m_1$.
- *32.** Show that the deferred acceptance algorithm given in the preamble to Exercise 61 of Section 3.1, always produces a male optimal and female pessimal matching.
- 33.** Define what it means for a matching to be female optimal and for a matching to be male pessimal.
- *34.** Show that when women do the proposing in the deferred acceptance algorithm, the matching produced is female optimal and male pessimal.
- In Exercises 35 and 36 we consider variations on the problem of finding stable matchings of men and women described in the preamble to Exercise 61 in Section 3.1.
- *35.** In this exercise we consider matching problems where there may be different numbers of men and women, so that it is impossible to match everyone with a member of the opposite gender.
- a)** Extend the definition of a stable matching from that given in the preamble to Exercise 60 in Section 3.1 to cover the case where there are unequal numbers of men and women. Avoid all cases where a man and a woman would prefer each other to their current situation, including those involving unmatched people. (Assume that an unmatched person prefers a match with a member of the opposite gender to remaining unmatched.)
- b)** Adapt the deferred acceptance algorithm to find stable matchings, using the definition of stable matchings from part (a), when there are different numbers of men and women.
- c)** Prove that all matchings produced by the algorithm from part (b) are stable, according to the definition from part (a).
- *36.** In this exercise we consider matching problems where some man-woman pairs are not allowed.
- a)** Extend the definition of a stable matching to cover the situation where there are the same number of men and women, but certain pairs of men and women are forbidden. Avoid all cases where a man and a woman would prefer each other to their current situation, including those involving unmatched people.
- b)** Adapt the deferred acceptance algorithm to find stable matchings when there are the same number of men and women, but certain man-woman pairs are forbidden. Be sure to consider people who are unmatched at the end of the algorithm. (Assume that an unmatched person prefers a match with a member of the opposite gender who is not a forbidden partner to remaining unmatched.)
- c)** Prove that all matchings produced by the algorithm from (b) are stable, according to the definition in part (a).

Exercises 37–40 deal with the problem of scheduling n jobs on a single processor. To complete job j , the processor must run job j for time t_j without interruption. Each job has a deadline d_j . If we start job j at time s_j , it will be completed at time $e_j = s_j + t_j$. The **lateness** of the job measures how long it finishes after its deadline, that is, the lateness of job j is $\max(0, e_j - d_j)$. We wish to devise a greedy algorithm that minimizes the maximum lateness of a job among the n jobs.

37. Suppose we have five jobs with specified required times and deadlines: $t_1 = 25, d_1 = 50; t_2 = 15, d_2 = 60; t_3 = 20, d_3 = 60; t_4 = 5, d_4 = 55; t_5 = 10, d_5 = 75$. Find the maximum lateness of any job when the jobs are scheduled in this order (and they start at time 0): Job 3, Job 1, Job 4, Job 2, Job 5. Answer the same question for the schedule Job 5, Job 4, Job 3, Job 1, Job 2.
 38. The **slackness** of a job requiring time t and with deadline d is $d - t$, the difference between its deadline and the time it requires. Find an example that shows that scheduling jobs by increasing slackness does not always yield a schedule with the smallest possible maximum lateness.
 39. Find an example that shows that scheduling jobs in order of increasing time required does not always yield a schedule with the smallest possible maximum lateness.
 - *40. Prove that scheduling jobs in order of increasing deadlines always produces a schedule that minimizes the maximum lateness of a job. [Hint: First show that for a schedule to be optimal, jobs must be scheduled with no idle time between them and so that no job is scheduled before another with an earlier deadline.]
 41. Suppose that we have a knapsack with total capacity of W kg. We also have n items where item j has mass w_j . The **knapsack problem** asks for a subset of these n items with the largest possible total mass not exceeding W .
 - a) Devise a brute-force algorithm for solving the knapsack problem.
 - b) Solve the knapsack problem when the capacity of the knapsack is 18 kg and there are five items: a 5-kg
- sleeping bag, an 8-kg tent, a 7-kg food pack, a 4-kg container of water, and an 11-kg portable stove.
- In Exercises 42–46 we will study the problem of load balancing. The input to the problem is a collection of p processors and n jobs, t_j is the time required to run job j , jobs run without interruption on a single machine until finished, and a processor can run only one job at a time. The **load** L_k of processor k is the sum over all jobs assigned to processor k of the times required to run these jobs. The **makespan** is the maximum load over all the p processors. The load balancing problem asks for an assignment of jobs to processors to minimize the makespan.
42. Suppose we have three processors and five jobs requiring times $t_1 = 3, t_2 = 5, t_3 = 4, t_4 = 7$, and $t_5 = 8$. Solve the load balancing problem for this input by finding the assignment of the five jobs to the three processors that minimizes the makespan.
 43. Suppose that L^* is the minimum makespan when p processors are given n jobs, where t_j is the time required to run job j .
 - a) Show that $L^* \geq \max_{j=1,2,\dots,n} t_j$.
 - b) Show that $L^* \geq \frac{1}{p} \sum_{j=1}^n t_j$.
 44. Write out in pseudocode the greedy algorithm that goes through the jobs in order and assigns each job to the processor with the smallest load at that point in the algorithm.
 45. Run the algorithm from Exercise 44 on the input given in Exercise 42.
- An **approximation algorithm** for an optimization problem produces a solution guaranteed to be close to an optimal solution. More precisely, suppose that the optimization problem asks for an input S that minimizes $F(X)$ where F is some function of the input X . If an algorithm always finds an input T with $F(T) \leq cF(S)$ where c is a fixed positive real number, the algorithm is called a **c -approximation algorithm** for the problem.
- *46. Prove that the algorithm from Exercise 44 is a 2-approximation algorithm for the load balancing problem. [Hint: Use both parts of Exercise 43.]

Computer Projects

Write programs with these inputs and outputs.

1. Given a list of n integers, find the largest integer in the list.
2. Given a list of n integers, find the first and last occurrences of the largest integer in the list.
3. Given a list of n distinct integers, determine the position of an integer in the list using a linear search.
4. Given an ordered list of n distinct integers, determine the position of an integer in the list using a binary search.
5. Given a list of n integers, sort them using a bubble sort.
6. Given a list of n integers, sort them using an insertion sort.
7. Given an integer n , use the greedy algorithm to find the change for n cents using quarters, dimes, nickels, and pennies.
8. Given the starting and ending times of n talks, use the appropriate greedy algorithm to schedule the most talks possible in a single lecture hall.

- 9.** Given an ordered list of n integers and an integer x in the list, find the number of comparisons used to determine the position of x in the list using a linear search and using a binary search.
- 10.** Given a list of integers, determine the number of comparisons used by the bubble sort and by the insertion sort to sort this list.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

- 1.** We know that n^b is $O(d^n)$ when b and d are positive numbers with $d \geq 2$. Give values of the constants C and k such that $n^b \leq Cd^n$ whenever $n > k$ for each of these sets of values: $b = 10, d = 2$; $b = 20, d = 3$; $b = 1000, d = 7$.
- 2.** Compute the change for different values of n with coins of different denominations using the greedy algorithm

and determine whether the smallest number of coins was used. Can you find conditions so that the greedy algorithm is guaranteed to use the fewest coins possible?

- 3.** Using a generator of random orderings of the integers $1, 2, \dots, n$, find the number of comparisons used by the bubble sort, insertion sort, binary insertion sort, and selection sort to sort these integers.

Writing Projects

Respond to these with essays using outside sources.

- 1.** Examine the history of the word *algorithm* and describe the use of this word in early writings.
- 2.** Look up Bachmann's original introduction of big- O notation. Explain how he and others have used this notation.
- 3.** Explain how sorting algorithms can be classified into a taxonomy based on the underlying principle on which they are based.
- 4.** Describe the radix sort algorithm.
- 5.** Describe the historic trends in how quickly processors can perform operations and use these trends to estimate how quickly processors will be able to perform operations in the next twenty years.
- 6.** Develop a detailed list of algorithmic paradigms and provide examples using each of these paradigms.
- 7.** Explain what the Turing Award is and describe the criteria used to select winners. List six past winners of the award and why they received the award.
- 8.** Describe what is meant by a parallel algorithm. Explain how the pseudocode used in this book can be extended to handle parallel algorithms.
- 9.** Explain how the complexity of parallel algorithms can be measured. Give some examples to illustrate this concept, showing how a parallel algorithm can work more quickly than one that does not operate in parallel.
- 10.** Describe six different NP-complete problems.
- 11.** Demonstrate how one of the many different NP-complete problems can be reduced to the satisfiability problem.

4

Number Theory and Cryptography

- 4.1** Divisibility and Modular Arithmetic
- 4.2** Integer Representations and Algorithms
- 4.3** Primes and Greatest Common Divisors
- 4.4** Solving Congruences
- 4.5** Applications of Congruences
- 4.6** Cryptography

The part of mathematics devoted to the study of the set of integers and their properties is known as number theory. In this chapter we will develop some of the important concepts of number theory including many of those used in computer science. As we develop number theory, we will use the proof methods developed in Chapter 1 to prove many theorems.

We will first introduce the notion of divisibility of integers, which we use to introduce modular, or clock, arithmetic. Modular arithmetic operates with the remainders of integers when they are divided by a fixed positive integer, called the modulus. We will prove many important results about modular arithmetic which we will use extensively in this chapter.

Integers can be represented with any positive integer b greater than 1 as a base. In this chapter we discuss base b representations of integers and give an algorithm for finding them. In particular, we will discuss binary, octal, and hexadecimal (base 2, 8, and 16) representations. We will describe algorithms for carrying out arithmetic using these representations and study their complexity. These algorithms were the first procedures called algorithms.

We will discuss prime numbers, the positive integers that have only 1 and themselves as positive divisors. We will prove that there are infinitely many primes; the proof we give is considered to be one of the most beautiful proofs in mathematics. We will discuss the distribution of primes and many famous open questions concerning primes. We will introduce the concept of greatest common divisors and study the Euclidean algorithm for computing them. This algorithm was first described thousands of years ago. We will introduce the fundamental theorem of arithmetic, a key result which tells us that every positive integer has a unique factorization into primes.

We will explain how to solve linear congruences, as well as systems of linear congruences, which we solve using the famous Chinese remainder theorem. We will introduce the notion of pseudoprimes, which are composite integers masquerading as primes, and show how this notion can help us rapidly generate prime numbers.

This chapter introduces several important applications of number theory. In particular, we will use number theory to generate pseudorandom numbers, to assign memory locations to computer files, and to find check digits used to detect errors in various kinds of identification numbers. We also introduce the subject of cryptography. Number theory plays an essentially role both in classical cryptography, first used thousands of years ago, and modern cryptography, which plays an essential role in electronic communication. We will show how the ideas we develop can be used in cryptographical protocols, introducing protocols for sharing keys and for sending signed messages. Number theory, once considered the purest of subjects, has become an essential tool in providing computer and Internet security.

4.1 Divisibility and Modular Arithmetic

Introduction

The ideas that we will develop in this section are based on the notion of divisibility. Division of an integer by a positive integer produces a quotient and a remainder. Working with these remainders leads to modular arithmetic, which plays an important role in mathematics and which is used throughout computer science. We will discuss some important applications of modular arithmetic

later in this chapter, including generating pseudorandom numbers, assigning computer memory locations to files, constructing check digits, and encrypting messages.

Division

When one integer is divided by a second nonzero integer, the quotient may or may not be an integer. For example, $12/3 = 4$ is an integer, whereas $11/4 = 2.75$ is not. This leads to Definition 1.

DEFINITION 1

If a and b are integers with $a \neq 0$, we say that a divides b if there is an integer c such that $b = ac$, or equivalently, if $\frac{b}{a}$ is an integer. When a divides b we say that a is a *factor* or *divisor* of b , and that b is a *multiple* of a . The notation $a | b$ denotes that a divides b . We write $a \nmid b$ when a does not divide b .

Remark: We can express $a | b$ using quantifiers as $\exists c(ac = b)$, where the universe of discourse is the set of integers.

In Figure 1 a number line indicates which integers are divisible by the positive integer d .

EXAMPLE 1 Determine whether $3 | 7$ and whether $3 | 12$.

Solution: We see that $3 \nmid 7$, because $7/3$ is not an integer. On the other hand, $3 | 12$ because $12/3 = 4$. 

EXAMPLE 2 Let n and d be positive integers. How many positive integers not exceeding n are divisible by d ?



Solution: The positive integers divisible by d are all the integers of the form dk , where k is a positive integer. Hence, the number of positive integers divisible by d that do not exceed n equals the number of integers k with $0 < dk \leq n$, or with $0 < k \leq n/d$. Therefore, there are $\lfloor n/d \rfloor$ positive integers not exceeding n that are divisible by d . 

Some of the basic properties of divisibility of integers are given in Theorem 1.

THEOREM 1

Let a , b , and c be integers, where $a \neq 0$. Then

- (i) if $a | b$ and $a | c$, then $a | (b + c)$;
- (ii) if $a | b$, then $a | bc$ for all integers c ;
- (iii) if $a | b$ and $b | c$, then $a | c$.

Proof: We will give a direct proof of (i). Suppose that $a | b$ and $a | c$. Then, from the definition of divisibility, it follows that there are integers s and t with $b = as$ and $c = at$. Hence,

$$b + c = as + at = a(s + t).$$

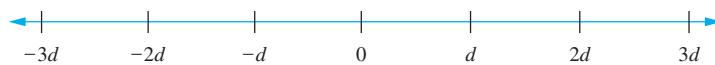


FIGURE 1 Integers Divisible by the Positive Integer d .

Therefore, a divides $b + c$. This establishes part (i) of the theorem. The proofs of parts (ii) and (iii) are left as Exercises 3 and 4. \triangleleft

Theorem 1 has this useful consequence.

COROLLARY 1

If a , b , and c are integers, where $a \neq 0$, such that $a | b$ and $a | c$, then $a | mb + nc$ whenever m and n are integers.

Proof: We will give a direct proof. By part (ii) of Theorem 1 we see that $a | mb$ and $a | nc$ whenever m and n are integers. By part (i) of Theorem 1 it follows that $a | mb + nc$. \triangleleft

The Division Algorithm

When an integer is divided by a positive integer, there is a quotient and a remainder, as the division algorithm shows.

THEOREM 2

THE DIVISION ALGORITHM Let a be an integer and d a positive integer. Then there are unique integers q and r , with $0 \leq r < d$, such that $a = dq + r$.

We defer the proof of the division algorithm to Section 5.2. (See Example 5 and Exercise 37.)

Remark: Theorem 2 is not really an algorithm. (Why not?) Nevertheless, we use its traditional name.

DEFINITION 2

In the equality given in the division algorithm, d is called the *divisor*, a is called the *dividend*, q is called the *quotient*, and r is called the *remainder*. This notation is used to express the quotient and remainder:

$$q = a \text{ div } d, \quad r = a \text{ mod } d.$$

Remark: Note that both $a \text{ div } d$ and $a \text{ mod } d$ for a fixed d are functions on the set of integers. Furthermore, when a is an integer and d is a positive integer, we have $a \text{ div } d = \lfloor a/d \rfloor$ and $a \text{ mod } d = a - d$. (See exercise 18.)

Examples 3 and 4 illustrate the division algorithm.

EXAMPLE 3 What are the quotient and remainder when 101 is divided by 11?

Solution: We have

$$101 = 11 \cdot 9 + 2.$$

Hence, the quotient when 101 is divided by 11 is $9 = 101 \text{ div } 11$, and the remainder is $2 = 101 \text{ mod } 11$. \triangleleft

EXAMPLE 4 What are the quotient and remainder when -11 is divided by 3 ?

Solution: We have

$$-11 = 3(-4) + 1.$$



Hence, the quotient when -11 is divided by 3 is $-4 = -11 \text{ div } 3$, and the remainder is $1 = -11 \text{ mod } 3$.

Note that the remainder cannot be negative. Consequently, the remainder is *not* -2 , even though

$$-11 = 3(-3) - 2,$$

because $r = -2$ does not satisfy $0 \leq r < 3$.

Note that the integer a is divisible by the integer d if and only if the remainder is zero when a is divided by d .

Remark: A programming language may have one, or possibly two, operators for modular arithmetic, denoted by `mod` (in BASIC, Maple, Mathematica, EXCEL, and SQL), `%` (in C, C++, Java, and Python), `rem` (in Ada and Lisp), or something else. Be careful when using them, because for $a < 0$, some of these operators return $a - m\lfloor a/m \rfloor$ instead of $a \text{ mod } m = a - m\lfloor a/m \rfloor$ (as shown in Exercise 18). Also, unlike $a \text{ mod } m$, some of these operators are defined when $m < 0$, and even when $m = 0$.

Modular Arithmetic

In some situations we care only about the remainder of an integer when it is divided by some specified positive integer. For instance, when we ask what time it will be (on a 24-hour clock) 50 hours from now, we care only about the remainder when 50 plus the current hour is divided by 24. Because we are often interested only in remainders, we have special notations for them. We have already introduced the notation $a \text{ mod } m$ to represent the remainder when an integer a is divided by the positive integer m . We now introduce a different, but related, notation that indicates that two integers have the same remainder when they are divided by the positive integer m .

DEFINITION 3

If a and b are integers and m is a positive integer, then a is *congruent to b modulo m* if m divides $a - b$. We use the notation $a \equiv b \pmod{m}$ to indicate that a is congruent to b modulo m . We say that $a \equiv b \pmod{m}$ is a **congruence** and that m is its **modulus** (plural **moduli**). If a and b are not congruent modulo m , we write $a \not\equiv b \pmod{m}$.

Although both notations $a \equiv b \pmod{m}$ and $a \text{ mod } m = b$ include “mod,” they represent fundamentally different concepts. The first represents a relation on the set of integers, whereas the second represents a function. However, the relation $a \equiv b \pmod{m}$ and the **mod** m function are closely related, as described in Theorem 3.

THEOREM 3

Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if $a \text{ mod } m = b \text{ mod } m$.

The proof of Theorem 3 is left as Exercises 15 and 16. Recall that $a \text{ mod } m$ and $b \text{ mod } m$ are the remainders when a and b are divided by m , respectively. Consequently, Theorem 3 also says that $a \equiv b \pmod{m}$ if and only if a and b have the same remainder when divided by m .

EXAMPLE 5 Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

Solution: Because 6 divides $17 - 5 = 12$, we see that $17 \equiv 5 \pmod{6}$. However, because $24 - 14 = 10$ is not divisible by 6, we see that $24 \not\equiv 14 \pmod{6}$. \blacktriangleleft

The great German mathematician Karl Friedrich Gauss developed the concept of congruences at the end of the eighteenth century. The notion of congruences has played an important role in the development of number theory.

Theorem 4 provides a useful way to work with congruences.

THEOREM 4

Let m be a positive integer. The integers a and b are congruent modulo m if and only if there is an integer k such that $a = b + km$.

Proof: If $a \equiv b \pmod{m}$, by the definition of congruence (Definition 3), we know that $m \mid (a - b)$. This means that there is an integer k such that $a - b = km$, so that $a = b + km$. Conversely, if there is an integer k such that $a = b + km$, then $km = a - b$. Hence, m divides $a - b$, so that $a \equiv b \pmod{m}$. \blacktriangleleft

The set of all integers congruent to an integer a modulo m is called the **congruence class** of a modulo m . In Chapter 9 we will show that there are m pairwise disjoint equivalence classes modulo m and that the union of these equivalence classes is the set of integers.

Theorem 5 shows that additions and multiplications preserve congruences.



KARL FRIEDRICH GAUSS (1777–1855) Karl Friedrich Gauss, the son of a bricklayer, was a child prodigy. He demonstrated his potential at the age of 10, when he quickly solved a problem assigned by a teacher to keep the class busy. The teacher asked the students to find the sum of the first 100 positive integers. Gauss realized that this sum could be found by forming 50 pairs, each with the sum 101: $1 + 100, 2 + 99, \dots, 50 + 51$. This brilliance attracted the sponsorship of patrons, including Duke Ferdinand of Brunswick, who made it possible for Gauss to attend Caroline College and the University of Göttingen. While a student, he invented the method of least squares, which is used to estimate the most likely value of a variable from experimental results. In 1796 Gauss made a fundamental discovery in geometry, advancing a subject that had not advanced since ancient times. He showed that a 17-sided regular polygon could be drawn using just a ruler and compass.

In 1799 Gauss presented the first rigorous proof of the fundamental theorem of algebra, which states that a polynomial of degree n has exactly n roots (counting multiplicities). Gauss achieved worldwide fame when he successfully calculated the orbit of the first asteroid discovered, Ceres, using scanty data.

Gauss was called the Prince of Mathematics by his contemporary mathematicians. Although Gauss is noted for his many discoveries in geometry, algebra, analysis, astronomy, and physics, he had a special interest in number theory, which can be seen from his statement “Mathematics is the queen of the sciences, and the theory of numbers is the queen of mathematics.” Gauss laid the foundations for modern number theory with the publication of his book *Disquisitiones Arithmeticae* in 1801.

THEOREM 5

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then

$$a + c \equiv b + d \pmod{m} \quad \text{and} \quad ac \equiv bd \pmod{m}.$$

Proof: We use a direct proof. Because $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, by Theorem 4 there are integers s and t with $b = a + sm$ and $d = c + tm$. Hence,

$$b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$$

and

$$bd = (a + sm)(c + tm) = ac + m(at + cs + stm).$$

Hence,

$$a + c \equiv b + d \pmod{m} \quad \text{and} \quad ac \equiv bd \pmod{m}. \quad \triangleleft$$

EXAMPLE 6 Because $7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$, it follows from Theorem 5 that

$$18 = 7 + 11 \equiv 2 + 1 = 3 \pmod{5}$$

and that

$$77 = 7 \cdot 11 \equiv 2 \cdot 1 = 2 \pmod{5}. \quad \triangleleft$$

You cannot always divide both sides of a congruence by the same number!



We must be careful working with congruences. Some properties we may expect to be true are not valid. For example, if $ac \equiv bc \pmod{m}$, the congruence $a \equiv b \pmod{m}$ may be false. Similarly, if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, the congruence $a^c \equiv b^d \pmod{m}$ may be false. (See Exercise 37.)

Corollary 2 shows how to find the values of the **mod** m function at the sum and product of two integers using the values of this function at each of these integers. We will use this result in Section 5.4.

COROLLARY 2

Let m be a positive integer and let a and b be integers. Then

$$(a + b) \pmod{m} = ((a \pmod{m}) + (b \pmod{m})) \pmod{m}$$

and

$$ab \pmod{m} = ((a \pmod{m})(b \pmod{m})) \pmod{m}.$$



Proof: By the definitions of **mod** m and of congruence modulo m , we know that $a \equiv (a \pmod{m}) \pmod{m}$ and $b \equiv (b \pmod{m}) \pmod{m}$. Hence, Theorem 5 tells us that

$$a + b \equiv (a \pmod{m}) + (b \pmod{m}) \pmod{m}$$

and

$$ab \equiv (a \pmod{m})(b \pmod{m}) \pmod{m}.$$

The equalities in this corollary follow from these last two congruences by Theorem 3. \triangleleft

Arithmetic Modulo m

We can define arithmetic operations on \mathbf{Z}_m , the set of nonnegative integers less than m , that is, the set $\{0, 1, \dots, m - 1\}$. In particular, we define addition of these integers, denoted by $+_m$ by

$$a +_m b = (a + b) \text{ mod } m,$$

where the addition on the right-hand side of this equation is the ordinary addition of integers, and we define multiplication of these integers, denoted by \cdot_m by

$$a \cdot_m b = (a \cdot b) \text{ mod } m,$$

where the multiplication on the right-hand side of this equation is the ordinary multiplication of integers. The operations $+_m$ and \cdot_m are called addition and multiplication modulo m and when we use these operations, we are said to be doing **arithmetic modulo m** .

EXAMPLE 7 Use the definition of addition and multiplication in \mathbf{Z}_m to find $7 +_{11} 9$ and $7 \cdot_{11} 9$.

Solution: Using the definition of addition modulo 11, we find that

$$7 +_{11} 9 = (7 + 9) \text{ mod } 11 = 16 \text{ mod } 11 = 5,$$

and

$$7 \cdot_{11} 9 = (7 \cdot 9) \text{ mod } 11 = 63 \text{ mod } 11 = 8.$$

Hence $7 +_{11} 9 = 5$ and $7 \cdot_{11} 9 = 8$. 

The operations $+_m$ and \cdot_m satisfy many of the same properties of ordinary addition and multiplication of integers. In particular, they satisfy these properties:

Closure If a and b belong to \mathbf{Z}_m , then $a +_m b$ and $a \cdot_m b$ belong to \mathbf{Z}_m .

Associativity If a , b , and c belong to \mathbf{Z}_m , then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \cdot_m b) \cdot_m c = a \cdot_m (b \cdot_m c)$.

Commutativity If a and b belong to \mathbf{Z}_m , then $a +_m b = b +_m a$ and $a \cdot_m b = b \cdot_m a$.

Identity elements The elements 0 and 1 are identity elements for addition and multiplication modulo m , respectively. That is, if a belongs to \mathbf{Z}_m , then $a +_m 0 = 0 +_m a = a$ and $a \cdot_m 1 = 1 \cdot_m a = a$.

Additive inverses If $a \neq 0$ belongs to \mathbf{Z}_m , then $m - a$ is an additive inverse of a modulo m and 0 is its own additive inverse. That is $a +_m (m - a) = 0$ and $0 +_m 0 = 0$.

Distributivity If a , b , and c belong to \mathbf{Z}_m , then $a \cdot_m (b +_m c) = (a \cdot_m b) +_m (a \cdot_m c)$ and $(a +_m b) \cdot_m c = (a \cdot_m c) +_m (b \cdot_m c)$.

These properties follow from the properties we have developed for congruences and remainders modulo m , together with the properties of integers; we leave their proofs as Exercises 42–44. Note that we have listed the property that every element of \mathbf{Z}_m has an additive inverse, but no analogous property for multiplicative inverses has been included. This is because multiplicative inverses do not always exist modulo m . For instance, there is no multiplicative inverse of 2 modulo 6, as the reader can verify. We will return to the question of when an integer has a multiplicative inverse modulo m later in this chapter.

Remark: Because \mathbf{Z}_m with the operations of addition and multiplication modulo m satisfies the properties listed, \mathbf{Z}_m with modular addition is said to be a **commutative group** and \mathbf{Z}_m with both of these operations is said to be a **commutative ring**. Note that the set of integers with ordinary addition and multiplication also forms a commutative ring. Groups and rings are studied in courses that cover abstract algebra.

Remark: In Exercise 30, and in later sections, we will use the notations $+$ and \cdot for $+_m$ and \cdot_m without the subscript m on the symbol for the operator whenever we work with \mathbf{Z}_m .

Exercises

1. Does 17 divide each of these numbers?
 a) 68 b) 84 c) 357 d) 1001
2. Prove that if a is an integer other than 0, then
 a) 1 divides a . b) a divides 0.
3. Prove that part (ii) of Theorem 1 is true.
4. Prove that part (iii) of Theorem 1 is true.
5. Show that if $a \mid b$ and $b \mid a$, where a and b are integers, then $a = b$ or $a = -b$.
6. Show that if a, b, c , and d are integers, where $a \neq 0$, such that $a \mid c$ and $b \mid d$, then $ab \mid cd$.
7. Show that if a, b , and c are integers, where $a \neq 0$ and $c \neq 0$, such that $ac \mid bc$, then $a \mid b$.
8. Prove or disprove that if $a \mid bc$, where a, b , and c are positive integers and $a \neq 0$, then $a \mid b$ or $a \mid c$.
9. What are the quotient and remainder when
 - a) 19 is divided by 7?
 - b) -111 is divided by 11?
 - c) 789 is divided by 23?
 - d) 1001 is divided by 13?
 - e) 0 is divided by 19?
 - f) 3 is divided by 5?
 - g) -1 is divided by 3?
 - h) 4 is divided by 1?
10. What are the quotient and remainder when
 - a) 44 is divided by 8?
 - b) 777 is divided by 21?
 - c) -123 is divided by 19?
 - d) -1 is divided by 23?
 - e) -2002 is divided by 87?
 - f) 0 is divided by 17?
 - g) 1,234,567 is divided by 1001?
 - h) -100 is divided by 101?
11. What time does a 12-hour clock read
 - a) 80 hours after it reads 11:00?
 - b) 40 hours before it reads 12:00?
 - c) 100 hours after it reads 6:00?
12. What time does a 24-hour clock read
 - a) 100 hours after it reads 2:00?
 - b) 45 hours before it reads 12:00?
 - c) 168 hours after it reads 19:00?
13. Suppose that a and b are integers, $a \equiv 4 \pmod{13}$, and $b \equiv 9 \pmod{13}$. Find the integer c with $0 \leq c \leq 12$ such that
 - a) $c \equiv 9a \pmod{13}$.
 - b) $c \equiv 11b \pmod{13}$.
 - c) $c \equiv a + b \pmod{13}$.
 - d) $c \equiv 2a + 3b \pmod{13}$.
 - e) $c \equiv a^2 + b^2 \pmod{13}$.
 - f) $c \equiv a^3 - b^3 \pmod{13}$.
14. Suppose that a and b are integers, $a \equiv 11 \pmod{19}$, and $b \equiv 3 \pmod{19}$. Find the integer c with $0 \leq c \leq 18$ such that
 - a) $c \equiv 13a \pmod{19}$.
 - b) $c \equiv 8b \pmod{19}$.
 - c) $c \equiv a - b \pmod{19}$.
 - d) $c \equiv 7a + 3b \pmod{19}$.
 - e) $c \equiv 2a^2 + 3b^2 \pmod{19}$.
 - f) $c \equiv a^3 + 4b^3 \pmod{19}$.
15. Let m be a positive integer. Show that $a \equiv b \pmod{m}$ if $a \text{ mod } m = b \text{ mod } m$.
16. Let m be a positive integer. Show that $a \text{ mod } m = b \text{ mod } m$ if $a \equiv b \pmod{m}$.
17. Show that if n and k are positive integers, then $\lceil n/k \rceil = \lfloor (n-1)/k \rfloor + 1$.
18. Show that if a is an integer and d is an integer greater than 1, then the quotient and remainder obtained when a is divided by d are $\lfloor a/d \rfloor$ and $a - d\lfloor a/d \rfloor$, respectively.
19. Find a formula for the integer with smallest absolute value that is congruent to an integer a modulo m , where m is a positive integer.
20. Evaluate these quantities.

a) $-17 \text{ mod } 2$	b) $144 \text{ mod } 7$
c) $-101 \text{ mod } 13$	d) $199 \text{ mod } 19$
21. Evaluate these quantities.

a) $13 \text{ mod } 3$	b) $-97 \text{ mod } 11$
c) $155 \text{ mod } 19$	d) $-221 \text{ mod } 23$
22. Find $a \text{ div } m$ and $a \text{ mod } m$ when
 - a) $a = -111, m = 99$.
 - b) $a = -9999, m = 101$.
 - c) $a = 10299, m = 999$.
 - d) $a = 123456, m = 1001$.

- 23.** Find $a \text{ div } m$ and $a \text{ mod } m$ when
- $a = 228, m = 119$.
 - $a = 9009, m = 223$.
 - $a = -10101, m = 333$.
 - $a = -765432, m = 38271$.
- 24.** Find the integer a such that
- $a \equiv 43 \pmod{23}$ and $-22 \leq a \leq 0$.
 - $a \equiv 17 \pmod{29}$ and $-14 \leq a \leq 14$.
 - $a \equiv -11 \pmod{21}$ and $90 \leq a \leq 110$.
- 25.** Find the integer a such that
- $a \equiv -15 \pmod{27}$ and $-26 \leq a \leq 0$.
 - $a \equiv 24 \pmod{31}$ and $-15 \leq a \leq 15$.
 - $a \equiv 99 \pmod{41}$ and $100 \leq a \leq 140$.
- 26.** List five integers that are congruent to 4 modulo 12.
- 27.** List all integers between -100 and 100 that are congruent to -1 modulo 25.
- 28.** Decide whether each of these integers is congruent to 3 modulo 7.
- 37
 - 66
 - 17
 - 67
- 29.** Decide whether each of these integers is congruent to 5 modulo 17.
- 80
 - 103
 - 29
 - 122
- 30.** Find each of these values.
- $(177 \text{ mod } 31 + 270 \text{ mod } 31) \text{ mod } 31$
 - $(177 \text{ mod } 31 \cdot 270 \text{ mod } 31) \text{ mod } 31$
- 31.** Find each of these values.
- $(-133 \text{ mod } 23 + 261 \text{ mod } 23) \text{ mod } 23$
 - $(457 \text{ mod } 23 \cdot 182 \text{ mod } 23) \text{ mod } 23$
- 32.** Find each of these values.
- $(19^2 \text{ mod } 41) \text{ mod } 9$
 - $(32^3 \text{ mod } 13)^2 \text{ mod } 11$
 - $(7^3 \text{ mod } 23)^2 \text{ mod } 31$
 - $(21^2 \text{ mod } 15)^3 \text{ mod } 22$
- 33.** Find each of these values.
- $(99^2 \text{ mod } 32)^3 \text{ mod } 15$
 - $(3^4 \text{ mod } 17)^2 \text{ mod } 11$
 - $(19^3 \text{ mod } 23)^2 \text{ mod } 31$
 - $(89^3 \text{ mod } 79)^4 \text{ mod } 26$
- 34.** Show that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, where a, b, c, d , and m are integers with $m \geq 2$, then $a - c \equiv b - d \pmod{m}$.
- 35.** Show that if $n \mid m$, where n and m are integers greater than 1, and if $a \equiv b \pmod{m}$, where a and b are integers, then $a \equiv b \pmod{n}$.
- 36.** Show that if a, b, c , and m are integers such that $m \geq 2$, $c > 0$, and $a \equiv b \pmod{m}$, then $ac \equiv bc \pmod{mc}$.
- 37.** Find counterexamples to each of these statements about congruences.
- If $ac \equiv bc \pmod{m}$, where a, b, c , and m are integers with $m \geq 2$, then $a \equiv b \pmod{m}$.
 - If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, where a, b, c, d , and m are integers with c and d positive and $m \geq 2$, then $a^c \equiv b^d \pmod{m}$.
- 38.** Show that if n is an integer then $n^2 \equiv 0$ or $1 \pmod{4}$.
- 39.** Use Exercise 38 to show that if m is a positive integer of the form $4k + 3$ for some nonnegative integer k , then m is not the sum of the squares of two integers.
- 40.** Prove that if n is an odd positive integer, then $n^2 \equiv 1 \pmod{8}$.
- 41.** Show that if a, b, k , and m are integers such that $k \geq 1$, $m \geq 2$, and $a \equiv b \pmod{m}$, then $a^k \equiv b^k \pmod{m}$.
- 42.** Show that \mathbf{Z}_m with addition modulo m , where $m \geq 2$ is an integer, satisfies the closure, associative, and commutative properties, 0 is an additive identity, and for every nonzero $a \in \mathbf{Z}_m$, $m - a$ is an inverse of a modulo m .
- 43.** Show that \mathbf{Z}_m with multiplication modulo m , where $m \geq 2$ is an integer, satisfies the closure, associative, and commutativity properties, and 1 is a multiplicative identity.
- 44.** Show that the distributive property of multiplication over addition holds for \mathbf{Z}_m , where $m \geq 2$ is an integer.
- 45.** Write out the addition and multiplication tables for \mathbf{Z}_5 (where by addition and multiplication we mean $+_5$ and \cdot_5).
- 46.** Write out the addition and multiplication tables for \mathbf{Z}_6 (where by addition and multiplication we mean $+_6$ and \cdot_6).
- 47.** Determine whether each of the functions $f(a) = a \text{ div } d$ and $g(a) = a \text{ mod } d$, where d is a fixed positive integer, from the set of integers to the set of integers, is one-to-one, and determine whether each of these functions is onto.

4.2 Integer Representations and Algorithms

Introduction

Integers can be expressed using any integer greater than one as a base, as we will show in this section. Although we commonly use decimal (base 10), representations, binary (base 2), octal (base 8), and hexadecimal (base 16) representations are often used, especially in computer science. Given a base b and an integer n , we will show how to construct the base b representation of this integer. We will also explain how to quickly convert between binary and octal and between binary and hexadecimal notations.

As mentioned in Section 3.1, the term *algorithm* originally referred to procedures for performing arithmetic operations using the decimal representations of integers. These algorithms, adapted for use with binary representations, are the basis for computer arithmetic. They provide good illustrations of the concept of an algorithm and the complexity of algorithms. For these reasons, they will be discussed in this section.

We will also introduce an algorithm for finding $a \text{ div } d$ and $a \text{ mod } d$ where a and d are integers with $d > 1$. Finally, we will describe an efficient algorithm for modular exponentiation, which is a particularly important algorithm for cryptography, as we will see in Section 4.6.

Representations of Integers

In everyday life we use decimal notation to express integers. For example, 965 is used to denote $9 \cdot 10^2 + 6 \cdot 10 + 5$. However, it is often convenient to use bases other than 10. In particular, computers usually use binary notation (with 2 as the base) when carrying out arithmetic, and octal (base 8) or hexadecimal (base 16) notation when expressing characters, such as letters or digits. In fact, we can use any integer greater than 1 as the base when expressing integers. This is stated in Theorem 1.

THEOREM 1

Let b be an integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0,$$

where k is a nonnegative integer, a_0, a_1, \dots, a_k are nonnegative integers less than b , and $a_k \neq 0$.

A proof of this theorem can be constructed using mathematical induction, a proof method that is discussed in Section 5.1. It can also be found in [Ro10]. The representation of n given in Theorem 1 is called the **base b expansion of n** . The base b expansion of n is denoted by $(a_k a_{k-1} \dots a_1 a_0)_b$. For instance, $(245)_8$ represents $2 \cdot 8^2 + 4 \cdot 8 + 5 = 165$. Typically, the subscript 10 is omitted for base 10 expansions of integers because base 10, or **decimal expansions**, are commonly used to represent integers.

BINARY EXPANSIONS Choosing 2 as the base gives **binary expansions** of integers. In binary notation each digit is either a 0 or a 1. In other words, the binary expansion of an integer is just a bit string. Binary expansions (and related expansions that are variants of binary expansions) are used by computers to represent and do arithmetic with integers.

EXAMPLE 1 What is the decimal expansion of the integer that has $(10101111)_2$ as its binary expansion?

Solution: We have

$$\begin{aligned}(10101111)_2 &= 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 \\ &\quad + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351.\end{aligned}$$

OCTAL AND HEXADECIMAL EXPANSIONS Among the most important bases in computer science are base 2, base 8, and base 16. Base 8 expansions are called **octal** expansions and base 16 expansions are **hexadecimal** expansions.

EXAMPLE 2 What is the decimal expansion of the number with octal expansion $(7016)_8$?

Solution: Using the definition of a base b expansion with $b = 8$ tells us that

$$(7016)_8 = 7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8 + 6 = 3598.$$

Sixteen different digits are required for hexadecimal expansions. Usually, the hexadecimal digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F, where the letters A through F represent the digits corresponding to the numbers 10 through 15 (in decimal notation).

EXAMPLE 3 What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

Solution: Using the definition of a base b expansion with $b = 16$ tells us that

$$(2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16 + 11 = 175627.$$

Each hexadecimal digit can be represented using four bits. For instance, we see that $(1110\ 0101)_2 = (E5)_{16}$ because $(1110)_2 = (E)_{16}$ and $(0101)_2 = (5)_{16}$. **Bytes**, which are bit strings of length eight, can be represented by two hexadecimal digits.

BASE CONVERSION We will now describe an algorithm for constructing the base b expansion of an integer n . First, divide n by b to obtain a quotient and remainder, that is,

$$n = bq_0 + a_0, \quad 0 \leq a_0 < b.$$

The remainder, a_0 , is the rightmost digit in the base b expansion of n . Next, divide q_0 by b to obtain

$$q_0 = bq_1 + a_1, \quad 0 \leq a_1 < b.$$

We see that a_1 is the second digit from the right in the base b expansion of n . Continue this process, successively dividing the quotients by b , obtaining additional base b digits as the remainders. This process terminates when we obtain a quotient equal to zero. It produces the base b digits of n from the right to the left.

EXAMPLE 4 Find the octal expansion of $(12345)_{10}$.



Solution: First, divide 12345 by 8 to obtain

$$12345 = 8 \cdot 1543 + 1.$$

Successively dividing quotients by 8 gives

$$\begin{aligned} 1543 &= 8 \cdot 192 + 7, \\ 192 &= 8 \cdot 24 + 0, \\ 24 &= 8 \cdot 3 + 0, \\ 3 &= 8 \cdot 0 + 3. \end{aligned}$$

The successive remainders that we have found, 1, 7, 0, 0, and 3, are the digits from the right to the left of 12345 in base 8. Hence,

$$(12345)_{10} = (30071)_8.$$

EXAMPLE 5 Find the hexadecimal expansion of $(177130)_{10}$.

Solution: First divide 177130 by 16 to obtain

$$177130 = 16 \cdot 11070 + 10.$$

Successively dividing quotients by 16 gives

$$\begin{aligned} 11070 &= 16 \cdot 691 + 14, \\ 691 &= 16 \cdot 43 + 3, \\ 43 &= 16 \cdot 2 + 11, \\ 2 &= 16 \cdot 0 + 2. \end{aligned}$$

The successive remainders that we have found, 10, 14, 3, 11, 2, give us the digits from the right to the left of 177130 in the hexadecimal (base 16) expansion of $(177130)_{10}$. It follows that

$$(177130)_{10} = (2B3EA)_{16}.$$

(Recall that the integers 10, 11, and 14 correspond to the hexadecimal digits A, B, and E, respectively.) 

EXAMPLE 6 Find the binary expansion of $(241)_{10}$.

Solution: First divide 241 by 2 to obtain

$$241 = 2 \cdot 120 + 1.$$

Successively dividing quotients by 2 gives

$$\begin{aligned} 120 &= 2 \cdot 60 + 0, \\ 60 &= 2 \cdot 30 + 0, \\ 30 &= 2 \cdot 15 + 0, \\ 15 &= 2 \cdot 7 + 1, \\ 7 &= 2 \cdot 3 + 1, \\ 3 &= 2 \cdot 1 + 1, \\ 1 &= 2 \cdot 0 + 1. \end{aligned}$$

The successive remainders that we have found, 1, 0, 0, 0, 1, 1, 1, 1, are the digits from the right to the left in the binary (base 2) expansion of $(241)_{10}$. Hence,

$$(241)_{10} = (1111\ 0001)_2.$$

The pseudocode given in Algorithm 1 finds the base b expansion $(a_{k-1} \dots a_1 a_0)_b$ of the integer n .

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

ALGORITHM 1 Constructing Base b Expansions.

```

procedure base  $b$  expansion( $n, b$ : positive integers with  $b > 1$ )
   $q := n$ 
   $k := 0$ 
  while  $q \neq 0$ 
     $a_k := q \bmod b$ 
     $q := q \text{ div } b$ 
     $k := k + 1$ 
  return  $(a_{k-1}, \dots, a_1, a_0)$   $\{(a_{k-1} \dots a_1 a_0)_b$  is the base  $b$  expansion of  $n\}$ 

```

In Algorithm 1, q represents the quotient obtained by successive divisions by b , starting with $q = n$. The digits in the base b expansion are the remainders of these divisions and are given by $q \bmod b$. The algorithm terminates when a quotient $q = 0$ is reached.

Remark: Note that Algorithm 1 can be thought of as a greedy algorithm, as the base b digits are taken as large as possible in each step.

CONVERSION BETWEEN BINARY, OCTAL, AND HEXADECIMAL EXPANSIONS

Conversion between binary and octal and between binary and hexadecimal expansions is extremely easy because each octal digit corresponds to a block of three binary digits and each hexadecimal digit corresponds to a block of four binary digits, with these correspondences shown in Table 1 without initial 0s shown. (We leave it as Exercises 13–16 to show that this is the case.) This conversion is illustrated in Example 7.

EXAMPLE 7 Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$ and the binary expansions of $(765)_8$ and $(A8D)_{16}$.

Solution: To convert $(11\ 1110\ 1011\ 1100)_2$ into octal notation we group the binary digits into blocks of three, adding initial zeros at the start of the leftmost block if necessary. These blocks, from left to right, are 011, 111, 010, 111, and 100, corresponding to 3, 7, 2, 7, and 4, respectively. Consequently, $(11\ 1110\ 1011\ 1100)_2 = (37274)_8$. To convert $(11\ 1110\ 1011\ 1100)_2$ into hexadecimal notation we group the binary digits into blocks of four, adding initial zeros at the start of the leftmost block if necessary. These blocks, from left to right, are 0011, 1110, 1011, and 1100, corresponding to the hexadecimal digits 3, E, B, and C, respectively. Consequently, $(11\ 1110\ 1011\ 1100)_2 = (3EBC)_{16}$.

To convert $(765)_8$ into binary notation, we replace each octal digit by a block of three binary digits. These blocks are 111, 110, and 101. Hence, $(765)_8 = (1\ 1111\ 0101)_2$. To convert $(A8D)_{16}$ into binary notation, we replace each hexadecimal digit by a block of four binary digits. These blocks are 1010, 1000, and 1101. Hence, $(A8D)_{16} = (1010\ 1000\ 1101)_2$. 

Algorithms for Integer Operations

The algorithms for performing operations with integers using their binary expansions are extremely important in computer arithmetic. We will describe algorithms for the addition and the multiplication of two integers expressed in binary notation. We will also analyze the computational complexity of these algorithms, in terms of the actual number of bit operations used. Throughout this discussion, suppose that the binary expansions of a and b are

$$a = (a_{n-1}a_{n-2}\dots a_1a_0)_2, \quad b = (b_{n-1}b_{n-2}\dots b_1b_0)_2,$$

so that a and b each have n bits (putting bits equal to 0 at the beginning of one of these expansions if necessary).

We will measure the complexity of algorithms for integer arithmetic in terms of the number of bits in these numbers.

ADDITION ALGORITHM Consider the problem of adding two integers in binary notation. A procedure to perform addition can be based on the usual method for adding numbers with pencil and paper. This method proceeds by adding pairs of binary digits together with carries, when they occur, to compute the sum of two integers. This procedure will now be specified in detail.

To add a and b , first add their rightmost bits. This gives

$$a_0 + b_0 = c_0 \cdot 2 + s_0,$$

where s_0 is the rightmost bit in the binary expansion of $a + b$ and c_0 is the **carry**, which is either 0 or 1. Then add the next pair of bits and the carry,

$$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1,$$

where s_1 is the next bit (from the right) in the binary expansion of $a + b$, and c_1 is the carry. Continue this process, adding the corresponding bits in the two binary expansions and the carry, to determine the next bit from the right in the binary expansion of $a + b$. At the last stage, add a_{n-1} , b_{n-1} , and c_{n-2} to obtain $c_{n-1} \cdot 2 + s_{n-1}$. The leading bit of the sum is $s_n = c_{n-1}$. This procedure produces the binary expansion of the sum, namely, $a + b = (s_ns_{n-1}s_{n-2}\dots s_1s_0)_2$.

EXAMPLE 8 Add $a = (1110)_2$ and $b = (1011)_2$.

Solution: Following the procedure specified in the algorithm, first note that

$$a_0 + b_0 = 0 + 1 = 0 \cdot 2 + 1,$$

so that $c_0 = 0$ and $s_0 = 1$. Then, because

$$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 \cdot 2 + 0,$$

it follows that $c_1 = 1$ and $s_1 = 0$. Continuing,

$$a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 \cdot 2 + 0,$$

so that $c_2 = 1$ and $s_2 = 0$. Finally, because

$$\begin{array}{r} 111 \\ 110 \\ +101 \\ \hline 11001 \end{array}$$

follows that $c_3 = 1$ and $s_3 = 1$. This means that $s_4 = c_3 = 1$. Therefore, $s = a + b = (11001)_2$. This addition is displayed in Figure 1, where carries are shown in blue. 

FIGURE 1
Adding $(1110)_2$ and $(1011)_2$.

The algorithm for addition can be described using pseudocode as follows.

ALGORITHM 2 Addition of Integers.

```

procedure add( $a, b$ : positive integers)
{the binary expansions of  $a$  and  $b$  are  $(a_{n-1}a_{n-2}\dots a_1a_0)_2$ 
 and  $(b_{n-1}b_{n-2}\dots b_1b_0)_2$ , respectively}
 $c := 0$ 
for  $j := 0$  to  $n - 1$ 
     $d := \lfloor(a_j + b_j + c)/2\rfloor$ 
     $s_j := a_j + b_j + c - 2d$ 
     $c := d$ 
 $s_n := c$ 
return  $(s_0, s_1, \dots, s_n)$  {the binary expansion of the sum is  $(s_ns_{n-1}\dots s_0)_2$ }
```

Next, the number of additions of bits used by Algorithm 2 will be analyzed.

EXAMPLE 9 How many additions of bits are required to use Algorithm 2 to add two integers with n bits (or less) in their binary representations?

Solution: Two integers are added by successively adding pairs of bits and, when it occurs, a carry. Adding each pair of bits and the carry requires two additions of bits. Thus, the total number of additions of bits used is less than twice the number of bits in the expansion. Hence, the number of additions of bits used by Algorithm 2 to add two n -bit integers is $O(n)$. 

MULTIPLICATION ALGORITHM Next, consider the multiplication of two n -bit integers a and b . The conventional algorithm (used when multiplying with pencil and paper) works as follows. Using the distributive law, we see that

$$\begin{aligned} ab &= a(b_02^0 + b_12^1 + \dots + b_{n-1}2^{n-1}) \\ &= a(b_02^0) + a(b_12^1) + \dots + a(b_{n-1}2^{n-1}). \end{aligned}$$

We can compute ab using this equation. We first note that $ab_j = a$ if $b_j = 1$ and $ab_j = 0$ if $b_j = 0$. Each time we multiply a term by 2, we shift its binary expansion one place to the left and add a zero at the tail end of the expansion. Consequently, we can obtain $(ab_j)2^j$ by shifting the binary expansion of ab_j j places to the left, adding j zero bits at the tail end of this binary expansion. Finally, we obtain ab by adding the n integers ab_j2^j , $j = 0, 1, 2, \dots, n - 1$.

Algorithm 3 displays this procedure for multiplication.

ALGORITHM 3 Multiplication of Integers.

```

procedure multiply( $a, b$ : positive integers)
{the binary expansions of  $a$  and  $b$  are  $(a_{n-1}a_{n-2}\dots a_1a_0)_2$ 
 and  $(b_{n-1}b_{n-2}\dots b_1b_0)_2$ , respectively}
for  $j := 0$  to  $n - 1$ 
    if  $b_j = 1$  then  $c_j := a$  shifted  $j$  places
    else  $c_j := 0$ 
{ $c_0, c_1, \dots, c_{n-1}$  are the partial products}
 $p := 0$ 
for  $j := 0$  to  $n - 1$ 
     $p := p + c_j$ 
return  $p$  { $p$  is the value of  $ab$ }

```

Example 10 illustrates the use of this algorithm.

EXAMPLE 10 Find the product of $a = (110)_2$ and $b = (101)_2$.

Solution: First note that

$$ab_0 \cdot 2^0 = (110)_2 \cdot 1 \cdot 2^0 = (110)_2,$$

$$ab_1 \cdot 2^1 = (110)_2 \cdot 0 \cdot 2^1 = (0000)_2,$$

$$\begin{array}{r} 110 \\ \times 101 \\ \hline 110 \\ 000 \\ \hline 1110 \end{array}$$

and

$$ab_2 \cdot 2^2 = (110)_2 \cdot 1 \cdot 2^2 = (1100)_2.$$

To find the product, add $(110)_2$, $(0000)_2$, and $(1100)_2$. Carrying out these additions (using Algorithm 2, including initial zero bits when necessary) shows that $ab = (1110)_2$. This multiplication is displayed in Figure 2. 

FIGURE 2
Multiplying
 $(110)_2$ and $(101)_2$.

Next, we determine the number of additions of bits and shifts of bits used by Algorithm 3 to multiply two integers.

EXAMPLE 11 How many additions of bits and shifts of bits are used to multiply a and b using Algorithm 3?

Solution: Algorithm 3 computes the products of a and b by adding the partial products $c_0, c_1, c_2, \dots, c_{n-1}$. When $b_j = 1$, we compute the partial product c_j by shifting the binary expansion of a by j bits. When $b_j = 0$, no shifts are required because $c_j = 0$. Hence, to find all n of the integers $ab_j 2^j$, $j = 0, 1, \dots, n - 1$, requires at most

$$0 + 1 + 2 + \dots + n - 1$$

shifts. Hence, by Example 5 in Section 3.2 the number of shifts required is $O(n^2)$.

To add the integers ab_j from $j = 0$ to $j = n - 1$ requires the addition of an n -bit integer, an $(n + 1)$ -bit integer, \dots , and a $(2n)$ -bit integer. We know from Example 9 that each of these additions requires $O(n)$ additions of bits. Consequently, a total of $O(n^2)$ additions of bits are required for all n additions. 

Surprisingly, there are more efficient algorithms than the conventional algorithm for multiplying integers. One such algorithm, which uses $O(n^{1.585})$ bit operations to multiply n -bit numbers, will be described in Section 8.3.

ALGORITHM FOR div AND mod Given integers a and d , $d > 0$, we can find $q = a \text{ div } d$ and $r = a \text{ mod } d$ using Algorithm 4. In this brute-force algorithm, when a is positive we subtract d from a as many times as necessary until what is left is less than d . The number of times we perform this subtraction is the quotient and what is left over after all these subtractions is the remainder. Algorithm 4 also covers the case where a is negative. This algorithm finds the quotient q and remainder r when $|a|$ is divided by d . Then, when $a < 0$ and $r > 0$, it uses these to find the quotient $-(q + 1)$ and remainder $d - r$ when a is divided by d . We leave it to the reader (Exercise 59) to show that, assuming that $a > d$, this algorithm uses $O(q \log a)$ bit operations.

ALGORITHM 4 Computing div and mod.

```

procedure division algorithm( $a$ : integer,  $d$ : positive integer)
 $q := 0$ 
 $r := |a|$ 
while  $r \geq d$ 
     $r := r - d$ 
     $q := q + 1$ 
if  $a < 0$  and  $r > 0$  then
     $r := d - r$ 
     $q := -(q + 1)$ 
return  $(q, r)$  { $q = a \text{ div } d$  is the quotient,  $r = a \text{ mod } d$  is the remainder}

```

There are more efficient algorithms than Algorithm 4 for determining the quotient $q = a \text{ div } d$ and the remainder $r = a \text{ mod } d$ when a positive integer a is divided by a positive integer d (see [Kn98] for details). These algorithms require $O(\log a \cdot \log d)$ bit operations. If both of the binary expansions of a and d contain n or fewer bits, then we can replace $\log a \cdot \log d$ by n^2 . This means that we need $O(n^2)$ bit operations to find the quotient and remainder when a is divided by d .

Modular Exponentiation

In cryptography it is important to be able to find $b^n \text{ mod } m$ efficiently, where b , n , and m are large integers. It is impractical to first compute b^n and then find its remainder when divided by m because b^n will be a huge number. Instead, we can use an algorithm that employs the binary expansion of the exponent n .

Before we present this algorithm, we illustrate its basic idea. We will explain how to use the binary expansion of n , say $n = (a_{k-1} \dots a_1 a_0)_2$, to compute b^n . First, note that

$$b^n = b^{a_{k-1} \cdot 2^{k-1} + \dots + a_1 \cdot 2 + a_0} = b^{a_{k-1} \cdot 2^{k-1}} \dots b^{a_1 \cdot 2} \cdot b^{a_0}.$$

This shows that to compute b^n , we need only compute the values of b , b^2 , $(b^2)^2 = b^4$, $(b^4)^2 = b^8$, \dots , b^{2^k} . Once we have these values, we multiply the terms b^{2^j} in this list, where $a_j = 1$. (For efficiency, after multiplying by each term, we reduce the result modulo m .) This gives us b^n . For example, to compute 3^{11} we first note that $11 = (1011)_2$, so that $3^{11} = 3^8 3^2 3^1$. By successively squaring, we find that $3^2 = 9$, $3^4 = 9^2 = 81$, and $3^8 = (81)^2 = 6561$. Consequently, $3^{11} = 3^8 3^2 3^1 = 6561 \cdot 9 \cdot 3 = 177,147$.

 Be sure to reduce modulo m after each multiplication!

The algorithm successively finds $b \pmod m$, $b^2 \pmod m$, $b^4 \pmod m, \dots, b^{2^{k-1}} \pmod m$ and multiplies together those terms $b^{2^j} \pmod m$ where $a_j = 1$, finding the remainder of the product when divided by m after each multiplication. Pseudocode for this algorithm is shown in Algorithm 5. Note that in Algorithm 5 we can use the most efficient algorithm available to compute values of the **mod** function, not necessarily Algorithm 4.

ALGORITHM 5 Modular Exponentiation.

```

procedure modular_exponentiation( $b$ : integer,  $n = (a_{k-1}a_{k-2}\dots a_1a_0)_2$ ,
                                 $m$ : positive integers)
     $x := 1$ 
     $power := b \pmod m$ 
    for  $i := 0$  to  $k - 1$ 
        if  $a_i = 1$  then  $x := (x \cdot power) \pmod m$ 
         $power := (power \cdot power) \pmod m$ 
    return  $x \{x \text{ equals } b^n \pmod m\}$ 
```

We illustrate how Algorithm 5 works in Example 12.

EXAMPLE 12 Use Algorithm 5 to find $3^{644} \pmod{645}$.

Solution: Algorithm 5 initially sets $x = 1$ and $power = 3 \pmod{645} = 3$. In the computation of $3^{644} \pmod{645}$, this algorithm determines $3^{2^j} \pmod{645}$ for $j = 1, 2, \dots, 9$ by successively squaring and reducing modulo 645. If $a_j = 1$ (where a_j is the bit in the j th position in the binary expansion of 644, which is $(1010000100)_2$), it multiplies the current value of x by $3^{2^j} \pmod{645}$ and reduces the result modulo 645. Here are the steps used:

- $i = 0$: Because $a_0 = 0$, we have $x = 1$ and $power = 3^2 \pmod{645} = 9 \pmod{645} = 9$;
- $i = 1$: Because $a_1 = 0$, we have $x = 1$ and $power = 9^2 \pmod{645} = 81 \pmod{645} = 81$;
- $i = 2$: Because $a_2 = 1$, we have $x = 1 \cdot 81 \pmod{645} = 81$ and $power = 81^2 \pmod{645} = 6561 \pmod{645} = 111$;
- $i = 3$: Because $a_3 = 0$, we have $x = 81$ and $power = 111^2 \pmod{645} = 12,321 \pmod{645} = 66$;
- $i = 4$: Because $a_4 = 0$, we have $x = 81$ and $power = 66^2 \pmod{645} = 4356 \pmod{645} = 486$;
- $i = 5$: Because $a_5 = 0$, we have $x = 81$ and $power = 486^2 \pmod{645} = 236,196 \pmod{645} = 126$;
- $i = 6$: Because $a_6 = 0$, we have $x = 81$ and $power = 126^2 \pmod{645} = 15,876 \pmod{645} = 396$;
- $i = 7$: Because $a_7 = 1$, we find that $x = (81 \cdot 396) \pmod{645} = 471$ and $power = 396^2 \pmod{645} = 156,816 \pmod{645} = 81$;
- $i = 8$: Because $a_8 = 0$, we have $x = 471$ and $power = 81^2 \pmod{645} = 6561 \pmod{645} = 111$;
- $i = 9$: Because $a_9 = 1$, we find that $x = (471 \cdot 111) \pmod{645} = 36$.

This shows that following the steps of Algorithm 5 produces the result $3^{644} \pmod{645} = 36$.

Algorithm 5 is quite efficient; it uses $O((\log m)^2 \log n)$ bit operations to find $b^n \pmod m$ (see Exercise 58).

Exercises

1. Convert the decimal expansion of each of these integers to a binary expansion.
 - a) 231
 - b) 4532
 - c) 97644
2. Convert the decimal expansion of each of these integers to a binary expansion.
 - a) 321
 - b) 1023
 - c) 100632
3. Convert the binary expansion of each of these integers to a decimal expansion.

a) (1 1111) ₂	b) (10 0000 0001) ₂
c) (1 0101 0101) ₂	d) (110 1001 0001 0000) ₂
4. Convert the binary expansion of each of these integers to a decimal expansion.

a) (1 0111) ₂	b) (10 1011 0101) ₂
c) (11 1011 1110) ₂	d) (111 1100 0001 1111) ₂
5. Convert the octal expansion of each of these integers to a binary expansion.

a) (572) ₈	b) (1604) ₈
c) (423) ₈	d) (2417) ₈
6. Convert the binary expansion of each of these integers to an octal expansion.

a) (1111 0111) ₂	b) (1010 1010 1010) ₂
c) (111 0111 0111 0111) ₂	d) (101 0101 0101 0101) ₂
7. Convert the hexadecimal expansion of each of these integers to a binary expansion.

a) (80E) ₁₆	b) (135AB) ₁₆
c) (ABBA) ₁₆	d) (DEFACED) ₁₆
8. Convert (BADFACED)₁₆ from its hexadecimal expansion to its binary expansion.
9. Convert (ABCDEF)₁₆ from its hexadecimal expansion to its binary expansion.
10. Convert each of the integers in Exercise 6 from a binary expansion to a hexadecimal expansion.
11. Convert (1011 0111 1011)₂ from its binary expansion to its hexadecimal expansion.
12. Convert (1 1000 0110 0011)₂ from its binary expansion to its hexadecimal expansion.
13. Show that the hexadecimal expansion of a positive integer can be obtained from its binary expansion by grouping together blocks of four binary digits, adding initial zeros if necessary, and translating each block of four binary digits into a single hexadecimal digit.
14. Show that the binary expansion of a positive integer can be obtained from its hexadecimal expansion by translating each hexadecimal digit into a block of four binary digits.
15. Show that the octal expansion of a positive integer can be obtained from its binary expansion by grouping together blocks of three binary digits, adding initial zeros if necessary, and translating each block of three binary digits into a single octal digit.
16. Show that the binary expansion of a positive integer can be obtained from its octal expansion by translating each octal digit into a block of three binary digits.
17. Convert (7345321)₈ to its binary expansion and (10 1011 1011)₂ to its octal expansion.
18. Give a procedure for converting from the hexadecimal expansion of an integer to its octal expansion using binary notation as an intermediate step.
19. Give a procedure for converting from the octal expansion of an integer to its hexadecimal expansion using binary notation as an intermediate step.
20. Explain how to convert from binary to base 64 expansions and from base 64 expansions to binary expansions and from octal to base 64 expansions and from base 64 expansions to octal expansions.
21. Find the sum and the product of each of these pairs of numbers. Express your answers as a binary expansion.

a) (100 0111) ₂ , (111 0111) ₂	b) (1110 1111) ₂ , (1011 1101) ₂
c) (10 1010 1010) ₂ , (1 1111 0000) ₂	d) (10 0000 0001) ₂ , (11 1111 1111) ₂
22. Find the sum and product of each of these pairs of numbers. Express your answers as a base 3 expansion.

a) (112) ₃ , (210) ₃	b) (2112) ₃ , (12021) ₃
c) (20001) ₃ , (1111) ₃	d) (120021) ₃ , (2002) ₃
23. Find the sum and product of each of these pairs of numbers. Express your answers as an octal expansion.

a) (763) ₈ , (147) ₈	b) (6001) ₈ , (272) ₈
c) (1111) ₈ , (777) ₈	d) (54321) ₈ , (3456) ₈
24. Find the sum and product of each of these pairs of numbers. Express your answers as a hexadecimal expansion.

a) (1AE) ₁₆ , (BBC) ₁₆	b) (20CBA) ₁₆ , (A01) ₁₆
c) (ABCDE) ₁₆ , (1111) ₁₆	d) (E0000E) ₁₆ , (BAAA) ₁₆
25. Use Algorithm 5 to find $7^{644} \pmod{645}$.
26. Use Algorithm 5 to find $11^{644} \pmod{645}$.
27. Use Algorithm 5 to find $3^{2003} \pmod{99}$.
28. Use Algorithm 5 to find $123^{1001} \pmod{101}$.
29. Show that every positive integer can be represented uniquely as the sum of distinct powers of 2. [Hint: Consider binary expansions of integers.]

- 30.** It can be shown that every integer can be uniquely represented in the form

$$e_k 3^k + e_{k-1} 3^{k-1} + \cdots + e_1 3 + e_0,$$

where $e_j = -1, 0$, or 1 for $j = 0, 1, 2, \dots, k$. Expansions of this type are called **balanced ternary expansions**. Find the balanced ternary expansions of

- a) 5. b) 13. c) 37. d) 79.

- 31.** Show that a positive integer is divisible by 3 if and only if the sum of its decimal digits is divisible by 3.

- 32.** Show that a positive integer is divisible by 11 if and only if the difference of the sum of its decimal digits in even-numbered positions and the sum of its decimal digits in odd-numbered positions is divisible by 11.

- 33.** Show that a positive integer is divisible by 3 if and only if the difference of the sum of its binary digits in even-numbered positions and the sum of its binary digits in odd-numbered positions is divisible by 3.

One's complement representations of integers are used to simplify computer arithmetic. To represent positive and negative integers with absolute value less than 2^{n-1} , a total of n bits is used. The leftmost bit is used to represent the sign. A 0 bit in this position is used for positive integers, and a 1 bit in this position is used for negative integers. For positive integers, the remaining bits are identical to the binary expansion of the integer. For negative integers, the remaining bits are obtained by first finding the binary expansion of the absolute value of the integer, and then taking the complement of each of these bits, where the complement of a 1 is a 0 and the complement of a 0 is a 1.

- 34.** Find the one's complement representations, using bit strings of length six, of the following integers.

- a) 22 b) 31 c) -7 d) -19

- 35.** What integer does each of the following one's complement representations of length five represent?

- a) 11001 b) 01101
c) 10001 d) 11111

- 36.** If m is a positive integer less than 2^{n-1} , how is the one's complement representation of $-m$ obtained from the one's complement of m , when bit strings of length n are used?

- 37.** How is the one's complement representation of the sum of two integers obtained from the one's complement representations of these integers?

- 38.** How is the one's complement representation of the difference of two integers obtained from the one's complement representations of these integers?

- 39.** Show that the integer m with one's complement representation $(a_{n-1}a_{n-2}\dots a_1a_0)$ can be found using the equation $m = -a_{n-1}(2^{n-1} - 1) + a_{n-2}2^{n-2} + \cdots + a_1 \cdot 2 + a_0$.

Two's complement representations of integers are also used to simplify computer arithmetic and are used more commonly

than one's complement representations. To represent an integer x with $-2^{n-1} \leq x \leq 2^{n-1} - 1$ for a specified positive integer n , a total of n bits is used. The leftmost bit is used to represent the sign. A 0 bit in this position is used for positive integers, and a 1 bit in this position is used for negative integers, just as in one's complement expansions. For a positive integer, the remaining bits are identical to the binary expansion of the integer. For a negative integer, the remaining bits are the bits of the binary expansion of $2^{n-1} - |x|$. Two's complement expansions of integers are often used by computers because addition and subtraction of integers can be performed easily using these expansions, where these integers can be either positive or negative.

- 40.** Answer Exercise 34, but this time find the two's complement expansion using bit strings of length six.

- 41.** Answer Exercise 35 if each expansion is a two's complement expansion of length five.

- 42.** Answer Exercise 36 for two's complement expansions.

- 43.** Answer Exercise 37 for two's complement expansions.

- 44.** Answer Exercise 38 for two's complement expansions.

- 45.** Show that the integer m with two's complement representation $(a_{n-1}a_{n-2}\dots a_1a_0)$ can be found using the equation $m = -a_{n-1} \cdot 2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 \cdot 2 + a_0$.

- 46.** Give a simple algorithm for forming the two's complement representation of an integer from its one's complement representation.

- 47.** Sometimes integers are encoded by using four-digit binary expansions to represent each decimal digit. This produces the **binary coded decimal** form of the integer. For instance, 791 is encoded in this way by 011110010001. How many bits are required to represent a number with n decimal digits using this type of encoding?

A **Cantor expansion** is a sum of the form

$$a_n n! + a_{n-1}(n-1)! + \cdots + a_2 2! + a_1 1!,$$

where a_i is an integer with $0 \leq a_i \leq i$ for $i = 1, 2, \dots, n$.

- 48.** Find the Cantor expansions of

- | | |
|----------|---------------|
| a) 2. | b) 7. |
| c) 19. | d) 87. |
| e) 1000. | f) 1,000,000. |

- *49.** Describe an algorithm that finds the Cantor expansion of an integer.

- *50.** Describe an algorithm to add two integers from their Cantor expansions.

- 51.** Add $(10111)_2$ and $(11010)_2$ by working through each step of the algorithm for addition given in the text.

- 52.** Multiply $(1110)_2$ and $(1010)_2$ by working through each step of the algorithm for multiplication given in the text.

- 53.** Describe an algorithm for finding the difference of two binary expansions.

- 54.** Estimate the number of bit operations used to subtract two binary expansions.

- 55.** Devise an algorithm that, given the binary expansions of the integers a and b , determines whether $a > b$, $a = b$, or $a < b$.
- 56.** How many bit operations does the comparison algorithm from Exercise 55 use when the larger of a and b has n bits in its binary expansion?
- 57.** Estimate the complexity of Algorithm 1 for finding the base b expansion of an integer n in terms of the number of divisions used.
- *58.** Show that Algorithm 5 uses $O((\log m)^2 \log n)$ bit operations to find $b^n \bmod m$.
- 59.** Show that Algorithm 4 uses $O(q \log a)$ bit operations, assuming that $a > d$.

4.3 Primes and Greatest Common Divisors

Introduction

In Section 4.1 we studied the concept of divisibility of integers. One important concept based on divisibility is that of a prime number. A prime is an integer greater than 1 that is divisible by no positive integers other than 1 and itself. The study of prime numbers goes back to ancient times. Thousands of years ago it was known that there are infinitely many primes; the proof of this fact, found in the works of Euclid, is famous for its elegance and beauty.

We will discuss the distribution of primes among the integers. We will describe some of the results about primes found by mathematicians in the last 400 years. In particular, we will introduce an important theorem, the fundamental theorem of arithmetic. This theorem, which asserts that every positive integer can be written uniquely as the product of primes in nondecreasing order, has many interesting consequences. We will also discuss some of the many old conjectures about primes that remain unsettled today.

Primes have become essential in modern cryptographic systems, and we will develop some of their properties important in cryptography. For example, finding large primes is essential in modern cryptography. The length of time required to factor large integers into their prime factors is the basis for the strength of some important modern cryptographic systems.

In this section we will also study the greatest common divisor of two integers, as well as the least common multiple of two integers. We will develop an important algorithm for computing greatest common divisors, called the Euclidean algorithm.

Primes

Every integer greater than 1 is divisible by at least two integers, because a positive integer is divisible by 1 and by itself. Positive integers that have exactly two different positive integer factors are called **primes**.

DEFINITION 1

An integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p . A positive integer that is greater than 1 and is not prime is called *composite*.

Remark: The integer n is composite if and only if there exists an integer a such that $a | n$ and $1 < a < n$.

EXAMPLE 1

The integer 7 is prime because its only positive factors are 1 and 7, whereas the integer 9 is composite because it is divisible by 3. 

The primes are the building blocks of positive integers, as the fundamental theorem of arithmetic shows. The proof will be given in Section 5.2.

THEOREM 1

THE FUNDAMENTAL THEOREM OF ARITHMETIC Every integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of nondecreasing size.

Example 2 gives some prime factorizations of integers.

EXAMPLE 2

The prime factorizations of 100, 641, 999, and 1024 are given by



$$\begin{aligned}100 &= 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 5^2, \\641 &= 641, \\999 &= 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37, \\1024 &= 2 \cdot 2 = 2^{10}. \end{aligned}$$



Trial Division

It is often important to show that a given integer is prime. For instance, in cryptology, large primes are used in some methods for making messages secret. One procedure for showing that an integer is prime is based on the following observation.

THEOREM 2

If n is a composite integer, then n has a prime divisor less than or equal to \sqrt{n} .

Proof: If n is composite, by the definition of a composite integer, we know that it has a factor a with $1 < a < n$. Hence, by the definition of a factor of a positive integer, we have $n = ab$, where b is a positive integer greater than 1. We will show that $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$. If $a > \sqrt{n}$ and $b > \sqrt{n}$, then $ab > \sqrt{n} \cdot \sqrt{n} = n$, which is a contradiction. Consequently, $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$. Because both a and b are divisors of n , we see that n has a positive divisor not exceeding \sqrt{n} . This divisor is either prime or, by the fundamental theorem of arithmetic, has a prime divisor less than itself. In either case, n has a prime divisor less than or equal to \sqrt{n} . ◀

From Theorem 2, it follows that an integer is prime if it is not divisible by any prime less than or equal to its square root. This leads to the brute-force algorithm known as **trial division**. To use trial division we divide n by all primes not exceeding \sqrt{n} and conclude that n is prime if it is not divisible by any of these primes. In Example 3 we use trial division to show that 101 is prime.

EXAMPLE 3

Show that 101 is prime.

Solution: The only primes not exceeding $\sqrt{101}$ are 2, 3, 5, and 7. Because 101 is not divisible by 2, 3, 5, or 7 (the quotient of 101 and each of these integers is not an integer), it follows that 101 is prime. ◀

Because every integer has a prime factorization, it would be useful to have a procedure for finding this prime factorization. Consider the problem of finding the prime factorization of n . Begin by dividing n by successive primes, starting with the smallest prime, 2. If n has a prime factor, then by Theorem 3 a prime factor p not exceeding \sqrt{n} will be found. So, if no prime

factor not exceeding \sqrt{n} is found, then n is prime. Otherwise, if a prime factor p is found, continue by factoring n/p . Note that n/p has no prime factors less than p . Again, if n/p has no prime factor greater than or equal to p and not exceeding its square root, then it is prime. Otherwise, if it has a prime factor q , continue by factoring $n/(pq)$. This procedure is continued until the factorization has been reduced to a prime. This procedure is illustrated in Example 4.

EXAMPLE 4 Find the prime factorization of 7007.

Solution: To find the prime factorization of 7007, first perform divisions of 7007 by successive primes, beginning with 2. None of the primes 2, 3, and 5 divides 7007. However, 7 divides 7007, with $7007/7 = 1001$. Next, divide 1001 by successive primes, beginning with 7. It is immediately seen that 7 also divides 1001, because $1001/7 = 143$. Continue by dividing 143 by successive primes, beginning with 7. Although 7 does not divide 143, 11 does divide 143, and $143/11 = 13$. Because 13 is prime, the procedure is completed. It follows that $7007 = 7 \cdot 1001 = 7 \cdot 7 \cdot 143 = 7 \cdot 7 \cdot 11 \cdot 13$. Consequently, the prime factorization of 7007 is $7 \cdot 7 \cdot 11 \cdot 13 = 7^2 \cdot 11 \cdot 13$. 

Prime numbers were studied in ancient times for philosophical reasons. Today, there are highly practical reasons for their study. In particular, large primes play a crucial role in cryptography, as we will see in Section 4.6.



The Sieve of Eratosthenes

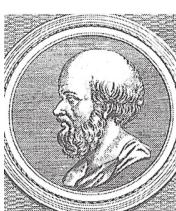
Note that composite integers not exceeding 100 must have a prime factor not exceeding 10. Because the only primes less than 10 are 2, 3, 5, and 7, the primes not exceeding 100 are these four primes and those positive integers greater than 1 and not exceeding 100 that are divisible by none of 2, 3, 5, or 7.



The **sieve of Eratosthenes** is used to find all primes not exceeding a specified positive integer. For instance, the following procedure is used to find the primes not exceeding 100. We begin with the list of all integers between 1 and 100. To begin the sieving process, the integers that are divisible by 2, other than 2, are deleted. Because 3 is the first integer greater than 2 that is left, all those integers divisible by 3, other than 3, are deleted. Because 5 is the next integer left after 3, those integers divisible by 5, other than 5, are deleted. The next integer left is 7, so those integers divisible by 7, other than 7, are deleted. Because all composite integers not exceeding 100 are divisible by 2, 3, 5, or 7, all remaining integers except 1 are prime. In Table 1, the panels display those integers deleted at each stage, where each integer divisible by 2, other than 2, is underlined in the first panel, each integer divisible by 3, other than 3, is underlined in the second panel, each integer divisible by 5, other than 5, is underlined in the third panel, and each integer divisible by 7, other than 7, is underlined in the fourth panel. The integers not underlined are the primes not exceeding 100. We conclude that the primes less than 100 are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, and 97.



THE INFINITE OF PRIMES It has long been known that there are infinitely many primes. This means that whenever p_1, p_2, \dots, p_n are the n smallest primes, we know there is a larger



ERATOSTHENES (276 B.C.E.–194 B.C.E.) It is known that Eratosthenes was born in Cyrene, a Greek colony west of Egypt, and spent time studying at Plato's Academy in Athens. We also know that King Ptolemy II invited Eratosthenes to Alexandria to tutor his son and that later Eratosthenes became chief librarian at the famous library at Alexandria, a central repository of ancient wisdom. Eratosthenes was an extremely versatile scholar, writing on mathematics, geography, astronomy, history, philosophy, and literary criticism. Besides his work in mathematics, he is most noted for his chronology of ancient history and for his famous measurement of the size of the earth.

TABLE 1 The Sieve of Eratosthenes.

Integers divisible by 2 other than 2 receive an underline.										Integers divisible by 3 other than 3 receive an underline.									
1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>	1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>
11	<u>12</u>	13	<u>14</u>	15	<u>16</u>	17	<u>18</u>	19	<u>20</u>	11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>
21	<u>22</u>	23	<u>24</u>	25	<u>26</u>	27	<u>28</u>	29	<u>30</u>	<u>21</u>	<u>22</u>	23	<u>24</u>	25	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>
31	<u>32</u>	33	<u>34</u>	35	<u>36</u>	37	<u>38</u>	39	<u>40</u>	31	<u>32</u>	33	<u>34</u>	35	<u>36</u>	37	<u>38</u>	39	<u>40</u>
41	<u>42</u>	43	<u>44</u>	45	<u>46</u>	47	<u>48</u>	49	<u>50</u>	41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>
51	<u>52</u>	53	<u>54</u>	55	<u>56</u>	57	<u>58</u>	59	<u>60</u>	<u>51</u>	<u>52</u>	53	<u>54</u>	55	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>
61	<u>62</u>	63	<u>64</u>	65	<u>66</u>	67	<u>68</u>	69	<u>70</u>	61	<u>62</u>	63	<u>64</u>	65	<u>66</u>	67	<u>68</u>	69	<u>70</u>
71	<u>72</u>	73	<u>74</u>	75	<u>76</u>	77	<u>78</u>	79	<u>80</u>	71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>
81	<u>82</u>	83	<u>84</u>	85	<u>86</u>	87	<u>88</u>	89	<u>90</u>	81	<u>82</u>	83	<u>84</u>	85	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>
91	<u>92</u>	93	<u>94</u>	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>	91	<u>92</u>	<u>93</u>	<u>94</u>	95	<u>96</u>	97	<u>98</u>	99	<u>100</u>
Integers divisible by 5 other than 5 receive an underline.										Integers divisible by 7 other than 7 receive an underline; integers in color are prime.									
1	2	3	<u>4</u>	5	<u>6</u>	7	<u>8</u>	9	<u>10</u>	1	<u>2</u>	3	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	9	<u>10</u>
11	<u>12</u>	13	<u>14</u>	<u>15</u>	<u>16</u>	17	<u>18</u>	19	<u>20</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>
<u>21</u>	<u>22</u>	23	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	29	<u>30</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>
31	<u>32</u>	33	<u>34</u>	<u>35</u>	<u>36</u>	37	<u>38</u>	39	<u>40</u>	31	<u>32</u>	33	<u>34</u>	<u>35</u>	<u>36</u>	<u>37</u>	<u>38</u>	39	<u>40</u>
41	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	47	<u>48</u>	49	<u>50</u>	<u>41</u>	<u>42</u>	43	<u>44</u>	<u>45</u>	<u>46</u>	<u>47</u>	<u>48</u>	49	<u>50</u>
<u>51</u>	<u>52</u>	53	<u>54</u>	<u>55</u>	<u>56</u>	<u>57</u>	<u>58</u>	59	<u>60</u>	<u>51</u>	<u>52</u>	<u>53</u>	<u>54</u>	<u>55</u>	<u>56</u>	<u>57</u>	<u>58</u>	<u>59</u>	<u>60</u>
61	<u>62</u>	63	<u>64</u>	<u>65</u>	<u>66</u>	67	<u>68</u>	69	<u>70</u>	<u>61</u>	<u>62</u>	<u>63</u>	<u>64</u>	<u>65</u>	<u>66</u>	<u>67</u>	<u>68</u>	69	<u>70</u>
71	<u>72</u>	73	<u>74</u>	<u>75</u>	<u>76</u>	77	<u>78</u>	79	<u>80</u>	<u>71</u>	<u>72</u>	<u>73</u>	<u>74</u>	<u>75</u>	<u>76</u>	<u>77</u>	<u>78</u>	<u>79</u>	<u>80</u>
81	<u>82</u>	83	<u>84</u>	<u>85</u>	<u>86</u>	<u>87</u>	<u>88</u>	89	<u>90</u>	81	<u>82</u>	<u>83</u>	<u>84</u>	<u>85</u>	<u>86</u>	<u>87</u>	<u>88</u>	<u>89</u>	<u>90</u>
91	<u>92</u>	93	<u>94</u>	<u>95</u>	<u>96</u>	97	<u>98</u>	99	<u>100</u>	91	<u>92</u>	<u>93</u>	<u>94</u>	<u>95</u>	<u>96</u>	<u>97</u>	<u>98</u>	99	<u>100</u>

prime not listed. We will prove this fact using a proof given by Euclid in his famous mathematics text, *The Elements*. This simple, yet elegant, proof is considered by many mathematicians to be among the most beautiful proofs in mathematics. It is the first proof presented in the book *Proofs from THE BOOK* [AiZi10], where THE BOOK refers to the imagined collection of perfect proofs that the famous mathematician Paul Erdős claimed is maintained by God. By the way, there are a vast number of different proofs than there are an infinitude of primes, and new ones are published surprisingly frequently.

THEOREM 3

There are infinitely many primes.



Proof: We will prove this theorem using a proof by contradiction. We assume that there are only finitely many primes, p_1, p_2, \dots, p_n . Let

$$Q = p_1 p_2 \cdots p_n + 1.$$

By the fundamental theorem of arithmetic, Q is prime or else it can be written as the product of two or more primes. However, none of the primes p_j divides Q , for if $p_j \mid Q$, then p_j divides

$Q - p_1 p_2 \cdots p_n = 1$. Hence, there is a prime not in the list p_1, p_2, \dots, p_n . This prime is either Q , if it is prime, or a prime factor of Q . This is a contradiction because we assumed that we have listed all the primes. Consequently, there are infinitely many primes. \triangleleft

Remark: Note that in this proof we do *not* state that Q is prime! Furthermore, in this proof, we have given a nonconstructive existence proof that given any n primes, there is a prime not in this list. For this proof to be constructive, we would have had to explicitly give a prime not in our original list of n primes.

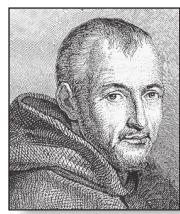
Because there are infinitely many primes, given any positive integer there are primes greater than this integer. There is an ongoing quest to discover larger and larger prime numbers; for almost all the last 300 years, the largest prime known has been an integer of the special form $2^p - 1$, where p is also prime. (Note that $2^n - 1$ cannot be prime when n is not prime; see Exercise 9.) Such primes are called **Mersenne primes**, after the French monk Marin Mersenne, who studied them in the seventeenth century. The reason that the largest known prime has usually been a Mersenne prime is that there is an extremely efficient test, known as the Lucas–Lehmer test, for determining whether $2^p - 1$ is prime. Furthermore, it is not currently possible to test numbers not of this or certain other special forms anywhere near as quickly to determine whether they are prime.

EXAMPLE 5 The numbers $2^2 - 1 = 3$, $2^3 - 1 = 7$, $2^5 - 1 = 31$ and $2^7 - 1 = 127$ are Mersenne primes, while $2^{11} - 1 = 2047$ is not a Mersenne prime because $2047 = 23 \cdot 89$. \triangleleft



Progress in finding Mersenne primes has been steady since computers were invented. As of early 2011, 47 Mersenne primes were known, with 16 found since 1990. The largest Mersenne prime known (again as of early 2011) is $2^{43,112,609} - 1$, a number with nearly 13 million decimal digits, which was shown to be prime in 2008. A communal effort, the Great Internet Mersenne Prime Search (GIMPS), is devoted to the search for new Mersenne primes. You can join this search, and if you are lucky, find a new Mersenne prime and possibly even win a cash prize. By the way, even the search for Mersenne primes has practical implications. One quality control test for supercomputers has been to replicate the Lucas–Lehmer test that establishes the primality of a large Mersenne prime. (See [Ro10] for more information about the quest for finding Mersenne primes.)

THE DISTRIBUTION OF PRIMES Theorem 3 tells us that there are infinitely many primes. However, how many primes are less than a positive number x ? This question interested mathematicians for many years; in the late eighteenth century, mathematicians produced large tables



MARIN MERSENNE (1588–1648) Mersenne was born in Maine, France, into a family of laborers and attended the College of Mans and the Jesuit College at La Flèche. He continued his education at the Sorbonne, studying theology from 1609 to 1611. He joined the religious order of the Minims in 1611, a group whose name comes from the word *minimi* (the members of this group were extremely humble; they considered themselves the least of all religious orders). Besides prayer, the members of this group devoted their energy to scholarship and study. In 1612 he became a priest at the Place Royale in Paris; between 1614 and 1618 he taught philosophy at the Minim Convent at Nevers. He returned to Paris in 1619, where his cell in the Minims de l'Annociade became a place for meetings of French scientists, philosophers, and mathematicians, including Fermat and Pascal. Mersenne corresponded extensively with scholars throughout Europe, serving as a clearinghouse for mathematical and scientific knowledge, a function later served by mathematical journals (and today also by the Internet). Mersenne wrote books covering mechanics, mathematical physics, mathematics, music, and acoustics. He studied prime numbers and tried unsuccessfully to construct a formula representing all primes. In 1644 Mersenne claimed that $2^p - 1$ is prime for $p = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$ but is composite for all other primes less than 257. It took over 300 years to determine that Mersenne's claim was wrong five times. Specifically, $2^p - 1$ is not prime for $p = 67$ and $p = 257$ but is prime for $p = 61$, $p = 87$, and $p = 107$. It is also noteworthy that Mersenne defended two of the most famous men of his time, Descartes and Galileo, from religious critics. He also helped expose alchemists and astrologers as frauds.

of prime numbers to gather evidence concerning the distribution of primes. Using this evidence, the great mathematicians of the day, including Gauss and Legendre, conjectured, but did not prove, Theorem 4.

THEOREM 4

THE PRIME NUMBER THEOREM The ratio of the number of primes not exceeding x and $x/\ln x$ approaches 1 as x grows without bound. (Here $\ln x$ is the natural logarithm of x .)



The prime number theorem was first proved in 1896 by the French mathematician Jacques Hadamard and the Belgian mathematician Charles-Jean-Gustave-Nicholas de la Vallée-Poussin using the theory of complex variables. Although proofs not using complex variables have been found, all known proofs of the prime number theorem are quite complicated.

We can use the prime number theorem to estimate the odds that a randomly chosen number is prime. The prime number theorem tells us that the number of primes not exceeding x can be approximated by $x/\ln x$. Consequently, the odds that a randomly selected positive integer less than n is prime are approximately $(n/\ln n)/n = 1/\ln n$. Sometimes we need to find a prime with a particular number of digits. We would like an estimate of how many integers with a particular number of digits we need to select before we encounter a prime. Using the prime number theorem and calculus, it can be shown that the probability that an integer n is prime is also approximately $1/\ln n$. For example, the odds that an integer near 10^{1000} is prime are approximately $1/\ln 10^{1000}$, which is approximately $1/2300$. (Of course, by choosing only odd numbers, we double our chances of finding a prime.)

Using trial division with Theorem 2 gives procedures for factoring and for primality testing. However, these procedures are not efficient algorithms; many much more practical and efficient algorithms for these tasks have been developed. Factoring and primality testing have become important in the applications of number theory to cryptography. This has led to a great interest in developing efficient algorithms for both tasks. Clever procedures have been devised in the last 30 years for efficiently generating large primes. Moreover, in 2002, an important theoretical discovery was made by Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. They showed there is a polynomial-time algorithm in the number of bits in the binary expansion of an integer for determining whether a positive integer is prime. Algorithms based on their work use $O((\log n)^6)$ bit operations to determine whether a positive integer n is prime.

However, even though powerful new factorization methods have been developed in the same time frame, factoring large numbers remains extraordinarily more time-consuming than primality testing. No polynomial-time algorithm for factoring integers is known. Nevertheless, the challenge of factoring large numbers interests many people. There is a communal effort on the Internet to factor large numbers, especially those of the special form $k^n \pm 1$, where k is a small positive integer and n is a large positive integer (such numbers are called *Cunningham numbers*). At any given time, there is a list of the “Ten Most Wanted” large numbers of this type awaiting factorization.

PRIMES AND ARITHMETIC PROGRESSIONS Every odd integer is in one of the two arithmetic progressions $4k + 1$ or $4k + 3$, $k = 1, 2, \dots$. Because we know that there are infinitely many primes, we can ask whether there are infinitely many primes in both of these arithmetic progressions. The primes 5, 13, 17, 29, 37, 41, ... are in the arithmetic progression $4k + 1$; the primes 3, 7, 11, 19, 23, 31, 43, ... are in the arithmetic progression $4k + 3$. Looking at the evidence hints that there may be infinitely many primes in both progressions. What about other arithmetic progressions $ak + b$, $k = 1, 2, \dots$, where no integer greater than one divides both a and b ? Do they contain infinitely many primes? The answer was provided by the German mathematician G. Lejeune Dirichlet, who proved that every such arithmetic progression contains infinitely many primes. His proof, and all proofs found later, are beyond the scope of this book.

However, it is possible to prove special cases of Dirichlet's theorem using the ideas developed in this book. For example, Exercises 54 and 55 ask for proofs that there are infinitely many primes in the arithmetic progressions $3k + 2$ and $4k + 3$, where k is a positive integer. (The hint for each of these exercises supplies the basic idea needed for the proof.)

We have explained that every arithmetic progression $ak + b$, $k = 1, 2, \dots$, where a and b have no common factor greater than one, contains infinitely many primes. But are there long arithmetic progressions made up of just primes? For example, some exploration shows that 5, 11, 17, 23, 29 is an arithmetic progression of five primes and 199, 409, 619, 829, 1039, 1249, 1459, 1669, 1879, 2089 is an arithmetic progression of ten primes. In the 1930s, the famous mathematician Paul Erdős conjectured that for every positive integer n greater than two, there is an arithmetic progression of length n made up entirely of primes. In 2006, Ben Green and Terence Tao were able to prove this conjecture. Their proof, considered to be a mathematical tour de force, is a nonconstructive proof that combines powerful ideas from several advanced areas of mathematics.

Conjectures and Open Problems About Primes

Number theory is noted as a subject for which it is easy to formulate conjectures, some of which are difficult to prove and others that remained open problems for many years. We will describe some conjectures in number theory and discuss their status in Examples 6–9.

EXAMPLE 6



It would be useful to have a function $f(n)$ such that $f(n)$ is prime for all positive integers n . If we had such a function, we could find large primes for use in cryptography and other applications. Looking for such a function, we might check out different polynomial functions, as some mathematicians did several hundred years ago. After a lot of computation we may encounter the polynomial $f(n) = n^2 - n + 41$. This polynomial has the interesting property that $f(n)$ is prime for all positive integers n not exceeding 40. [We have $f(1) = 41$, $f(2) = 43$, $f(3) = 47$, $f(4) = 53$, and so on.] This can lead us to the conjecture that $f(n)$ is prime for all positive integers n . Can we settle this conjecture?

Solution: Perhaps not surprisingly, this conjecture turns out to be false; we do not have to look far to find a positive integer n for which $f(n)$ is composite, because $f(41) = 41^2 - 41 + 41 = 41^2$. Because $f(n) = n^2 - n + 41$ is prime for all positive integers n with $1 \leq n \leq 40$, we might



TERENCE TAO (BORN 1975) Tao was born in Australia. His father is a pediatrician and his mother taught mathematics at a Hong Kong secondary school. Tao was a child prodigy, teaching himself arithmetic at the age of two. At 10, he became the youngest contestant at the International Mathematical Olympiad (IMO); he won an IMO gold medal at 13. Tao received his bachelors and masters degrees when he was 17, and began graduate studies at Princeton, receiving his Ph.D. in three years. In 1996 he became a faculty member at UCLA, where he continues to work.

Tao is extremely versatile; he enjoys working on problems in diverse areas, including harmonic analysis, partial differential equations, number theory, and combinatorics. You can follow his work by reading his blog where he discusses progress on various problems. His most famous result is the Green-Tao theorem, which says that there are arbitrarily long arithmetic progressions of primes. Tao has made important contributions to the applications of mathematics, such as developing a method for reconstructing digital images using the least possible amount of information. Tao has an amazing reputation among mathematicians; he has become a Mr. Fix-It for researchers in mathematics. The well-known mathematician Charles Fefferman, himself a child prodigy, has said that "if you're stuck on a problem, then one way out is to interest Terence Tao." In 2006 Tao was awarded a Fields Medal, the most prestigious award for mathematicians under the age of 40. He was also awarded a MacArthur Fellowship in 2006, and in 2008, he received the Allan T. Waterman award, which came with a \$500,000 cash prize to support research work of scientists early in their career. Tao's wife Laura is an engineer at the Jet Propulsion Laboratory.

be tempted to find a different polynomial with the property that $f(n)$ is prime for *all* positive integers n . However, there is no such polynomial. It can be shown that for every polynomial $f(n)$ with integer coefficients, there is a positive integer y such that $f(y)$ is composite. (See Exercise 23 in the Supplementary Exercises.)

Many famous problems about primes still await ultimate resolution by clever people. We describe a few of the most accessible and better known of these open problems in Examples 7–9. Number theory is noted for its wealth of easy-to-understand conjectures that resist attack by all but the most sophisticated techniques, or simply resist all attacks. We present these conjectures to show that many questions that seem relatively simple remain unsettled even in the twenty-first century.

EXAMPLE 7

Goldbach's Conjecture In 1742, Christian Goldbach, in a letter to Leonhard Euler, conjectured that every odd integer $n, n > 5$, is the sum of three primes. Euler replied that this conjecture is equivalent to the conjecture that every even integer $n, n > 2$, is the sum of two primes (see Exercise 21 in the Supplementary Exercises). The conjecture that every even integer $n, n > 2$, is the sum of two primes is now called **Goldbach's conjecture**. We can check this conjecture for small even numbers. For example, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 5 + 3$, $10 = 7 + 3$, $12 = 7 + 5$, and so on. Goldbach's conjecture was verified by hand calculations for numbers up to the millions prior to the advent of computers. With computers it can be checked for extremely large numbers. As of mid 2011, the conjecture has been checked for all positive even integers up to $1.6 \cdot 10^{18}$.



Although no proof of Goldbach's conjecture has been found, most mathematicians believe it is true. Several theorems have been proved, using complicated methods from analytic number theory far beyond the scope of this book, establishing results weaker than Goldbach's conjecture. Among these are the result that every even integer greater than 2 is the sum of at most six primes (proved in 1995 by O. Ramaré) and that every sufficiently large positive integer is the sum of a prime and a number that is either prime or the product of two primes (proved in 1966 by J. R. Chen). Perhaps Goldbach's conjecture will be settled in the not too distant future.

EXAMPLE 8



There are many conjectures asserting that there are infinitely many primes of certain special forms. A conjecture of this sort is the conjecture that there are infinitely many primes of the form $n^2 + 1$, where n is a positive integer. For example, $5 = 2^2 + 1$, $17 = 4^2 + 1$, $37 = 6^2 + 1$, and so on. The best result currently known is that there are infinitely many positive integers n such that $n^2 + 1$ is prime or the product of at most two primes (proved by Henryk Iwaniec in 1973 using advanced techniques from analytic number theory, far beyond the scope of this book).

EXAMPLE 9



The Twin Prime Conjecture **Twin primes** are pairs of primes that differ by 2, such as 3 and 5, 5 and 7, 11 and 13, 17 and 19, and 4967 and 4969. The twin prime conjecture asserts that there are infinitely many twin primes. The strongest result proved concerning twin primes is that there are infinitely many pairs p and $p + 2$, where p is prime and $p + 2$ is prime or the product of two primes (proved by J. R. Chen in 1966). The world's record for twin primes, as of mid 2011, consists of the numbers $65,516,468,355 \cdot 2^{333,333} \pm 1$, which have 100,355 decimal digits.



CHRISTIAN GOLDBACH (1690–1764) Christian Goldbach was born in Königsberg, Prussia, the city noted for its famous bridge problem (which will be studied in Section 10.5). He became professor of mathematics at the Academy in St. Petersburg in 1725. In 1728 Goldbach went to Moscow to tutor the son of the Tsar. He entered the world of politics when, in 1742, he became a staff member in the Russian Ministry of Foreign Affairs. Goldbach is best known for his correspondence with eminent mathematicians, including Euler and Bernoulli, for his famous conjectures in number theory, and for several contributions to analysis.

Greatest Common Divisors and Least Common Multiples

The largest integer that divides both of two integers is called the **greatest common divisor** of these integers.

DEFINITION 2

Let a and b be integers, not both zero. The largest integer d such that $d \mid a$ and $d \mid b$ is called the *greatest common divisor* of a and b . The greatest common divisor of a and b is denoted by $\gcd(a, b)$.

The greatest common divisor of two integers, not both zero, exists because the set of common divisors of these integers is nonempty and finite. One way to find the greatest common divisor of two integers is to find all the positive common divisors of both integers and then take the largest divisor. This is done in Examples 10 and 11. Later, a more efficient method of finding greatest common divisors will be given.

EXAMPLE 10 What is the greatest common divisor of 24 and 36?

Solution: The positive common divisors of 24 and 36 are 1, 2, 3, 4, 6, and 12. Hence, $\gcd(24, 36) = 12$. 

EXAMPLE 11 What is the greatest common divisor of 17 and 22?

Solution: The integers 17 and 22 have no positive common divisors other than 1, so that $\gcd(17, 22) = 1$. 

Because it is often important to specify that two integers have no common positive divisor other than 1, we have Definition 3.

DEFINITION 3

The integers a and b are *relatively prime* if their greatest common divisor is 1.

EXAMPLE 12 By Example 11 it follows that the integers 17 and 22 are relatively prime, because $\gcd(17, 22) = 1$. 

Because we often need to specify that no two integers in a set of integers have a common positive divisor greater than 1, we make Definition 4.

DEFINITION 4

The integers a_1, a_2, \dots, a_n are *pairwise relatively prime* if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

EXAMPLE 13 Determine whether the integers 10, 17, and 21 are pairwise relatively prime and whether the integers 10, 19, and 24 are pairwise relatively prime.

Solution: Because $\gcd(10, 17) = 1$, $\gcd(10, 21) = 1$, and $\gcd(17, 21) = 1$, we conclude that 10, 17, and 21 are pairwise relatively prime.

Because $\gcd(10, 24) = 2 > 1$, we see that 10, 19, and 24 are not pairwise relatively prime. 

Another way to find the greatest common divisor of two positive integers is to use the prime factorizations of these integers. Suppose that the prime factorizations of the positive integers a and b are

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}, \quad b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n},$$

where each exponent is a nonnegative integer, and where all primes occurring in the prime factorization of either a or b are included in both factorizations, with zero exponents if necessary. Then $\gcd(a, b)$ is given by

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_n^{\min(a_n, b_n)},$$

where $\min(x, y)$ represents the minimum of the two numbers x and y . To show that this formula for $\gcd(a, b)$ is valid, we must show that the integer on the right-hand side divides both a and b , and that no larger integer also does. This integer does divide both a and b , because the power of each prime in the factorization does not exceed the power of this prime in either the factorization of a or that of b . Further, no larger integer can divide both a and b , because the exponents of the primes in this factorization cannot be increased, and no other primes can be included.

EXAMPLE 14 Because the prime factorizations of 120 and 500 are $120 = 2^3 \cdot 3 \cdot 5$ and $500 = 2^2 \cdot 5^3$, the greatest common divisor is

$$\gcd(120, 500) = 2^{\min(3, 2)} 3^{\min(1, 0)} 5^{\min(1, 3)} = 2^2 3^0 5^1 = 20. \quad \blacktriangleleft$$

Prime factorizations can also be used to find the **least common multiple** of two integers.

DEFINITION 5

The *least common multiple* of the positive integers a and b is the smallest positive integer that is divisible by both a and b . The least common multiple of a and b is denoted by $\text{lcm}(a, b)$.

The least common multiple exists because the set of integers divisible by both a and b is nonempty (as ab belongs to this set, for instance), and every nonempty set of positive integers has a least element (by the well-ordering property, which will be discussed in Section 5.2). Suppose that the prime factorizations of a and b are as before. Then the least common multiple of a and b is given by

$$\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \cdots p_n^{\max(a_n, b_n)},$$

where $\max(x, y)$ denotes the maximum of the two numbers x and y . This formula is valid because a common multiple of a and b has at least $\max(a_i, b_i)$ factors of p_i in its prime factorization, and the least common multiple has no other prime factors besides those in a and b .

EXAMPLE 15 What is the least common multiple of $2^3 3^5 7^2$ and $2^4 3^3$?

Solution: We have

$$\text{lcm}(2^3 3^5 7^2, 2^4 3^3) = 2^{\max(3, 4)} 3^{\max(5, 3)} 7^{\max(2, 0)} = 2^4 3^5 7^2. \quad \blacktriangleleft$$

Theorem 5 gives the relationship between the greatest common divisor and least common multiple of two integers. It can be proved using the formulae we have derived for these quantities. The proof of this theorem is left as Exercise 31.

THEOREM 5

Let a and b be positive integers. Then

$$ab = \gcd(a, b) \cdot \text{lcm}(a, b).$$

The Euclidean Algorithm



Computing the greatest common divisor of two integers directly from the prime factorizations of these integers is inefficient. The reason is that it is time-consuming to find prime factorizations. We will give a more efficient method of finding the greatest common divisor, called the **Euclidean algorithm**. This algorithm has been known since ancient times. It is named after the ancient Greek mathematician Euclid, who included a description of this algorithm in his book *The Elements*.

Before describing the Euclidean algorithm, we will show how it is used to find $\gcd(91, 287)$. First, divide 287, the larger of the two integers, by 91, the smaller, to obtain

$$287 = 91 \cdot 3 + 14.$$

Any divisor of 91 and 287 must also be a divisor of $287 - 91 \cdot 3 = 14$. Also, any divisor of 91 and 14 must also be a divisor of $287 = 91 \cdot 3 + 14$. Hence, the greatest common divisor of 91 and 287 is the same as the greatest common divisor of 91 and 14. This means that the problem of finding $\gcd(91, 287)$ has been reduced to the problem of finding $\gcd(91, 14)$.

Next, divide 91 by 14 to obtain

$$91 = 14 \cdot 6 + 7.$$

Because any common divisor of 91 and 14 also divides $91 - 14 \cdot 6 = 7$ and any common divisor of 14 and 7 divides 91, it follows that $\gcd(91, 14) = \gcd(14, 7)$.

Continue by dividing 14 by 7, to obtain

$$14 = 7 \cdot 2.$$

Because 7 divides 14, it follows that $\gcd(14, 7) = 7$. Furthermore, because $\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$, the original problem has been solved.

We now describe how the Euclidean algorithm works in generality. We will use successive divisions to reduce the problem of finding the greatest common divisor of two positive integers to the same problem with smaller integers, until one of the integers is zero.

The Euclidean algorithm is based on the following result about greatest common divisors and the division algorithm.



EUCLID (325 B.C.E.–265 B.C.E.) Euclid was the author of the most successful mathematics book ever written, *The Elements*, which appeared in over 1000 different editions from ancient to modern times. Little is known about Euclid's life, other than that he taught at the famous academy at Alexandria in Egypt. Apparently, Euclid did not stress applications. When a student asked what he would get by learning geometry, Euclid explained that knowledge was worth acquiring for its own sake and told his servant to give the student a coin "because he must make a profit from what he learns."

LEMMA 1 Let $a = bq + r$, where a, b, q , and r are integers. Then $\gcd(a, b) = \gcd(b, r)$.

Proof: If we can show that the common divisors of a and b are the same as the common divisors of b and r , we will have shown that $\gcd(a, b) = \gcd(b, r)$, because both pairs must have the same *greatest* common divisor.

So suppose that d divides both a and b . Then it follows that d also divides $a - bq = r$ (from Theorem 1 of Section 4.1). Hence, any common divisor of a and b is also a common divisor of b and r .

Likewise, suppose that d divides both b and r . Then d also divides $bq + r = a$. Hence, any common divisor of b and r is also a common divisor of a and b .

Consequently, $\gcd(a, b) = \gcd(b, r)$. □

Suppose that a and b are positive integers with $a \geq b$. Let $r_0 = a$ and $r_1 = b$. When we successively apply the division algorithm, we obtain

$$\begin{aligned} r_0 &= r_1q_1 + r_2 & 0 \leq r_2 < r_1, \\ r_1 &= r_2q_2 + r_3 & 0 \leq r_3 < r_2, \\ &\vdots & \\ &\vdots & \\ r_{n-2} &= r_{n-1}q_{n-1} + r_n & 0 \leq r_n < r_{n-1}, \\ r_{n-1} &= r_nq_n. \end{aligned}$$

Eventually a remainder of zero occurs in this sequence of successive divisions, because the sequence of remainders $a = r_0 > r_1 > r_2 > \dots \geq 0$ cannot contain more than a terms. Furthermore, it follows from Lemma 1 that

$$\begin{aligned} \gcd(a, b) &= \gcd(r_0, r_1) = \gcd(r_1, r_2) = \dots = \gcd(r_{n-2}, r_{n-1}) \\ &= \gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n. \end{aligned}$$

Hence, the greatest common divisor is the last nonzero remainder in the sequence of divisions.

EXAMPLE 16 Find the greatest common divisor of 414 and 662 using the Euclidean algorithm.

Solution: Successive uses of the division algorithm give:

$$\begin{aligned} 662 &= 414 \cdot 1 + 248 \\ 414 &= 248 \cdot 1 + 166 \\ 248 &= 166 \cdot 1 + 82 \\ 166 &= 82 \cdot 2 + 2 \\ 82 &= 2 \cdot 41. \end{aligned}$$

Hence, $\gcd(414, 662) = 2$, because 2 is the last nonzero remainder. □

The Euclidean algorithm is expressed in pseudocode in Algorithm 1.

ALGORITHM 1 The Euclidean Algorithm.

```

procedure gcd( $a, b$ : positive integers)
   $x := a$ 
   $y := b$ 
  while  $y \neq 0$ 
     $r := x \bmod y$ 
     $x := y$ 
     $y := r$ 
  return  $x\{gcd(a, b) \text{ is } x\}$ 

```

In Algorithm 1, the initial values of x and y are a and b , respectively. At each stage of the procedure, x is replaced by y , and y is replaced by $x \bmod y$, which is the remainder when x is divided by y . This process is repeated as long as $y \neq 0$. The algorithm terminates when $y = 0$, and the value of x at that point, the last nonzero remainder in the procedure, is the greatest common divisor of a and b .

We will study the time complexity of the Euclidean algorithm in Section 5.3, where we will show that the number of divisions required to find the greatest common divisor of a and b , where $a \geq b$, is $O(\log b)$.

gcds as Linear Combinations

An important result we will use throughout the remainder of this section is that the greatest common divisor of two integers a and b can be expressed in the form

$$sa + tb,$$

where s and t are integers. In other words, $\gcd(a, b)$ can be expressed as a **linear combination** with integer coefficients of a and b . For example, $\gcd(6, 14) = 2$, and $2 = (-2) \cdot 6 + 1 \cdot 14$. We state this fact as Theorem 6.

THEOREM 6

BÉZOUT'S THEOREM If a and b are positive integers, then there exist integers s and t such that $\gcd(a, b) = sa + tb$.



ÉTIENNE BÉZOUT (1730–1783) Bézout was born in Nemours, France, where his father was a magistrate. Reading the writings of the great mathematician Leonhard Euler enticed him to become a mathematician. In 1758 he was appointed to a position at the Académie des Sciences in Paris; in 1763 he was appointed examiner of the Gardes de la Marine, where he was assigned the task of writing mathematics textbooks. This assignment led to a four-volume textbook completed in 1767. Bézout is well known for his six-volume comprehensive textbook on mathematics. His textbooks were extremely popular and were studied by many generations of students hoping to enter the École Polytechnique, the famous engineering and science school. His books were translated into English and used in North America, including at Harvard.

His most important original work was published in 1779 in the book *Théorie générale des équations algébriques*, where he introduced important methods for solving simultaneous polynomial equations in many unknowns. The most well-known result in this book is now called *Bézout's theorem*, which in its general form tells us that the number of common points on two plane algebraic curves equals the product of the degrees of these curves. Bézout is also credited with inventing the determinant (which was called the Bézoutian by the great English mathematician James Joseph Sylvester). He was considered to be a kind person with a warm heart, although he had a reserved and somber personality. He was happily married and a father.

DEFINITION 6

If a and b are positive integers, then integers s and t such that $\gcd(a, b) = sa + tb$ are called *Bézout coefficients* of a and b (after Étienne Bézout, a French mathematician of the eighteenth century). Also, the equation $\gcd(a, b) = sa + tb$ is called *Bézout's identity*.

We will not give a formal proof of Theorem 6 here (see Exercise 36 in Section 5.2 and [Ro10] for proofs). We will provide an example of a general method that can be used to find a linear combination of two integers equal to their greatest common divisor. (In this section, we will assume that a linear combination has integer coefficients.) The method proceeds by working backward through the divisions of the Euclidean algorithm, so this method requires a forward pass and a backward pass through the steps of the Euclidean algorithm. (In the exercises we will describe an algorithm called the **extended Euclidean algorithm**, which can be used to express $\gcd(a, b)$ as a linear combination of a and b using a single pass through the steps of the Euclidean algorithm; see the preamble to Exercise 41.)

EXAMPLE 17 Express $\gcd(252, 198) = 18$ as a linear combination of 252 and 198.

Solution: To show that $\gcd(252, 198) = 18$, the Euclidean algorithm uses these divisions:

$$\begin{aligned} 252 &= 1 \cdot 198 + 54 \\ 198 &= 3 \cdot 54 + 36 \\ 54 &= 1 \cdot 36 + 18 \\ 36 &= 2 \cdot 18. \end{aligned}$$

Using the next-to-last division (the third division), we can express $\gcd(252, 198) = 18$ as a linear combination of 54 and 36. We find that

$$18 = 54 - 1 \cdot 36.$$

The second division tells us that

$$36 = 198 - 3 \cdot 54.$$

Substituting this expression for 36 into the previous equation, we can express 18 as a linear combination of 54 and 198. We have

$$18 = 54 - 1 \cdot 36 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198.$$

The first division tells us that

$$54 = 252 - 1 \cdot 198.$$

Substituting this expression for 54 into the previous equation, we can express 18 as a linear combination of 252 and 198. We conclude that

$$18 = 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198,$$

completing the solution. 

We will use Theorem 6 to develop several useful results. One of our goals will be to prove the part of the fundamental theorem of arithmetic asserting that a positive integer has at most one prime factorization. We will show that if a positive integer has a factorization into primes, where the primes are written in nondecreasing order, then this factorization is unique.

First, we need to develop some results about divisibility.

LEMMA 2

If a , b , and c are positive integers such that $\gcd(a, b) = 1$ and $a \mid bc$, then $a \mid c$.

Proof: Because $\gcd(a, b) = 1$, by Bézout's theorem there are integers s and t such that

$$sa + tb = 1.$$

Multiplying both sides of this equation by c , we obtain

$$sac + tbc = c.$$

We can now use Theorem 1 of Section 4.1 to show that $a \mid c$. By part (ii) of that theorem, $a \mid tbc$. Because $a \mid sac$ and $a \mid tbc$, by part (i) of that theorem, we conclude that a divides $sac + tbc$. Because $sac + tbc = c$, we conclude that $a \mid c$, completing the proof. \triangle

We will use the following generalization of Lemma 2 in the proof of uniqueness of prime factorizations. (The proof of Lemma 3 is left as Exercise 64 in Section 5.1, because it can be most easily carried out using the method of mathematical induction, covered in that section.)

LEMMA 3

If p is a prime and $p \mid a_1a_2 \cdots a_n$, where each a_i is an integer, then $p \mid a_i$ for some i .

We can now show that a factorization of an integer into primes is unique. That is, we will show that every integer can be written as the product of primes in nondecreasing order in at most one way. This is part of the fundamental theorem of arithmetic. We will prove the other part, that every integer has a factorization into primes, in Section 5.2.

Proof (of the uniqueness of the prime factorization of a positive integer): We will use a proof by contradiction. Suppose that the positive integer n can be written as the product of primes in two different ways, say, $n = p_1p_2 \cdots p_s$ and $n = q_1q_2 \cdots q_t$, each p_i and q_j are primes such that $p_1 \leq p_2 \leq \cdots \leq p_s$ and $q_1 \leq q_2 \leq \cdots \leq q_t$.

When we remove all common primes from the two factorizations, we have

$$p_{i_1}p_{i_2} \cdots p_{i_u} = q_{j_1}q_{j_2} \cdots q_{j_v},$$

where no prime occurs on both sides of this equation and u and v are positive integers. By Lemma 3 it follows that p_{i_1} divides q_{j_k} for some k . Because no prime divides another prime, this is impossible. Consequently, there can be at most one factorization of n into primes in nondecreasing order. \triangle

Lemma 2 can also be used to prove a result about dividing both sides of a congruence by the same integer. We have shown (Theorem 5 in Section 4.1) that we can multiply both sides of a congruence by the same integer. However, dividing both sides of a congruence by an integer does not always produce a valid congruence, as Example 18 shows.

EXAMPLE 18

The congruence $14 \equiv 8 \pmod{6}$ holds, but both sides of this congruence cannot be divided by 2 to produce a valid congruence because $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \pmod{6}$. \triangle

Although we cannot divide both sides of a congruence by any integer to produce a valid congruence, we can if this integer is relatively prime to the modulus. Theorem 7 establishes this important fact. We use Lemma 2 in the proof.

THEOREM 7

Let m be a positive integer and let a , b , and c be integers. If $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.

Proof: Because $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c(a - b)$. By Lemma 2, because $\gcd(c, m) = 1$, it follows that $m \mid a - b$. We conclude that $a \equiv b \pmod{m}$. \triangleleft

Exercises

c) $17, 17^{17}$
e) 0, 5

d) $2^2 \cdot 7, 5^3 \cdot 13$
f) $2 \cdot 3 \cdot 5 \cdot 7, 2 \cdot 3 \cdot 5 \cdot 7$

25. What are the greatest common divisors of these pairs of integers?

a) $3^7 \cdot 5^3 \cdot 7^3, 2^{11} \cdot 3^5 \cdot 5^9$
b) $11 \cdot 13 \cdot 17, 2^9 \cdot 3^7 \cdot 5^5 \cdot 7^3$
c) $23^{31}, 23^{17}$
d) $41 \cdot 43 \cdot 53, 41 \cdot 43 \cdot 53$
e) $3^{13} \cdot 5^{17}, 2^{12} \cdot 7^{21}$
f) 1111, 0

26. What is the least common multiple of each pair in Exercise 24?

27. What is the least common multiple of each pair in Exercise 25?

28. Find $\gcd(1000, 625)$ and $\text{lcm}(1000, 625)$ and verify that $\gcd(1000, 625) \cdot \text{lcm}(1000, 625) = 1000 \cdot 625$.

29. Find $\gcd(92928, 123552)$ and $\text{lcm}(92928, 123552)$, and verify that $\gcd(92928, 123552) \cdot \text{lcm}(92928, 123552) = 92928 \cdot 123552$. [Hint: First find the prime factorizations of 92928 and 123552.]

30. If the product of two integers is $2^7 3^8 5^{27} 11$ and their greatest common divisor is $2^3 3^4 5$, what is their least common multiple?

31. Show that if a and b are positive integers, then $ab = \gcd(a, b) \cdot \text{lcm}(a, b)$. [Hint: Use the prime factorizations of a and b and the formulae for $\gcd(a, b)$ and $\text{lcm}(a, b)$ in terms of these factorizations.]

32. Use the Euclidean algorithm to find

a) $\gcd(1, 5)$.
c) $\gcd(123, 277)$.
e) $\gcd(1529, 14038)$.
b) $\gcd(100, 101)$.
d) $\gcd(1529, 14039)$.
f) $\gcd(11111, 111111)$.

33. Use the Euclidean algorithm to find

a) $\gcd(12, 18)$.
c) $\gcd(1001, 1331)$.
e) $\gcd(1000, 5040)$.
b) $\gcd(111, 201)$.
d) $\gcd(12345, 54321)$.
f) $\gcd(9888, 6060)$.

34. How many divisions are required to find $\gcd(21, 34)$ using the Euclidean algorithm?

35. How many divisions are required to find $\gcd(34, 55)$ using the Euclidean algorithm?

- *36. Show that if a and b are both positive integers, then $(2^a - 1) \bmod (2^b - 1) = 2^a \bmod b - 1$.

- *37. Use Exercise 36 to show that if a and b are positive integers, then $\gcd(2^a - 1, 2^b - 1) = 2^{\gcd(a, b)} - 1$. [Hint: Show that the remainders obtained when the Euclidean algorithm is used to compute $\gcd(2^a - 1, 2^b - 1)$ are of the form $2^r - 1$, where r is a remainder arising when the Euclidean algorithm is used to find $\gcd(a, b)$.]

38. Use Exercise 37 to show that the integers $2^{35} - 1, 2^{34} - 1, 2^{33} - 1, 2^{31} - 1, 2^{29} - 1$, and $2^{23} - 1$ are pairwise relatively prime.

39. Using the method followed in Example 17, express the greatest common divisor of each of these pairs of integers as a linear combination of these integers.

a) 10, 11
d) 34, 55
g) 123, 2347
b) 21, 44
e) 117, 213
h) 3454, 4666
c) 36, 48
f) 0, 223
i) 9999, 11111

40. Using the method followed in Example 17, express the greatest common divisor of each of these pairs of integers as a linear combination of these integers.

a) 9, 11
d) 21, 55
g) 2002, 2339
b) 33, 44
e) 101, 203
h) 3457, 4669
c) 35, 78
f) 124, 323
i) 10001, 13422

The **extended Euclidean algorithm** can be used to express $\gcd(a, b)$ as a linear combination with integer coefficients of the integers a and b . We set $s_0 = 1, s_1 = 0, t_0 = 0$, and $t_1 = 1$ and let $s_j = s_{j-2} - q_{j-1}s_{j-1}$ and $t_j = t_{j-2} - q_{j-1}t_{j-1}$ for $j = 2, 3, \dots, n$, where the q_j are the quotients in the divisions used when the Euclidean algorithm finds $\gcd(a, b)$, as shown in the text. It can be shown (see [Ro10]) that $\gcd(a, b) = s_n a + t_n b$. The main advantage of the extended Euclidean algorithm is that it uses one pass through the steps of the Euclidean algorithm to find Bézout coefficients of a and b , unlike the method in the text which uses two passes.

41. Use the extended Euclidean algorithm to express $\gcd(26, 91)$ as a linear combination of 26 and 91.
42. Use the extended Euclidean algorithm to express $\gcd(252, 356)$ as a linear combination of 252 and 356.
43. Use the extended Euclidean algorithm to express $\gcd(144, 89)$ as a linear combination of 144 and 89.
44. Use the extended Euclidean algorithm to express $\gcd(1001, 100001)$ as a linear combination of 1001 and 100001.

45. Describe the extended Euclidean algorithm using pseudocode.

46. Find the smallest positive integer with exactly n different positive factors when n is
a) 3.
b) 4.
c) 5.
d) 6.
e) 10.

47. Can you find a formula or rule for the n th term of a sequence related to the prime numbers or prime factorizations so that the initial terms of the sequence have these values?

a) 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, ...
b) 1, 2, 3, 2, 5, 2, 7, 2, 3, 2, 11, 2, 13, 2, ...
c) 1, 2, 2, 3, 2, 4, 2, 4, 3, 4, 2, 6, 2, 4, ...
d) 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, ...
e) 1, 2, 3, 3, 5, 5, 7, 7, 7, 7, 11, 11, 13, 13, ...
f) 1, 2, 6, 30, 210, 2310, 30030, 510510, 9699690, 223092870, ...

48. Can you find a formula or rule for the n th term of a sequence related to the prime numbers or prime factorizations so that the initial terms of the sequence have these values?

a) 2, 2, 3, 5, 7, 7, 11, 11, 11, 13, 13, ...
b) 0, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 6, 6, ...
c) 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, ...
d) 1, -1, -1, 0, -1, 1, -1, 0, 0, 1, -1, 0, -1, 1, 1, ...
e) 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, ...
f) 4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, ...
49. Prove that the product of any three consecutive integers is divisible by 6.

- 50.** Show that if a, b , and m are integers such that $m \geq 2$ and $a \equiv b \pmod{m}$, then $\gcd(a, m) = \gcd(b, m)$.
- *51.** Prove or disprove that $n^2 - 79n + 1601$ is prime whenever n is a positive integer.
- 52.** Prove or disprove that $p_1 p_2 \cdots p_n + 1$ is prime for every positive integer n , where p_1, p_2, \dots, p_n are the n smallest prime numbers.
- 53.** Show that there is a composite integer in every arithmetic progression $ak + b$, $k = 1, 2, \dots$ where a and b are positive integers.
- 54.** Adapt the proof in the text that there are infinitely many primes to prove that there are infinitely many primes of the form $3k + 2$, where k is a nonnegative integer. [Hint: Suppose that there are only finitely many such primes q_1, q_2, \dots, q_n , and consider the number $3q_1 q_2 \cdots q_n - 1$.]
- 55.** Adapt the proof in the text that there are infinitely many primes to prove that there are infinitely many primes of the form $4k + 3$, where k is a nonnegative integer. [Hint: Suppose that there are only finitely many such primes q_1, q_2, \dots, q_n , and consider the number $4q_1 q_2 \cdots q_n - 1$.]
- *56.** Prove that the set of positive rational numbers is countable by setting up a function that assigns to a rational number p/q with $\gcd(p, q) = 1$ the base 11 number formed by the decimal representation of p followed by the base 11 digit A, which corresponds to the decimal number 10, followed by the decimal representation of q .
- *57.** Prove that the set of positive rational numbers is countable by showing that the function K is a one-to-one correspondence between the set of positive rational numbers and the set of positive integers if $K(m/n) = p_1^{2a_1} p_2^{2a_2} \cdots p_s^{2a_s} q_1^{2b_1-1} q_2^{2b_2-1} \cdots q_t^{2b_t-1}$, where $\gcd(m, n) = 1$ and the prime-power factorizations of m and n are $m = p_1^{a_1} p_2^{a_2} \cdots p_s^{a_s}$ and $n = q_1^{b_1} q_2^{b_2} \cdots q_t^{b_t}$.

4.4 Solving Congruences

Introduction

Solving linear congruences, which have the form $ax \equiv b \pmod{m}$, is an essential task in the study of number theory and its applications, just as solving linear equations plays an important role in calculus and linear algebra. To solve linear congruences, we employ inverses modulo m . We explain how to work backwards through the steps of the Euclidean algorithm to find inverses modulo m . Once we have found an inverse of a modulo m , we solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides of the congruence by this inverse.

Simultaneous systems of linear congruence have been studied since ancient times. For example, the Chinese mathematician Sun-Tsu studied them in the first century. We will show how to solve systems of linear congruences modulo pairwise relatively prime moduli. The result we will prove is called the Chinese remainder theorem, and our proof will give a method to find all solutions of such systems of congruences. We will also show how to use the Chinese remainder theorem as a basis for performing arithmetic with large integers.

We will introduce a useful result of Fermat, known as Fermat's little theorem, which states that if p is prime and p does not divide a , then $a^{p-1} \equiv 1 \pmod{p}$. We will examine the converse of this statement, which will lead us to the concept of a pseudoprime. A pseudoprime m to the base a is a composite integer m that masquerades as a prime by satisfying the congruence $a^{m-1} \equiv 1 \pmod{m}$. We will also give an example of a Carmichael number, which is a composite integer that is a pseudoprime to all bases a relatively prime to it.

We also introduce the notion of discrete logarithms, which are analogous to ordinary logarithms. To define discrete logarithms we must first define primitive roots. A primitive root of a prime p is an integer r such that every integer not divisible by p is congruent to a power of r modulo p . If r is a primitive root of p and $r^e \equiv a \pmod{p}$, then e is the discrete logarithm of a modulo p to the base r . Finding discrete logarithms turns out to be an extremely difficult problem in general. The difficulty of this problem is the basis for the security of many cryptographic systems.

Linear Congruences

A congruence of the form

$$ax \equiv b \pmod{m},$$

where m is a positive integer, a and b are integers, and x is a variable, is called a **linear congruence**. Such congruences arise throughout number theory and its applications.

How can we solve the linear congruence $ax \equiv b \pmod{m}$, that is, how can we find all integers x that satisfy this congruence? One method that we will describe uses an integer \bar{a} such that $\bar{a}a \equiv 1 \pmod{m}$, if such an integer exists. Such an integer \bar{a} is said to be an **inverse** of a modulo m . Theorem 1 guarantees that an inverse of a modulo m exists whenever a and m are relatively prime.

THEOREM 1

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m . (That is, there is a unique positive integer \bar{a} less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to \bar{a} modulo m .)

Proof: By Theorem 6 of Section 4.3, because $\gcd(a, m) = 1$, there are integers s and t such that

$$sa + tm = 1.$$

This implies that

$$sa + tm \equiv 1 \pmod{m}.$$

Because $tm \equiv 0 \pmod{m}$, it follows that

$$sa \equiv 1 \pmod{m}.$$

Consequently, s is an inverse of a modulo m . That this inverse is unique modulo m is left as Exercise 7. □

Using inspection to find an inverse of a modulo m is easy when m is small. To find this inverse, we look for a multiple of a that exceeds a multiple of m by 1. For example, to find an inverse of 3 modulo 7, we can find $j \cdot 3$ for $j = 1, 2, \dots, 6$, stopping when we find a multiple of 3 that is one more than a multiple of 7. We can speed this approach up if we note that $2 \cdot 3 \equiv -1 \pmod{7}$. This means that $(-2) \cdot 3 \equiv 1 \pmod{7}$. Hence, $5 \cdot 3 \equiv 1 \pmod{7}$, so 5 is an inverse of 3 modulo 7.

We can design a more efficient algorithm than brute force to find an inverse of a modulo m when $\gcd(a, m) = 1$ using the steps of the Euclidean algorithm. By reversing these steps as in Example 17 of Section 4.3, we can find a linear combination $sa + tm = 1$ where s and t are integers. Reducing both sides of this equation modulo m tells us that s is an inverse of a modulo m . We illustrate this procedure in Example 1.

EXAMPLE 1 Find an inverse of 3 modulo 7 by first finding Bézout coefficients of 3 and 7. (Note that we have already shown that 5 is an inverse of 3 modulo 7 by inspection.)

Solution: Because $\gcd(3, 7) = 1$, Theorem 1 tells us that an inverse of 3 modulo 7 exists. The Euclidean algorithm ends quickly when used to find the greatest common divisor of 3 and 7:

$$7 = 2 \cdot 3 + 1.$$

From this equation we see that

$$-2 \cdot 3 + 1 \cdot 7 = 1.$$

This shows that -2 and 1 are Bézout coefficients of 3 and 7. We see that -2 is an inverse of 3 modulo 7. Note that every integer congruent to -2 modulo 7 is also an inverse of 3, such as 5, -9 , 12, and so on. 

EXAMPLE 2 Find an inverse of 101 modulo 4620.

Solution: For completeness, we present all steps used to compute an inverse of 101 modulo 4620. (Only the last step goes beyond methods developed in Section 4.3 and illustrated in Example 17 in that section.) First, we use the Euclidean algorithm to show that $\gcd(101, 4620) = 1$. Then we will reverse the steps to find Bézout coefficients a and b such that $101a + 4620b = 1$. It will then follow that a is an inverse of 101 modulo 4620. The steps used by the Euclidean algorithm to find $\gcd(101, 4620)$ are

$$\begin{aligned} 4620 &= 45 \cdot 101 + 75 \\ 101 &= 1 \cdot 75 + 26 \\ 75 &= 2 \cdot 26 + 23 \\ 26 &= 1 \cdot 23 + 3 \\ 23 &= 7 \cdot 3 + 2 \\ 3 &= 1 \cdot 2 + 1 \\ 2 &= 2 \cdot 1. \end{aligned}$$

Because the last nonzero remainder is 1, we know that $\gcd(101, 4620) = 1$. We can now find the Bézout coefficients for 101 and 4620 by working backwards through these steps, expressing $\gcd(101, 4620) = 1$ in terms of each successive pair of remainders. In each step we eliminate the remainder by expressing it as a linear combination of the divisor and the dividend. We obtain

$$\begin{aligned} 1 &= 3 - 1 \cdot 2 \\ &= 3 - 1 \cdot (23 - 7 \cdot 3) = -1 \cdot 23 + 8 \cdot 3 \\ &= -1 \cdot 23 + 8 \cdot (26 - 1 \cdot 23) = 8 \cdot 26 - 9 \cdot 23 \\ &= 8 \cdot 26 - 9 \cdot (75 - 2 \cdot 26) = -9 \cdot 75 + 26 \cdot 26 \\ &= -9 \cdot 75 + 26 \cdot (101 - 1 \cdot 75) = 26 \cdot 101 - 35 \cdot 75 \\ &= 26 \cdot 101 - 35 \cdot (4620 - 45 \cdot 101) = -35 \cdot 4620 + 1601 \cdot 101. \end{aligned}$$

That $-35 \cdot 4620 + 1601 \cdot 101 = 1$ tells us that -35 and 1601 are Bézout coefficients of 4620 and 101, and 1601 is an inverse of 101 modulo 4620. 

Once we have an inverse \bar{a} of a modulo m , we can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides of the linear congruence by \bar{a} , as Example 3 illustrates.

EXAMPLE 3 What are the solutions of the linear congruence $3x \equiv 4 \pmod{7}$?

Solution: By Example 1 we know that -2 is an inverse of 3 modulo 7 . Multiplying both sides of the congruence by -2 shows that

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}.$$

Because $-6 \equiv 1 \pmod{7}$ and $-8 \equiv 6 \pmod{7}$, it follows that if x is a solution, then $x \equiv -8 \equiv 6 \pmod{7}$.

We need to determine whether every x with $x \equiv 6 \pmod{7}$ is a solution. Assume that $x \equiv 6 \pmod{7}$. Then, by Theorem 5 of Section 4.1, it follows that

$$3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7},$$

which shows that all such x satisfy the congruence. We conclude that the solutions to the congruence are the integers x such that $x \equiv 6 \pmod{7}$, namely, $6, 13, 20, \dots$ and $-1, -8, -15, \dots$ 

The Chinese Remainder Theorem



Systems of linear congruences arise in many contexts. For example, as we will see later, they are the basis for a method that can be used to perform arithmetic with large integers. Such systems can even be found as word puzzles in the writings of ancient Chinese and Hindu mathematicians, such as that given in Example 4.

EXAMPLE 4 In the first century, the Chinese mathematician Sun-Tsu asked:

There are certain things whose number is unknown. When divided by 3, the remainder is 2; when divided by 5, the remainder is 3; and when divided by 7, the remainder is 2. What will be the number of things?

This puzzle can be translated into the following question: What are the solutions of the systems of congruences

$$\begin{aligned} x &\equiv 2 \pmod{3}, \\ x &\equiv 3 \pmod{5}, \\ x &\equiv 2 \pmod{7}? \end{aligned}$$

We will solve this system, and with it Sun-Tsu's puzzle, later in this section. 

The *Chinese remainder theorem*, named after the Chinese heritage of problems involving systems of linear congruences, states that when the moduli of a system of linear congruences are pairwise relatively prime, there is a unique solution of the system modulo the product of the moduli.

THEOREM 2

THE CHINESE REMAINDER THEOREM Let m_1, m_2, \dots, m_n be pairwise relatively prime positive integers greater than one and a_1, a_2, \dots, a_n arbitrary integers. Then the system

$$x \equiv a_1 \pmod{m_1},$$

$$x \equiv a_2 \pmod{m_2},$$

.

.

$$x \equiv a_n \pmod{m_n}$$

has a unique solution modulo $m = m_1 m_2 \cdots m_n$. (That is, there is a solution x with $0 \leq x < m$, and all other solutions are congruent modulo m to this solution.)

Proof: To establish this theorem, we need to show that a solution exists and that it is unique modulo m . We will show that a solution exists by describing a way to construct this solution; showing that the solution is unique modulo m is Exercise 30.

To construct a simultaneous solution, first let

$$M_k = m/m_k$$

for $k = 1, 2, \dots, n$. That is, M_k is the product of the moduli except for m_k . Because m_i and m_k have no common factors greater than 1 when $i \neq k$, it follows that $\gcd(m_k, M_k) = 1$. Consequently, by Theorem 1, we know that there is an integer y_k , an inverse of M_k modulo m_k , such that

$$M_k y_k \equiv 1 \pmod{m_k}.$$

To construct a simultaneous solution, form the sum

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n.$$

We will now show that x is a simultaneous solution. First, note that because $M_j \equiv 0 \pmod{m_k}$ whenever $j \neq k$, all terms except the k th term in this sum are congruent to 0 modulo m_k . Because $M_k y_k \equiv 1 \pmod{m_k}$ we see that

$$x \equiv a_k M_k y_k \equiv a_k \pmod{m_k},$$

for $k = 1, 2, \dots, n$. We have shown that x is a simultaneous solution to the n congruences. \triangleleft

Example 5 illustrates how to use the construction given in our proof of the Chinese remainder theorem to solve a system of congruences. We will solve the system given in Example 4, arising in Sun-Tsu's puzzle.

EXAMPLE 5

To solve the system of congruences in Example 4, first let $m = 3 \cdot 5 \cdot 7 = 105$, $M_1 = m/3 = 35$, $M_2 = m/5 = 21$, and $M_3 = m/7 = 15$. We see that 2 is an inverse of $M_1 = 35$ modulo 3, because $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod{3}$; 1 is an inverse of $M_2 = 21$ modulo 5, because $21 \equiv 1 \pmod{5}$; and 1 is an inverse of $M_3 = 15$ (mod 7), because $15 \equiv 1 \pmod{7}$. The solutions to this system are those x such that

$$\begin{aligned} x &\equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \\ &= 233 \equiv 23 \pmod{105}. \end{aligned}$$

It follows that 23 is the smallest positive integer that is a simultaneous solution. We conclude that 23 is the smallest positive integer that leaves a remainder of 2 when divided by 3, a remainder of 3 when divided by 5, and a remainder of 2 when divided by 7. 

Although the construction in Theorem 2 provides a general method for solving systems of linear congruences with pairwise relatively prime moduli, it can be easier to solve a system using a different method. Example 6 illustrates the use of a method known as **back substitution**.

EXAMPLE 6 Use the method of back substitution to find all integers x such that $x \equiv 1 \pmod{5}$, $x \equiv 2 \pmod{6}$, and $x \equiv 3 \pmod{7}$.

Solution: By Theorem 4 in Section 4.1, the first congruence can be rewritten as an equality, $x = 5t + 1$ where t is an integer. Substituting this expression for x into the second congruence tells us that

$$5t + 1 \equiv 2 \pmod{6},$$

which can be easily solved to show that $t \equiv 5 \pmod{6}$ (as the reader should verify). Using Theorem 4 in Section 4.1 again, we see that $t = 6u + 5$ where u is an integer. Substituting this expression for t back into the equation $x = 5t + 1$ tells us that $x = 5(6u + 5) + 1 = 30u + 26$. We insert this into the third equation to obtain

$$30u + 26 \equiv 3 \pmod{7}.$$

Solving this congruence tells us that $u \equiv 6 \pmod{7}$ (as the reader should verify). Hence, Theorem 4 in Section 4.1 tells us that $u = 7v + 6$ where v is an integer. Substituting this expression for u into the equation $x = 30u + 26$ tells us that $x = 30(7v + 6) + 26 = 210v + 206$. Translating this back into a congruence, we find the solution to the simultaneous congruences,

$$x \equiv 206 \pmod{210}. \quad \blacktriangleleft$$

Computer Arithmetic with Large Integers

Suppose that m_1, m_2, \dots, m_n are pairwise relatively prime moduli and let m be their product. By the Chinese remainder theorem, we can show (see Exercise 28) that an integer a with $0 \leq a < m$ can be uniquely represented by the n -tuple consisting of its remainders upon division by m_i , $i = 1, 2, \dots, n$. That is, we can uniquely represent a by

$$(a \bmod m_1, a \bmod m_2, \dots, a \bmod m_n).$$

EXAMPLE 7 What are the pairs used to represent the nonnegative integers less than 12 when they are represented by the ordered pair where the first component is the remainder of the integer upon division by 3 and the second component is the remainder of the integer upon division by 4?

Solution: We have the following representations, obtained by finding the remainder of each integer when it is divided by 3 and by 4:

$$\begin{array}{lll} 0 = (0, 0) & 4 = (1, 0) & 8 = (2, 0) \\ 1 = (1, 1) & 5 = (2, 1) & 9 = (0, 1) \\ 2 = (2, 2) & 6 = (0, 2) & 10 = (1, 2) \\ 3 = (0, 3) & 7 = (1, 3) & 11 = (2, 3). \end{array} \quad \blacktriangleleft$$

To perform arithmetic with large integers, we select moduli m_1, m_2, \dots, m_n , where each m_i is an integer greater than 2, $\gcd(m_i, m_j) = 1$ whenever $i \neq j$, and $m = m_1 m_2 \cdots m_n$ is greater than the results of the arithmetic operations we want to carry out.

Once we have selected our moduli, we carry out arithmetic operations with large integers by performing componentwise operations on the n -tuples representing these integers using their remainders upon division by m_i , $i = 1, 2, \dots, n$. Once we have computed the value of each component in the result, we recover its value by solving a system of n congruences modulo m_i , $i = 1, 2, \dots, n$. This method of performing arithmetic with large integers has several valuable features. First, it can be used to perform arithmetic with integers larger than can ordinarily be carried out on a computer. Second, computations with respect to the different moduli can be done in parallel, speeding up the arithmetic.

EXAMPLE 8

Suppose that performing arithmetic with integers less than 100 on a certain processor is much quicker than doing arithmetic with larger integers. We can restrict almost all our computations to integers less than 100 if we represent integers using their remainders modulo pairwise relatively prime integers less than 100. For example, we can use the moduli of 99, 98, 97, and 95. (These integers are relatively prime pairwise, because no two have a common factor greater than 1.)

By the Chinese remainder theorem, every nonnegative integer less than $99 \cdot 98 \cdot 97 \cdot 95 = 89,403,930$ can be represented uniquely by its remainders when divided by these four moduli. For example, we represent 123,684 as $(33, 8, 9, 89)$, because $123,684 \pmod{99} = 33$; $123,684 \pmod{98} = 8$; $123,684 \pmod{97} = 9$; and $123,684 \pmod{95} = 89$. Similarly, we represent 413,456 as $(32, 92, 42, 16)$.

To find the sum of 123,684 and 413,456, we work with these 4-tuples instead of these two integers directly. We add the 4-tuples componentwise and reduce each component with respect to the appropriate modulus. This yields

$$\begin{aligned} (33, 8, 9, 89) + (32, 92, 42, 16) \\ = (65 \pmod{99}, 100 \pmod{98}, 51 \pmod{97}, 105 \pmod{95}) \\ = (65, 2, 51, 10). \end{aligned}$$

To find the sum, that is, the integer represented by $(65, 2, 51, 10)$, we need to solve the system of congruences

$$\begin{aligned} x &\equiv 65 \pmod{99}, \\ x &\equiv 2 \pmod{98}, \\ x &\equiv 51 \pmod{97}, \\ x &\equiv 10 \pmod{95}. \end{aligned}$$

It can be shown (see Exercise 53) that 537,140 is the unique nonnegative solution of this system less than 89,403,930. Consequently, 537,140 is the sum. Note that it is only when we have to recover the integer represented by $(65, 2, 51, 10)$ that we have to do arithmetic with integers larger than 100. 

Particularly good choices for moduli for arithmetic with large integers are sets of integers of the form $2^k - 1$, where k is a positive integer, because it is easy to do binary arithmetic modulo such integers, and because it is easy to find sets of such integers that are pairwise relatively prime. [The second reason is a consequence of the fact that $\gcd(2^a - 1, 2^b - 1) = 2^{\gcd(a, b)} - 1$, as Exercise 37 in Section 4.3 shows.] Suppose, for instance, that we can do arithmetic with integers less than 2^{35} easily on our computer, but that working with larger integers requires special procedures. We can use pairwise relatively prime moduli less than 2^{35} to perform arithmetic with integers as large as their product. For example, as Exercise 38 in Section 4.3 shows, the integers $2^{35} - 1, 2^{34} - 1, 2^{33} - 1, 2^{31} - 1, 2^{29} - 1$, and $2^{23} - 1$ are pairwise relatively prime. Because the product of these six moduli exceeds 2^{184} , we can perform arithmetic with integers as large as 2^{184} (as long as the results do not exceed this number) by doing arithmetic modulo each of these six moduli, none of which exceeds 2^{35} .

Fermat's Little Theorem

The great French mathematician Pierre de Fermat made many important discoveries in number theory. One of the most useful of these states that p divides $a^{p-1} - 1$ whenever p is prime and a is an integer not divisible by p . Fermat announced this result in a letter to one of his correspondents. However, he did not include a proof in the letter, stating that he feared the proof would be too long. Although Fermat never published a proof of this fact, there is little doubt that he knew how to prove it, unlike the result known as Fermat's last theorem. The first published proof is credited to Leonhard Euler. We now state this theorem in terms of congruences.

THEOREM 3

FERMAT'S LITTLE THEOREM If p is prime and a is an integer not divisible by p , then

$$a^{p-1} \equiv 1 \pmod{p}.$$

Furthermore, for every integer a we have

$$a^p \equiv a \pmod{p}.$$

Remark: Fermat's little theorem tells us that if $a \in \mathbf{Z}_p$, then $a^{p-1} = 1$ in \mathbf{Z}_p .

The proof of Theorem 3 is outlined in Exercise 19.

Fermat's little theorem is extremely useful in computing the remainders modulo p of large powers of integers, as Example 9 illustrates.

EXAMPLE 9

Find $7^{222} \pmod{11}$.

Solution: We can use Fermat's little theorem to evaluate $7^{222} \pmod{11}$ rather than using the fast modular exponentiation algorithm. By Fermat's little theorem we know that $7^{10} \equiv 1 \pmod{11}$, so $(7^{10})^k \equiv 1 \pmod{11}$ for every positive integer k . To take advantage of this last congruence, we divide the exponent 222 by 10, finding that $222 = 22 \cdot 10 + 2$. We now see that

$$7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} 7^2 \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}.$$

It follows that $7^{222} \pmod{11} = 5$. 

Example 9 illustrated how we can use Fermat's little theorem to compute $a^n \pmod{p}$, where p is prime and $p \nmid a$. First, we use the division algorithm to find the quotient q and remainder r when n is divided by $p-1$, so that $n = q(p-1) + r$ where $0 \leq r < p-1$. It follows that $a^n = a^{q(p-1)+r} = (a^{p-1})^q a^r \equiv 1^q a^r \equiv a^r \pmod{p}$. Hence, to find $a^n \pmod{p}$, we only need to compute $a^r \pmod{p}$. We will take advantage of this simplification many times in our study of number theory.

Pseudoprimes

In Section 4.2 we showed that an integer n is prime when it is not divisible by any prime p with $p \leq \sqrt{n}$. Unfortunately, using this criterion to show that a given integer is prime is inefficient. It requires that we find all primes not exceeding \sqrt{n} and that we carry out trial division by each such prime to see whether it divides n .

Are there more efficient ways to determine whether an integer is prime? According to some sources, ancient Chinese mathematicians believed that n was an odd prime if and only if

$$2^{n-1} \equiv 1 \pmod{n}.$$

If this were true, it would provide an efficient primality test. Why did they believe this congruence could be used to determine whether an integer $n > 2$ is prime? First, they observed that the congruence holds whenever n is an odd prime. For example, 5 is prime and

$$2^{5-1} = 2^4 = 16 \equiv 1 \pmod{5}.$$

By Fermat's little theorem, we know that this observation was correct, that is, $2^{n-1} \equiv 1 \pmod{n}$ whenever n is an odd prime. Second, they never found a composite integer n for which the congruence holds. However, the ancient Chinese were only partially correct. They were correct in thinking that the congruence holds whenever n is prime, but they were incorrect in concluding that n is necessarily prime if the congruence holds.

Unfortunately, there are composite integers n such that $2^{n-1} \equiv 1 \pmod{n}$. Such integers are called **pseudoprimes** to the base 2.

EXAMPLE 10 The integer 341 is a pseudoprime to the base 2 because it is composite ($341 = 11 \cdot 31$) and as Exercise 37 shows

$$2^{340} \equiv 1 \pmod{341}.$$

We can use an integer other than 2 as the base when we study pseudoprimes.

DEFINITION 1

Let b be a positive integer. If n is a composite positive integer, and $b^{n-1} \equiv 1 \pmod{n}$, then n is called a *pseudoprime to the base b* .

Given a positive integer n , determining whether $2^{n-1} \equiv 1 \pmod{n}$ is a useful test that provides some evidence concerning whether n is prime. In particular, if n satisfies this congruence, then it is either prime or a pseudoprime to the base 2; if n does not satisfy this congruence, it is composite. We can perform similar tests using bases b other than 2 and obtain more evidence as to whether n is prime. If n passes all such tests, it is either prime or a pseudoprime to all the bases b we have chosen. Furthermore, among the positive integers not exceeding x , where x is a positive real number, compared to primes there are relatively few pseudoprimes to the base b , where b is a positive integer. For example, among the positive integers less than 10^{10} there are 455,052,512 primes, but only 14,884 pseudoprimes to the base 2. Unfortunately, we



PIERRE DE FERMAT (1601–1665) Pierre de Fermat, one of the most important mathematicians of the seventeenth century, was a lawyer by profession. He is the most famous amateur mathematician in history. Fermat published little of his mathematical discoveries. It is through his correspondence with other mathematicians that we know of his work. Fermat was one of the inventors of analytic geometry and developed some of the fundamental ideas of calculus. Fermat, along with Pascal, gave probability theory a mathematical basis. Fermat formulated what was the most famous unsolved problem in mathematics. He asserted that the equation $x^n + y^n = z^n$ has no nontrivial positive integer solutions when n is an integer greater than 2. For more than 300 years, no proof (or counterexample) was found. In his copy of the works of the ancient Greek mathematician Diophantus, Fermat wrote that he had a proof but that it would not fit in the margin. Because the first proof, found by Andrew Wiles in 1994, relies on sophisticated, modern mathematics, most people think that Fermat thought he had a proof, but that the proof was incorrect. However, he may have been tempting others to look for a proof, not being able to find one himself.

cannot distinguish between primes and pseudoprimes just by choosing sufficiently many bases, because there are composite integers n that pass all tests with bases b such that $\gcd(b, n) = 1$. This leads to Definition 2.

DEFINITION 2

A composite integer n that satisfies the congruence $b^{n-1} \equiv 1 \pmod{n}$ for all positive integers b with $\gcd(b, n) = 1$ is called a *Carmichael number*. (These numbers are named after Robert Carmichael, who studied them in the early twentieth century.)

EXAMPLE 11

The integer 561 is a Carmichael number. To see this, first note that 561 is composite because $561 = 3 \cdot 11 \cdot 17$. Next, note that if $\gcd(b, 561) = 1$, then $\gcd(b, 3) = \gcd(b, 11) = \gcd(b, 17) = 1$.

Using Fermat's little theorem we find that

$$b^2 \equiv 1 \pmod{3}, \quad b^{10} \equiv 1 \pmod{11}, \quad \text{and } b^{16} \equiv 1 \pmod{17}.$$

It follows that

$$\begin{aligned} b^{560} &= (b^2)^{280} \equiv 1 \pmod{3}, \\ b^{560} &= (b^{10})^{56} \equiv 1 \pmod{11}, \\ b^{560} &= (b^{16})^{35} \equiv 1 \pmod{17}. \end{aligned}$$

By Exercise 29, it follows that $b^{560} \equiv 1 \pmod{561}$ for all positive integers b with $\gcd(b, 561) = 1$. Hence 561 is a Carmichael number. \blacktriangleleft

Although there are infinitely many Carmichael numbers, more delicate tests, described in the exercise set, can be devised that can be used as the basis for efficient probabilistic primality tests. Such tests can be used to quickly show that it is almost certainly the case that a given integer is prime. More precisely, if an integer is not prime, then the probability that it passes a series of tests is close to 0. We will describe such a test in Chapter 7 and discuss the notions from probability theory that this test relies on. These probabilistic primality tests can be used, and are used, to find large primes extremely rapidly on computers.

Primitive Roots and Discrete Logarithms

In the set of positive real numbers, if $b > 1$, and $x = b^y$, we say that y is the logarithm of x to the base b . Here, we will show that we can also define the concept of logarithms modulo p of positive integers where p is a prime. Before we do so, we need a definition.

DEFINITION 3

A *primitive root* modulo a prime p is an integer r in \mathbb{Z}_p such that every nonzero element of \mathbb{Z}_p is a power of r .



ROBERT DANIEL CARMICHAEL (1879–1967) Robert Daniel Carmichael was born in Alabama. He received his undergraduate degree from Lineville College in 1898 and his Ph.D. in 1911 from Princeton. Carmichael held positions at Indiana University from 1911 until 1915 and at the University of Illinois from 1915 until 1947. Carmichael was an active researcher in a wide variety of areas, including number theory, real analysis, differential equations, mathematical physics, and group theory. His Ph.D. thesis, written under the direction of G. D. Birkhoff, is considered the first significant American contribution to the subject of differential equations.

EXAMPLE 12 Determine whether 2 and 3 are primitive roots modulo 11.

Solution: When we compute the powers of 2 in \mathbf{Z}_{11} , we obtain $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5, 2^5 = 10, 2^6 = 9, 2^7 = 7, 2^8 = 3, 2^9 = 6, 2^{10} = 1$. Because every element of \mathbf{Z}_{11} is a power of 2, 2 is a primitive root of 11.

When we compute the powers of 3 modulo 11, we obtain $3^1 = 3, 3^2 = 9, 3^3 = 5, 3^4 = 4, 3^5 = 1$. We note that this pattern repeats when we compute higher powers of 3. Because not all elements of \mathbf{Z}_{11} are powers of 3, we conclude that 3 is not a primitive root of 11. 

An important fact in number theory is that there is a primitive root modulo p for every prime p . We refer the reader to [Ro10] for a proof of this fact. Suppose that p is prime and r is a primitive root modulo p . If a is an integer between 1 and $p - 1$, that is, an element of \mathbf{Z}_p , we know that there is an unique exponent e such that $r^e = a$ in \mathbf{Z}_p , that is, $r^e \pmod p = a$.

DEFINITION 4

Suppose that p is a prime, r is a primitive root modulo p , and a is an integer between 1 and $p - 1$ inclusive. If $r^e \pmod p = a$ and $0 \leq e \leq p - 1$, we say that e is the *discrete logarithm* of a modulo p to the base r and we write $\log_r a = e$ (where the prime p is understood).

EXAMPLE 13 Find the discrete logarithms of 3 and 5 modulo 11 to the base 2.

Solution: When we computed the powers of 2 modulo 11 in Example 12, we found that $2^8 = 3$ and $2^4 = 5$ in \mathbf{Z}_{11} . Hence, the discrete logarithms of 3 and 5 modulo 11 to the base 2 are 8 and 4, respectively. (These are the powers of 2 that equal 3 and 5, respectively, in \mathbf{Z}_{11} .) We write $\log_2 3 = 8$ and $\log_2 5 = 4$ (where the modulus 11 is understood and not explicitly noted in the notation). 

The discrete logarithm problem is hard!

The **discrete logarithm problem** takes as input a prime p , a primitive root r modulo p , and a positive integer $a \in \mathbf{Z}_p$; its output is the discrete logarithm of a modulo p to the base r . Although this problem might seem not to be that difficult, it turns out that no polynomial time algorithm is known for solving it. The difficulty of this problem plays an important role in cryptography, as we will see in Section 4.6

Exercises

1. Show that 15 is an inverse of 7 modulo 26.
2. Show that 937 is an inverse of 13 modulo 2436.
3. By inspection (as discussed prior to Example 1), find an inverse of 4 modulo 9.
4. By inspection (as discussed prior to Example 1), find an inverse of 2 modulo 17.
5. Find an inverse of a modulo m for each of these pairs of relatively prime integers using the method followed in Example 2.
 - a) $a = 4, m = 9$
 - b) $a = 19, m = 141$
 - c) $a = 55, m = 89$
 - d) $a = 89, m = 232$
6. Find an inverse of a modulo m for each of these pairs of relatively prime integers using the method followed in Example 2.
 - a) $a = 2, m = 17$
 - b) $a = 34, m = 89$
- c) $a = 144, m = 233$
- d) $a = 200, m = 1001$
- *7. Show that if a and m are relatively prime positive integers, then the inverse of a modulo m is unique modulo m . [Hint: Assume that there are two solutions b and c of the congruence $ax \equiv 1 \pmod m$. Use Theorem 7 of Section 4.3 to show that $b \equiv c \pmod m$.]
8. Show that an inverse of a modulo m , where a is an integer and $m > 2$ is a positive integer, does not exist if $\gcd(a, m) > 1$.
9. Solve the congruence $4x \equiv 5 \pmod 9$ using the inverse of 4 modulo 9 found in part (a) of Exercise 5.
10. Solve the congruence $2x \equiv 7 \pmod {17}$ using the inverse of 2 modulo 7 found in part (a) of Exercise 6.
11. Solve each of these congruences using the modular inverses found in parts (b), (c), and (d) of Exercise 5.
 - a) $19x \equiv 4 \pmod {141}$
 - b) $55x \equiv 34 \pmod {89}$
 - c) $89x \equiv 2 \pmod {232}$

- 12.** Solve each of these congruences using the modular inverses found in parts (b), (c), and (d) of Exercise 6.
- $34x \equiv 77 \pmod{89}$
 - $144x \equiv 4 \pmod{233}$
 - $200x \equiv 13 \pmod{1001}$
- 13.** Find the solutions of the congruence $15x^2 + 19x \equiv 5 \pmod{11}$. [Hint: Show the congruence is equivalent to the congruence $15x^2 + 19x + 6 \equiv 0 \pmod{11}$. Factor the left-hand side of the congruence; show that a solution of the quadratic congruence is a solution of one of the two different linear congruences.]
- 14.** Find the solutions of the congruence $12x^2 + 25x \equiv 10 \pmod{11}$. [Hint: Show the congruence is equivalent to the congruence $12x^2 + 25x + 12 \equiv 0 \pmod{11}$. Factor the left-hand side of the congruence; show that a solution of the quadratic congruence is a solution of one of two different linear congruences.]
- *15.** Show that if m is an integer greater than 1 and $ac \equiv bc \pmod{m}$, then $a \equiv b \pmod{m/\gcd(c, m)}$.
- 16. a)** Show that the positive integers less than 11, except 1 and 10, can be split into pairs of integers such that each pair consists of integers that are inverses of each other modulo 11.
b) Use part (a) to show that $10! \equiv -1 \pmod{11}$.
- 17.** Show that if p is prime, the only solutions of $x^2 \equiv 1 \pmod{p}$ are integers x such that $x \equiv 1 \pmod{p}$ or $x \equiv -1 \pmod{p}$.
- *18. a)** Generalize the result in part (a) of Exercise 16; that is, show that if p is a prime, the positive integers less than p , except 1 and $p - 1$, can be split into $(p - 3)/2$ pairs of integers such that each pair consists of integers that are inverses of each other. [Hint: Use the result of Exercise 17.]
b) From part (a) conclude that $(p - 1)! \equiv -1 \pmod{p}$ whenever p is prime. This result is known as **Wilson's theorem**.
c) What can we conclude if n is a positive integer such that $(n - 1)! \not\equiv -1 \pmod{n}$?
- *19.** This exercise outlines a proof of Fermat's little theorem.
- Suppose that a is not divisible by the prime p . Show that no two of the integers $1 \cdot a, 2 \cdot a, \dots, (p - 1)a$ are congruent modulo p .
 - Conclude from part (a) that the product of $1, 2, \dots, p - 1$ is congruent modulo p to the product of $a, 2a, \dots, (p - 1)a$. Use this to show that
- $$(p - 1)! \equiv a^{p-1}(p - 1)! \pmod{p}.$$
- Use Theorem 7 of Section 4.3 to show from part (b) that $a^{p-1} \equiv 1 \pmod{p}$ if $p \nmid a$. [Hint: Use Lemma 3 of Section 4.3 to show that p does not divide $(p - 1)!$ and then use Theorem 7 of Section 4.3. Alternatively, use Wilson's theorem from Exercise 18(b).]
 - Use part (c) to show that $a^p \equiv a \pmod{p}$ for all integers a .
- 20.** Use the construction in the proof of the Chinese remainder theorem to find all solutions to the system of congruences $x \equiv 2 \pmod{3}$, $x \equiv 1 \pmod{4}$, and $x \equiv 3 \pmod{5}$.
- 21.** Use the construction in the proof of the Chinese remainder theorem to find all solutions to the system of congruences $x \equiv 1 \pmod{2}$, $x \equiv 2 \pmod{3}$, $x \equiv 3 \pmod{5}$, and $x \equiv 4 \pmod{11}$.
- 22.** Solve the system of congruence $x \equiv 3 \pmod{6}$ and $x \equiv 4 \pmod{7}$ using the method of back substitution.
- 23.** Solve the system of congruences in Exercise 20 using the method of back substitution.
- 24.** Solve the system of congruences in Exercise 21 using the method of back substitution.
- 25.** Write out in pseudocode an algorithm for solving a simultaneous system of linear congruences based on the construction in the proof of the Chinese remainder theorem.
- *26.** Find all solutions, if any, to the system of congruences $x \equiv 5 \pmod{6}$, $x \equiv 3 \pmod{10}$, and $x \equiv 8 \pmod{15}$.
- *27.** Find all solutions, if any, to the system of congruences $x \equiv 7 \pmod{9}$, $x \equiv 4 \pmod{12}$, and $x \equiv 16 \pmod{21}$.
- 28.** Use the Chinese remainder theorem to show that an integer a , with $0 \leq a < m = m_1m_2 \cdots m_n$, where the positive integers m_1, m_2, \dots, m_n are pairwise relatively prime, can be represented uniquely by the n -tuple $(a \pmod{m_1}, a \pmod{m_2}, \dots, a \pmod{m_n})$.
- *29.** Let m_1, m_2, \dots, m_n be pairwise relatively prime integers greater than or equal to 2. Show that if $a \equiv b \pmod{m_i}$ for $i = 1, 2, \dots, n$, then $a \equiv b \pmod{m}$, where $m = m_1m_2 \cdots m_n$. (This result will be used in Exercise 30 to prove the Chinese remainder theorem. Consequently, do not use the Chinese remainder theorem to prove it.)
- *30.** Complete the proof of the Chinese remainder theorem by showing that the simultaneous solution of a system of linear congruences modulo pairwise relatively prime moduli is unique modulo the product of these moduli. [Hint: Assume that x and y are two simultaneous solutions. Show that $m_i \mid x - y$ for all i . Using Exercise 29, conclude that $m = m_1m_2 \cdots m_n \mid x - y$.]
- 31.** Which integers leave a remainder of 1 when divided by 2 and also leave a remainder of 1 when divided by 3?
- 32.** Which integers are divisible by 5 but leave a remainder of 1 when divided by 3?
- 33.** Use Fermat's little theorem to find $7^{121} \pmod{13}$.
- 34.** Use Fermat's little theorem to find $23^{1002} \pmod{41}$.
- 35.** Use Fermat's little theorem to show that if p is prime and $p \nmid a$, then a^{p-2} is an inverse of a modulo p .
- 36.** Use Exercise 35 to find an inverse of 5 modulo 41.
- 37. a)** Show that $2^{340} \equiv 1 \pmod{11}$ by Fermat's little theorem and noting that $2^{340} = (2^{10})^{34}$.
b) Show that $2^{340} \equiv 1 \pmod{31}$ using the fact that $2^{340} = (2^5)^{68} = 32^{68}$.
c) Conclude from parts (a) and (b) that $2^{340} \equiv 1 \pmod{341}$.

- 38.** a) Use Fermat's little theorem to compute $3^{302} \pmod{5}$, $3^{302} \pmod{7}$, and $3^{302} \pmod{11}$.
 b) Use your results from part (a) and the Chinese remainder theorem to find $3^{302} \pmod{385}$. (Note that $385 = 5 \cdot 7 \cdot 11$.)
- 39.** a) Use Fermat's little theorem to compute $5^{2003} \pmod{7}$, $5^{2003} \pmod{11}$, and $5^{2003} \pmod{13}$.
 b) Use your results from part (a) and the Chinese remainder theorem to find $5^{2003} \pmod{1001}$. (Note that $1001 = 7 \cdot 11 \cdot 13$.)
- 40.** Show with the help of Fermat's little theorem that if n is a positive integer, then 42 divides $n^7 - n$.
- 41.** Show that if p is an odd prime, then every divisor of the Mersenne number $2^p - 1$ is of the form $2kp + 1$, where k is a nonnegative integer. [Hint: Use Fermat's little theorem and Exercise 37 of Section 4.3.]
- 42.** Use Exercise 41 to determine whether $M_{13} = 2^{13} - 1 = 8191$ and $M_{23} = 2^{23} - 1 = 8,388,607$ are prime.
- 43.** Use Exercise 41 to determine whether $M_{11} = 2^{11} - 1 = 2047$ and $M_{17} = 2^{17} - 1 = 131,071$ are prime.
- Let n be a positive integer and let $n - 1 = 2^s t$, where s is a nonnegative integer and t is an odd positive integer. We say that n passes **Miller's test for the base b** if either $b^t \equiv 1 \pmod{n}$ or $b^{2^j t} \equiv -1 \pmod{n}$ for some j with $0 \leq j \leq s - 1$. It can be shown (see [Ro10]) that a composite integer n passes Miller's test for fewer than $n/4$ bases b with $1 < b < n$. A composite positive integer n that passes Miller's test to the base b is called a **strong pseudoprime to the base b** .
- ***44.** Show that if n is prime and b is a positive integer with $n \not\mid b$, then n passes Miller's test to the base b .
- 45.** Show that 2047 is a strong pseudoprime to the base 2 by showing that it passes Miller's test to the base 2, but is composite.
- 46.** Show that 1729 is a Carmichael number.
- 47.** Show that 2821 is a Carmichael number.
- ***48.** Show that if $n = p_1 p_2 \cdots p_k$, where p_1, p_2, \dots, p_k are distinct primes that satisfy $p_j - 1 \mid n - 1$ for $j = 1, 2, \dots, k$, then n is a Carmichael number.
- 49.** a) Use Exercise 48 to show that every integer of the form $(6m+1)(12m+1)(18m+1)$, where m is a positive integer and $6m+1$, $12m+1$, and $18m+1$ are all primes, is a Carmichael number.
 b) Use part (a) to show that 172,947,529 is a Carmichael number.
- 50.** Find the nonnegative integer a less than 28 represented by each of these pairs, where each pair represents $(a \pmod{4}, a \pmod{7})$.
- | | | |
|------------------|------------------|------------------|
| a) (0, 0) | b) (1, 0) | c) (1, 1) |
| d) (2, 1) | e) (2, 2) | f) (0, 3) |
| g) (2, 0) | h) (3, 5) | i) (3, 6) |
- 51.** Express each nonnegative integer a less than 15 as a pair $(a \pmod{3}, a \pmod{5})$.
- 52.** Explain how to use the pairs found in Exercise 51 to add 4 and 7.
- 53.** Solve the system of congruences that arises in Example 8.

- 54.** Show that 2 is a primitive root of 19.
- 55.** Find the discrete logarithms of 5 and 6 to the base 2 modulo 19.
- 56.** Let p be an odd prime and r a primitive root of p . Show that if a and b are positive integers in \mathbb{Z}_p , then $\log_r(ab) \equiv \log_r a + \log_r b \pmod{p-1}$.
- 57.** Write out a table of discrete logarithms modulo 17 with respect to the primitive root 3.

If m is a positive integer, the integer a is a **quadratic residue** of m if $\gcd(a, m) = 1$ and the congruence $x^2 \equiv a \pmod{m}$ has a solution. In other words, a quadratic residue of m is an integer relatively prime to m that is a perfect square modulo m . If a is not a quadratic residue of m and $\gcd(a, m) = 1$, we say that it is a **quadratic nonresidue** of m . For example, 2 is a quadratic residue of 7 because $\gcd(2, 7) = 1$ and $3^2 \equiv 2 \pmod{7}$ and 3 is a quadratic nonresidue of 7 because $\gcd(3, 7) = 1$ and $x^2 \equiv 3 \pmod{7}$ has no solution.

- 58.** Which integers are quadratic residues of 11?
- 59.** Show that if p is an odd prime and a is an integer not divisible by p , then the congruence $x^2 \equiv a \pmod{p}$ has either no solutions or exactly two incongruent solutions modulo p .
- 60.** Show that if p is an odd prime, then there are exactly $(p-1)/2$ quadratic residues of p among the integers $1, 2, \dots, p-1$.
- If p is an odd prime and a is an integer not divisible by p , the **Legendre symbol** $\left(\frac{a}{p}\right)$ is defined to be 1 if a is a quadratic residue of p and -1 otherwise.
- 61.** Show that if p is an odd prime and a and b are integers with $a \equiv b \pmod{p}$, then

$$\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right).$$

- 62.** Prove **Euler's criterion**, which states that if p is an odd prime and a is a positive integer not divisible by p , then

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

- [Hint: If a is a quadratic residue modulo p , apply Fermat's little theorem; otherwise, apply Wilson's theorem, given in Exercise 18(b).]
- 63.** Use Exercise 62 to show that if p is an odd prime and a and b are integers not divisible by p , then
- $$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$$
- 64.** Show that if p is an odd prime, then -1 is a quadratic residue of p if $p \equiv 1 \pmod{4}$, and -1 is not a quadratic residue of p if $p \equiv 3 \pmod{4}$. [Hint: Use Exercise 62.]
- 65.** Find all solutions of the congruence $x^2 \equiv 29 \pmod{35}$. [Hint: Find the solutions of this congruence modulo 5 and modulo 7, and then use the Chinese remainder theorem.]

- 66.** Find all solutions of the congruence $x^2 \equiv 16 \pmod{105}$.
 [Hint: Find the solutions of this congruence modulo 3, modulo 5, and modulo 7, and then use the Chinese remainder theorem.]

- 67.** Describe a brute force algorithm for solving the discrete logarithm problem and find the worst-case and average-case time complexity of this algorithm.

4.5 Applications of Congruences

Congruences have many applications to discrete mathematics, computer science, and many other disciplines. We will introduce three applications in this section: the use of congruences to assign memory locations to computer files, the generation of pseudorandom numbers, and check digits.

Suppose that a customer identification number is ten digits long. To retrieve customer files quickly, we do not want to assign a memory location to a customer record using the ten-digit identification number. Instead, we want to use a smaller integer associated to the identification number. This can be done using what is known as a hashing function. In this section we will show how we can use modular arithmetic to do hashing.

Constructing sequences of random numbers is important for randomized algorithms, for simulations, and for many other purposes. Constructing a sequence of truly random numbers is extremely difficult, or perhaps impossible, because any method for generating what are supposed to be random numbers may generate numbers with hidden patterns. As a consequence, methods have been developed for finding sequences of numbers that have many desirable properties of random numbers, and which can be used for various purposes in place of random numbers. In this section we will show how to use congruences to generate sequences of pseudorandom numbers. The advantage is that the pseudorandom numbers so generated are constructed quickly; the disadvantage is that they have too much predictability to be used for many tasks.

Congruences also can be used to produce check digits for identification numbers of various kinds, such as code numbers used to identify retail products, numbers used to identify books, airline ticket numbers, and so on. We will explain how to construct check digits using congruences for a variety of types of identification numbers. We will show that these check digits can be used to detect certain kinds of common errors made when identification numbers are printed.

Hashing Functions



The central computer at an insurance company maintains records for each of its customers. How can memory locations be assigned so that customer records can be retrieved quickly? The solution to this problem is to use a suitably chosen **hashing function**. Records are identified using a **key**, which uniquely identifies each customer's records. For instance, customer records are often identified using the Social Security number of the customer as the key. A hashing function h assigns memory location $h(k)$ to the record that has k as its key.

In practice, many different hashing functions are used. One of the most common is the function

$$h(k) = k \bmod m$$

where m is the number of available memory locations.

Hashing functions should be easily evaluated so that files can be quickly located. The hashing function $h(k) = k \bmod m$ meets this requirement; to find $h(k)$, we need only compute the remainder when k is divided by m . Furthermore, the hashing function should be onto, so that all memory locations are possible. The function $h(k) = k \bmod m$ also satisfies this property.

EXAMPLE 1 Find the memory locations assigned by the hashing function $h(k) = k \bmod 111$ to the records of customers with Social Security numbers 064212848 and 037149212.

Solution: The record of the customer with Social Security number 064212848 is assigned to memory location 14, because

$$h(064212848) = 064212848 \bmod 111 = 14.$$

Similarly, because

$$h(037149212) = 037149212 \bmod 111 = 65,$$

the record of the customer with Social Security number 037149212 is assigned to memory location 65. 

Because a hashing function is not one-to-one (because there are more possible keys than memory locations), more than one file may be assigned to a memory location. When this happens, we say that a **collision** occurs. One way to resolve a collision is to assign the first free location following the occupied memory location assigned by the hashing function.

EXAMPLE 2 After making the assignments of records to memory locations in Example 1, assign a memory location to the record of the customer with Social Security number 107405723.

Solution: First note that the hashing function $h(k) = k \bmod 111$ maps the Social Security number 107405723 to location 14, because

$$h(107405723) = 107405723 \bmod 111 = 14.$$

However, this location is already occupied (by the file of the customer with Social Security number 064212848). But, because memory location 15, the first location following memory location 14, is free, we assign the record of the customer with Social Security number 107405723 to this location. 

In Example 1 we used a **linear probing function**, namely $h(k, i) = h(k) + i \bmod m$, to look for the first free memory location, where i runs from 0 to $m - 1$. There are many other ways to resolve collisions that are discussed in the references on hashing functions given at the end of the book.

Pseudorandom Numbers

Randomly chosen numbers are often needed for computer simulations. Different methods have been devised for generating numbers that have properties of randomly chosen numbers. Because numbers generated by systematic methods are not truly random, they are called **pseudorandom numbers**.



The most commonly used procedure for generating pseudorandom numbers is the **linear congruential method**. We choose four integers: the **modulus** m , **multiplier** a , **increment** c , and **seed** x_0 , with $2 \leq a < m$, $0 \leq c < m$, and $0 \leq x_0 < m$. We generate a sequence of pseudorandom numbers $\{x_n\}$, with $0 \leq x_n < m$ for all n , by successively using the recursively defined function

$$x_{n+1} = (ax_n + c) \bmod m.$$

(This is an example of a recursive definition, discussed in Section 5.3. In that section we will show that such sequences are well defined.)

Many computer experiments require the generation of pseudorandom numbers between 0 and 1. To generate such numbers, we divide numbers generated with a linear congruential generator by the modulus: that is, we use the numbers x_n/m .

EXAMPLE 3 Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.

Solution: We compute the terms of this sequence by successively using the recursively defined function $x_{n+1} = (7x_n + 4) \bmod 9$, beginning by inserting the seed $x_0 = 3$ to find x_1 . We find that

$$\begin{aligned}x_1 &= 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7, \\x_2 &= 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8, \\x_3 &= 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6, \\x_4 &= 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1, \\x_5 &= 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2, \\x_6 &= 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0, \\x_7 &= 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = 4 \bmod 9 = 4, \\x_8 &= 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5, \\x_9 &= 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3.\end{aligned}$$

Because $x_9 = x_0$ and because each term depends only on the previous term, we see that the sequence

$$3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \dots$$

is generated. This sequence contains nine different numbers before repeating. 

Most computers do use linear congruential generators to generate pseudorandom numbers. Often, a linear congruential generator with increment $c = 0$ is used. Such a generator is called a **pure multiplicative generator**. For example, the pure multiplicative generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16,807$ is widely used. With these values, it can be shown that $2^{31} - 2$ numbers are generated before repetition begins.

Pseudorandom numbers generated by linear congruential generators have long been used for many tasks. Unfortunately, it has been shown that sequences of pseudorandom numbers generated in this way do not share some important statistical properties that true random numbers have. Because of this, it is not advisable to use them for some tasks, such as large simulations. For such sensitive tasks, other methods are used to produce sequences of pseudorandom numbers, either using some sort of algorithm or sampling numbers arising from a random physical phenomenon. For more details on pseudorandom number, see [Kn97] and [Re10].

Check Digits

Congruences are used to check for errors in digit strings. A common technique for detecting errors in such strings is to add an extra digit at the end of the string. This final digit, or check digit, is calculated using a particular function. Then, to determine whether a digit string is correct, a check is made to see whether this final digit has the correct value. We begin with an application of this idea for checking the correctness of bit strings.

EXAMPLE 4 Parity Check Bits Digital information is represented by bit string, split into blocks of a specified size. Before each block is stored or transmitted, an extra bit, called a **parity check bit**, can be appended to each block. The parity check bit x_{n+1} for the bit string $x_1x_2\dots x_n$ is defined by

$$x_{n+1} = x_1 + x_2 + \dots + x_n \pmod{2}.$$

It follows that x_{n+1} is 0 if there are an even number of 1 bits in the block of n bits and it is 1 if there are an odd number of 1 bits in the block of n bits. When we examine a string that includes a parity check bit, we know that there is an error in it if the parity check bit is wrong. However, when the parity check bit is correct, there still may be an error. A parity check can detect an odd number of errors in the previous bits, but not an even number of errors. (See Exercise 14.)

Suppose we receive in a transmission the bit strings 01100101 and 11010110, each ending with a parity check bit. Should we accept these bit strings as correct?

Solution: Before accepting these strings as correct, we examine their parity check bits. The parity check bit of the first string is 1. Because $0 + 1 + 1 + 0 + 0 + 1 + 0 \equiv 1 \pmod{2}$, the parity check bit is correct. The parity check bit of the second string is 0. We find that $1 + 1 + 0 + 1 + 0 + 1 + 1 \equiv 1 \pmod{2}$, so the parity check is incorrect. We conclude that the first string may have been transmitted correctly and we know for certain that the second string was transmitted incorrectly. We accept the first string as correct (even though it still may contain an even number of errors), but we reject the second string. 

Check bits computed using congruences are used extensively to verify the correctness of various kinds of identification numbers. Examples 5 and 6 show how check bits are computed for codes that identify products (Universal Product Codes) and books (International Standard Book Numbers). The preambles to Exercises 18, 28, and 32 introduce the use of congruences to find and use check digits in money order numbers, airline ticket numbers, and identification numbers for periodicals, respectively. Note that congruences are also used to compute check digits for bank account numbers, drivers license numbers, credit card numbers, and many other types of identification numbers.

EXAMPLE 5 UPCs Retail products are identified by their **Universal Product Codes (UPCs)**. The most common form of a UPC has 12 decimal digits: the first digit identifies the product category, the next five digits identify the manufacturer, the following five identify the particular product, and the last digit is a check digit. The check digit is determined by the congruence

$$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}.$$

Answer these questions:

- (a) Suppose that the first 11 digits of a UPC are 79357343104. What is the check digit?
- (b) Is 041331021641 a valid UPC?

Solution: (a) We insert the digits of 79357343104 into the congruence for UPC check digits. This gives $3 \cdot 7 + 9 + 3 \cdot 3 + 5 + 3 \cdot 7 + 3 + 3 \cdot 4 + 3 + 3 \cdot 1 + 0 + 3 \cdot 4 + x_{12} \equiv 0 \pmod{10}$. Simplifying, we have $21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$. Hence, $98 + x_{12} \equiv 0 \pmod{10}$. It follows that $x_{12} \equiv 2 \pmod{10}$, so the check digit is 2.

(b) To check whether 041331021641 is valid, we insert the digits into the congruence these digits must satisfy. This gives $3 \cdot 0 + 4 + 3 \cdot 1 + 3 + 3 \cdot 3 + 1 + 3 \cdot 0 + 2 + 3 \cdot 1 + 6 + 3 \cdot 4 + 1 \equiv 0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 \equiv 4 \not\equiv 0 \pmod{10}$. Hence, 041331021641 is not a valid UPC. 

EXAMPLE 6

Remember that the check digit of an ISBN-10 can be an X!

ISBNs All books are identified by an **International Standard Book Number (ISBN-10)**, a 10-digit code $x_1x_2 \dots x_{10}$, assigned by the publisher. (Recently, a 13-digit code known as ISBN-13 was introduced to identify a larger number of published works; see the preamble to Exercise 42 in the Supplementary Exercises.) An ISBN-10 consists of blocks identifying the language, the publisher, the number assigned to the book by its publishing company, and finally, a check digit that is either a digit or the letter X (used to represent 10). This check digit is selected so that

$$x_{10} \equiv \sum_{i=1}^9 ix_i \pmod{11},$$

or equivalently, so that

$$\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}.$$

Answer these questions about ISBN-10s:

- (a) The first nine digits of the ISBN-10 of the sixth edition of this book are 007288008. What is the check digit?
- (b) Is 084930149X a valid ISBN-10?

Solution: (a) The check digit is determined by the congruence $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$. Inserting the digits 007288008 gives $x_{10} \equiv 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 7 + 4 \cdot 2 + 5 \cdot 8 + 6 \cdot 8 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 8 \pmod{11}$. This means that $x_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$, so $x_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $x_{10} = 2$.

(b) To see whether 084930149X is a valid ISBN-10, we see if $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}$. We see that $1 \cdot 0 + 2 \cdot 8 + 3 \cdot 4 + 4 \cdot 9 + 5 \cdot 3 + 6 \cdot 0 + 7 \cdot 1 + 8 \cdot 4 + 9 \cdot 9 + 10 \cdot 10 = 0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$. Hence, 084930149X is not a valid ISBN-10. 

Several kinds of errors often arise in identification numbers. A **single error**, an error in one digit of an identification number, is perhaps the most common type of error. Another common kind of error is a **transposition error**, which occurs when two digits are accidentally interchanged. For each type of identification number, including a check digit, we would like to be able to detect these common types of errors, as well as other types of errors. We will investigate whether the check digit for ISBNs can detect single errors and transposition errors. Whether check digits for UPCs can detect these kinds of errors is left as Exercises 26 and 27.

Suppose that $x_1x_2 \dots x_{10}$ is a valid ISBN (so that $\sum_{i=1}^{10} ix_i \equiv 0 \pmod{10}$). We will show that we can detect a single error and a transposition of two digits (where we include the possibility that one of the two digits is the check digit X, representing 10). Suppose that this ISBN has been printed with a single error as $y_1y_2 \dots y_{10}$. If there is a single error, then, for some integer j , $y_i = x_i$ for $i \neq j$ and $y_j = x_j + a$ where $-10 \leq a \leq 10$ and $a \neq 0$. Note that $a = y_j - x_j$ is the error in the j th place. It then follows that

$$\sum_{i=1}^{10} iy_i = \left(\sum_{i=1}^{10} ix_i \right) + ja \equiv ja \not\equiv 0 \pmod{11}.$$

Publishers sometimes do not calculate ISBNs correctly for their books, as was done for an earlier edition of this text.

These last two congruences hold because $\sum_{i=1}^{10} x_i \equiv 0 \pmod{10}$ and $11 \nmid ja$, because $11 \nmid j$ and $11 \nmid a$. We conclude that $y_1y_2\dots y_{10}$ is not a valid ISBN. So, we have detected the single error.

Now suppose that two unequal digits have been transposed. It follows that there are distinct integers j and k such that $y_j = x_k$ and $y_k = x_j$, and $y_i = x_i$ for $i \neq j$ and $i \neq k$. Hence,

$$\sum_{i=1}^{10} iy_i = \left(\sum_{i=1}^{10} ix_i \right) + (jx_k - jx_j) + (kx_j - kx_k) \equiv (j-k)(x_k - x_j) \not\equiv 0 \pmod{11},$$

because $\sum_{i=1}^{10} x_i \equiv 0 \pmod{10}$ and $11 \nmid (j-k)$ and $11 \nmid (x_k - x_j)$. We see that $y_1y_2\dots y_{10}$ is not a valid ISBN. Thus, we can detect the interchange of two unequal digits.

Exercises

1. Which memory locations are assigned by the hashing function $h(k) = k \pmod{97}$ to the records of insurance company customers with these Social Security numbers?
 - a) 034567981
 - b) 183211232
 - c) 220195744
 - d) 987255335
2. Which memory locations are assigned by the hashing function $h(k) = k \pmod{101}$ to the records of insurance company customers with these Social Security numbers?
 - a) 104578690
 - b) 432222187
 - c) 372201919
 - d) 501338753
3. A parking lot has 31 visitor spaces, numbered from 0 to 30. Visitors are assigned parking spaces using the hashing function $h(k) = k \pmod{31}$, where k is the number formed from the first three digits on a visitor's license plate.
 - a) Which spaces are assigned by the hashing function to cars that have these first three digits on their license plates: 317, 918, 007, 100, 111, 310?
 - b) Describe a procedure visitors should follow to find a free parking space, when the space they are assigned is occupied.

Another way to resolve collisions in hashing is to use *double hashing*. We use an initial hashing function $h(k) = k \pmod{p}$ where p is prime. We also use a second hashing function $g(k) = (k+1) \pmod{(p-2)}$. When a collision occurs, we use a probing sequence $h(k, i) = (h(k) + i \cdot g(k)) \pmod{p}$.

4. Use the double hashing procedure we have described with $p = 4969$ to assign memory locations to files for employees with social security numbers $k_1 = 132489971$, $k_2 = 509496993$, $k_3 = 546332190$, $k_4 = 034367980$, $k_5 = 047900151$, $k_6 = 329938157$, $k_7 = 212228844$, $k_8 = 325510778$, $k_9 = 353354519$, $k_{10} = 053708912$.
5. What sequence of pseudorandom numbers is generated using the linear congruential generator $x_{n+1} = (3x_n + 2) \pmod{13}$ with seed $x_0 = 1$?
6. What sequence of pseudorandom numbers is generated using the linear congruential generator $x_{n+1} = (4x_n + 1) \pmod{7}$ with seed $x_0 = 3$?

7. What sequence of pseudorandom numbers is generated using the pure multiplicative generator $x_{n+1} = 3x_n \pmod{11}$ with seed $x_0 = 2$?

8. Write an algorithm in pseudocode for generating a sequence of pseudorandom numbers using a linear congruential generator.

The **middle-square method** for generating pseudorandom numbers begins with an n -digit integer. This number is squared, initial zeros are appended to ensure that the result has $2n$ digits, and its middle n digits are used to form the next number in the sequence. This process is repeated to generate additional terms.

9. Find the first eight terms of the sequence of four-digit pseudorandom numbers generated by the middle square method starting with 2357.

10. Explain why both 3792 and 2916 would be bad choices for the initial term of a sequence of four-digit pseudorandom numbers generated by the middle square method.

The **power generator** is a method for generating pseudorandom numbers. To use the power generator, parameters p and d are specified, where p is a prime, d is a positive integer such that $p \nmid d$, and a seed x_0 is specified. The pseudorandom numbers x_1, x_2, \dots are generated using the recursive definition $x_{n+1} = x_n^d \pmod{p}$.

11. Find the sequence of pseudorandom numbers generated by the power generator with $p = 7$, $d = 3$, and seed $x_0 = 2$.

12. Find the sequence of pseudorandom numbers generated by the power generator with $p = 11$, $d = 2$, and seed $x_0 = 3$.

13. Suppose you received these bit strings over a communications link, where the last bit is a parity check bit. In which string are you sure there is an error?

- a) 0000011111
- b) 10101010101
- c) 11111100000
- d) 10111101111

14. Prove that a parity check bit can detect an error in a string if and only if the string contains an odd number of errors.

15. The first nine digits of the ISBN-10 of the European version of the fifth edition of this book are 0-07-119881. What is the check digit for that book?
16. The ISBN-10 of the sixth edition of *Elementary Number Theory and Its Applications* is 0-321-500Q1-8, where Q is a digit. Find the value of Q .
17. Determine whether the check digit of the ISBN-10 for this textbook (the seventh edition of *Discrete Mathematics and its Applications*) was computed correctly by the publisher.

The United States Postal Service (USPS) sells money orders identified by an 11-digit number $x_1x_2 \dots x_{11}$. The first ten digits identify the money order; x_{11} is a check digit that satisfies $x_{11} = x_1 + x_2 + \dots + x_{10} \pmod{9}$.

18. Find the check digit for the USPS money orders that have identification number that start with these ten digits.
- 7555618873
 - 6966133421
 - 8018927435
 - 3289744134
19. Determine whether each of these numbers is a valid USPS money order identification number.
- 74051489623
 - 88382013445
 - 56152240784
 - 66606631178
20. One digit in each of these identification numbers of a postal money order is smudged. Can you recover the smudged digit, indicated by a Q , in each of these numbers?
- $Q1223139784$
 - 6702120 Q 988
 - 27 Q 41007734
 - 213279032 Q 1
21. One digit in each of these identification numbers of a postal money order is smudged. Can you recover the smudged digit, indicated by a Q , in each of these numbers?
- 493212 Q 0688
 - 850 Q 9103858
 - 2 Q 941007734
 - 66687 Q 03201
22. Determine which single digit errors are detected by the USPS money order code.
23. Determine which transposition errors are detected by the USPS money order code.
24. Determine the check digit for the UPCs that have these initial 11 digits.
- 73232184434
 - 63623991346
 - 04587320720
 - 93764323341
25. Determine whether each of the strings of 12 digits is a valid UPC code.

- 036000291452
- 012345678903
- 782421843014
- 726412175425

26. Does the check digit of a UPC code detect all single errors? Prove your answer or find a counterexample.
27. Determine which transposition errors the check digit of a UPC code finds.

Some airline tickets have a 15-digit identification number $a_1a_2 \dots a_{15}$ where a_{15} is a check digit that equals $a_1a_2 \dots a_{14} \pmod{7}$.

28. Find the check digit a_{15} that follows each of these initial 14 digits of an airline ticket identification number.

- 10237424413392
- 00032781811234
- 00611232134231
- 00193222543435

29. Determine whether each of these 15-digit numbers is a valid airline ticket identification number.

- 101333341789013
- 007862342770445
- 113273438882531
- 000122347322871

30. Which errors in a single digit of a 15-digit airline ticket identification number can be detected?

- *31. Can the accidental transposition of two consecutive digits in an airline ticket identification number be detected using the check digit?

Periodicals are identified using an **International Standard Serial Number (ISSN)**. An ISSN consists of two blocks of four digits. The last digit in the second block is a check digit. This check digit is determined by the congruence $d_8 \equiv 3d_1 + 4d_2 + 5d_3 + 6d_4 + 7d_5 + 8d_6 + 9d_7 \pmod{11}$. When $d_8 \equiv 10 \pmod{11}$, we use the letter X to represent d_8 in the code.

32. For each of these initial seven digits of an ISSN, determine the check digit (which may be the letter X).

- 1570-868
- 1553-734
- 1089-708
- 1383-811

33. Are each of these eight-digit codes possible ISSNs? That is, do they end with a correct check digit?

- 1059-1027
- 0002-9890
- 1530-8669
- 1007-120X

34. Does the check digit of an ISSN detect every single error in an ISSN? Justify your answer with either a proof or a counterexample.

35. Does the check digit of an ISSN detect every error where two consecutive digits are accidentally interchanged? Justify your answer with either a proof or a counterexample.

4.6 Cryptography

Introduction

Number theory plays a key role in cryptography, the subject of transforming information so that it cannot be easily recovered without special knowledge. Number theory is the basis of many classical ciphers, first used thousands of years ago, and used extensively until the 20th century. These ciphers encrypt messages by changing each letter to a different letter, or each block of letters to a different block of letters. We will discuss some classical ciphers, including shift ciphers, which replace each letter by the letter a fixed number of positions later in the alphabet, wrapping around to the beginning of the alphabet when necessary. The classical ciphers we will discuss are examples of private key ciphers where knowing how to encrypt allows someone to also decrypt messages. With a private key cipher, two parties who wish to communicate in secret must share a secret key. The classical ciphers we will discuss are also vulnerable to cryptanalysis, which seeks to recover encrypted information without access to the secret information used to encrypt the message. We will show how to cryptanalyze messages sent using shift ciphers.

Number theory is also important in public key cryptography, a type of cryptography invented in the 1970s. In public key cryptography, knowing how to encrypt does not also tell someone how to decrypt. The most widely used public key system, called the RSA cryptosystem, encrypts messages using modular exponentiation, where the modulus is the product of two large primes. Knowing how to encrypt requires that someone know the modulus and an exponent. (It does not require that the two prime factors of the modulus be known.) As far as it is known, knowing how to decrypt requires someone to know how to invert the encryption function, which can only be done in a practical amount of time when someone knows these two large prime factors. In this chapter we will explain how the RSA cryptosystem works, including how to encrypt and decrypt messages.

The subject of cryptography also includes the subject of cryptographic protocols, which are exchanges of messages carried out by two or more parties to achieve a specific security goal. We will discuss two important protocols in this chapter. One allows two people to share a common secret key. The other can be used to send signed messages so that a recipient can be sure that they were sent by the purported sender.

Classical Cryptography

One of the earliest known uses of cryptography was by Julius Caesar. He made messages secret by shifting each letter three letters forward in the alphabet (sending the last three letters of the alphabet to the first three). For instance, using this scheme the letter B is sent to E and the letter X is sent to A. This is an example of **encryption**, that is, the process of making a message secret.

To express Caesar's encryption process mathematically, first replace each letter by an element of \mathbb{Z}_{26} , that is, an integer from 0 to 25 equal to one less than its position in the alphabet. For example, replace A by 0, K by 10, and Z by 25. Caesar's encryption method can be represented by the function f that assigns to the nonnegative integer p , $p \leq 25$, the integer $f(p)$ in the set $\{0, 1, 2, \dots, 25\}$ with

$$f(p) = (p + 3) \bmod 26.$$

In the encrypted version of the message, the letter represented by p is replaced with the letter represented by $(p + 3) \bmod 26$.

EXAMPLE 1 What is the secret message produced from the message “MEET YOU IN THE PARK” using the Caesar cipher?

Solution: First replace the letters in the message with numbers. This produces

$$12 \ 4 \ 4 \ 19 \quad 24 \ 14 \ 20 \quad 8 \ 13 \quad 19 \ 7 \ 4 \quad 15 \ 0 \ 17 \ 10.$$

Now replace each of these numbers p by $f(p) = (p + 3) \bmod 26$. This gives

$$15 \ 7 \ 7 \ 22 \quad 1 \ 17 \ 23 \quad 11 \ 16 \quad 22 \ 10 \ 7 \quad 18 \ 3 \ 20 \ 13.$$

Translating this back to letters produces the encrypted message “PHHW BRX LQ WKH SDUN.” 

To recover the original message from a secret message encrypted by the Caesar cipher, the function f^{-1} , the inverse of f , is used. Note that the function f^{-1} sends an integer p from \mathbf{Z}_{26} , to $f^{-1}(p) = (p - 3) \bmod 26$. In other words, to find the original message, each letter is shifted back three letters in the alphabet, with the first three letters sent to the last three letters of the alphabet. The process of determining the original message from the encrypted message is called **decryption**.

There are various ways to generalize the Caesar cipher. For example, instead of shifting the numerical equivalent of each letter by 3, we can shift the numerical equivalent of each letter by k , so that

$$f(p) = (p + k) \bmod 26.$$

Such a cipher is called a *shift cipher*. Note that decryption can be carried out using

$$f^{-1}(p) = (p - k) \bmod 26.$$

Here the integer k is called a **key**. We illustrate the use of a shift cipher in Examples 2 and 3.

EXAMPLE 2 Encrypt the plaintext message “STOP GLOBAL WARMING” using the shift cipher with shift $k = 11$.

Solution: To encrypt the message “STOP GLOBAL WARMING” we first translate each letter to the corresponding element of \mathbf{Z}_{26} . This produces the string

$$18 \ 19 \ 14 \ 15 \quad 6 \ 11 \ 14 \ 1 \ 0 \ 11 \quad 22 \ 0 \ 17 \ 12 \ 8 \ 13 \ 6.$$

We now apply the shift $f(p) = (p + 11) \bmod 26$ to each number in this string. We obtain

$$3 \ 4 \ 25 \ 0 \quad 17 \ 22 \ 25 \ 12 \ 11 \ 22 \quad 7 \ 11 \ 2 \ 23 \ 19 \ 24 \ 17.$$

Translating this last string back to letters, we obtain the ciphertext “DEZA RWZMLW HLCX-TYR.” 

EXAMPLE 3 Decrypt the ciphertext message “LEWLÝPLUJL PZ H NYLHA ALHJOLY” that was encrypted with the shift cipher with shift $k = 7$.

Solution: To decrypt the ciphertext “LEWLÝPLUJL PZ H NYLHA ALHJOLY” we first translate the letters back to elements of \mathbf{Z}_{26} . We obtain

$$11 \ 4 \ 22 \ 11 \ 24 \ 15 \ 11 \ 20 \ 9 \ 11 \quad 15 \ 25 \quad 7 \quad 13 \ 24 \ 11 \ 7 \ 0 \quad 0 \ 11 \ 7 \ 9 \ 14 \ 11 \ 24.$$

Next, we shift each of these numbers by $-k = -7$ modulo 26 to obtain

$$4 \ 23 \ 15 \ 4 \ 17 \ 8 \ 4 \ 13 \ 2 \ 4 \quad 8 \ 18 \quad 0 \quad 6 \ 17 \ 4 \ 0 \ 19 \quad 19 \ 4 \ 0 \ 2 \ 7 \ 4 \ 17.$$

Finally, we translate these numbers back to letters to obtain the plaintext. We obtain “EXPERIENCE IS A GREAT TEACHER.” 

We can generalize shift ciphers further to slightly enhance security by using a function of the form

$$f(p) = (ap + b) \text{ mod } 26,$$

where a and b are integers, chosen so that f is a bijection. (The function $f(p) = (ap + b) \text{ mod } 26$ is a bijection if and only if $\gcd(a, 26) = 1$.) Such a mapping is called an *affine transformation*, and the resulting cipher is called an *affine cipher*.

EXAMPLE 4 What letter replaces the letter K when the function $f(p) = (7p + 3) \text{ mod } 26$ is used for encryption?

Solution: First, note that 10 represents K. Then, using the encryption function specified, it follows that $f(10) = (7 \cdot 10 + 3) \text{ mod } 26 = 21$. Because 21 represents V, K is replaced by V in the encrypted message. 

We will now show how to decrypt messages encrypted using an affine cipher. Suppose that $c = (ap + b) \text{ mod } 26$ with $\gcd(a, 26) = 1$. To decrypt we need to show how to express p in terms of c . To do this, we apply the encrypting congruence $c \equiv ap + b \pmod{26}$, and solve it for p . To do this, we first subtract b from both sides, to obtain $c - b \equiv ap \pmod{26}$. Because $\gcd(a, 26) = 1$, we know that there is an inverse \bar{a} of a modulo 26. Multiplying both sides of the last equation by \bar{a} gives us $\bar{a}(c - b) \equiv \bar{a}ap \pmod{26}$. Because $\bar{a}a \equiv 1 \pmod{26}$, this tells us that $p \equiv \bar{a}(c - b) \pmod{26}$. This determines p because p belongs to \mathbb{Z}_{26} .

CRYPTANALYSIS The process of recovering plaintext from ciphertext without knowledge of both the encryption method and the key is known as **cryptanalysis** or **breaking codes**. In general, cryptanalysis is a difficult process, especially when the encryption method is unknown. We will not discuss cryptanalysis in general, but we will explain how to break messages that were encrypted using a shift cipher.

If we know that a ciphertext message was produced by enciphering a message using a shift cipher, we can try to recover the message by shifting all characters of the ciphertext by each of the 26 possible shifts (including a shift of zero characters). One of these is guaranteed to be the plaintext. However, we can use a more intelligent approach, which we can build upon to cryptanalyze ciphertext resulting from other ciphers. The main tool for cryptanalyzing ciphertext encrypted using a shift cipher is the count of the frequency of letters in the ciphertext. The nine most common letters in English text and their approximate relative frequencies are E 13%, T 9%, A 8%, O 8%, I 7%, N 7%, S 7%, H 6%, and R 6%. To cryptanalyze ciphertext that we know was produced using a shift cipher, we first find the relative frequencies of letters in the ciphertext. We list the most common letters in the ciphertext in frequency order; we hypothesize that the most common letter in the ciphertext is produced by encrypting E. Then, we determine the value of the shift under this hypothesis, say k . If the message produced by shifting the ciphertext by $-k$ makes sense, we presume that our hypothesis is correct and that we have the correct value of k . If it does not make sense, we next consider the hypothesis that the most common letter in the ciphertext is produced by encrypting T, the second most common letter in English; we find k under this hypothesis, shift the letters of the message by $-k$, and see whether the resulting message makes sense. If it does not, we continue the process working our way through the letters from most common to least common.

Mathematicians make the best code breakers. Their work in World War II changed the course of the war.

EXAMPLE 5 Suppose that we intercepted the ciphertext message ZNK KGXRE HOXJ MKZY ZNK CUXS that we know was produced by a shift cipher. What was the original plaintext message?

Solution: Because we know that the intercepted ciphertext message was encrypted using a shift cipher, we begin by calculating the frequency of letters in the ciphertext. We find that the most common letter in the ciphertext is K. So, we hypothesize that the shift cipher sent the plaintext letter E to the ciphertext letter K. If this hypothesis is correct, we know that $10 = 4 + k \bmod 26$, so $k = 6$. Next, we shift the letters of the message by -6 , obtaining THE EARLY BIRD GETS THE WORM. Because this message makes sense, we assume that the hypothesis that $k = 6$ is correct. 



BLOCK CIPHERS Shift ciphers and affine ciphers proceed by replacing each letter of the alphabet by another letter in the alphabet. Because of this, these ciphers are called **character or monoalphabetic ciphers**. Encryption methods of this kind are vulnerable to attacks based on the analysis of letter frequency in the ciphertext, as we just illustrated. We can make it harder to successfully attack ciphertext by replacing blocks of letters with other blocks of letters instead of replacing individual characters with individual characters; such ciphers are called **block ciphers**.

We will now introduce a simple type of block cipher, called the **transposition cipher**. As a key we use a permutation σ of the set $\{1, 2, \dots, m\}$ for some positive integer m , that is, a one-to-one function from $\{1, 2, \dots, m\}$ to itself. To encrypt a message we first split its letters into blocks of size m . (If the number of letters in the message is not divisible by m we add some random letters at the end to fill out the final block.) We encrypt the block $p_1 p_2 \dots p_m$ as $c_1 c_2 \dots c_m = p_{\sigma(1)} p_{\sigma(2)} \dots, p_{\sigma(m)}$. To decrypt a ciphertext block $c_1 c_2 \dots c_m$, we transpose its letters using the permutation σ^{-1} , the inverse of σ . Example 6 illustrates encryption and decryption for a transposition cipher.

EXAMPLE 6 Using the transposition cipher based on the permutation σ of the set $\{1, 2, 3, 4\}$ with $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 4$, and $\sigma(4) = 2$,

(a) Encrypt the plaintext message PIRATE ATTACK.

(b) Decrypt the ciphertext message SWUE TRAE OEHS, which was encrypted using this cipher.

Solution: (a) We first split the letters of the plaintext into blocks of four letters. We obtain PIRA TEAT TACK. To encrypt each block, we send the first letter to the third position, the second letter to the first position, the third letter to the fourth position, and the fourth letter to the second position. We obtain IAPR ETIA AKTC.

(b) We note that σ^{-1} , the inverse of σ , sends 1 to 2, sends 2 to 4, sends 3 to 1, and sends 4 to 3. Applying $\sigma^{-1}(m)$ to each block gives us the plaintext: USEW ATER HOSE. (Grouping together these letters to form common words, we surmise that the plaintext is USE WATER HOSE.) 

CRYPTOSYSTEMS We have defined two families of ciphers: shift ciphers and affine ciphers. We now introduce the notion of a cryptosystem, which provides a general structure for defining new families of ciphers.

DEFINITION 1

A *cryptosystem* is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where \mathcal{P} is the set of plaintext strings, \mathcal{C} is the set of ciphertext strings, \mathcal{K} is the *keyspace* (the set of all possible keys), \mathcal{E} is the set of encryption functions, and \mathcal{D} is the set of decryption functions. We denote by E_k the encryption function in \mathcal{E} corresponding to the key k and D_k the decryption function in \mathcal{D} that decrypts ciphertext that was encrypted using E_k , that is $D_k(E_k(p)) = p$, for all plaintext strings p .

We now illustrate the use of the definition of a cryptosystem.

EXAMPLE 7 Describe the family of shift ciphers as a cryptosystem.

Solution: To encrypt a string of English letters with a shift cipher, we first translate each letter to an integer between 0 and 25, that is, to an element of \mathbf{Z}_{26} . We then shift each of these integers by a fixed integer modulo 26, and finally, we translate the integers back to letters. To apply the definition of a cryptosystem to shift ciphers, we assume that our messages are already integers, that is, elements of \mathbf{Z}_{26} . That is, we assume that the translation between letters and integers is outside of the cryptosystem. Consequently, both the set of plaintext strings \mathcal{P} and the set of ciphertext strings \mathcal{C} are the set of strings of elements of \mathbf{Z}_{26} . The set of keys \mathcal{K} is the set of possible shifts, so $\mathcal{K} = \mathbf{Z}_{26}$. The set \mathcal{E} consists of functions of the form $E_k(p) = (p + k) \bmod 26$, and the set \mathcal{D} of decryption functions is the same as the set of encrypting functions where $D_k(p) = (p - k) \bmod 26$. 

The concept of a cryptosystem is useful in the discussion of additional families of ciphers and is used extensively in cryptography.

Public Key Cryptography

All classical ciphers, including shift ciphers and affine ciphers, are examples of **private key cryptosystems**. In a private key cryptosystem, once you know an encryption key, you can quickly find the decryption key. So, knowing how to encrypt messages using a particular key allows you to decrypt messages that were encrypted using this key. For example, when a shift cipher is used with encryption key k , the plaintext integer p is sent to

$$c = (p + k) \bmod 26.$$

Decryption is carried out by shifting by $-k$; that is,

$$p = (c - k) \bmod 26.$$

So knowing how to encrypt with a shift cipher also tells you how to decrypt.

When a private key cryptosystem is used, two parties who wish to communicate in secret must share a secret key. Because anyone who knows this key can both encrypt and decrypt messages, two people who want to communicate securely need to securely exchange this key. (We will introduce a method for doing this later in this section.) The shift cipher and affine cipher cryptosystems are private key cryptosystems. They are quite simple and are extremely vulnerable to cryptanalysis. However, the same is not true of many modern private key cryptosystems. In particular, the current US government standard for private key cryptography, the Advanced Encryption Standard (AES), is extremely complex and is considered to be highly resistant to cryptanalysis. (See [St06] for details on AES and other modern private key cryptosystems.) AES is widely used in government and commercial communications. However, it still shares the property that for secure communications keys be shared. Furthermore, for extra security, a new key is used for each communication session between two parties, which requires a method for generating keys and securely sharing them.

To avoid the need for keys to be shared by every pair of parties that wish to communicate securely, in the 1970s cryptologists introduced the concept of **public key cryptosystems**. When such cryptosystems are used, knowing how to send an encrypted message does not help decrypt messages. In such a system, everyone can have a publicly known encryption key. Only the decryption keys are kept secret, and only the intended recipient of a message can decrypt it, because, as far as it is currently known, knowledge of the encryption key does not let someone recover the plaintext message without an extraordinary amount of work (such as billions of years of computer time).

The RSA Cryptosystem

M.I.T. is also known as the 'Tute.

Unfortunately, no one calls this the Cocks cryptosystem.

In 1976, three researchers at the Massachusetts Institute of Technology—Ronald Rivest, Adi Shamir, and Leonard Adleman—introduced to the world a public key cryptosystem, known as the **RSA system**, from the initials of its inventors. As often happens with cryptographic discoveries, the RSA system had been discovered several years earlier in secret government research in the United Kingdom. Clifford Cocks, working in secrecy at the United Kingdom's Government Communications Headquarters (GCHQ), had discovered this cryptosystem in 1973. However, his invention was unknown to the outside world until the late 1990s, when he was allowed to share classified GCHQ documents from the early 1970s. (An excellent account of this earlier discovery, as well as the work of Rivest, Shamir, and Adleman, can be found in [Si99].)

In the RSA cryptosystem, each individual has an encryption key (n, e) where $n = pq$, the modulus is the product of two large primes p and q , say with 200 digits each, and an exponent e that is relatively prime to $(p - 1)(q - 1)$. To produce a usable key, two large primes must be found. This can be done quickly on a computer using probabilistic primality tests, referred to earlier in this section. However, the product of these primes $n = pq$, with approximately 400 digits, cannot, as far as is currently known, be factored in a reasonable length of time. As we will see, this is an important reason why decryption cannot, as far as is currently known, be done quickly without a separate decryption key.

RSA Encryption

To encrypt messages using a particular key (n, e) , we first translate a plaintext message M into sequences of integers. To do this, we first translate each plaintext letter into a two-digit number, using the same translation we employed for shift ciphers, with one key difference. That is, we include an initial zero for the letters A through J, so that A is translated into 00, B into 01, . . . , and J into 09. Then, we concatenate these two-digit numbers into strings of digits. Next, we divide this string into equally sized blocks of $2N$ digits, where $2N$ is the largest even number such that the number 2525 . . . 25 with $2N$ digits does not exceed n . (When necessary, we pad the plaintext message with dummy Xs to make the last block the same size as all other blocks.)

After these steps, we have translated the plaintext message M into a sequence of integers m_1, m_2, \dots, m_k for some integer k . Encryption proceeds by transforming each block m_i to a ciphertext block c_i . This is done using the function

$$C = M^e \pmod{n}.$$

(To perform the encryption, we use an algorithm for fast modular exponentiation, such as Algorithm 5 in Section 4.2.) We leave the encrypted message as blocks of numbers and send these to the intended recipient. Because the RSA cryptosystem encrypts blocks of characters into blocks of characters, it is a block cipher.



CLIFFORD COCKS (BORN 1950) Clifford Cocks, born in Cheshire, England, was a talented mathematics student. In 1968 he won a silver medal at the International Mathematical Olympiad. Cocks attended King's College, Cambridge, studying mathematics. He also spent a short time at Oxford University working in number theory. In 1973 he decided not to complete his graduate work, instead taking a mathematical job at the Government Communications Headquarters (GCHQ) of British intelligence. Two months after joining GCHQ, Cocks learned about public key cryptography from an internal GCHQ report written by James Ellis. Cocks used his number theory knowledge to invent what is now called the RSA cryptosystem. He quickly realized that a public key cryptosystem could be based on the difficulty of reversing the process of multiplying two large primes. In 1997 he was allowed to reveal declassified GCHQ internal documents describing his discovery. Cocks is also known for his invention of a secure identity based encryption scheme, which uses information about a user's identity as a public key. In 2001, Cocks became the Chief Mathematician at GCHQ. He has also set up the Heilbronn Institute for Mathematical Research, a partnership between GCHQ and the University of Bristol.

Example 8 illustrates how RSA encryption is performed. For practical reasons we use small primes p and q in this example, rather than primes with 200 or more digits. Although the cipher described in this example is not secure, it does illustrate the techniques used in the RSA cipher.

EXAMPLE 8 Encrypt the message STOP using the RSA cryptosystem with key $(2537, 13)$. Note that $2537 = 43 \cdot 59$, $p = 43$ and $q = 59$ are primes, and

$$\gcd(e, (p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1.$$

Solution: To encrypt, we first translate the letters in STOP into their numerical equivalents. We then group these numbers into blocks of four digits (because $2525 < 2537 < 252525$), to obtain

$$1819 \quad 1415.$$

We encrypt each block using the mapping

$$C = M^{13} \pmod{2537}.$$

Computations using fast modular multiplication show that $1819^{13} \pmod{2537} = 2081$ and $1415^{13} \pmod{2537} = 2182$. The encrypted message is 2081 2182. 

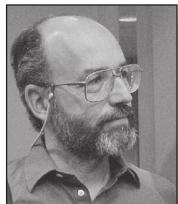
RSA Decryption

The plaintext message can be quickly recovered from a ciphertext message when the decryption key d , an inverse of e modulo $(p-1)(q-1)$, is known. [Such an inverse exists because $\gcd(e, (p-1)(q-1)) = 1$.] To see this, note that if $de \equiv 1 \pmod{(p-1)(q-1)}$, there is an integer k such that $de = 1 + k(p-1)(q-1)$. It follows that

$$C^d \equiv (M^e)^d = M^{de} = M^{1+k(p-1)(q-1)} \pmod{n}.$$



RONALD RIVEST (BORN 1948) Ronald Rivest received a B.A. from Yale in 1969 and his Ph.D. in computer science from Stanford in 1974. Rivest is a computer science professor at M.I.T. and was a cofounder of RSA Data Security, which held the patent on the RSA cryptosystem that he invented together with Adi Shamir and Leonard Adleman. Areas that Rivest has worked in besides cryptography include machine learning, VLSI design, and computer algorithms. He is a coauthor of a popular text on algorithms ([CoLeRiSt09]).



ADI SHAMIR (BORN 1952) Adi Shamir was born in Tel Aviv, Israel. His undergraduate degree is from Tel Aviv University (1972) and his Ph.D. is from the Weizmann Institute of Science (1977). Shamir was a research assistant at the University of Warwick and an assistant professor at M.I.T. He is currently a professor in the Applied Mathematics Department at the Weizmann Institute and leads a group studying computer security. Shamir's contributions to cryptography, besides the RSA cryptosystem, include cracking knapsack cryptosystems, cryptanalysis of the Data Encryption Standard (DES), and the design of many cryptographic protocols.



LEONARD ADLEMAN (BORN 1945) Leonard Adleman was born in San Francisco, California. He received a B.S. in mathematics (1968) and his Ph.D. in computer science (1976) from the University of California, Berkeley. Adleman was a member of the mathematics faculty at M.I.T. from 1976 until 1980, where he was a coinventor of the RSA cryptosystem, and in 1980 he took a position in the computer science department at the University of Southern California (USC). He was appointed to a chaired position at USC in 1985. Adleman has worked on computer security, computational complexity, immunology, and molecular biology. He invented the term "computer virus." Adleman's recent work on DNA computing has sparked great interest. He was a technical adviser for the movie *Sneakers*, in which computer security played an important role.

By Fermat's little theorem [assuming that $\gcd(M, p) = \gcd(M, q) = 1$, which holds except in rare cases, which we cover in Exercise 28], it follows that $M^{p-1} \equiv 1 \pmod{p}$ and $M^{q-1} \equiv 1 \pmod{q}$. Consequently,

$$C^d \equiv M \cdot (M^{p-1})^{k(q-1)} \equiv M \cdot 1 = M \pmod{p}$$

and

$$C^d \equiv M \cdot (M^{q-1})^{k(p-1)} \equiv M \cdot 1 = M \pmod{q}.$$

Because $\gcd(p, q) = 1$, it follows by the Chinese remainder theorem that

$$C^d \equiv M \pmod{pq}.$$

Example 9 illustrates how to decrypt messages sent using the RSA cryptosystem.

EXAMPLE 9 We receive the encrypted message 0981 0461. What is the decrypted message if it was encrypted using the RSA cipher from Example 8?

Solution: The message was encrypted using the RSA cryptosystem with $n = 43 \cdot 59$ and exponent 13. As Exercise 2 in Section 4.4 shows, $d = 937$ is an inverse of 13 modulo $42 \cdot 58 = 2436$. We use 937 as our decryption exponent. Consequently, to decrypt a block C , we compute

$$M = C^{937} \pmod{2537}.$$

To decrypt the message, we use the fast modular exponentiation algorithm to compute $0981^{937} \pmod{2537} = 0704$ and $0461^{937} \pmod{2537} = 1115$. Consequently, the numerical version of the original message is 0704 1115. Translating this back to English letters, we see that the message is HELP. 

RSA as a Public Key System



Why is the RSA cryptosystem suitable for public key cryptography? First, it is possible to rapidly construct a public key by finding two large primes p and q , each with more than 200 digits, and to find an integer e relatively prime to $(p - 1)(q - 1)$. When we know the factorization of the modulus n , that is, when we know p and q , we can quickly find an inverse d of e modulo $(p - 1)(q - 1)$. [This is done by using the Euclidean algorithm to find Bézout coefficients s and t for d and $(p - 1)(q - 1)$, which shows that the inverse of d modulo $(p - 1)(q - 1)$ is $s \pmod{(p - 1)(q - 1)}$.] Knowing d lets us decrypt messages sent using our key. However, no method is known to decrypt messages that is not based on finding a factorization of n , or that does not also lead to the factorization of n .

Factorization is believed to be a difficult problem, as opposed to finding large primes p and q , which can be done quickly. The most efficient factorization methods known (as of 2010) require billions of years to factor 400-digit integers. Consequently, when p and q are 200-digit primes, it is believed that messages encrypted using $n = pq$ as the modulus cannot be found in a reasonable time unless the primes p and q are known.

Although no polynomial-time algorithm is known for factoring large integers, active research is under way to find new ways to efficiently factor integers. Integers that were thought, as recently as several years ago, to be far too large to be factored in a reasonable amount of time can now be factored routinely. Integers with more than 150 digits, as well as some with more than 200 digits, have been factored using team efforts. When new factorization techniques are found,

it will be necessary to use larger primes to ensure secrecy of messages. Unfortunately, messages that were considered secure earlier can be saved and subsequently decrypted by unintended recipients when it becomes feasible to factor the $n = pq$ in the key used for RSA encryption.

The RSA method is now widely used. However, the most commonly used cryptosystems are private key cryptosystems. The use of public key cryptography, via the RSA system, is growing. Nevertheless, there are applications that use both private key and public key systems. For example, a public key cryptosystem, such as RSA, can be used to distribute private keys to pairs of individuals when they wish to communicate. These people then use a private key system for encryption and decryption of messages.

Cryptographic Protocols

So far we have shown how cryptography can be used to make messages secure. However, there are many other important applications of cryptography. Among these applications are **cryptographic protocols**, which are exchanges of messages carried out by two or more parties to achieve a particular security goal. In particular, we will show how cryptography can be used to allow two people to exchange a secret key over an insecure communication channel. We will also show how cryptography can be used to send signed secret messages so that the recipient can be sure that the message came from the purported sender. We refer the reader to [St05] for thorough discussions of a variety of cryptographic protocols.

KEY EXCHANGE We now discuss a protocol that two parties can use to exchange a secret key over an insecure communications channel without having shared any information in the past. Generating a key that two parties can share is important for many applications of cryptography. For example, for two people to send secure messages to each other using a private key cryptosystem they need to share a common key. The protocol we will describe is known as the **Diffie-Hellman key agreement protocol**, after Whitfield Diffie and Martin Hellman, who described it in 1976. However, this protocol was invented in 1974 by Malcolm Williamson in secret work at the British GCHQ. It was not until 1997 that his discovery was made public.

Suppose that Alice and Bob want to share a common key. The protocol follows these steps, where the computations are done in \mathbf{Z}_p .

- (1) Alice and Bob agree to use a prime p and a primitive root a of p .
- (2) Alice chooses a secret integer k_1 and sends $a^{k_1} \pmod p$ to Bob.
- (3) Bob chooses a secret integer k_2 and sends $a^{k_2} \pmod p$ to Alice.
- (4) Alice computes $(a^{k_2})^{k_1} \pmod p$.
- (5) Bob computes $(a^{k_1})^{k_2} \pmod p$.

At the end of this protocol, Alice and Bob have computed their shared key, namely

$$(a^{k_2})^{k_1} \pmod p = (a^{k_1})^{k_2} \pmod p.$$

To analyze the security of this protocol, note that the messages sent in steps (1), (2), and (3) are not assumed to be sent securely. We can even assume that these communications were in the clear and that their contents are public information. So, p , a , $a^{k_1} \pmod p$, and $a^{k_2} \pmod p$ are assumed to be public information. The protocol ensures that k_1 , k_2 , and the common key $(a^{k_2})^{k_1} \pmod p = (a^{k_1})^{k_2} \pmod p$ are kept secret. To find the secret information from this public information requires that an adversary solves instances of the discrete logarithm problem,

because the adversary would need to find k_1 and k_2 from $a^{k_1} \pmod p$ and $a^{k_2} \pmod p$, respectively. Furthermore, no other method is known for finding the shared key using just the public information. We have remarked that this is thought to be computationally infeasible when p and a are sufficiently large. With the computing power available now, this system is considered unbreakable when p has more than 300 decimal digits and k_1 and k_2 have more than 100 decimal digits each.

DIGITAL SIGNATURES Not only can cryptography be used to secure the confidentiality of a message, but it also can be used so that the recipient of the message knows that it came from the person they think it came from. We first show how a message can be sent so that a recipient of the message will be sure that the message came from the purported sender of the message. In particular, we can show how this can be accomplished using the RSA cryptosystem to apply a **digital signature** to a message.

Suppose that Alice's RSA public key is (n, e) and her private key is d . Alice encrypts a plaintext message x using the encryption function $E_{(n,e)}(x) = x^e \pmod n$. She decrypts a ciphertext message y using the decryption function $D_{(n,e)} = x^d \pmod n$. Alice wants to send the message M so that everyone who receives the message knows that it came from her. Just as in RSA encryption, she translates the letters into their numerical equivalents and splits the resulting string into blocks m_1, m_2, \dots, m_k such that each block is the same size which is as large as possible so that $0 \leq m_i \leq n$ for $i = 1, 2, \dots, k$. She then applies her *decryption function* $D_{(n,e)}$ to each block, obtaining $D_{n,e}(m_i), i = 1, 2, \dots, k$. She sends the result to all intended recipients of the message.

When a recipient receives her message, they apply Alice's encryption function $E_{(n,e)}$ to each block, which everyone has available because Alice's key (n, e) is public information. The result is the original plaintext block because $E_{(n,e)}(D_{(n,e)}(x)) = x$. So, Alice can send her message to as many people as she wants and by signing it in this way, every recipient can be sure it came from Alice. Example 10 illustrates this protocol.



EXAMPLE 10 Suppose Alice's public RSA cryptosystem key is the same as in Example 8. That is, $n = 43 \cdot 59 = 2537$ and $e = 13$. Her decryption key is $d = 937$, as described in Example 9. She wants to send the message "MEET AT NOON" to her friends so that they are sure it came from her. What should she send?

Solution: Alice first translates the message into blocks of digits, obtaining 1204 0419 0019 1314 1413 (as the reader should verify). She then applies her decryption transformation $D_{(2537,13)}(x) = x^{937} \pmod{2537}$ to each block. Using fast modular exponentiation (with the help of a computational aid), she finds that $1204^{937} \pmod{2537} = 817$, $419^{937} \pmod{2537} = 555$, $19^{937} \pmod{2537} = 1310$, $1314^{937} \pmod{2537} = 2173$, and $1413^{937} \pmod{2537} = 1026$.

So, the message she sends, split into blocks, is 0817 0555 1310 2173 1026. When one of her friends gets this message, they apply her encryption transformation $E_{(2537,13)}$ to each block. When they do this, they obtain the blocks of digits of the original message which they translate back to English letters. 

We have shown that signed messages can be sent using the RSA cryptosystem. We can extend this by sending signed secret messages. To do this, the sender applies RSA encryption using the publicly known encryption key of an intended recipient to each block that was encrypted using sender's decryption transformation. The recipient then first applies his private decryption transformation and then the sender's public encryption transformation. (Exercise 32 asks for this protocol to be carried out.)

Exercises

1. Encrypt the message DO NOT PASS GO by translating the letters into numbers, applying the given encryption function, and then translating the numbers back into letters.
 - $f(p) = (p + 3) \text{ mod } 26$ (the Caesar cipher)
 - $f(p) = (p + 13) \text{ mod } 26$
 - $f(p) = (3p + 7) \text{ mod } 26$
 2. Encrypt the message STOP POLLUTION by translating the letters into numbers, applying the given encryption function, and then translating the numbers back into letters.
 - $f(p) = (p + 4) \text{ mod } 26$
 - $f(p) = (p + 21) \text{ mod } 26$
 - $f(p) = (17p + 22) \text{ mod } 26$
 3. Encrypt the message WATCH YOUR STEP by translating the letters into numbers, applying the given encryption function, and then translating the numbers back into letters.
 - $f(p) = (p + 14) \text{ mod } 26$
 - $f(p) = (14p + 21) \text{ mod } 26$
 - $f(p) = (-7p + 1) \text{ mod } 26$
 4. Decrypt these messages that were encrypted using the Caesar cipher.
 - EOXH MHDQV
 - WHVW WRGDB
 - HDW GLP VXP
 5. Decrypt these messages encrypted using the shift cipher $f(p) = (p + 10) \text{ mod } 26$.
 - CEBBOXNOB XYG
 - LO WI PBSOXN
 - DSWO PYB PEX
 6. Suppose that when a long string of text is encrypted using a shift cipher $f(p) = (p + k) \text{ mod } 26$, the most common letter in the ciphertext is X. What is the most likely value for k assuming that the distribution of letters in the text is typical of English text?
 7. Suppose that when a string of English text is encrypted using a shift cipher $f(p) = (p + k) \text{ mod } 26$, the resulting ciphertext is DY CVOOZ ZOBMRKXMO DY NBOKW. What was the original plaintext string?
 8. Suppose that the ciphertext DVE CFMV KF NFEUVI, REU KYRK ZJ KYV JVVU FW JTZVETV was produced by encrypting a plaintext message using a shift cipher. What is the original plaintext?
 9. Suppose that the ciphertext ERC WYJJMGMIRXPC EHZERGIH XIGLRSPSKC MW MRHMWXM-RKYMWLEFPI JVSQ QEKG was produced by encrypting a plaintext message using a shift cipher. What is the original plaintext?
 10. Determine whether there is a key for which the enciphering function for the shift cipher is the same as the deciphering function.
 11. What is the decryption function for an affine cipher if the encryption function is $c = (15p + 13) \text{ mod } 26$?
 - *12. Find all pairs of integers keys (a, b) for affine ciphers for which the encryption function $c = (ap + b) \text{ mod } 26$ is the same as the corresponding decryption function.
 13. Suppose that the most common letter and the second most common letter in a long ciphertext produced by encrypting a plaintext using an affine cipher $f(p) = (ap + b) \text{ mod } 26$ are Z and J, respectively. What are the most likely values of a and b ?
 14. Encrypt the message GRIZZLY BEARS using blocks of five letters and the transposition cipher based on the permutation of $\{1, 2, 3, 4, 5\}$ with $\sigma(1) = 3, \sigma(2) = 5, \sigma(3) = 1, \sigma(4) = 2$, and $\sigma(5) = 4$. For this exercise, use the letter X as many times as necessary to fill out the final block of fewer than five letters.
 15. Decrypt the message EABW EFRO ATMR ASIN which is the ciphertext produced by encrypting a plaintext message using the transposition cipher with blocks of four letters and the permutation σ of $\{1, 2, 3, 4\}$ defined by $\sigma(1) = 3, \sigma(2) = 1, \sigma(3) = 4$, and $\sigma(4) = 2$.
 - *16. Suppose that you know that a ciphertext was produced by encrypting a plaintext message with a transposition cipher. How might you go about breaking it?
 17. Suppose you have intercepted a ciphertext message and when you determine the frequencies of letters in this message, you find the frequencies are similar to the frequency of letters in English text. Which type of cipher do you suspect was used?
- The **Vigenère cipher** is a block cipher, with a key that is a string of letters with numerical equivalents $k_1 k_2 \dots k_m$, where $k_i \in \mathbf{Z}_{26}$ for $i = 1, 2, \dots, m$. Suppose that the numerical equivalents of the letters of a plaintext block are $p_1 p_2 \dots p_m$. The corresponding numerical ciphertext block is $(p_1 + k_1) \text{ mod } 26 (p_2 + k_2) \text{ mod } 26 \dots (p_m + k_m) \text{ mod } 26$. Finally, we translate back to letters. For example, suppose that the key string is RED, with numerical equivalents 17 4 3. Then, the plaintext ORANGE, with numerical equivalents 14 17 00 13 06 04, is encrypted by first splitting it into two blocks 14 17 00 and 13 06 04. Then, in each block we shift the first letter by 17, the second by 4, and the third by 3. We obtain 5 21 03 and 04 10 07. The ciphertext is FVDEKH.
18. Use the Vigenère cipher with key BLUE to encrypt the message SNOWFALL.
 19. The ciphertext OIKYWVHBX was produced by encrypting a plaintext message using the Vigenère cipher with key HOT. What is the plaintext message?

- 20.** Express the Vigenère cipher as a cryptosystem.

To break a Vigenère cipher by recovering a plaintext message from the ciphertext message without having the key, the first step is to figure out the length of the key string. The second step is to figure out each character of the key string by determining the corresponding shift. Exercises 21 and 22 deal with these two aspects.

- 21.** Suppose that when a long string of text is encrypted using a Vigenère cipher, the same string is found in the ciphertext starting at several different positions. Explain how this information can be used to help determine the length of the key.

- 22.** Once the length of the key string of a Vigenère cipher is known, explain how to determine each of its characters. Assume that the plaintext is long enough so that the frequency of its letters is reasonably close to the frequency of letters in typical English text.

- *23.** Show that we can easily factor n when we know that n is the product of two primes, p and q , and we know the value of $(p - 1)(q - 1)$.

In Exercises 24–27 first express your answers without computing modular exponentiations. Then use a computational aid to complete these computations.

- 24.** Encrypt the message ATTACK using the RSA system with $n = 43 \cdot 59$ and $e = 13$, translating each letter into integers and grouping together pairs of integers, as done in Example 8.

- 25.** Encrypt the message UPLOAD using the RSA system with $n = 53 \cdot 61$ and $e = 17$, translating each letter into integers and grouping together pairs of integers, as done in Example 8.

- 26.** What is the original message encrypted using the RSA system with $n = 53 \cdot 61$ and $e = 17$ if the encrypted message is 3185 2038 2460 2550? (To decrypt, first find the decryption exponent d , which is the inverse of $e = 17$ modulo $52 \cdot 60$.)

- 27.** What is the original message encrypted using the RSA system with $n = 43 \cdot 59$ and $e = 13$ if the encrypted message is 0667 1947 0671? (To decrypt, first find the decryption exponent d which is the inverse of $e = 13$ modulo $42 \cdot 58$.)

- *28.** Suppose that (n, e) is an RSA encryption key, with $n = pq$ where p and q are large primes and $\gcd(e, (p - 1)(q - 1)) = 1$. Furthermore, suppose that d is an inverse of e modulo $(p - 1)(q - 1)$. Suppose that $C \equiv M^e \pmod{pq}$. In the text we showed that RSA decryption, that is, the congruence $C^d \equiv M \pmod{pq}$ holds when $\gcd(M, pq) = 1$. Show that this decryption congruence also holds when $\gcd(M, pq) > 1$. [Hint: Use congruences modulo p and modulo q and apply the Chinese remainder theorem.]

- 29.** Describe the steps that Alice and Bob follow when they use the Diffie-Hellman key exchange protocol to generate a shared key. Assume that they use the prime $p = 23$ and take $a = 5$, which is a primitive root of 23, and that Alice selects $k_1 = 8$ and Bob selects $k_2 = 5$. (You may want to use some computational aid.)

- 30.** Describe the steps that Alice and Bob follow when they use the Diffie-Hellman key exchange protocol to generate a shared key. Assume that they use the prime $p = 101$ and take $a = 2$, which is a primitive root of 101, and that Alice selects $k_1 = 7$ and Bob selects $k_2 = 9$. (You may want to use some computational aid.)

In Exercises 31–32 suppose that Alice and Bob have these public keys and corresponding private keys: $(n_{\text{Alice}}, e_{\text{Alice}}) = (2867, 7) = (61 \cdot 47, 7)$, $d_{\text{Alice}} = 1183$ and $(n_{\text{Bob}}, e_{\text{Bob}}) = (3127, 21) = (59 \cdot 53, 21)$, $d_{\text{Bob}} = 1149$. First express your answers without carrying out the calculations. Then, using a computational aid, if available, perform the calculation to get the numerical answers.

- 31.** Alice wants to send to all her friends, including Bob, the message “SELL EVERYTHING” so that he knows that she sent it. What should she send to her friends, assuming she signs the message using the RSA cryptosystem.

- 32.** Alice wants to send to Bob the message “BUY NOW” so that he knows that she sent it and so that only Bob can read it. What should she send to Bob, assuming she signs the message and then encrypts it using Bob’s public key?

- 33.** We describe a basic key exchange protocol using private key cryptography upon which more sophisticated protocols for key exchange are based. Encryption within the protocol is done using a private key cryptosystem (such as AES) that is considered secure. The protocol involves three parties, Alice and Bob, who wish to exchange a key, and a trusted third party Cathy. Assume that Alice has a secret key k_{Alice} that only she and Cathy know, and Bob has a secret key k_{Bob} which only he and Cathy know. The protocol has three steps:

(i) Alice sends the trusted third party Cathy the message “request a shared key with Bob” encrypted using Alice’s key k_{Alice} .

(ii) Cathy sends back to Alice a key $k_{\text{Alice}, \text{Bob}}$, which she generates, encrypted using the key k_{Alice} , followed by this same key $k_{\text{Alice}, \text{Bob}}$, encrypted using Bob’s key, k_{Bob} .

(iii) Alice sends to Bob the key $k_{\text{Alice}, \text{Bob}}$ encrypted using k_{Bob} , known only to Bob and to Cathy.

Explain why this protocol allows Alice and Bob to share the secret key $k_{\text{Alice}, \text{Bob}}$, known only to them and to Cathy.

Key Terms and Results

TERMS

- $a | b$ (a divides b):** there is an integer c such that $b = ac$
- a and b are congruent modulo m :** m divides $a - b$
- modular arithmetic:** arithmetic done modulo an integer $m \geq 2$
- prime:** an integer greater than 1 with exactly two positive integer divisors
- composite:** an integer greater than 1 that is not prime
- Mersenne prime:** a prime of the form $2^p - 1$, where p is prime
- $\gcd(a, b)$ (greatest common divisor of a and b):** the largest integer that divides both a and b
- relatively prime integers:** integers a and b such that $\gcd(a, b) = 1$
- pairwise relatively prime integers:** a set of integers with the property that every pair of these integers is relatively prime
- $\text{lcm}(a, b)$ (least common multiple of a and b):** the smallest positive integer that is divisible by both a and b
- $a \text{ mod } b$:** the remainder when the integer a is divided by the positive integer b
- $a \equiv b \pmod{m}$ (a is congruent to b modulo m):** $a - b$ is divisible by m
- $n = (a_k a_{k-1} \dots a_1 a_0)_b$:** the base b representation of n
- binary representation:** the base 2 representation of an integer
- octal representation:** the base 8 representation of an integer
- hexadecimal representation:** the base 16 representation of an integer
- linear combination of a and b with integer coefficients:** an expression of the form $sa + tb$, where s and t are integers
- Bézout coefficients of a and b :** integers s and t such that the Bézout identity $sa + tb = \gcd(a, b)$ holds
- inverse of a modulo m :** an integer \bar{a} such that $\bar{a}a \equiv 1 \pmod{m}$
- linear congruence:** a congruence of the form $ax \equiv b \pmod{m}$, where x is an integer variable
- pseudoprime to the base b :** a composite integer n such that $b^{n-1} \equiv 1 \pmod{n}$
- Carmichael number:** a composite integer n such that n is a pseudoprime to the base b for all positive integers b with $\gcd(b, n) = 1$
- primitive root of a prime p :** an integer r in \mathbb{Z}_p such that every integer not divisible by p is congruent modulo p to a power of r
- discrete logarithm of a to the base r modulo p :** the integer e with $0 \leq e \leq p - 1$ such that $r^e \equiv a \pmod{p}$
- encryption:** the process of making a message secret
- decryption:** the process of returning a secret message to its original form
- encryption key:** a value that determines which of a family of encryption functions is to be used
- shift cipher:** a cipher that encrypts the plaintext letter p as $(p + k) \pmod{m}$ for an integer k
- affine cipher:** a cipher that encrypts the plaintext letter p as $(ap + b) \pmod{m}$ for integers a and b with $\gcd(a, 26) = 1$
- character cipher:** a cipher that encrypts characters one by one
- block cipher:** a cipher that encrypts blocks of characters of a fixed size

cryptanalysis: the process of recovering the plaintext from ciphertext without knowledge of the encryption method, or with knowledge of the encryption method, but not the key

cryptosystem: a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ where \mathcal{P} is the set of plaintext messages, \mathcal{C} is the set of ciphertext messages, \mathcal{K} is the set of keys, \mathcal{E} is the set of encryption functions, and \mathcal{D} is the set of decryption functions

private key encryption: encryption where both encryption keys and decryption keys must be kept secret

public key encryption: encryption where encryption keys are public knowledge, but decryption keys are kept secret

RSA cryptosystem: the cryptosystem where \mathcal{P} and \mathcal{C} are both \mathbb{Z}_{26} , \mathcal{K} is the set of pairs $k = (n, e)$ where $n = pq$ where p and q are large primes and e is a positive integer, $E_k(p) = p^e \pmod{n}$, and $D_k(c) = c^d \pmod{n}$ where d is the inverse of e modulo $(p-1)(q-1)$

key exchange protocol: a protocol used for two parties to generate a shared key

digital signature: a method that a recipient can use to determine that the purported sender of a message actually sent the message

RESULTS

division algorithm: Let a and d be integers with d positive. Then there are unique integers q and r with $0 \leq r < d$ such that $a = dq + r$.

Let b be an integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form $n = a_kb^k + a_{k-1}b^{k-1} + \dots + a_1b + a_0$.

The algorithm for finding the base b expansion of an integer (see Algorithm 1 in Section 4.2)

The conventional algorithms for addition and multiplication of integers (given in Section 4.2)

The modular exponentiation algorithm (see Algorithm 5 in Section 4.2)

Euclidean algorithm: for finding greatest common divisors by successively using the division algorithm (see Algorithm 1 in Section 4.3)

Bézout's theorem: If a and b are positive integers, then $\gcd(a, b)$ is a linear combination of a and b .

sieve of Eratosthenes: A procedure for finding all primes not exceeding a specified number n , described in Section 4.3

fundamental theorem of arithmetic: Every positive integer can be written uniquely as the product of primes, where the prime factors are written in order of increasing size.

If a and b are positive integers, then $ab = \gcd(a, b) \cdot \text{lcm}(a, b)$.

If m is a positive integer and $\gcd(a, m) = 1$, then a has a unique inverse modulo m .

Chinese remainder theorem: A system of linear congruences modulo pairwise relatively prime integers has a unique solution modulo the product of these moduli.

Fermat's little theorem: If p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p}$.

Review Questions

1. Find $210 \text{ div } 17$ and $210 \text{ mod } 17$.
2. a) Define what it means for a and b to be congruent modulo 7.
b) Which pairs of the integers $-11, -8, -7, -1, 0, 3$, and 17 are congruent modulo 7?
c) Show that if a and b are congruent modulo 7, then $10a + 13$ and $-4b + 20$ are also congruent modulo 7.
3. Show that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$.
4. Describe a procedure for converting decimal (base 10) expansions of integers into hexadecimal expansions.
5. Convert $(1101\ 1001\ 0101\ 1011)_2$ to octal and hexadecimal representations.
6. Convert $(7206)_8$ and $(A0EB)_{16}$ to a binary representation.
7. State the fundamental theorem of arithmetic.
8. a) Describe a procedure for finding the prime factorization of an integer.
b) Use this procedure to find the prime factorization of 80,707.
9. a) Define the greatest common divisor of two integers.
b) Describe at least three different ways to find the greatest common divisor of two integers. When does each method work best?
c) Find the greatest common divisor of 1,234,567 and 7,654,321.
d) Find the greatest common divisor of $2^3 3^5 5^7 7^9 11$ and $2^9 3^7 5^5 7^3 13$.
10. a) How can you find a linear combination (with integer coefficients) of two integers that equals their greatest common divisor?
- b) Express $\gcd(84, 119)$ as a linear combination of 84 and 119.
11. a) What does it mean for \bar{a} to be an inverse of a modulo m ?
b) How can you find an inverse of a modulo m when m is a positive integer and $\gcd(a, m) = 1$?
c) Find an inverse of 7 modulo 19.
12. a) How can an inverse of a modulo m be used to solve the congruence $ax \equiv b \pmod{m}$ when $\gcd(a, m) = 1$?
b) Solve the linear congruence $7x \equiv 13 \pmod{19}$.
13. a) State the Chinese remainder theorem.
b) Find the solutions to the system $x \equiv 1 \pmod{4}$, $x \equiv 2 \pmod{5}$, and $x \equiv 3 \pmod{7}$.
14. Suppose that $2^{n-1} \equiv 1 \pmod{n}$. Is n necessarily prime?
15. Use Fermat's little theorem to evaluate $9^{200} \pmod{19}$.
16. Explain how the check digit is found for a 10-digit ISBN.
17. Encrypt the message APPLES AND ORANGES using a shift cipher with key $k = 13$.
18. a) What is the difference between a public key and a private key cryptosystem?
b) Explain why using shift ciphers is a private key system.
c) Explain why the RSA cryptosystem is a public key system.
19. Explain how encryption and decryption are done in the RSA cryptosystem.
20. Describe how two parties can share a secret key using the Diffie-Hellman key exchange protocol.

Supplementary Exercises

1. The odometer on a car goes up to 100,000 miles. The present owner of a car bought it when the odometer read 43,179 miles. He now wants to sell it; when you examine the car for possible purchase, you notice that the odometer reads 89,697 miles. What can you conclude about how many miles he drove the car, assuming that the odometer always worked correctly?
2. a) Explain why $n \text{ div } 7$ equals the number of complete weeks in n days.
b) Explain why $n \text{ div } 24$ equals the number of complete days in n hours.
3. Find four numbers congruent to 5 modulo 17.
4. Show that if a and d are positive integers, then there are integers q and r such that $a = dq + r$ where $-d/2 < r \leq d/2$.
- *5. Show that if $ac \equiv bc \pmod{m}$, where a, b, c , and m are integers with $m > 2$, and $d = \gcd(m, c)$, then $a \equiv b \pmod{m/d}$.
6. Show that the sum of the squares of two odd integers cannot be the square of an integer.
7. Show that if $n^2 + 1$ is a perfect square, where n is an integer, then n is even.
8. Prove that there are no solutions in integers x and y to the equation $x^2 - 5y^2 = 2$. [Hint: Consider this equation modulo 5.]
9. Develop a test for divisibility of a positive integer n by 8 based on the binary expansion of n .
10. Develop a test for divisibility of a positive integer n by 3 based on the binary expansion of n .
11. Devise an algorithm for guessing a number between 1 and $2^n - 1$ by successively guessing each bit in its binary expansion.
12. Determine the complexity, in terms of the number of guesses, needed to determine a number between 1 and $2^n - 1$ by successively guessing the bits in its binary expansion.
13. Show that an integer is divisible by 9 if and only if the sum of its decimal digits is divisible by 9.

- **14.** Show that if a and b are positive irrational numbers such that $1/a + 1/b = 1$, then every positive integer can be uniquely expressed as either $\lfloor ka \rfloor$ or $\lfloor kb \rfloor$ for some positive integer k .
- 15.** Prove there are infinitely many primes by showing that $Q_n = n! + 1$ must have a prime factor greater than n whenever n is a positive integer.
- 16.** Find a positive integer n for which $Q_n = n! + 1$ is not prime.
- 17.** Use Dirichlet's theorem, which states there are infinitely many primes in every arithmetic progression $ak + b$ where $\gcd(a, b) = 1$, to show that there are infinitely many primes that have a decimal expansion ending with a 1.
- 18.** Prove that if n is a positive integer such that the sum of the divisors of n is $n + 1$, then n is prime.
- *19.** Show that every integer greater than 11 is the sum of two composite integers.
- 20.** Find the five smallest consecutive composite integers.
- 21.** Show that Goldbach's conjecture, which states that every even integer greater than 2 is the sum of two primes, is equivalent to the statement that every integer greater than 5 is the sum of three primes.
- 22.** Find an arithmetic progression of length six beginning with 7 that contains only primes.
- *23.** Prove that if $f(x)$ is a nonconstant polynomial with integer coefficients, then there is an integer y such that $f(y)$ is composite. [Hint: Assume that $f(x_0) = p$ is prime. Show that p divides $f(x_0 + kp)$ for all integers k . Obtain a contradiction of the fact that a polynomial of degree n , where $n > 1$, takes on each value at most n times.]
- *24.** How many zeros are at the end of the binary expansion of $100_{10}!$?
- 25.** Use the Euclidean algorithm to find the greatest common divisor of 10,223 and 33,341.
- 26.** How many divisions are required to find $\gcd(144, 233)$ using the Euclidean algorithm?
- 27.** Find $\gcd(2n + 1, 3n + 2)$, where n is a positive integer. [Hint: Use the Euclidean algorithm.]
- 28.** **a)** Show that if a and b are positive integers with $a \geq b$, then $\gcd(a, b) = a$ if $a = b$, $\gcd(a, b) = 2 \gcd(a/2, b/2)$ if a and b are even, $\gcd(a, b) = \gcd(a/2, b)$ if a is even and b is odd, and $\gcd(a, b) = \gcd(a - b, b)$ if both a and b are odd.
b) Explain how to use (a) to construct an algorithm for computing the greatest common divisor of two positive integers that uses only comparisons, subtractions, and shifts of binary expansions, without using any divisions.
c) Find $\gcd(1202, 4848)$ using this algorithm.
- 29.** Adapt the proof that there are infinitely many primes (Theorem 3 in Section 4.3) to show that there are infinitely many primes in the arithmetic progression $6k + 5, k = 1, 2, \dots$
- 30.** Explain why you cannot directly adapt the proof that there are infinitely many primes (Theorem 3 in Section 4.3) to show that there are infinitely many primes in the arithmetic progression $3k + 1, k = 1, 2, \dots$
- 31.** Explain why you cannot directly adapt the proof that there are infinitely many primes (Theorem 3 in Section 4.3) to show that there are infinitely many primes in the arithmetic progression $4k + 1, k = 1, 2, \dots$
- 32.** Show that if the smallest prime factor p of the positive integer n is larger than $\sqrt[3]{n}$, then n/p is prime or equal to 1.
- A set of integers is called **mutually relatively prime** if the greatest common divisor of these integers is 1.
- 33.** Determine whether the integers in each of these sets are mutually relatively prime.
a) 8, 10, 12 **b)** 12, 15, 25
c) 15, 21, 28 **d)** 21, 24, 28, 32
- 34.** Find a set of four mutually relatively prime integers such that no two of them are relatively prime.
- *35.** For which positive integers n is $n^4 + 4^n$ prime?
- 36.** Show that the system of congruences $x \equiv 2 \pmod{6}$ and $x \equiv 3 \pmod{9}$ has no solutions.
- 37.** Find all solutions of the system of congruences $x \equiv 4 \pmod{6}$ and $x \equiv 13 \pmod{15}$.
- *38.** **a)** Show that the system of congruences $x \equiv a_1 \pmod{m_1}$ and $x \equiv a_2 \pmod{m_2}$, where a_1, a_2, m_1 , and m_2 are integers with $m_1 > 0$ and $m_2 > 0$, has a solution if and only if $\gcd(m_1, m_2) \mid a_1 - a_2$.
b) Show that if the system in part (a) has a solution, then it is unique modulo $\text{lcm}(m_1, m_2)$.
- 39.** Prove that 30 divides $n^9 - n$ for every nonnegative integer n .
- 40.** Prove that $n^{12} - 1$ is divisible by 35 for every integer n for which $\gcd(n, 35) = 1$.
- 41.** Show that if p and q are distinct prime numbers, then $p^{q-1} + q^{p-1} \equiv 1 \pmod{pq}$.
- The check digit a_{13} for an ISBN-13 with initial digits $a_1 a_2 \dots a_{12}$ is determined by the congruence $(a_1 + a_3 + \dots + a_{13}) + 3(a_2 + a_4 + \dots + a_{12}) \equiv 0 \pmod{10}$.
- 42.** Determine whether each of these 13-digit numbers is a valid ISBN-13.
a) 978-0-073-20679-1
b) 978-0-45424-521-1
c) 978-3-16-148410-0
d) 978-0-201-10179-9
- 43.** Show that the check digit of an ISBN-13 can always detect a single error.
- 44.** Show that there are transpositions of two digits that are not detected by an ISBN-13.
- A **routing transit number (RTN)** is a bank code used in the United States which appears on the bottom of checks. The most common form of an RTN has nine digits, where the last digit is a check digit. If $d_1 d_2 \dots d_9$ is a valid RTN,

the congruence $3(d_1 + d_4 + d_7) + 7(d_2 + d_5 + d_8) + (d_3 + d_6 + d_9) \equiv 0 \pmod{10}$ must hold.

45. Show that if $d_1 d_2 \dots d_9$ is a valid RTN, then $d_9 = 7(d_1 + d_4 + d_7) + 3(d_2 + d_5 + d_8) + 9(d_3 + d_6) \pmod{10}$. Furthermore, use this formula to find the check digit that follows the eight digits 11100002 in a valid RTN.
46. Show that the check digit of an RTN can detect all single errors and determine which transposition errors an RTN check digit can catch and which ones it cannot catch.
47. The encrypted version of a message is LJMKG MG-MXF QEXMW. If it was encrypted using the affine cipher $f(p) = (7p + 10) \pmod{26}$, what was the original message?

Autokey ciphers are ciphers where the n th letter of the plaintext is shifted by the numerical equivalent of the n th letter of a keystream. The keystream begins with a seed letter; its subsequent letters are constructed using either the plaintext or the ciphertext. When the plaintext is used, each character of the

keystream, after the first, is the previous letter of the plaintext. When the ciphertext is used, each subsequent character of the keystream, after the first, is the previous letter of the ciphertext computed so far. In both cases, plaintext letters are encrypted by shifting each character by the numerical equivalent of the corresponding keystream letter.

48. Use the autokey cipher to encrypt the message NOW IS THE TIME TO DECIDE (ignoring spaces) using
 - a) the keystream with seed X followed by letters of the plaintext.
 - b) the keystream with seed X followed by letters of the ciphertext.
49. Use the autokey cipher to encrypt the message THE DREAM OF REASON (ignoring spaces) using
 - a) the keystream with seed X followed by letters of the plaintext.
 - b) the keystream with seed X followed by letters of the ciphertext.

Computer Projects

Write programs with these inputs and outputs.

1. Given integers n and b , each greater than 1, find the base b expansion of this integer.
2. Given the positive integers a , b , and m with $m > 1$, find $a^b \pmod{m}$.
3. Given a positive integer, find the Cantor expansion of this integer (see the preamble to Exercise 48 of Section 4.2).
4. Given a positive integer, determine whether it is prime using trial division.
5. Given a positive integer, find the prime factorization of this integer.
6. Given two positive integers, find their greatest common divisor using the Euclidean algorithm.
7. Given two positive integers, find their least common multiple.
8. Given positive integers a and b , find Bézout coefficients s and t of a and b .
9. Given relatively prime positive integers a and b , find an inverse of a modulo b .
10. Given n linear congruences modulo pairwise relatively prime moduli, find the simultaneous solution of these congruences modulo the product of these moduli.
11. Given a positive integer N , a modulus m , a multiplier a , an increment c , and a seed x_0 , where $0 \leq a < m$, $0 \leq c < m$, and $0 \leq x_0 < m$, generate the sequence of N pseudo-random numbers using the linear congruential generator $x_{n+1} = (ax_n + c) \pmod{m}$.
12. Given a set of identification numbers, use a hash function to assign them to memory locations where there are k memory locations.
13. Compute the check digit when given the first nine digits of an ISBN-10.
14. Given a message and a positive integer k less than 26, encrypt this message using the shift cipher with key k ; and given a message encrypted using a shift cipher with key k , decrypt this message.
15. Given a message and positive integers a and b less than 26 with $\gcd(a, 26) = 1$, encrypt this message using an affine cipher with key (a, b) ; and given a message encrypted using the affine cipher with key (a, b) , decrypt this message, by first finding the decryption key and then applying the appropriate decryption transformation.
16. Find the original plaintext message from the ciphertext message produced by encrypting the plaintext message using a shift cipher. Do this using a frequency count of letters in the ciphertext.
- *17. Construct a valid RSA encryption key by finding two primes p and q with 200 digits each and an integer $e > 1$ relatively prime to $(p - 1)(q - 1)$.
18. Given a message and an integer $n = pq$ where p and q are odd primes and an integer $e > 1$ relatively prime to $(p - 1)(q - 1)$, encrypt the message using the RSA cryptosystem with key (n, e) .
19. Given a valid RSA key (n, e) , and the primes p and q with $n = pq$, find the associated decryption key d .
20. Given a message encrypted using the RSA cryptosystem with key (n, e) and the associated decryption key d , decrypt this message.
21. Generate a shared key using the Diffie-Hellman key exchange protocol.
22. Given the RSA public and private keys of two parties, send a signed secret message from one of the parties to the other.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Determine whether $2^p - 1$ is prime for each of the primes not exceeding 100.
2. Test a range of large Mersenne numbers $2^p - 1$ to determine whether they are prime. (You may want to use software from the GIMPS project.)
3. Determine whether $Q_n = p_1 p_2 \cdots p_n + 1$ is prime where p_1, p_2, \dots, p_n are the n smallest primes, for as many positive integer n as possible.
4. Look for polynomials in one variables whose values at long runs of consecutive integers are all primes.
5. Find as many primes of the form $n^2 + 1$ where n is a positive integer as you can. It is not known whether there are infinitely many such primes.
6. Find 10 different primes each with 100 digits.
7. How many primes are there less than 1,000,000, less than 10,000,000, and less than 100,000,000? Can you propose an estimate for the number of primes less than x where x is a positive integer?
8. Find a prime factor of each of 10 different 20-digit odd integers, selected at random. Keep track of how long it takes to find a factor of each of these integers. Do the same thing for 10 different 30-digit odd integers, 10 different 40-digit odd integers, and so on, continuing as long as possible.
9. Find all pseudoprimes to the base 2 that do not exceed 10,000.

Writing Projects

Respond to these with essays using outside sources.

1. Describe the Lucas–Lehmer test for determining whether a Mersenne number is prime. Discuss the progress of the GIMPS project in finding Mersenne primes using this test.
2. Explain how probabilistic primality tests are used in practice to produce extremely large numbers that are almost certainly prime. Do such tests have any potential drawbacks?
3. The question of whether there are infinitely many Carmichael numbers was solved recently after being open for more than 75 years. Describe the ingredients that went into the proof that there are infinitely many such numbers.
4. Summarize the current status of factoring algorithms in terms of their complexity and the size of numbers that can currently be factored. When do you think that it will be feasible to factor 200-digit numbers?
5. Describe the algorithms that are actually used by modern computers to add, subtract, multiply, and divide positive integers.
6. Describe the history of the Chinese remainder theorem. Describe some of the relevant problems posed in Chinese and Hindu writings and how the Chinese remainder theorem applies to them.
7. When are the numbers of a sequence truly random numbers, and not pseudorandom? What shortcomings have been observed in simulations and experiments in which pseudorandom numbers have been used? What are the properties that pseudorandom numbers can have that random numbers should not have?
8. Explain how a check digit is found for an International Bank Account Number (IBAN) and discuss the types of errors that can be found using this check digit.
9. Describe the Luhn algorithm for finding the check digit of a credit card number and discuss the types of errors that can be found using this check digit.
10. Show how a congruence can be used to tell the day of the week for any given date.
11. Describe how public key cryptography is being applied. Are the ways it is applied secure given the status of factoring algorithms? Will information kept secure using public key cryptography become insecure in the future?
12. Describe how public key cryptography can be used to produce signed secret messages so that the recipient is relatively sure the message was sent by the person expected to have sent it.
13. Describe the Rabin public key cryptosystem, explaining how to encrypt and how to decrypt messages and why it is suitable for use as a public key cryptosystem.
- *14. Explain why it would not be suitable to use p , where p is a large prime, as the modulus for encryption in the RSA cryptosystem. That is, explain how someone could, without excessive computation, find a private key from the corresponding public key if the modulus were a large prime, rather than the product of two large primes.
15. Explain what is meant by a cryptographic hash function? What are the important properties such a function must have?

5

Induction and Recursion

- 5.1** Mathematical Induction
- 5.2** Strong Induction and Well-Ordering
- 5.3** Recursive Definitions and Structural Induction
- 5.4** Recursive Algorithms
- 5.5** Program Correctness

Many mathematical statements assert that a property is true for all positive integers. Examples of such statements are that for every positive integer n : $n! \leq n^n$, $n^3 - n$ is divisible by 3; a set with n elements has 2^n subsets; and the sum of the first n positive integers is $n(n + 1)/2$. A major goal of this chapter, and the book, is to give the student a thorough understanding of mathematical induction, which is used to prove results of this kind.

Proofs using mathematical induction have two parts. First, they show that the statement holds for the positive integer 1. Second, they show that if the statement holds for a positive integer then it must also hold for the next larger integer. Mathematical induction is based on the rule of inference that tells us that if $P(1)$ and $\forall k(P(k) \rightarrow P(k + 1))$ are true for the domain of positive integers, then $\forall n P(n)$ is true. Mathematical induction can be used to prove a tremendous variety of results. Understanding how to read and construct proofs by mathematical induction is a key goal of learning discrete mathematics.

In Chapter 2 we explicitly defined sets and functions. That is, we described sets by listing their elements or by giving some property that characterizes these elements. We gave formulae for the values of functions. There is another important way to define such objects, based on mathematical induction. To define functions, some initial terms are specified, and a rule is given for finding subsequent values from values already known. (We briefly touched on this sort of definition in Chapter 2 when we showed how sequences can be defined using recurrence relations.) Sets can be defined by listing some of their elements and giving rules for constructing elements from those already known to be in the set. Such definitions, called *recursive definitions*, are used throughout discrete mathematics and computer science. Once we have defined a set recursively, we can use a proof method called structural induction to prove results about this set.

When a procedure is specified for solving a problem, this procedure must *always* solve the problem correctly. Just testing to see that the correct result is obtained for a set of input values does not show that the procedure always works correctly. The correctness of a procedure can be guaranteed only by proving that it always yields the correct result. The final section of this chapter contains an introduction to the techniques of program verification. This is a formal technique to verify that procedures are correct. Program verification serves as the basis for attempts under way to prove in a mechanical fashion that programs are correct.

5.1 Mathematical Induction

Introduction

Suppose that we have an infinite ladder, as shown in Figure 1, and we want to know whether we can reach every step on this ladder. We know two things:

1. We can reach the first rung of the ladder.
2. If we can reach a particular rung of the ladder, then we can reach the next rung.

Can we conclude that we can reach every rung? By (1), we know that we can reach the first rung of the ladder. Moreover, because we can reach the first rung, by (2), we can also reach the second rung; it is the next rung after the first rung. Applying (2) again, because we can reach the second rung, we can also reach the third rung. Continuing in this way, we can show that we

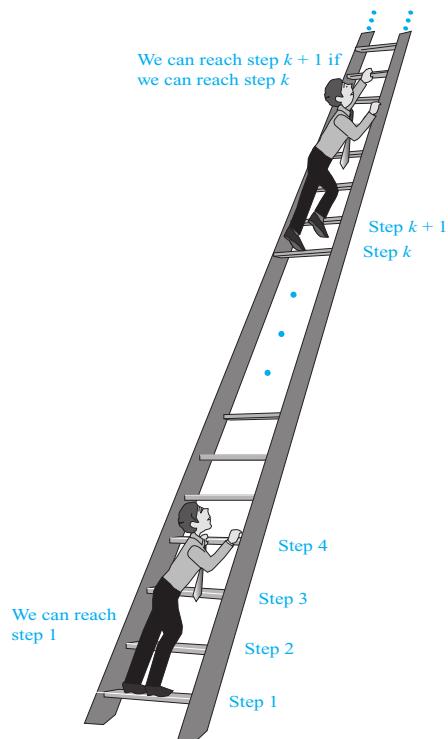


FIGURE 1 Climbing an Infinite Ladder.

can reach the fourth rung, the fifth rung, and so on. For example, after 100 uses of (2), we know that we can reach the 101st rung. But can we conclude that we are able to reach every rung of this infinite ladder? The answer is yes, something we can verify using an important proof technique called **mathematical induction**. That is, we can show that $P(n)$ is true for every positive integer n , where $P(n)$ is the statement that we can reach the n th rung of the ladder.

Mathematical induction is an extremely important proof technique that can be used to prove assertions of this type. As we will see in this section and in subsequent sections of this chapter and later chapters, mathematical induction is used extensively to prove results about a large variety of discrete objects. For example, it is used to prove results about the complexity of algorithms, the correctness of certain types of computer programs, theorems about graphs and trees, as well as a wide range of identities and inequalities.

In this section, we will describe how mathematical induction can be used and why it is a valid proof technique. It is extremely important to note that mathematical induction can be used only to prove results obtained in some other way. It is *not* a tool for discovering formulae or theorems.

Mathematical Induction



In general, mathematical induction * can be used to prove statements that assert that $P(n)$ is true for all positive integers n , where $P(n)$ is a propositional function. A proof by mathematical

*Unfortunately, using the terminology “mathematical induction” clashes with the terminology used to describe different types of reasoning. In logic, **deductive reasoning** uses rules of inference to draw conclusions from premises, whereas **inductive reasoning** makes conclusions only supported, but not ensured, by evidence. Mathematical proofs, including arguments that use mathematical induction, are deductive, not inductive.

induction has two parts, a **basis step**, where we show that $P(1)$ is true, and an **inductive step**, where we show that for all positive integers k , if $P(k)$ is true, then $P(k + 1)$ is true.

PRINCIPLE OF MATHEMATICAL INDUCTION To prove that $P(n)$ is true for all positive integers n , where $P(n)$ is a propositional function, we complete two steps:

BASIS STEP: We verify that $P(1)$ is true.

INDUCTIVE STEP: We show that the conditional statement $P(k) \rightarrow P(k + 1)$ is true for all positive integers k .

To complete the inductive step of a proof using the principle of mathematical induction, we assume that $P(k)$ is true for an arbitrary positive integer k and show that under this assumption, $P(k + 1)$ must also be true. The assumption that $P(k)$ is true is called the **inductive hypothesis**. Once we complete both steps in a proof by mathematical induction, we have shown that $P(n)$ is true for all positive integers, that is, we have shown that $\forall n P(n)$ is true where the quantification is over the set of positive integers. In the inductive step, we show that $\forall k(P(k) \rightarrow P(k + 1))$ is true, where again, the domain is the set of positive integers.

Expressed as a rule of inference, this proof technique can be stated as

$$(P(1) \wedge \forall k(P(k) \rightarrow P(k + 1))) \rightarrow \forall n P(n),$$

when the domain is the set of positive integers. Because mathematical induction is such an important technique, it is worthwhile to explain in detail the steps of a proof using this technique. The first thing we do to prove that $P(n)$ is true for all positive integers n is to show that $P(1)$ is true. This amounts to showing that the particular statement obtained when n is replaced by 1 in $P(n)$ is true. Then we must show that $P(k) \rightarrow P(k + 1)$ is true for every positive integer k . To prove that this conditional statement is true for every positive integer k , we need to show that $P(k + 1)$ cannot be false when $P(k)$ is true. This can be accomplished by assuming that $P(k)$ is true and showing that *under this hypothesis* $P(k + 1)$ must also be true.

Remark: In a proof by mathematical induction it is *not* assumed that $P(k)$ is true for all positive integers! It is only shown that *if it is assumed* that $P(k)$ is true, then $P(k + 1)$ is also true. Thus, a proof by mathematical induction is not a case of begging the question, or circular reasoning.

When we use mathematical induction to prove a theorem, we first show that $P(1)$ is true. Then we know that $P(2)$ is true, because $P(1)$ implies $P(2)$. Further, we know that $P(3)$ is true, because $P(2)$ implies $P(3)$. Continuing along these lines, we see that $P(n)$ is true for every positive integer n .



HISTORICAL NOTE The first known use of mathematical induction is in the work of the sixteenth-century mathematician Francesco Maurolico (1494–1575). Maurolico wrote extensively on the works of classical mathematics and made many contributions to geometry and optics. In his book *Arithmetoricorum Libri Duo*, Maurolico presented a variety of properties of the integers together with proofs of these properties. To prove some of these properties, he devised the method of mathematical induction. His first use of mathematical induction in this book was to prove that the sum of the first n odd positive integers equals n^2 . Augustus De Morgan is credited with the first presentation in 1838 of formal proofs using mathematical induction, as well as introducing the terminology “mathematical induction.” Maurolico’s proofs were informal and he never used the word “induction.” See [Gu11] to learn more about the history of the method of mathematical induction.

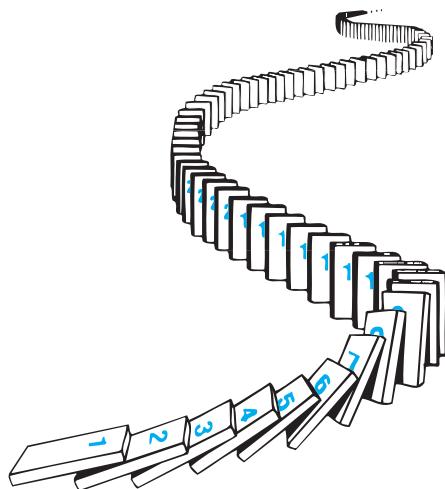


FIGURE 2 Illustrating How Mathematical Induction Works Using Dominoes.

WAYS TO REMEMBER HOW MATHEMATICAL INDUCTION WORKS Thinking of the infinite ladder and the rules for reaching steps can help you remember how mathematical induction works. Note that statements (1) and (2) for the infinite ladder are exactly the basis step and inductive step, respectively, of the proof that $P(n)$ is true for all positive integers n , where $P(n)$ is the statement that we can reach the n th rung of the ladder. Consequently, we can invoke mathematical induction to conclude that we can reach every rung.

Another way to illustrate the principle of mathematical induction is to consider an infinite row of dominoes, labeled $1, 2, 3, \dots, n, \dots$, where each domino is standing up. Let $P(n)$ be the proposition that domino n is knocked over. If the first domino is knocked over—i.e., if $P(1)$ is true—and if, whenever the k th domino is knocked over, it also knocks the $(k + 1)$ st domino over—i.e., if $P(k) \rightarrow P(k + 1)$ is true for all positive integers k —then all the dominoes are knocked over. This is illustrated in Figure 2.

Why Mathematical Induction is Valid

Why is mathematical induction a valid proof technique? The reason comes from the well-ordering property, listed in Appendix 1, as an axiom for the set of positive integers, which states that every nonempty subset of the set of positive integers has a least element. So, suppose we know that $P(1)$ is true and that the proposition $P(k) \rightarrow P(k + 1)$ is true for all positive integers k . To show that $P(n)$ must be true for all positive integers n , assume that there is at least one positive integer for which $P(n)$ is false. Then the set S of positive integers for which $P(n)$ is false is nonempty. Thus, by the well-ordering property, S has a least element, which will be denoted by m . We know that m cannot be 1, because $P(1)$ is true. Because m is positive and greater than 1, $m - 1$ is a positive integer. Furthermore, because $m - 1$ is less than m , it is not in S , so $P(m - 1)$ must be true. Because the conditional statement $P(m - 1) \rightarrow P(m)$ is also true, it must be the case that $P(m)$ is true. This contradicts the choice of m . Hence, $P(n)$ must be true for every positive integer n .

The Good and the Bad of Mathematical Induction

An important point needs to be made about mathematical induction before we commence a study of its use. The good thing about mathematical induction is that it can be used to prove

You can prove a theorem by mathematical induction even if you do not have the slightest idea why it is true!

a conjecture once it is has been made (and is true). The bad thing about it is that it cannot be used to find new theorems. Mathematicians sometimes find proofs by mathematical induction unsatisfying because they do not provide insights as to why theorems are true. Many theorems can be proved in many ways, including by mathematical induction. Proofs of these theorems by methods other than mathematical induction are often preferred because of the insights they bring.

Examples of Proofs by Mathematical Induction

Many theorems assert that $P(n)$ is true for all positive integers n , where $P(n)$ is a propositional function. Mathematical induction is a technique for proving theorems of this kind. In other words, mathematical induction can be used to prove statements of the form $\forall n P(n)$, where the domain is the set of positive integers. Mathematical induction can be used to prove an extremely wide variety of theorems, each of which is a statement of this form. (Remember, many mathematical assertions include an implicit universal quantifier. The statement “if n is a positive integer, then $n^3 - n$ is divisible by 3” is an example of this. Making the implicit universal quantifier explicit yields the statement “for every positive integer n , $n^3 - n$ is divisible by 3.”)

We will use how theorems are proved using mathematical induction. The theorems we will prove include summation formulae, inequalities, identities for combinations of sets, divisibility results, theorems about algorithms, and some other creative results. In this section and in later sections, we will employ mathematical induction to prove many other types of results, including the correctness of computer programs and algorithms. Mathematical induction can be used to prove a wide variety of theorems, not just summation formulae, inequalities, and other types of examples we illustrate here. (For proofs by mathematical induction of many more interesting and diverse results, see the *Handbook of Mathematical Induction* by David Gunderson [Gu11]. This book is part of the extensive CRC Series in Discrete Mathematics, many of which may be of interest to readers. The author is the Series Editor of these books).

Note that there are many opportunities for errors in induction proofs. We will describe some incorrect proofs by mathematical induction at the end of this section and in the exercises. To avoid making errors in proofs by mathematical induction, try to follow the guidelines for such proofs given at the end of this section.

Look for the $\stackrel{\text{IH}}{=}$ symbol to see where the inductive hypothesis is used.

SEEING WHERE THE INDUCTIVE HYPOTHESIS IS USED To help the reader understand each of the mathematical induction proofs in this section, we will note where the inductive hypothesis is used. We indicate this use in three different ways: by explicit mention in the text, by inserting the acronym IH (for inductive hypothesis) over an equals sign or a sign for an inequality, or by specifying the inductive hypothesis as the reason for a step in a multi-line display.

PROVING SUMMATION FORMULAE We begin by using mathematical induction to prove several summation formulae. As we will see, mathematical induction is particularly well suited for proving that such formulae are valid. However, summation formulae can be proven in other ways. This is not surprising because there are often different ways to prove a theorem. The major disadvantage of using mathematical induction to prove a summation formula is that you cannot use it to derive this formula. That is, you must already have the formula before you attempt to prove it by mathematical induction.

Examples 1–4 illustrate how to use mathematical induction to prove summation formulae. The first summation formula we will prove by mathematical induction, in Example 1, is a closed formula for the sum of the smallest n positive integers.

EXAMPLE 1 Show that if n is a positive integer, then



$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Solution: Let $P(n)$ be the proposition that the sum of the first n positive integers, $1 + 2 + \cdots + n = \frac{n(n+1)}{2}$, is $n(n+1)/2$. We must do two things to prove that $P(n)$ is true for $n = 1, 2, 3, \dots$. Namely, we must show that $P(1)$ is true and that the conditional statement $P(k)$ implies $P(k+1)$ is true for $k = 1, 2, 3, \dots$.

BASIS STEP: $P(1)$ is true, because $1 = \frac{1(1+1)}{2}$. (The left-hand side of this equation is 1 because 1 is the sum of the first positive integer. The right-hand side is found by substituting 1 for n in $n(n+1)/2$.)

INDUCTIVE STEP: For the inductive hypothesis we assume that $P(k)$ holds for an arbitrary positive integer k . That is, we assume that

$$1 + 2 + \cdots + k = \frac{k(k+1)}{2}.$$

Under this assumption, it must be shown that $P(k+1)$ is true, namely, that

$$1 + 2 + \cdots + k + (k+1) = \frac{(k+1)[(k+1)+1]}{2} = \frac{(k+1)(k+2)}{2}$$

is also true. When we add $k+1$ to both sides of the equation in $P(k)$, we obtain

$$\begin{aligned} 1 + 2 + \cdots + k + (k+1) &\stackrel{\text{IH}}{=} \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2}. \end{aligned}$$

This last equation shows that $P(k+1)$ is true under the assumption that $P(k)$ is true. This completes the inductive step.

We have completed the basis step and the inductive step, so by mathematical induction we know that $P(n)$ is true for all positive integers n . That is, we have proven that $1 + 2 + \cdots + n = n(n+1)/2$ for all positive integers n .

As we noted, mathematical induction is not a tool for finding theorems about all positive integers. Rather, it is a proof method for proving such results once they are conjectured. In Example 2, using mathematical induction to prove a summation formula, we will both formulate and then prove a conjecture.

EXAMPLE 2 Conjecture a formula for the sum of the first n positive odd integers. Then prove your conjecture using mathematical induction.

Solution: The sums of the first n positive odd integers for $n = 1, 2, 3, 4, 5$ are

$$\begin{array}{lll} 1 = 1, & 1 + 3 = 4, & 1 + 3 + 5 = 9, \\ 1 + 3 + 5 + 7 = 16, & 1 + 3 + 5 + 7 + 9 = 25. & \end{array}$$

From these values it is reasonable to conjecture that the sum of the first n positive odd integers is n^2 , that is, $1 + 3 + 5 + \dots + (2n - 1) = n^2$. We need a method to *prove* that this *conjecture* is correct, if in fact it is.

Let $P(n)$ denote the proposition that the sum of the first n odd positive integers is n^2 . Our conjecture is that $P(n)$ is true for all positive integers. To use mathematical induction to prove this conjecture, we must first complete the basis step; that is, we must show that $P(1)$ is true. Then we must carry out the inductive step; that is, we must show that $P(k + 1)$ is true when $P(k)$ is assumed to be true. We now attempt to complete these two steps.

BASIS STEP: $P(1)$ states that the sum of the first one odd positive integer is 1^2 . This is true because the sum of the first odd positive integer is 1. The basis step is complete.

INDUCTIVE STEP: To complete the inductive step we must show that the proposition $P(k) \rightarrow P(k + 1)$ is true for every positive integer k . To do this, we first assume the inductive hypothesis. The inductive hypothesis is the statement that $P(k)$ is true for an arbitrary positive integer k , that is,

$$1 + 3 + 5 + \dots + (2k - 1) = k^2.$$

(Note that the k th odd positive integer is $(2k - 1)$, because this integer is obtained by adding 2 a total of $k - 1$ times to 1.) To show that $\forall k (P(k) \rightarrow P(k + 1))$ is true, we must show that if $P(k)$ is true (the inductive hypothesis), then $P(k + 1)$ is true. Note that $P(k + 1)$ is the statement that

$$1 + 3 + 5 + \dots + (2k - 1) + (2k + 1) = (k + 1)^2.$$

So, assuming that $P(k)$ is true, it follows that

$$\begin{aligned} 1 + 3 + 5 + \dots + (2k - 1) + (2k + 1) &= [1 + 3 + \dots + (2k - 1)] + (2k + 1) \\ &\stackrel{\text{IH}}{=} k^2 + (2k + 1) \\ &= k^2 + 2k + 1 \\ &= (k + 1)^2. \end{aligned}$$

This shows that $P(k + 1)$ follows from $P(k)$. Note that we used the inductive hypothesis $P(k)$ in the second equality to replace the sum of the first k odd positive integers by k^2 .

We have now completed both the basis step and the inductive step. That is, we have shown that $P(1)$ is true and the conditional statement $P(k) \rightarrow P(k + 1)$ is true for all positive integers k . Consequently, by the principle of mathematical induction we can conclude that $P(n)$ is true for all positive integers n . That is, we know that $1 + 3 + 5 + \dots + (2n - 1) = n^2$ for all positive integers n . 

Often, we will need to show that $P(n)$ is true for $n = b, b + 1, b + 2, \dots$, where b is an integer other than 1. We can use mathematical induction to accomplish this, as long as we change the basis step by replacing $P(1)$ with $P(b)$. In other words, to use mathematical induction to show that $P(n)$ is true for $n = b, b + 1, b + 2, \dots$, where b is an integer other than 1, we show that $P(b)$ is true in the basis step. In the inductive step, we show that the conditional statement $P(k) \rightarrow P(k + 1)$ is true for $k = b, b + 1, b + 2, \dots$. Note that b can be negative, zero, or positive. Following the domino analogy we used earlier, imagine that we begin by knocking down the b th domino (the basis step), and as each domino falls, it knocks down the next domino (the inductive step). We leave it to the reader to show that this form of induction is valid (see Exercise 83).

We illustrate this notion in Example 3, which states that a summation formula is valid for all nonnegative integers. In this example, we need to prove that $P(n)$ is true for $n = 0, 1, 2, \dots$. So, the basis step in Example 3 shows that $P(0)$ is true.

EXAMPLE 3 Use mathematical induction to show that

$$1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$$

for all nonnegative integers n .

Solution: Let $P(n)$ be the proposition that $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$ for the integer n .

BASIS STEP: $P(0)$ is true because $2^0 = 1 = 2^1 - 1$. This completes the basis step.

INDUCTIVE STEP: For the inductive hypothesis, we assume that $P(k)$ is true for an arbitrary nonnegative integer k . That is, we assume that

$$1 + 2 + 2^2 + \cdots + 2^k = 2^{k+1} - 1.$$

To carry out the inductive step using this assumption, we must show that when we assume that $P(k)$ is true, then $P(k + 1)$ is also true. That is, we must show that

$$1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$$

assuming the inductive hypothesis $P(k)$. Under the assumption of $P(k)$, we see that

$$\begin{aligned} 1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} &= (1 + 2 + 2^2 + \cdots + 2^k) + 2^{k+1} \\ &\stackrel{\text{IH}}{=} (2^{k+1} - 1) + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \\ &= 2^{k+2} - 1. \end{aligned}$$

Note that we used the inductive hypothesis in the second equation in this string of equalities to replace $1 + 2 + 2^2 + \cdots + 2^k$ by $2^{k+1} - 1$. We have completed the inductive step.

Because we have completed the basis step and the inductive step, by mathematical induction we know that $P(n)$ is true for all nonnegative integers n . That is, $1 + 2 + \cdots + 2^n = 2^{n+1} - 1$ for all nonnegative integers n . 

The formula given in Example 3 is a special case of a general result for the sum of terms of a geometric progression (Theorem 1 in Section 2.4). We will use mathematical induction to provide an alternative proof of this formula.

EXAMPLE 4 Sums of Geometric Progressions Use mathematical induction to prove this formula for the sum of a finite number of terms of a geometric progression with initial term a and common ratio r :

$$\sum_{j=0}^n ar^j = a + ar + ar^2 + \cdots + ar^n = \frac{ar^{n+1} - a}{r - 1} \quad \text{when } r \neq 1,$$

where n is a nonnegative integer.

Solution: To prove this formula using mathematical induction, let $P(n)$ be the statement that the sum of the first $n + 1$ terms of a geometric progression in this formula is correct.

BASIS STEP: $P(0)$ is true, because

$$\frac{ar^{0+1} - a}{r - 1} = \frac{ar - a}{r - 1} = \frac{a(r - 1)}{r - 1} = a.$$

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(k)$ is true, where k is an arbitrary nonnegative integer. That is, $P(k)$ is the statement that

$$a + ar + ar^2 + \cdots + ar^k = \frac{ar^{k+1} - a}{r - 1}.$$

To complete the inductive step we must show that if $P(k)$ is true, then $P(k + 1)$ is also true. To show that this is the case, we first add ar^{k+1} to both sides of the equality asserted by $P(k)$. We find that

$$a + ar + ar^2 + \cdots + ar^k + ar^{k+1} \stackrel{\text{IH}}{=} \frac{ar^{k+1} - a}{r - 1} + ar^{k+1}.$$

Rewriting the right-hand side of this equation shows that

$$\begin{aligned} \frac{ar^{k+1} - a}{r - 1} + ar^{k+1} &= \frac{ar^{k+1} - a}{r - 1} + \frac{ar^{k+2} - ar^{k+1}}{r - 1} \\ &= \frac{ar^{k+2} - a}{r - 1}. \end{aligned}$$

Combining these last two equations gives

$$a + ar + ar^2 + \cdots + ar^k + ar^{k+1} = \frac{ar^{k+2} - a}{r - 1}.$$

This shows that if the inductive hypothesis $P(k)$ is true, then $P(k + 1)$ must also be true. This completes the inductive argument.

We have completed the basis step and the inductive step, so by mathematical induction $P(n)$ is true for all nonnegative integers n . This shows that the formula for the sum of the terms of a geometric series is correct. 

As previously mentioned, the formula in Example 3 is the case of the formula in Example 4 with $a = 1$ and $r = 2$. The reader should verify that putting these values for a and r into the general formula gives the same formula as in Example 3.

PROVING INEQUALITIES Mathematical induction can be used to prove a variety of inequalities that hold for all positive integers greater than a particular positive integer, as Examples 5–7 illustrate.

EXAMPLE 5 Use mathematical induction to prove the inequality

$$n < 2^n$$

for all positive integers n .



Solution: Let $P(n)$ be the proposition that $n < 2^n$.

BASIS STEP: $P(1)$ is true, because $1 < 2^1 = 2$. This completes the basis step.

INDUCTIVE STEP: We first assume the inductive hypothesis that $P(k)$ is true for an arbitrary positive integer k . That is, the inductive hypothesis $P(k)$ is the statement that $k < 2^k$. To complete the inductive step, we need to show that if $P(k)$ is true, then $P(k + 1)$, which is the statement that $k + 1 < 2^{k+1}$, is true. That is, we need to show that if $k < 2^k$, then $k + 1 < 2^{k+1}$. To show

that this conditional statement is true for the positive integer k , we first add 1 to both sides of $k < 2^k$, and then note that $1 \leq 2^k$. This tells us that

$$k + 1 < 2^k + 1 \stackrel{\text{IH}}{\leq} 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}.$$

This shows that $P(k + 1)$ is true, namely, that $k + 1 < 2^{k+1}$, based on the assumption that $P(k)$ is true. The induction step is complete.

Therefore, because we have completed both the basis step and the inductive step, by the principle of mathematical induction we have shown that $n < 2^n$ is true for all positive integers n . 

EXAMPLE 6 Use mathematical induction to prove that $2^n < n!$ for every integer n with $n \geq 4$. (Note that this inequality is false for $n = 1, 2$, and 3 .)

Solution: Let $P(n)$ be the proposition that $2^n < n!$.

BASIS STEP: To prove the inequality for $n \geq 4$ requires that the basis step be $P(4)$. Note that $P(4)$ is true, because $2^4 = 16 < 24 = 4!$

INDUCTIVE STEP: For the inductive step, we assume that $P(k)$ is true for an arbitrary integer k with $k \geq 4$. That is, we assume that $2^k < k!$ for the positive integer k with $k \geq 4$. We must show that under this hypothesis, $P(k + 1)$ is also true. That is, we must show that if $2^k < k!$ for an arbitrary positive integer k where $k \geq 4$, then $2^{k+1} < (k + 1)!$. We have

$$\begin{aligned} 2^{k+1} &= 2 \cdot 2^k && \text{by definition of exponent} \\ &< 2 \cdot k! && \text{by the inductive hypothesis} \\ &< (k + 1)k! && \text{because } 2 < k + 1 \\ &= (k + 1)! && \text{by definition of factorial function.} \end{aligned}$$

This shows that $P(k + 1)$ is true when $P(k)$ is true. This completes the inductive step of the proof.

We have completed the basis step and the inductive step. Hence, by mathematical induction $P(n)$ is true for all integers n with $n \geq 4$. That is, we have proved that $2^n < n!$ is true for all integers n with $n \geq 4$. 

An important inequality for the sum of the reciprocals of a set of positive integers will be proved in Example 7.

EXAMPLE 7 An Inequality for Harmonic Numbers The **harmonic numbers** H_j , $j = 1, 2, 3, \dots$, are defined by

$$H_j = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{j}.$$

For instance,

$$H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12}.$$

Use mathematical induction to show that

$$H_{2^n} \geq 1 + \frac{n}{2},$$

whenever n is a nonnegative integer.

Solution: To carry out the proof, let $P(n)$ be the proposition that $H_{2^n} \geq 1 + \frac{n}{2}$.

BASIS STEP: $P(0)$ is true, because $H_{2^0} = H_1 = 1 \geq 1 + \frac{0}{2}$.

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(k)$ is true, that is, $H_{2^k} \geq 1 + \frac{k}{2}$, where k is an arbitrary nonnegative integer. We must show that if $P(k)$ is true, then $P(k+1)$, which states that $H_{2^{k+1}} \geq 1 + \frac{k+1}{2}$, is also true. So, assuming the inductive hypothesis, it follows that

$$\begin{aligned} H_{2^{k+1}} &= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{2^k} + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}} && \text{by the definition of harmonic number} \\ &= H_{2^k} + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}} && \text{by the definition of } 2^k\text{th harmonic number} \\ &\geq \left(1 + \frac{k}{2}\right) + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}} && \text{by the inductive hypothesis} \\ &\geq \left(1 + \frac{k}{2}\right) + 2^k \cdot \frac{1}{2^{k+1}} && \text{because there are } 2^k \text{ terms} \\ &\geq \left(1 + \frac{k}{2}\right) + \frac{1}{2} && \text{canceling a common factor of } 2^k \text{ in second term} \\ &= 1 + \frac{k+1}{2}. \end{aligned}$$

This establishes the inductive step of the proof.

We have completed the basis step and the inductive step. Thus, by mathematical induction $P(n)$ is true for all nonnegative integers n . That is, the inequality $H_{2^n} \geq 1 + \frac{n}{2}$ for the harmonic numbers holds for all nonnegative integers n . 

Remark: The inequality established here shows that the **harmonic series**

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} + \cdots$$

is a divergent infinite series. This is an important example in the study of infinite series.

PROVING DIVISIBILITY RESULTS Mathematical induction can be used to prove divisibility results about integers. Although such results are often easier to prove using basic results in number theory, it is instructive to see how to prove such results using mathematical induction, as Examples 8 and 9 illustrate.

EXAMPLE 8 Use mathematical induction to prove that $n^3 - n$ is divisible by 3 whenever n is a positive integer. (Note that this is the statement with $p = 3$ of Fermat's little theorem, which is Theorem 3 of Section 4.4.)



Solution: To construct the proof, let $P(n)$ denote the proposition: " $n^3 - n$ is divisible by 3."

BASIS STEP: The statement $P(1)$ is true because $1^3 - 1 = 0$ is divisible by 3. This completes the basis step.

INDUCTIVE STEP: For the inductive hypothesis we assume that $P(k)$ is true; that is, we assume that $k^3 - k$ is divisible by 3 for an arbitrary positive integer k . To complete the inductive

step, we must show that when we assume the inductive hypothesis, it follows that $P(k+1)$, the statement that $(k+1)^3 - (k+1)$ is divisible by 3, is also true. That is, we must show that $(k+1)^3 - (k+1)$ is divisible by 3. Note that

$$\begin{aligned}(k+1)^3 - (k+1) &= (k^3 + 3k^2 + 3k + 1) - (k+1) \\ &= (k^3 - k) + 3(k^2 + k).\end{aligned}$$

Using the inductive hypothesis, we conclude that the first term $k^3 - k$ is divisible by 3. The second term is divisible by 3 because it is 3 times an integer. So, by part (i) of Theorem 1 in Section 4.1, we know that $(k+1)^3 - (k+1)$ is also divisible by 3. This completes the inductive step.

Because we have completed both the basis step and the inductive step, by the principle of mathematical induction we know that $n^3 - n$ is divisible by 3 whenever n is a positive integer. 

The next example presents a more challenging proof by mathematical induction of a divisibility result.

EXAMPLE 9 Use mathematical induction to prove that $7^{n+2} + 8^{2n+1}$ is divisible by 57 for every nonnegative integer n .



Solution: To construct the proof, let $P(n)$ denote the proposition: “ $7^{n+2} + 8^{2n+1}$ is divisible by 57.”

BASIS STEP: To complete the basis step, we must show that $P(0)$ is true, because we want to prove that $P(n)$ is true for every nonnegative integer. We see that $P(0)$ is true because $7^{0+2} + 8^{2 \cdot 0 + 1} = 7^2 + 8^1 = 57$ is divisible by 57. This completes the basis step.

INDUCTIVE STEP: For the inductive hypothesis we assume that $P(k)$ is true for an arbitrary nonnegative integer k ; that is, we assume that $7^{k+2} + 8^{2k+1}$ is divisible by 57. To complete the inductive step, we must show that when we assume that the inductive hypothesis $P(k)$ is true, then $P(k+1)$, the statement that $7^{(k+1)+2} + 8^{2(k+1)+1}$ is divisible by 57, is also true.

The difficult part of the proof is to see how to use the inductive hypothesis. To take advantage of the inductive hypothesis, we use these steps:

$$\begin{aligned}7^{(k+1)+2} + 8^{2(k+1)+1} &= 7^{k+3} + 8^{2k+3} \\ &= 7 \cdot 7^{k+2} + 8^2 \cdot 8^{2k+1} \\ &= 7 \cdot 7^{k+2} + 64 \cdot 8^{2k+1} \\ &= 7(7^{k+2} + 8^{2k+1}) + 57 \cdot 8^{2k+1}.\end{aligned}$$

We can now use the inductive hypothesis, which states that $7^{k+2} + 8^{2k+1}$ is divisible by 57. We will use parts (i) and (ii) of Theorem 1 in Section 4.1. By part (ii) of this theorem, and the inductive hypothesis, we conclude that the first term in this last sum, $7(7^{k+2} + 8^{2k+1})$, is divisible by 57. By part (ii) of this theorem, the second term in this sum, $57 \cdot 8^{2k+1}$, is divisible by 57. Hence, by part (i) of this theorem, we conclude that $7(7^{k+2} + 8^{2k+1}) + 57 \cdot 8^{2k+1} = 7^{k+3} + 8^{2k+3}$ is divisible by 57. This completes the inductive step.

Because we have completed both the basis step and the inductive step, by the principle of mathematical induction we know that $7^{n+2} + 8^{2n+1}$ is divisible by 57 for every nonnegative integer n . 

PROVING RESULTS ABOUT SETS Mathematical induction can be used to prove many results about sets. In particular, in Example 10 we prove a formula for the number of subsets of a finite set and in Example 11 we establish a set identity.

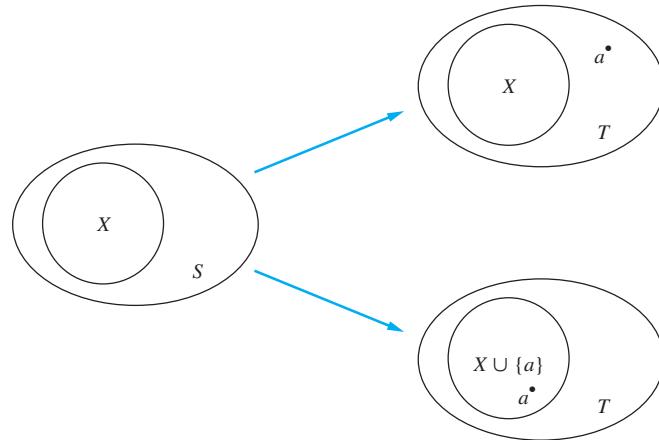


FIGURE 3 Generating Subsets of a Set with $k + 1$ Elements. Here $T = S \cup \{a\}$.

EXAMPLE 10 The Number of Subsets of a Finite Set Use mathematical induction to show that if S is a finite set with n elements, where n is a nonnegative integer, then S has 2^n subsets. (We will prove this result directly in several ways in Chapter 6.)

Solution: Let $P(n)$ be the proposition that a set with n elements has 2^n subsets.

BASIS STEP: $P(0)$ is true, because a set with zero elements, the empty set, has exactly $2^0 = 1$ subset, namely, itself.

INDUCTIVE STEP: For the inductive hypothesis we assume that $P(k)$ is true for an arbitrary nonnegative integer k , that is, we assume that every set with k elements has 2^k subsets. It must be shown that under this assumption, $P(k + 1)$, which is the statement that every set with $k + 1$ elements has 2^{k+1} subsets, must also be true. To show this, let T be a set with $k + 1$ elements. Then, it is possible to write $T = S \cup \{a\}$, where a is one of the elements of T and $S = T - \{a\}$ (and hence $|S| = k$). The subsets of T can be obtained in the following way. For each subset X of S there are exactly two subsets of T , namely, X and $X \cup \{a\}$. (This is illustrated in Figure 3.) These constitute all the subsets of T and are all distinct. We now use the inductive hypothesis to conclude that S has 2^k subsets, because it has k elements. We also know that there are two subsets of T for each subset of S . Therefore, there are $2 \cdot 2^k = 2^{k+1}$ subsets of T . This finishes the inductive argument.

Because we have completed the basis step and the inductive step, by mathematical induction it follows that $P(n)$ is true for all nonnegative integers n . That is, we have proved that a set with n elements has 2^n subsets whenever n is a nonnegative integer. \blacktriangleleft

EXAMPLE 11 Use mathematical induction to prove the following generalization of one of De Morgan's laws:

$$\overline{\bigcap_{j=1}^n A_j} = \bigcup_{j=1}^n \overline{A_j}$$

whenever A_1, A_2, \dots, A_n are subsets of a universal set U and $n \geq 2$.

Solution: Let $P(n)$ be the identity for n sets.

BASIS STEP: The statement $P(2)$ asserts that $\overline{A_1 \cap A_2} = \overline{A_1} \cup \overline{A_2}$. This is one of De Morgan's laws; it was proved in Example 11 of Section 2.2.

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(k)$ is true, where k is an arbitrary integer with $k \geq 2$; that is, it is the statement that

$$\overline{\bigcap_{j=1}^k A_j} = \bigcup_{j=1}^k \overline{A_j}$$

whenever A_1, A_2, \dots, A_k are subsets of the universal set U . To carry out the inductive step, we need to show that this assumption implies that $P(k+1)$ is true. That is, we need to show that if this equality holds for every collection of k subsets of U , then it must also hold for every collection of $k+1$ subsets of U . Suppose that $A_1, A_2, \dots, A_k, A_{k+1}$ are subsets of U . When the inductive hypothesis is assumed to hold, it follows that

$$\begin{aligned}\overline{\bigcap_{j=1}^{k+1} A_j} &= \overline{\left(\bigcap_{j=1}^k A_j\right) \cap A_{k+1}} && \text{by the definition of intersection} \\ &= \overline{\left(\bigcap_{j=1}^k A_j\right)} \cup \overline{A_{k+1}} && \text{by De Morgan's law (where the two sets are } \bigcap_{j=1}^k A_j \text{ and } A_{k+1} \text{)} \\ &= \left(\bigcup_{j=1}^k \overline{A_j}\right) \cup \overline{A_{k+1}} && \text{by the inductive hypothesis} \\ &= \bigcup_{j=1}^{k+1} \overline{A_j} && \text{by the definition of union.}\end{aligned}$$

This completes the inductive step.

Because we have completed both the basis step and the inductive step, by mathematical induction we know that $P(n)$ is true whenever n is a positive integer, $n \geq 2$. That is, we know that

$$\overline{\bigcap_{j=1}^n A_j} = \bigcup_{j=1}^n \overline{A_j}$$

whenever A_1, A_2, \dots, A_n are subsets of a universal set U and $n \geq 2$.

PROVING RESULTS ABOUT ALGORITHMS Next, we provide an example (somewhat more difficult than previous examples) that illustrates one of many ways mathematical induction is used in the study of algorithms. We will show how mathematical induction can be used to prove that a greedy algorithm we introduced in Section 3.1 always yields an optimal solution.

EXAMPLE 12

Recall the algorithm for scheduling talks discussed in Example 7 of Section 3.1. The input to this algorithm is a group of m proposed talks with preset starting and ending times. The goal is to schedule as many of these lectures as possible in the main lecture hall so that no two talks overlap. Suppose that talk t_j begins at time s_j and ends at time e_j . (No two lectures can proceed in the main lecture hall at the same time, but a lecture in this hall can begin at the same time another one ends.)

Without loss of generality, we assume that the talks are listed in order of nondecreasing ending time, so that $e_1 \leq e_2 \leq \dots \leq e_m$. The greedy algorithm proceeds by selecting at each stage a talk with the earliest ending time among all those talks that begin no sooner than when

 the last talk scheduled in the main lecture hall has ended. Note that a talk with the earliest end time is always selected first by the algorithm. We will show that this greedy algorithm is optimal in the sense that it always schedules the most talks possible in the main lecture hall. To prove the optimality of this algorithm we use mathematical induction on the variable n , the number of talks scheduled by the algorithm. We let $P(n)$ be the proposition that if the greedy algorithm schedules n talks in the main lecture hall, then it is not possible to schedule more than n talks in this hall.

BASIS STEP: Suppose that the greedy algorithm managed to schedule just one talk, t_1 , in the main lecture hall. This means that no other talk can start at or after e_1 , the end time of t_1 . Otherwise, the first such talk we come to as we go through the talks in order of nondecreasing end times could be added. Hence, at time e_1 each of the remaining talks needs to use the main lecture hall because they all start before e_1 and end after e_1 . It follows that no two talks can be scheduled because both need to use the main lecture hall at time e_1 . This shows that $P(1)$ is true and completes the basis step.

INDUCTIVE STEP: The inductive hypothesis is that $P(k)$ is true, where k is an arbitrary positive integer, that is, that the greedy algorithm always schedules the most possible talks when it selects k talks, where k is a positive integer, given any set of talks, no matter how many. We must show that $P(k + 1)$ follows from the assumption that $P(k)$ is true, that is, we must show that under the assumption of $P(k)$, the greedy algorithm always schedules the most possible talks when it selects $k + 1$ talks.

Now suppose that the greedy algorithm has selected $k + 1$ talks. Our first step in completing the inductive step is to show there is a schedule including the most talks possible that contains talk t_1 , a talk with the earliest end time. This is easy to see because a schedule that begins with the talk t_i in the list, where $i > 1$, can be changed so that talk t_1 replaces talk t_i . To see this, note that because $e_1 \leq e_i$, all talks that were scheduled to follow talk t_i can still be scheduled.

Once we included talk t_1 , scheduling the talks so that as many as possible are scheduled is reduced to scheduling as many talks as possible that begin at or after time e_1 . So, if we have scheduled as many talks as possible, the schedule of talks other than talk t_1 is an optimal schedule of the original talks that begin once talk t_1 has ended. Because the greedy algorithm schedules k talks when it creates this schedule, we can apply the inductive hypothesis to conclude that it has scheduled the most possible talks. It follows that the greedy algorithm has scheduled the most possible talks, $k + 1$, when it produced a schedule with $k + 1$ talks, so $P(k + 1)$ is true. This completes the inductive step.

We have completed the basis step and the inductive step. So, by mathematical induction we know that $P(n)$ is true for all positive integers n . This completes the proof of optimality. That is, we have proved that when the greedy algorithm schedules n talks, when n is a positive integer, then it is not possible to schedule more than n talks. 

CREATIVE USES OF MATHEMATICAL INDUCTION Mathematical induction can often be used in unexpected ways. We will illustrate two particularly clever uses of mathematical induction here, the first relating to survivors in a pie fight and the second relating to tilings with regular triominoes of checkerboards with one square missing.

EXAMPLE 13



Odd Pie Fights An odd number of people stand in a yard at mutually distinct distances. At the same time each person throws a pie at their nearest neighbor, hitting this person. Use mathematical induction to show that there is at least one survivor, that is, at least one person who is not hit by a pie. (This problem was introduced by Carmony [Ca79]. Note that this result is false when there are an even number of people; see Exercise 75.)

Solution: Let $P(n)$ be the statement that there is a survivor whenever $2n + 1$ people stand in a yard at distinct mutual distances and each person throws a pie at their nearest neighbor. To prove this result, we will show that $P(n)$ is true for all positive integers n . This follows because as n runs through all positive integers, $2n + 1$ runs through all odd integers greater than or equal

to 3. Note that one person cannot engage in a pie fight because there is no one else to throw the pie at.

BASIS STEP: When $n = 1$, there are $2n + 1 = 3$ people in the pie fight. Of the three people, suppose that the closest pair are A and B , and C is the third person. Because distances between pairs of people are different, the distance between A and C and the distance between B and C are both different from, and greater than, the distance between A and B . It follows that A and B throw pies at each other, while C throws a pie at either A or B , whichever is closer. Hence, C is not hit by a pie. This shows that at least one of the three people is not hit by a pie, completing the basis step.

INDUCTIVE STEP: For the inductive step, assume that $P(k)$ is true for an arbitrary odd integer k with $k \geq 3$. That is, assume that there is at least one survivor whenever $2k + 1$ people stand in a yard at distinct mutual distances and each throws a pie at their nearest neighbor. We must show that if the inductive hypothesis $P(k)$ is true, then $P(k + 1)$, the statement that there is at least one survivor whenever $2(k + 1) + 1 = 2k + 3$ people stand in a yard at distinct mutual distances and each throws a pie at their nearest neighbor, is also true.

So suppose that we have $2(k + 1) + 1 = 2k + 3$ people in a yard with distinct distances between pairs of people. Let A and B be the closest pair of people in this group of $2k + 3$ people. When each person throws a pie at the nearest person, A and B throw pies at each other. We have two cases to consider, (i) when someone else throws a pie at either A or B and (ii) when no one else throws a pie at either A or B .

Case (i): Because A and B throw pies at each other and someone else throws a pie at either A and B , at least three pies are thrown at A and B , and at most $(2k + 3) - 3 = 2k$ pies are thrown at the remaining $2k + 1$ people. This guarantees that at least one person is a survivor, for if each of these $2k + 1$ people was hit by at least one pie, a total of at least $2k + 1$ pies would have to be thrown at them. (The reasoning used in this last step is an example of the pigeonhole principle discussed further in Section 6.2.)

Case (ii): No one else throws a pie at either A and B . Besides A and B , there are $2k + 1$ people. Because the distances between pairs of these people are all different, we can use the inductive hypothesis to conclude that there is at least one survivor S when these $2k + 1$ people each throws a pie at their nearest neighbor. Furthermore, S is also not hit by either the pie thrown by A or the pie thrown by B because A and B throw their pies at each other, so S is a survivor because S is not hit by any of the pies thrown by these $2k + 3$ people.

We have completed both the basis step and the inductive step, using a proof by cases. So by mathematical induction it follows that $P(n)$ is true for all positive integers n . We conclude that whenever an odd number of people located in a yard at distinct mutual distances each throws a pie at their nearest neighbor, there is at least one survivor. 



In Section 1.8 we discussed the tiling of checkerboards by polyominoes. Example 14 illustrates how mathematical induction can be used to prove a result about covering checkerboards with right triominoes, pieces shaped like the letter “L.”

EXAMPLE 14

Let n be a positive integer. Show that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes, where these pieces cover three squares at a time, as shown in Figure 4.

Solution: Let $P(n)$ be the proposition that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using right triominoes. We can use mathematical induction to prove that $P(n)$ is true for all positive integers n .

BASIS STEP: $P(1)$ is true, because each of the four 2×2 checkerboards with one square removed can be tiled using one right triomino, as shown in Figure 5.

FIGURE 4 A Right Triomino.



FIGURE 5 Tiling 2×2 Checkerboards with One Square Removed.

INDUCTIVE STEP: The inductive hypothesis is the assumption that $P(k)$ is true for the positive integer k ; that is, it is the assumption that every $2^k \times 2^k$ checkerboard with one square removed can be tiled using right triominoes. It must be shown that under the assumption of the inductive hypothesis, $P(k+1)$ must also be true; that is, any $2^{k+1} \times 2^{k+1}$ checkerboard with one square removed can be tiled using right triominoes.

To see this, consider a $2^{k+1} \times 2^{k+1}$ checkerboard with one square removed. Split this checkerboard into four checkerboards of size $2^k \times 2^k$, by dividing it in half in both directions. This is illustrated in Figure 6. No square has been removed from three of these four checkerboards. The fourth $2^k \times 2^k$ checkerboard has one square removed, so we now use the inductive hypothesis to conclude that it can be covered by right triominoes. Now temporarily remove the square from each of the other three $2^k \times 2^k$ checkerboards that has the center of the original, larger checkerboard as one of its corners, as shown in Figure 7. By the inductive hypothesis, each of these three $2^k \times 2^k$ checkerboards with a square removed can be tiled by right triominoes. Furthermore, the three squares that were temporarily removed can be covered by one right triomino. Hence, the entire $2^{k+1} \times 2^{k+1}$ checkerboard can be tiled with right triominoes.

We have completed the basis step and the inductive step. Therefore, by mathematical induction $P(n)$ is true for all positive integers n . This shows that we can tile every $2^n \times 2^n$ checkerboard, where n is a positive integer, with one square removed, using right triominoes. \blacktriangleleft

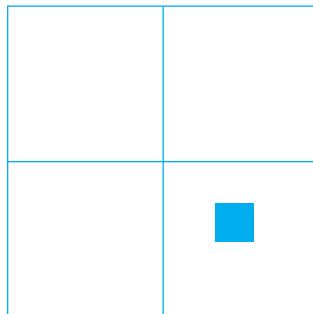


FIGURE 6 Dividing a $2^{k+1} \times 2^{k+1}$ Checkerboard into Four $2^k \times 2^k$ Checkerboards.

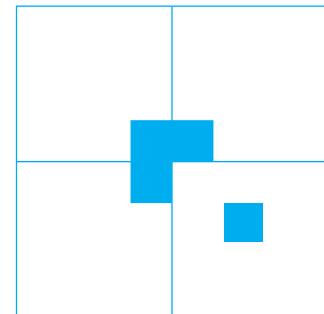


FIGURE 7 Tiling the $2^{k+1} \times 2^{k+1}$ Checkerboard with One Square Removed.

Mistaken Proofs By Mathematical Induction

As with every proof method, there are many opportunities for making errors when using mathematical induction. Many well-known mistaken, and often entertaining, proofs by mathematical induction of clearly false statements have been devised, as exemplified by Example 15 and Exercises 49–51. Often, it is not easy to find where the error in reasoning occurs in such mistaken proofs.

To uncover errors in proofs by mathematical induction, remember that in every such proof, both the basis step and the inductive step must be done correctly. Not completing the basis step in a supposed proof by mathematical induction can lead to mistaken proofs of clearly ridiculous statements such as “ $n = n + 1$ whenever n is a positive integer.” (We leave it to the reader to show that it is easy to construct a correct inductive step in an attempted proof of this statement.) Locating the error in a faulty proof by mathematical induction, as Example 15 illustrates, can be quite tricky, especially when the error is hidden in the basis step.

EXAMPLE 15 Find the error in this “proof” of the clearly false claim that every set of lines in the plane, no two of which are parallel, meet in a common point.

“Proof:” Let $P(n)$ be the statement that every set of n lines in the plane, no two of which are parallel, meet in a common point. We will attempt to prove that $P(n)$ is true for all positive integers $n \geq 2$.

BASIS STEP: The statement $P(2)$ is true because any two lines in the plane that are not parallel meet in a common point (by the definition of parallel lines).

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(k)$ is true for the positive integer k , that is, it is the assumption that every set of k lines in the plane, no two of which are parallel, meet in a common point. To complete the inductive step, we must show that if $P(k)$ is true, then $P(k+1)$ must also be true. That is, we must show that if every set of k lines in the plane, no two of which are parallel, meet in a common point, then every set of $k+1$ lines in the plane, no two of which are parallel, meet in a common point. So, consider a set of $k+1$ distinct lines in the plane. By the inductive hypothesis, the first k of these lines meet in a common point p_1 . Moreover, by the inductive hypothesis, the last k of these lines meet in a common point p_2 . We will show that p_1 and p_2 must be the same point. If p_1 and p_2 were different points, all lines containing both of them must be the same line because two points determine a line. This contradicts our assumption that all these lines are distinct. Thus, p_1 and p_2 are the same point. We conclude that the point $p_1 = p_2$ lies on all $k+1$ lines. We have shown that $P(k+1)$ is true assuming that $P(k)$ is true. That is, we have shown that if we assume that every k , $k \geq 2$, distinct lines meet in a common point, then every $k+1$ distinct lines meet in a common point. This completes the inductive step.

We have completed the basis step and the inductive step, and supposedly we have a correct proof by mathematical induction.

Solution: Examining this supposed proof by mathematical induction it appears that everything is in order. However, there is an error, as there must be. The error is rather subtle. Carefully looking at the inductive step shows that this step requires that $k \geq 3$. We cannot show that $P(2)$ implies $P(3)$. When $k = 2$, our goal is to show that every three distinct lines meet in a common point. The first two lines must meet in a common point p_1 and the last two lines must meet in a common point p_2 . But in this case, p_1 and p_2 do not have to be the same, because only the second line is common to both sets of lines. Here is where the inductive step fails. 

Guidelines for Proofs by Mathematical Induction

Examples 1–14 illustrate proofs by mathematical induction of a diverse collection of theorems. Each of these examples includes all the elements needed in a proof by mathematical induction. We have provided an example of an invalid proof by mathematical induction. Summarizing what we have learned from these examples, we can provide some useful guidelines for constructing correct proofs by mathematical induction. We now present these guidelines.

Template for Proofs by Mathematical Induction

1. Express the statement that is to be proved in the form “for all $n \geq b$, $P(n)$ ” for a fixed integer b .
2. Write out the words “Basis Step.” Then show that $P(b)$ is true, taking care that the correct value of b is used. This completes the first part of the proof.
3. Write out the words “Inductive Step.”
4. State, and clearly identify, the inductive hypothesis, in the form “assume that $P(k)$ is true for an arbitrary fixed integer $k \geq b$.”
5. State what needs to be proved under the assumption that the inductive hypothesis is true. That is, write out what $P(k + 1)$ says.
6. Prove the statement $P(k + 1)$ making use of the assumption $P(k)$. Be sure that your proof is valid for all integers k with $k \geq b$, taking care that the proof works for small values of k , including $k = b$.
7. Clearly identify the conclusion of the inductive step, such as by saying “this completes the inductive step.”
8. After completing the basis step and the inductive step, state the conclusion, namely that by mathematical induction, $P(n)$ is true for all integers n with $n \geq b$.

It is worthwhile to revisit each of the mathematical induction proofs in Examples 1–14 to see how these steps are completed. It will be helpful to follow these guidelines in the solutions of the exercises that ask for proofs by mathematical induction. The guidelines that we presented can be adapted for each of the variants of mathematical induction that we introduce in the exercises and later in this chapter.

Exercises

1. There are infinitely many stations on a train route. Suppose that the train stops at the first station and suppose that if the train stops at a station, then it stops at the next station. Show that the train stops at all stations.
 2. Suppose that you know that a golfer plays the first hole of a golf course with an infinite number of holes and that if this golfer plays one hole, then the golfer goes on to play the next hole. Prove that this golfer plays every hole on the course.
- Use mathematical induction in Exercises 3–17 to prove summation formulae. Be sure to identify where you use the inductive hypothesis.
3. Let $P(n)$ be the statement that $1^2 + 2^2 + \dots + n^2 = n(n + 1)(2n + 1)/6$ for the positive integer n .
 - a) What is the statement $P(1)$?
 - b) Show that $P(1)$ is true, completing the basis step of the proof.
 - c) What is the inductive hypothesis?
 - d) What do you need to prove in the inductive step?
 - e) Complete the inductive step, identifying where you use the inductive hypothesis.
 - f) Explain why these steps show that this formula is true whenever n is a positive integer.
 4. Let $P(n)$ be the statement that $1^3 + 2^3 + \dots + n^3 = (n(n + 1)/2)^2$ for the positive integer n .
 - a) What is the statement $P(1)$?
 - b) Show that $P(1)$ is true, completing the basis step of the proof.
 - c) What is the inductive hypothesis?
 - d) What do you need to prove in the inductive step?
 - e) Complete the inductive step, identifying where you use the inductive hypothesis.
 - f) Explain why these steps show that this formula is true whenever n is a positive integer.
 5. Prove that $1^2 + 3^2 + 5^2 + \dots + (2n + 1)^2 = (n + 1)(2n + 1)(2n + 3)/3$ whenever n is a nonnegative integer.
 6. Prove that $1 \cdot 1! + 2 \cdot 2! + \dots + n \cdot n! = (n + 1)! - 1$ whenever n is a positive integer.
 7. Prove that $3 + 3 \cdot 5 + 3 \cdot 5^2 + \dots + 3 \cdot 5^n = 3(5^{n+1} - 1)/4$ whenever n is a nonnegative integer.
 8. Prove that $2 - 2 \cdot 7 + 2 \cdot 7^2 - \dots + 2(-7)^n = (1 - (-7)^{n+1})/4$ whenever n is a nonnegative integer.

- 9.** a) Find a formula for the sum of the first n even positive integers.
 b) Prove the formula that you conjectured in part (a).

- 10.** a) Find a formula for

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \cdots + \frac{1}{n(n+1)}$$

by examining the values of this expression for small values of n .

- b) Prove the formula you conjectured in part (a).

- 11.** a) Find a formula for

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n}$$

by examining the values of this expression for small values of n .

- b) Prove the formula you conjectured in part (a).

- 12.** Prove that

$$\sum_{j=0}^n \left(-\frac{1}{2}\right)^j = \frac{2^{n+1} + (-1)^n}{3 \cdot 2^n}$$

whenever n is a nonnegative integer.

- 13.** Prove that $1^2 - 2^2 + 3^2 - \cdots + (-1)^{n-1} n^2 = (-1)^{n-1} n(n+1)/2$ whenever n is a positive integer.
14. Prove that for every positive integer n , $\sum_{k=1}^n k2^k = (n-1)2^{n+1} + 2$.

- 15.** Prove that for every positive integer n ,

$$1 \cdot 2 + 2 \cdot 3 + \cdots + n(n+1) = n(n+1)(n+2)/3.$$

- 16.** Prove that for every positive integer n ,

$$\begin{aligned} 1 \cdot 2 \cdot 3 + 2 \cdot 3 \cdot 4 + \cdots + n(n+1)(n+2) \\ = n(n+1)(n+2)(n+3)/4. \end{aligned}$$

- 17.** Prove that $\sum_{j=1}^n j^4 = n(n+1)(2n+1)(3n^2+3n-1)/30$ whenever n is a positive integer.

Use mathematical induction to prove the inequalities in Exercises 18–30.

- 18.** Let $P(n)$ be the statement that $n! < n^n$, where n is an integer greater than 1.
 a) What is the statement $P(2)$?
 b) Show that $P(2)$ is true, completing the basis step of the proof.
 c) What is the inductive hypothesis?
 d) What do you need to prove in the inductive step?
 e) Complete the inductive step.
 f) Explain why these steps show that this inequality is true whenever n is an integer greater than 1.

- 19.** Let $P(n)$ be the statement that

$$1 + \frac{1}{4} + \frac{1}{9} + \cdots + \frac{1}{n^2} < 2 - \frac{1}{n},$$

where n is an integer greater than 1.

- a) What is the statement $P(2)$?
 b) Show that $P(2)$ is true, completing the basis step of the proof.

- c) What is the inductive hypothesis?
 d) What do you need to prove in the inductive step?
 e) Complete the inductive step.
 f) Explain why these steps show that this inequality is true whenever n is an integer greater than 1.

- 20.** Prove that $3^n < n!$ if n is an integer greater than 6.

- 21.** Prove that $2^n > n^2$ if n is an integer greater than 4.

- 22.** For which nonnegative integers n is $n^2 \leq n!$? Prove your answer.

- 23.** For which nonnegative integers n is $2n + 3 \leq 2^n$? Prove your answer.

- 24.** Prove that $1/(2n) \leq [1 \cdot 3 \cdot 5 \cdots (2n-1)]/(2 \cdot 4 \cdots 2n)$ whenever n is a positive integer.

- *25.** Prove that if $h > -1$, then $1 + nh \leq (1+h)^n$ for all nonnegative integers n . This is called **Bernoulli's inequality**.

- *26.** Suppose that a and b are real numbers with $0 < b < a$. Prove that if n is a positive integer, then $a^n - b^n \leq na^{n-1}(a-b)$.

- *27.** Prove that for every positive integer n ,

$$1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \cdots + \frac{1}{\sqrt{n}} > 2(\sqrt{n+1} - 1).$$

- 28.** Prove that $n^2 - 7n + 12$ is nonnegative whenever n is an integer with $n \geq 3$.

In Exercises 29 and 30, H_n denotes the n th harmonic number.

- *29.** Prove that $H_{2^n} \leq 1 + n$ whenever n is a nonnegative integer.

- *30.** Prove that

$$H_1 + H_2 + \cdots + H_n = (n+1)H_n - n.$$

Use mathematical induction in Exercises 31–37 to prove divisibility facts.

- 31.** Prove that 2 divides $n^2 + n$ whenever n is a positive integer.

- 32.** Prove that 3 divides $n^3 + 2n$ whenever n is a positive integer.

- 33.** Prove that 5 divides $n^5 - n$ whenever n is a nonnegative integer.

- 34.** Prove that 6 divides $n^3 - n$ whenever n is a nonnegative integer.

- *35.** Prove that $n^2 - 1$ is divisible by 8 whenever n is an odd positive integer.

- *36.** Prove that 21 divides $4^{n+1} + 5^{2n-1}$ whenever n is a positive integer.

- *37.** Prove that if n is a positive integer, then 133 divides $11^{n+1} + 12^{2n-1}$.

Use mathematical induction in Exercises 38–46 to prove results about sets.

- 38.** Prove that if A_1, A_2, \dots, A_n and B_1, B_2, \dots, B_n are sets such that $A_j \subseteq B_j$ for $j = 1, 2, \dots, n$, then

$$\bigcup_{j=1}^n A_j \subseteq \bigcup_{j=1}^n B_j.$$

- 39.** Prove that if A_1, A_2, \dots, A_n and B_1, B_2, \dots, B_n are sets such that $A_j \subseteq B_j$ for $j = 1, 2, \dots, n$, then

$$\bigcap_{j=1}^n A_j \subseteq \bigcap_{j=1}^n B_j.$$

- 40.** Prove that if A_1, A_2, \dots, A_n and B are sets, then

$$\begin{aligned} (A_1 \cap A_2 \cap \dots \cap A_n) \cup B \\ = (A_1 \cup B) \cap (A_2 \cup B) \cap \dots \cap (A_n \cup B). \end{aligned}$$

- 41.** Prove that if A_1, A_2, \dots, A_n and B are sets, then

$$\begin{aligned} (A_1 \cup A_2 \cup \dots \cup A_n) \cap B \\ = (A_1 \cap B) \cup (A_2 \cap B) \cup \dots \cup (A_n \cap B). \end{aligned}$$

- 42.** Prove that if A_1, A_2, \dots, A_n and B are sets, then

$$\begin{aligned} (A_1 - B) \cap (A_2 - B) \cap \dots \cap (A_n - B) \\ = (A_1 \cap A_2 \cap \dots \cap A_n) - B. \end{aligned}$$

- 43.** Prove that if A_1, A_2, \dots, A_n are subsets of a universal set U , then

$$\overline{\bigcup_{k=1}^n A_k} = \bigcap_{k=1}^n \overline{A_k}.$$

- 44.** Prove that if A_1, A_2, \dots, A_n and B are sets, then

$$\begin{aligned} (A_1 - B) \cup (A_2 - B) \cup \dots \cup (A_n - B) \\ = (A_1 \cup A_2 \cup \dots \cup A_n) - B. \end{aligned}$$

- 45.** Prove that a set with n elements has $n(n - 1)/2$ subsets containing exactly two elements whenever n is an integer greater than or equal to 2.

- *46.** Prove that a set with n elements has $n(n - 1)(n - 2)/6$ subsets containing exactly three elements whenever n is an integer greater than or equal to 3.

In Exercises 47 and 48 we consider the problem of placing towers along a straight road, so that every building on the road receives cellular service. Assume that a building receives cellular service if it is within one mile of a tower.

- 47.** Devise a greedy algorithm that uses the minimum number of towers possible to provide cell service to d buildings located at positions x_1, x_2, \dots, x_d from the start of the road. [Hint: At each step, go as far as possible along the road before adding a tower so as not to leave any buildings without coverage.]

- *48.** Use mathematical induction to prove that the algorithm you devised in Exercise 47 produces an optimal solution, that is, that it uses the fewest towers possible to provide cellular service to all buildings.

Exercises 49–51 present incorrect proofs using mathematical induction. You will need to identify an error in reasoning in each exercise.

- 49.** What is wrong with this “proof” that all horses are the same color?

Let $P(n)$ be the proposition that all the horses in a set of n horses are the same color.

Basis Step: Clearly, $P(1)$ is true.

Inductive Step: Assume that $P(k)$ is true, so that all the horses in any set of k horses are the same color. Consider any $k + 1$ horses; number these as horses 1, 2, 3, ..., k , $k + 1$. Now the first k of these horses all must have the same color, and the last k of these must also have the same color. Because the set of the first k horses and the set of the last k horses overlap, all $k + 1$ must be the same color. This shows that $P(k + 1)$ is true and finishes the proof by induction.

- 50.** What is wrong with this “proof”?

Theorem For every positive integer n , $\sum_{i=1}^n i = (n + \frac{1}{2})^2/2$.

Basis Step: The formula is true for $n = 1$.

Inductive Step: Suppose that $\sum_{i=1}^n i = (n + \frac{1}{2})^2/2$. Then $\sum_{i=1}^{n+1} i = (\sum_{i=1}^n i) + (n + 1)$. By the inductive hypothesis, $\sum_{i=1}^{n+1} i = (n + \frac{1}{2})^2/2 + n + 1 = (n^2 + n + \frac{1}{4})/2 + n + 1 = (n^2 + 3n + \frac{9}{4})/2 = (n + \frac{3}{2})^2/2 = [(n + 1) + \frac{1}{2}]^2/2$, completing the inductive step.

- 51.** What is wrong with this “proof”?

Theorem For every positive integer n , if x and y are positive integers with $\max(x, y) = n$, then $x = y$.

Basis Step: Suppose that $n = 1$. If $\max(x, y) = 1$ and x and y are positive integers, we have $x = 1$ and $y = 1$.

Inductive Step: Let k be a positive integer. Assume that whenever $\max(x, y) = k$ and x and y are positive integers, then $x = y$. Now let $\max(x, y) = k + 1$, where x and y are positive integers. Then $\max(x - 1, y - 1) = k$, so by the inductive hypothesis, $x - 1 = y - 1$. It follows that $x = y$, completing the inductive step.

- 52.** Suppose that m and n are positive integers with $m > n$ and f is a function from $\{1, 2, \dots, m\}$ to $\{1, 2, \dots, n\}$. Use mathematical induction on the variable n to show that f is not one-to-one.

- *53.** Use mathematical induction to show that n people can divide a cake (where each person gets one or more separate pieces of the cake) so that the cake is divided fairly, that is, in the sense that each person thinks he or she got at least $(1/n)$ th of the cake. [Hint: For the inductive step, take a fair division of the cake among the first k people, have each person divide their share into what this person thinks are $k + 1$ equal portions, and then have the $(k + 1)$ st person select a portion from each of the k people. When showing this produces a fair division for $k + 1$ people, suppose that person $k + 1$ thinks that person i got p_i of the cake where $\sum_{i=1}^k p_i = 1$.]

- 54.** Use mathematical induction to show that given a set of $n + 1$ positive integers, none exceeding $2n$, there is at least one integer in this set that divides another integer in the set.

- *55.** A knight on a chessboard can move one space horizontally (in either direction) and two spaces vertically (in either direction) or two spaces horizontally (in either direction) and one space vertically (in either direction). Suppose that we have an infinite chessboard, made up

of all squares (m, n) where m and n are nonnegative integers that denote the row number and the column number of the square, respectively. Use mathematical induction to show that a knight starting at $(0, 0)$ can visit every square using a finite sequence of moves. [Hint: Use induction on the variable $s = m + n$.]

- 56.** Suppose that

$$\mathbf{A} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix},$$

where a and b are real numbers. Show that

$$\mathbf{A}^n = \begin{bmatrix} a^n & 0 \\ 0 & b^n \end{bmatrix}$$

for every positive integer n .

- 57.** (Requires calculus) Use mathematical induction to prove that the derivative of $f(x) = x^n$ equals nx^{n-1} whenever n is a positive integer. (For the inductive step, use the product rule for derivatives.)
- 58.** Suppose that \mathbf{A} and \mathbf{B} are square matrices with the property $\mathbf{AB} = \mathbf{BA}$. Show that $\mathbf{AB}^n = \mathbf{B}^n\mathbf{A}$ for every positive integer n .
- 59.** Suppose that m is a positive integer. Use mathematical induction to prove that if a and b are integers with $a \equiv b \pmod{m}$, then $a^k \equiv b^k \pmod{m}$ whenever k is a nonnegative integer.
- 60.** Use mathematical induction to show that $\neg(p_1 \vee p_2 \vee \dots \vee p_n)$ is equivalent to $\neg p_1 \wedge \neg p_2 \wedge \dots \wedge \neg p_n$ whenever p_1, p_2, \dots, p_n are propositions.

- *61.** Show that

$$\begin{aligned} [(p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3) \wedge \dots \wedge (p_{n-1} \rightarrow p_n)] \\ \rightarrow [(p_1 \wedge p_2 \wedge \dots \wedge p_{n-1}) \rightarrow p_n] \end{aligned}$$

is a tautology whenever p_1, p_2, \dots, p_n are propositions, where $n \geq 2$.

- *62.** Show that n lines separate the plane into $(n^2 + n + 2)/2$ regions if no two of these lines are parallel and no three pass through a common point.
- **63.** Let a_1, a_2, \dots, a_n be positive real numbers. The **arithmetic mean** of these numbers is defined by

$$A = (a_1 + a_2 + \dots + a_n)/n,$$

and the **geometric mean** of these numbers is defined by

$$G = (a_1 a_2 \cdots a_n)^{1/n}.$$

Use mathematical induction to prove that $A \geq G$.

- 64.** Use mathematical induction to prove Lemma 3 of Section 4.3, which states that if p is a prime and $p \mid a_1 a_2 \cdots a_n$, where a_i is an integer for $i = 1, 2, 3, \dots, n$, then $p \mid a_i$ for some integer i .
- 65.** Show that if n is a positive integer, then

$$\sum_{\{a_1, \dots, a_k\} \subseteq \{1, 2, \dots, n\}} \frac{1}{a_1 a_2 \cdots a_k} = n.$$

(Here the sum is over all nonempty subsets of the set of the n smallest positive integers.)

- *66.** Use the well-ordering property to show that the following form of mathematical induction is a valid method to prove that $P(n)$ is true for all positive integers n .

Basis Step: $P(1)$ and $P(2)$ are true.

Inductive Step: For each positive integer k , if $P(k)$ and $P(k+1)$ are both true, then $P(k+2)$ is true.

- 67.** Show that if A_1, A_2, \dots, A_n are sets where $n \geq 2$, and for all pairs of integers i and j with $1 \leq i < j \leq n$ either A_i is a subset of A_j or A_j is a subset of A_i , then there is an integer i , $1 \leq i \leq n$ such that A_i is a subset of A_j for all integers j with $1 \leq j \leq n$.

- *68.** A guest at a party is a **celebrity** if this person is known by every other guest, but knows none of them. There is at most one celebrity at a party, for if there were two, they would know each other. A particular party may have no celebrity. Your assignment is to find the celebrity, if one exists, at a party, by asking only one type of question—asking a guest whether they know a second guest. Everyone must answer your questions truthfully. That is, if Alice and Bob are two people at the party, you can ask Alice whether she knows Bob; she must answer correctly. Use mathematical induction to show that if there are n people at the party, then you can find the celebrity, if there is one, with $3(n - 1)$ questions. [Hint: First ask a question to eliminate one person as a celebrity. Then use the inductive hypothesis to identify a potential celebrity. Finally, ask two more questions to determine whether that person is actually a celebrity.]

Suppose there are n people in a group, each aware of a scandal no one else in the group knows about. These people communicate by telephone; when two people in the group talk, they share information about all scandals each knows about. For example, on the first call, two people share information, so by the end of the call, each of these people knows about two scandals. The **gossip problem** asks for $G(n)$, the minimum number of telephone calls that are needed for all n people to learn about all the scandals. Exercises 69–71 deal with the gossip problem.

- 69.** Find $G(1), G(2), G(3)$, and $G(4)$.

- 70.** Use mathematical induction to prove that $G(n) \leq 2n - 4$ for $n \geq 4$. [Hint: In the inductive step, have a new person call a particular person at the start and at the end.]

- **71.** Prove that $G(n) = 2n - 4$ for $n \geq 4$.

- *72.** Show that it is possible to arrange the numbers $1, 2, \dots, n$ in a row so that the average of any two of these numbers never appears between them. [Hint: Show that it suffices to prove this fact when n is a power of 2. Then use mathematical induction to prove the result when n is a power of 2.]

- *73.** Show that if I_1, I_2, \dots, I_n is a collection of open intervals on the real number line, $n \geq 2$, and every pair of these intervals has a nonempty intersection, that is, $I_i \cap I_j \neq \emptyset$ whenever $1 \leq i \leq n$ and $1 \leq j \leq n$, then the intersection of all these sets is nonempty, that is, $I_1 \cap I_2 \cap \dots \cap I_n \neq \emptyset$. (Recall that an **open interval** is

the set of real numbers x with $a < x < b$, where a and b are real numbers with $a < b$.)

Sometimes we cannot use mathematical induction to prove a result we believe to be true, but we can use mathematical induction to prove a stronger result. Because the inductive hypothesis of the stronger result provides more to work with, this process is called **inductive loading**. We use inductive loading in Exercise 74.

74. Suppose that we want to prove that

$$\frac{1}{2} \cdot \frac{3}{4} \cdots \frac{2n-1}{2n} < \frac{1}{\sqrt{3n}}$$

for all positive integers n .

a) Show that if we try to prove this inequality using mathematical induction, the basis step works, but the inductive step fails.

b) Show that mathematical induction can be used to prove the stronger inequality

$$\frac{1}{2} \cdot \frac{3}{4} \cdots \frac{2n-1}{2n} < \frac{1}{\sqrt{3n+1}}$$

for all integers greater than 1, which, together with a verification for the case where $n = 1$, establishes the weaker inequality we originally tried to prove using mathematical induction.

75. Let n be an even positive integer. Show that when n people stand in a yard at mutually distinct distances and each

person throws a pie at their nearest neighbor, it is possible that everyone is hit by a pie.

76. Construct a tiling using right triominoes of the 4×4 checkerboard with the square in the upper left corner removed.
 77. Construct a tiling using right triominoes of the 8×8 checkerboard with the square in the upper left corner removed.
 78. Prove or disprove that all checkerboards of these shapes can be completely covered using right triominoes whenever n is a positive integer.
 a) 3×2^n b) 6×2^n
 c) $3^n \times 3^n$ d) $6^n \times 6^n$
 *79. Show that a three-dimensional $2^n \times 2^n \times 2^n$ checkerboard with one $1 \times 1 \times 1$ cube missing can be completely covered by $2 \times 2 \times 2$ cubes with one $1 \times 1 \times 1$ cube removed.
 *80. Show that an $n \times n$ checkerboard with one square removed can be completely covered using right triominoes if $n > 5$, n is odd, and $3 \nmid n$.
 81. Show that a 5×5 checkerboard with a corner square removed can be tiled using right triominoes.
 *82. Find a 5×5 checkerboard with a square removed that cannot be tiled using right triominoes. Prove that such a tiling does not exist for this board.
 83. Use the principle of mathematical induction to show that $P(n)$ is true for $n = b, b + 1, b + 2, \dots$, where b is an integer, if $P(b)$ is true and the conditional statement $P(k) \rightarrow P(k + 1)$ is true for all integers k with $k \geq b$.

5.2 Strong Induction and Well-Ordering

Introduction

In Section 5.1 we introduced mathematical induction and we showed how to use it to prove a variety of theorems. In this section we will introduce another form of mathematical induction, called **strong induction**, which can often be used when we cannot easily prove a result using mathematical induction. The basis step of a proof by strong induction is the same as a proof of the same result using mathematical induction. That is, in a strong induction proof that $P(n)$ is true for all positive integers n , the basis step shows that $P(1)$ is true. However, the inductive steps in these two proof methods are different. In a proof by mathematical induction, the inductive step shows that if the inductive hypothesis $P(k)$ is true, then $P(k + 1)$ is also true. In a proof by strong induction, the inductive step shows that if $P(j)$ is true for all positive integers not exceeding k , then $P(k + 1)$ is true. That is, for the inductive hypothesis we assume that $P(j)$ is true for $j = 1, 2, \dots, k$.

The validity of both mathematical induction and strong induction follow from the well-ordering property in Appendix 1. In fact, mathematical induction, strong induction, and well-ordering are all equivalent principles (as shown in Exercises 41, 42, and 43). That is, the validity of each can be proved from either of the other two. This means that a proof using one of these two principles can be rewritten as a proof using either of the other two principles. Just as it is sometimes the case that it is much easier to see how to prove a result using strong induction rather than mathematical induction, it is sometimes easier to use well-ordering than one of the

two forms of mathematical induction. In this section we will give some examples of how the well-ordering property can be used to prove theorems.

Strong Induction

Before we illustrate how to use strong induction, we state this principle again.

STRONG INDUCTION To prove that $P(n)$ is true for all positive integers n , where $P(n)$ is a propositional function, we complete two steps:

BASIS STEP: We verify that the proposition $P(1)$ is true.

INDUCTIVE STEP: We show that the conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ is true for all positive integers k .

Note that when we use strong induction to prove that $P(n)$ is true for all positive integers n , our inductive hypothesis is the assumption that $P(j)$ is true for $j = 1, 2, \dots, k$. That is, the inductive hypothesis includes all k statements $P(1), P(2), \dots, P(k)$. Because we can use all k statements $P(1), P(2), \dots, P(k)$ to prove $P(k+1)$, rather than just the statement $P(k)$ as in a proof by mathematical induction, strong induction is a more flexible proof technique. Because of this, some mathematicians prefer to always use strong induction instead of mathematical induction, even when a proof by mathematical induction is easy to find.

You may be surprised that mathematical induction and strong induction are equivalent. That is, each can be shown to be a valid proof technique assuming that the other is valid. In particular, any proof using mathematical induction can also be considered to be a proof by strong induction because the inductive hypothesis of a proof by mathematical induction is part of the inductive hypothesis in a proof by strong induction. That is, if we can complete the inductive step of a proof using mathematical induction by showing that $P(k+1)$ follows from $P(k)$ for every positive integer k , then it also follows that $P(k+1)$ follows from all the statements $P(1), P(2), \dots, P(k)$, because we are assuming that not only $P(k)$ is true, but also more, namely, that the $k-1$ statements $P(1), P(2), \dots, P(k-1)$ are true. However, it is much more awkward to convert a proof by strong induction into a proof using the principle of mathematical induction. (See Exercise 42.)

Strong induction is sometimes called the **second principle of mathematical induction** or **complete induction**. When the terminology “complete induction” is used, the principle of mathematical induction is called **incomplete induction**, a technical term that is a somewhat unfortunate choice because there is nothing incomplete about the principle of mathematical induction; after all, it is a valid proof technique.

STRONG INDUCTION AND THE INFINITE LADDER To better understand strong induction, consider the infinite ladder in Section 5.1. Strong induction tells us that we can reach all rungs if

1. we can reach the first rung, and
2. for every integer k , if we can reach all the first k rungs, then we can reach the $(k+1)$ st rung.

That is, if $P(n)$ is the statement that we can reach the n th rung of the ladder, by strong induction we know that $P(n)$ is true for all positive integers n , because (1) tells us $P(1)$ is true, completing the basis step and (2) tells us that $P(1) \wedge P(2) \wedge \cdots \wedge P(k)$ implies $P(k+1)$, completing the inductive step.

Example 1 illustrates how strong induction can help us prove a result that cannot easily be proved using the principle of mathematical induction.

EXAMPLE 1 Suppose we can reach the first and second rungs of an infinite ladder, and we know that if we can reach a rung, then we can reach two rungs higher. Can we prove that we can reach every rung using the principle of mathematical induction? Can we prove that we can reach every rung using strong induction?

Solution: We first try to prove this result using the principle of mathematical induction.

BASIS STEP: The basis step of such a proof holds; here it simply verifies that we can reach the first rung.

ATTEMPTED INDUCTIVE STEP: The inductive hypothesis is the statement that we can reach the k th rung of the ladder. To complete the inductive step, we need to show that if we assume the inductive hypothesis for the positive integer k , namely, if we assume that we can reach the k th rung of the ladder, then we can show that we can reach the $(k + 1)$ st rung of the ladder. However, there is no obvious way to complete this inductive step because we do not know from the given information that we can reach the $(k + 1)$ st rung from the k th rung. After all, we only know that if we can reach a rung we can reach the rung two higher.

Now consider a proof using strong induction.

BASIS STEP: The basis step is the same as before; it simply verifies that we can reach the first rung.

INDUCTIVE STEP: The inductive hypothesis states that we can reach each of the first k rungs. To complete the inductive step, we need to show that if we assume that the inductive hypothesis is true, that is, if we can reach each of the first k rungs, then we can reach the $(k + 1)$ st rung. We already know that we can reach the second rung. We can complete the inductive step by noting that as long as $k \geq 2$, we can reach the $(k + 1)$ st rung from the $(k - 1)$ st rung because we know we can climb two rungs from a rung we can already reach, and because $k - 1 \leq k$, by the inductive hypothesis we can reach the $(k - 1)$ st rung. This completes the inductive step and finishes the proof by strong induction.

We have proved that if we can reach the first two rungs of an infinite ladder and for every positive integer k if we can reach all the first k rungs then we can reach the $(k + 1)$ st rung, then we can reach all rungs of the ladder. 

Examples of Proofs Using Strong Induction

Now that we have both mathematical induction and strong induction, how do we decide which method to apply in a particular situation? Although there is no cut-and-dried answer, we can supply some useful pointers. In practice, you should use mathematical induction when it is straightforward to prove that $P(k) \rightarrow P(k + 1)$ is true for all positive integers k . This is the case for all the proofs in the examples in Section 5.1. In general, you should restrict your use of the principle of mathematical induction to such scenarios. Unless you can clearly see that the inductive step of a proof by mathematical induction goes through, you should attempt a proof by strong induction. That is, use strong induction and not mathematical induction when you see how to prove that $P(k + 1)$ is true from the assumption that $P(j)$ is true for all positive integers j not exceeding k , but you cannot see how to prove that $P(k + 1)$ follows from just $P(k)$. Keep this in mind as you examine the proofs in this section. For each of these proofs, consider why strong induction works better than mathematical induction.

We will illustrate how strong induction is employed in Examples 2–4. In these examples, we will prove a diverse collection of results. Pay particular attention to the inductive step in each of these examples, where we show that a result $P(k + 1)$ follows under the assumption that $P(j)$ holds for all positive integers j not exceeding k , where $P(n)$ is a propositional function.

We begin with one of the most prominent uses of strong induction, the part of the fundamental theorem of arithmetic that tells us that every positive integer can be written as the product of primes.

EXAMPLE 2 Show that if n is an integer greater than 1, then n can be written as the product of primes.



Solution: Let $P(n)$ be the proposition that n can be written as the product of primes.

BASIS STEP: $P(2)$ is true, because 2 can be written as the product of one prime, itself. (Note that $P(2)$ is the first case we need to establish.)

INDUCTIVE STEP: The inductive hypothesis is the assumption that $P(j)$ is true for all integers j with $2 \leq j \leq k$, that is, the assumption that j can be written as the product of primes whenever j is a positive integer at least 2 and not exceeding k . To complete the inductive step, it must be shown that $P(k + 1)$ is true under this assumption, that is, that $k + 1$ is the product of primes.

There are two cases to consider, namely, when $k + 1$ is prime and when $k + 1$ is composite. If $k + 1$ is prime, we immediately see that $P(k + 1)$ is true. Otherwise, $k + 1$ is composite and can be written as the product of two positive integers a and b with $2 \leq a \leq b < k + 1$. Because both a and b are integers at least 2 and not exceeding k , we can use the inductive hypothesis to write both a and b as the product of primes. Thus, if $k + 1$ is composite, it can be written as the product of primes, namely, those primes in the factorization of a and those in the factorization of b .

Remark: Because 1 can be thought of as the *empty* product of no primes, we could have started the proof in Example 2 with $P(1)$ as the basis step. We chose not to do so because many people find this confusing.

Example 2 completes the proof of the fundamental theorem of arithmetic, which asserts that every nonnegative integer can be written uniquely as the product of primes in nondecreasing order. We showed in Section 4.3 that an integer has at most one such factorization into primes. Example 2 shows there is at least one such factorization.

Next, we show how strong induction can be used to prove that a player has a winning strategy in a game.

EXAMPLE 3 Consider a game in which two players take turns removing any positive number of matches they want from one of two piles of matches. The player who removes the last match wins the game. Show that if the two piles contain the same number of matches initially, the second player can always guarantee a win.

Solution: Let n be the number of matches in each pile. We will use strong induction to prove $P(n)$, the statement that the second player can win when there are initially n matches in each pile.

BASIS STEP: When $n = 1$, the first player has only one choice, removing one match from one of the piles, leaving a single pile with a single match, which the second player can remove to win the game.

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(j)$ is true for all j with $1 \leq j \leq k$, that is, the assumption that the second player can always win whenever there are j matches, where $1 \leq j \leq k$ in each of the two piles at the start of the game. We need to show that $P(k + 1)$ is true, that is, that the second player can win when there are initially $k + 1$ matches in each pile, under the assumption that $P(j)$ is true for $j = 1, 2, \dots, k$. So suppose that there are $k + 1$ matches in each of the two piles at the start of the game and suppose that the first player removes r matches ($1 \leq r \leq k$) from one of the piles, leaving $k + 1 - r$ matches in this pile. By removing the same number of matches from the other pile, the second player creates the

situation where there are two piles each with $k + 1 - r$ matches. Because $1 \leq k + 1 - r \leq k$, we can now use the inductive hypothesis to conclude that the second player can always win. We complete the proof by noting that if the first player removes all $k + 1$ matches from one of the piles, the second player can win by removing all the remaining matches. 

Using the principle of mathematical induction, instead of strong induction, to prove the results in Examples 2 and 3 is difficult. However, as Example 4 shows, some results can be readily proved using either the principle of mathematical induction or strong induction.

Before we present Example 4, note that we can slightly modify strong induction to handle a wider variety of situations. In particular, we can adapt strong induction to handle cases where the inductive step is valid only for integers greater than a particular integer. Let b be a fixed integer and j a fixed positive integer. The form of strong induction we need tells us that $P(n)$ is true for all integers n with $n \geq b$ if we can complete these two steps:

BASIS STEP: We verify that the propositions $P(b)$, $P(b + 1)$, \dots , $P(b + j)$ are true.

INDUCTIVE STEP: We show that $[P(b) \wedge P(b + 1) \wedge \dots \wedge P(k)] \rightarrow P(k + 1)$ is true for every integer $k \geq b + j$.

We will use this alternative form in the strong induction proof in Example 4. That this alternative form is equivalent to strong induction is left as Exercise 28.

EXAMPLE 4 Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.

Solution: We will prove this result using the principle of mathematical induction. Then we will present a proof using strong induction. Let $P(n)$ be the statement that postage of n cents can be formed using 4-cent and 5-cent stamps.

We begin by using the principle of mathematical induction.

BASIS STEP: Postage of 12 cents can be formed using three 4-cent stamps.

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(k)$ is true. That is, under this hypothesis, postage of k cents can be formed using 4-cent and 5-cent stamps. To complete the inductive step, we need to show that when we assume $P(k)$ is true, then $P(k + 1)$ is also true where $k \geq 12$. That is, we need to show that if we can form postage of k cents, then we can form postage of $k + 1$ cents. So, assume the inductive hypothesis is true; that is, assume that we can form postage of k cents using 4-cent and 5-cent stamps. We consider two cases, when at least one 4-cent stamp has been used and when no 4-cent stamps have been used. First, suppose that at least one 4-cent stamp was used to form postage of k cents. Then we can replace this stamp with a 5-cent stamp to form postage of $k + 1$ cents. But if no 4-cent stamps were used, we can form postage of k cents using only 5-cent stamps. Moreover, because $k \geq 12$, we needed at least three 5-cent stamps to form postage of k cents. So, we can replace three 5-cent stamps with four 4-cent stamps to form postage of $k + 1$ cents. This completes the inductive step.

Because we have completed the basis step and the inductive step, we know that $P(n)$ is true for all $n \geq 12$. That is, we can form postage of n cents, where $n \geq 12$ using just 4-cent and 5-cent stamps. This completes the proof by mathematical induction.

Next, we will use strong induction to prove the same result. In this proof, in the basis step we show that $P(12)$, $P(13)$, $P(14)$, and $P(15)$ are true, that is, that postage of 12, 13, 14, or 15 cents can be formed using just 4-cent and 5-cent stamps. In the inductive step we show how to get postage of $k + 1$ cents for $k \geq 15$ from postage of $k - 3$ cents.

BASIS STEP: We can form postage of 12, 13, 14, and 15 cents using three 4-cent stamps, two 4-cent stamps and one 5-cent stamp, one 4-cent stamp and two 5-cent stamps, and three 5-cent stamps, respectively. This shows that $P(12)$, $P(13)$, $P(14)$, and $P(15)$ are true. This completes the basis step.

INDUCTIVE STEP: The inductive hypothesis is the statement that $P(j)$ is true for $12 \leq j \leq k$, where k is an integer with $k \geq 15$. To complete the inductive step, we assume that we can form postage of j cents, where $12 \leq j \leq k$. We need to show that under the assumption that $P(k+1)$ is true, we can also form postage of $k+1$ cents. Using the inductive hypothesis, we can assume that $P(k-3)$ is true because $k-3 \geq 12$, that is, we can form postage of $k-3$ cents using just 4-cent and 5-cent stamps. To form postage of $k+1$ cents, we need only add another 4-cent stamp to the stamps we used to form postage of $k-3$ cents. That is, we have shown that if the inductive hypothesis is true, then $P(k+1)$ is also true. This completes the inductive step.

Because we have completed the basis step and the inductive step of a strong induction proof, we know by strong induction that $P(n)$ is true for all integers n with $n \geq 12$. That is, we know that every postage of n cents, where n is at least 12, can be formed using 4-cent and 5-cent stamps. This finishes the proof by strong induction.

(There are other ways to approach this problem besides those described here. Can you find a solution that does not use mathematical induction?) 

Using Strong Induction in Computational Geometry

Our next example of strong induction will come from **computational geometry**, the part of discrete mathematics that studies computational problems involving geometric objects. Computational geometry is used extensively in computer graphics, computer games, robotics, scientific calculations, and a vast array of other areas. Before we can present this result, we introduce some terminology, possibly familiar from earlier studies in geometry.

A **polygon** is a closed geometric figure consisting of a sequence of line segments s_1, s_2, \dots, s_n , called **sides**. Each pair of consecutive sides, s_i and s_{i+1} , $i = 1, 2, \dots, n-1$, as well as the last side s_n and the first side s_1 , of the polygon meet at a common endpoint, called a **vertex**. A polygon is called **simple** if no two nonconsecutive sides intersect. Every simple polygon divides the plane into two regions: its **interior**, consisting of the points inside the curve, and its **exterior**, consisting of the points outside the curve. This last fact is surprisingly complicated to prove. It is a special case of the famous Jordan curve theorem, which tells us that every simple curve divides the plane into two regions; see [Or00], for example.

A polygon is called **convex** if every line segment connecting two points in the interior of the polygon lies entirely inside the polygon. (A polygon that is not convex is said to be **nonconvex**.) Figure 1 displays some polygons; polygons (a) and (b) are convex, but polygons (c) and (d) are not. A **diagonal** of a simple polygon is a line segment connecting two nonconsecutive vertices of the polygon, and a diagonal is called an **interior diagonal** if it lies entirely inside the polygon, except for its endpoints. For example, in polygon (d), the line segment connecting a and f is an interior diagonal, but the line segment connecting a and d is a diagonal that is not an interior diagonal.

One of the most basic operations of computational geometry involves dividing a simple polygon into triangles by adding nonintersecting diagonals. This process is called **triangulation**. Note that a simple polygon can have many different triangulations, as shown in Figure 2. Perhaps the most basic fact in computational geometry is that it is possible to triangulate every simple

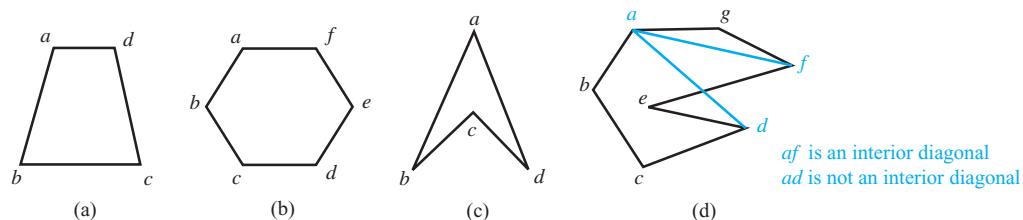
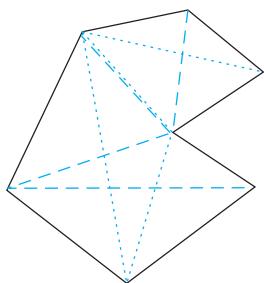


FIGURE 1 Convex and Nonconvex Polygons.



Two different triangulations of a simple polygon with seven sides into five triangles, shown with dotted lines and with dashed lines, respectively

FIGURE 2 Triangulations of a Polygon.

polygon, as we state in Theorem 1. Furthermore, this theorem tells us that every triangulation of a simple polygon with n sides includes $n - 2$ triangles.

THEOREM 1

A simple polygon with n sides, where n is an integer with $n \geq 3$, can be triangulated into $n - 2$ triangles.

It seems obvious that we should be able to triangulate a simple polygon by successively adding interior diagonals. Consequently, a proof by strong induction seems promising. However, such a proof requires this crucial lemma.

LEMMA 1

Every simple polygon with at least four sides has an interior diagonal.

Although Lemma 1 seems particularly simple, it is surprisingly tricky to prove. In fact, as recently as 30 years ago, a variety of incorrect proofs thought to be correct were commonly seen in books and articles. We defer the proof of Lemma 1 until after we prove Theorem 1. It is not uncommon to prove a theorem pending the later proof of an important lemma.

Proof (of Theorem 1): We will prove this result using strong induction. Let $T(n)$ be the statement that every simple polygon with n sides can be triangulated into $n - 2$ triangles.

BASIS STEP: $T(3)$ is true because a simple polygon with three sides is a triangle. We do not need to add any diagonals to triangulate a triangle; it is already triangulated into one triangle, itself. Consequently, every simple polygon with $n = 3$ has can be triangulated into $n - 2 = 3 - 2 = 1$ triangle.

INDUCTIVE STEP: For the inductive hypothesis, we assume that $T(j)$ is true for all integers j with $3 \leq j \leq k$. That is, we assume that we can triangulate a simple polygon with j sides into $j - 2$ triangles whenever $3 \leq j \leq k$. To complete the inductive step, we must show that when we assume the inductive hypothesis, $P(k + 1)$ is true, that is, that every simple polygon with $k + 1$ sides can be triangulated into $(k + 1) - 2 = k - 1$ triangles.

So, suppose that we have a simple polygon P with $k + 1$ sides. Because $k + 1 \geq 4$, Lemma 1 tells us that P has an interior diagonal ab . Now, ab splits P into two simple polygons Q , with s sides, and R , with t sides. The sides of Q and R are the sides of P , together with the side ab , which is a side of both Q and R . Note that $3 \leq s \leq k$ and $3 \leq t \leq k$ because both Q and R have at least one fewer side than P does (after all, each of these is formed from P by deleting at least two sides and replacing these sides by the diagonal ab). Furthermore, the number of sides of P is two less than the sum of the numbers of sides of Q and the number of

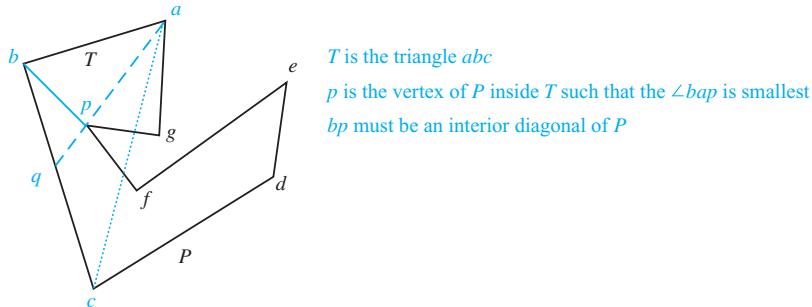


FIGURE 3 Constructing an Interior Diagonal of a Simple Polygon.

sides of R , because each side of P is a side of either Q or of R , but not both, and the diagonal ab is a side of both Q and R , but not P . That is, $k + 1 = s + t - 2$.

We now use the inductive hypothesis. Because both $3 \leq s \leq k$ and $3 \leq t \leq k$, by the inductive hypothesis we can triangulate Q and R into $s - 2$ and $t - 2$ triangles, respectively. Next, note that these triangulations together produce a triangulation of P . (Each diagonal added to triangulate one of these smaller polygons is also a diagonal of P .) Consequently, we can triangulate P into a total of $(s - 2) + (t - 2) = s + t - 4 = (k + 1) - 2$ triangles. This completes the proof by strong induction. That is, we have shown that every simple polygon with n sides, where $n \geq 3$, can be triangulated into $n - 2$ triangles. \triangleleft

We now return to our proof of Lemma 1. We present a proof published by Chung-Wu Ho [Ho75]. Note that although this proof may be omitted without loss of continuity, it does provide a correct proof of a result proved incorrectly by many mathematicians.



Proof: Suppose that P is a simple polygon drawn in the plane. Furthermore, suppose that b is the point of P or in the interior of P with the least y -coordinate among the vertices with the smallest x -coordinate. Then b must be a vertex of P , for if it is an interior point, there would have to be a vertex of P with a smaller x -coordinate. Two other vertices each share an edge with b , say a and c . It follows that the angle in the interior of P formed by ab and bc must be less than 180 degrees (otherwise, there would be points of P with smaller x -coordinates than b).

Now let T be the triangle $\triangle abc$. If there are no vertices of P on or inside T , we can connect a and c to obtain an interior diagonal. On the other hand, if there are vertices of P inside T , we will find a vertex p of P on or inside T such that bp is an interior diagonal. (This is the tricky part. Ho noted that in many published proofs of this lemma a vertex p was found such that bp was not necessarily an interior diagonal of P . See Exercise 21.) The key is to select a vertex p such that the angle $\angle bap$ is smallest. To see this, note that the ray starting at a and passing through p hits the line segment bc at a point, say q . It then follows that the triangle $\triangle baq$ cannot contain any vertices of P in its interior. Hence, we can connect b and p to produce an interior diagonal of P . Locating this vertex p is illustrated in Figure 3. \triangleleft

Proofs Using the Well-Ordering Property

The validity of both the principle of mathematical induction and strong induction follows from a fundamental axiom of the set of integers, the **well-ordering property** (see Appendix 1). The well-ordering property states that every nonempty set of nonnegative integers has a least element. We will show how the well-ordering property can be used directly in proofs. Furthermore, it can be shown (see Exercises 41, 42, and 43) that the well-ordering property, the principle of mathematical induction, and strong induction are all equivalent. That is, the validity of each of these three proof techniques implies the validity of the other two techniques. In Section 5.1 we

showed that the principle of mathematical induction follows from the well-ordering property. The other parts of this equivalence are left as Exercises 31, 42, and 43.

THE WELL-ORDERING PROPERTY Every nonempty set of nonnegative integers has a least element.

The well-ordering property can often be used directly in proofs.

EXAMPLE 5 Use the well-ordering property to prove the division algorithm. Recall that the division algorithm states that if a is an integer and d is a positive integer, then there are unique integers q and r with $0 \leq r < d$ and $a = dq + r$.



Solution: Let S be the set of nonnegative integers of the form $a - dq$, where q is an integer. This set is nonempty because $-dq$ can be made as large as desired (taking q to be a negative integer with large absolute value). By the well-ordering property, S has a least element $r = a - dq_0$.

The integer r is nonnegative. It is also the case that $r < d$. If it were not, then there would be a smaller nonnegative element in S , namely, $a - d(q_0 + 1)$. To see this, suppose that $r \geq d$. Because $a = dq_0 + r$, it follows that $a - d(q_0 + 1) = (a - dq_0) - d = r - d \geq 0$. Consequently, there are integers q and r with $0 \leq r < d$. The proof that q and r are unique is left as Exercise 37.

EXAMPLE 6 In a round-robin tournament every player plays every other player exactly once and each match has a winner and a loser. We say that the players p_1, p_2, \dots, p_m form a *cycle* if p_1 beats p_2 , p_2 beats p_3, \dots, p_{m-1} beats p_m , and p_m beats p_1 . Use the well-ordering principle to show that if there is a cycle of length m ($m \geq 3$) among the players in a round-robin tournament, there must be a cycle of three of these players.

Solution: We assume that there is no cycle of three players. Because there is at least one cycle in the round-robin tournament, the set of all positive integers n for which there is a cycle of length n is nonempty. By the well-ordering property, this set of positive integers has a least element k , which by assumption must be greater than three. Consequently, there exists a cycle of players $p_1, p_2, p_3, \dots, p_k$ and no shorter cycle exists.

Because there is no cycle of three players, we know that $k > 3$. Consider the first three elements of this cycle, p_1, p_2 , and p_3 . There are two possible outcomes of the match between p_1 and p_3 . If p_3 beats p_1 , it follows that p_1, p_2, p_3 is a cycle of length three, contradicting our assumption that there is no cycle of three players. Consequently, it must be the case that p_1 beats p_3 . This means that we can omit p_2 from the cycle $p_1, p_2, p_3, \dots, p_k$ to obtain the cycle $p_1, p_3, p_4, \dots, p_k$ of length $k - 1$, contradicting the assumption that the smallest cycle has length k . We conclude that there must be a cycle of length three.

Exercises

- Use strong induction to show that if you can run one mile or two miles, and if you can always run two more miles once you have run a specified number of miles, then you can run any number of miles.
- Use strong induction to show that all dominoes fall in an infinite arrangement of dominoes if you know that the first three dominoes fall, and that when a domino falls, the domino three farther down in the arrangement also falls.
- Let $P(n)$ be the statement that a postage of n cents can be formed using just 3-cent stamps and 5-cent stamps. The parts of this exercise outline a strong induction proof that $P(n)$ is true for $n \geq 8$.
 - Show that the statements $P(8)$, $P(9)$, and $P(10)$ are true, completing the basis step of the proof.
 - What is the inductive hypothesis of the proof?
 - What do you need to prove in the inductive step?
 - Complete the inductive step for $k \geq 10$.
 - Explain why these steps show that this statement is true whenever $n \geq 8$.
- Let $P(n)$ be the statement that a postage of n cents can be formed using just 4-cent stamps and 7-cent stamps. The

parts of this exercise outline a strong induction proof that $P(n)$ is true for $n \geq 18$.

- a) Show statements $P(18)$, $P(19)$, $P(20)$, and $P(21)$ are true, completing the basis step of the proof.
 - b) What is the inductive hypothesis of the proof?
 - c) What do you need to prove in the inductive step?
 - d) Complete the inductive step for $k \geq 21$.
 - e) Explain why these steps show that this statement is true whenever $n \geq 18$.
5. a) Determine which amounts of postage can be formed using just 4-cent and 11-cent stamps.
- b) Prove your answer to (a) using the principle of mathematical induction. Be sure to state explicitly your inductive hypothesis in the inductive step.
 - c) Prove your answer to (a) using strong induction. How does the inductive hypothesis in this proof differ from that in the inductive hypothesis for a proof using mathematical induction?
6. a) Determine which amounts of postage can be formed using just 3-cent and 10-cent stamps.
- b) Prove your answer to (a) using the principle of mathematical induction. Be sure to state explicitly your inductive hypothesis in the inductive step.
 - c) Prove your answer to (a) using strong induction. How does the inductive hypothesis in this proof differ from that in the inductive hypothesis for a proof using mathematical induction?
7. Which amounts of money can be formed using just two-dollar bills and five-dollar bills? Prove your answer using strong induction.
8. Suppose that a store offers gift certificates in denominations of 25 dollars and 40 dollars. Determine the possible total amounts you can form using these gift certificates. Prove your answer using strong induction.
- *9. Use strong induction to prove that $\sqrt{2}$ is irrational. [Hint: Let $P(n)$ be the statement that $\sqrt{2} \neq n/b$ for any positive integer b .]
10. Assume that a chocolate bar consists of n squares arranged in a rectangular pattern. The entire bar, a smaller rectangular piece of the bar, can be broken along a vertical or a horizontal line separating the squares. Assuming that only one piece can be broken at a time, determine how many breaks you must successively make to break the bar into n separate squares. Use strong induction to prove your answer.
11. Consider this variation of the game of Nim. The game begins with n matches. Two players take turns removing matches, one, two, or three at a time. The player removing the last match loses. Use strong induction to show that if each player plays the best strategy possible, the first player wins if $n = 4j$, $4j + 2$, or $4j + 3$ for some nonnegative integer j and the second player wins in the remaining case when $n = 4j + 1$ for some nonnegative integer j .

12. Use strong induction to show that every positive integer n can be written as a sum of distinct powers of two, that is, as a sum of a subset of the integers $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, and so on. [Hint: For the inductive step, separately consider the case where $k + 1$ is even and where it is odd. When it is even, note that $(k + 1)/2$ is an integer.]

*13. A jigsaw puzzle is put together by successively joining pieces that fit together into blocks. A move is made each time a piece is added to a block, or when two blocks are joined. Use strong induction to prove that no matter how the moves are carried out, exactly $n - 1$ moves are required to assemble a puzzle with n pieces.

14. Suppose you begin with a pile of n stones and split this pile into n piles of one stone each by successively splitting a pile of stones into two smaller piles. Each time you split a pile you multiply the number of stones in each of the two smaller piles you form, so that if these piles have r and s stones in them, respectively, you compute rs . Show that no matter how you split the piles, the sum of the products computed at each step equals $n(n - 1)/2$.

15. Prove that the first player has a winning strategy for the game of Chomp, introduced in Example 12 in Section 1.8, if the initial board is square. [Hint: Use strong induction to show that this strategy works. For the first move, the first player chomps all cookies except those in the left and top edges. On subsequent moves, after the second player has chomped cookies on either the top or left edge, the first player chomps cookies in the same relative positions in the left or top edge, respectively.]

*16. Prove that the first player has a winning strategy for the game of Chomp, introduced in Example 12 in Section 1.8, if the initial board is two squares wide, that is, a $2 \times n$ board. [Hint: Use strong induction. The first move of the first player should be to chomp the cookie in the bottom row at the far right.]

17. Use strong induction to show that if a simple polygon with at least four sides is triangulated, then at least two of the triangles in the triangulation have two sides that border the exterior of the polygon.

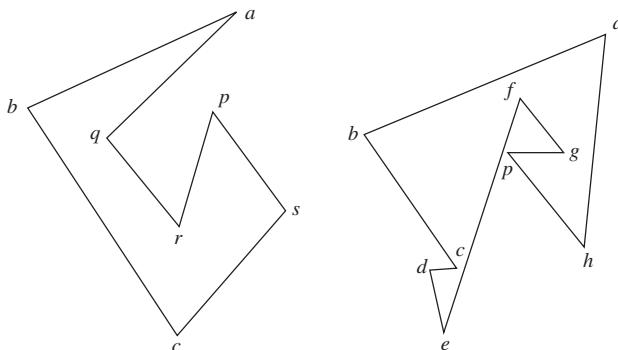
*18. Use strong induction to show that when a simple polygon P with consecutive vertices v_1, v_2, \dots, v_n is triangulated into $n - 2$ triangles, the $n - 2$ triangles can be numbered $1, 2, \dots, n - 2$ so that v_i is a vertex of triangle i for $i = 1, 2, \dots, n - 2$.

*19. **Pick's theorem** says that the area of a simple polygon P in the plane with vertices that are all lattice points (that is, points with integer coordinates) equals $I(P) + B(P)/2 - 1$, where $I(P)$ and $B(P)$ are the number of lattice points in the interior of P and on the boundary of P , respectively. Use strong induction on the number of vertices of P to prove Pick's theorem. [Hint: For the basis step, first prove the theorem for rectangles, then for right triangles, and finally for all triangles by noting that the area of a triangle is the area of a larger rectangle containing it with the areas of at most three triangles subtracted. For the inductive step, take advantage of Lemma 1.]

****20.** Suppose that P is a simple polygon with vertices v_1, v_2, \dots, v_n listed so that consecutive vertices are connected by an edge, and v_1 and v_n are connected by an edge. A vertex v_i is called an **ear** if the line segment connecting the two vertices adjacent to v_i is an interior diagonal of the simple polygon. Two ears v_i and v_j are called **nonoverlapping** if the interiors of the triangles with vertices v_i and its two adjacent vertices and v_j and its two adjacent vertices do not intersect. Prove that every simple polygon with at least four vertices has at least two nonoverlapping ears.

21. In the proof of Lemma 1 we mentioned that many incorrect methods for finding a vertex p such that the line segment bp is an interior diagonal of P have been published. This exercise presents some of the incorrect ways p has been chosen in these proofs. Show, by considering one of the polygons drawn here, that for each of these choices of p , the line segment bp is not necessarily an interior diagonal of P .

- a) p is the vertex of P such that the angle $\angle abp$ is smallest.
- b) p is the vertex of P with the least x -coordinate (other than b).
- c) p is the vertex of P that is closest to b .



Exercises 22 and 23 present examples that show inductive loading can be used to prove results in computational geometry.

***22.** Let $P(n)$ be the statement that when nonintersecting diagonals are drawn inside a convex polygon with n sides, at least two vertices of the polygon are not endpoints of any of these diagonals.

- a) Show that when we attempt to prove $P(n)$ for all integers n with $n \geq 3$ using strong induction, the inductive step does not go through.
- b) Show that we can prove that $P(n)$ is true for all integers n with $n \geq 3$ by proving by strong induction the stronger assertion $Q(n)$, for $n \geq 4$, where $Q(n)$ states that whenever nonintersecting diagonals are drawn inside a convex polygon with n sides, at least two *non-adjacent* vertices are not endpoints of any of these diagonals.

23. Let $E(n)$ be the statement that in a triangulation of a simple polygon with n sides, at least one of the triangles in the triangulation has two sides bordering the exterior of the polygon.

a) Explain where a proof using strong induction that $E(n)$ is true for all integers $n \geq 4$ runs into difficulties.

b) Show that we can prove that $E(n)$ is true for all integers $n \geq 4$ by proving by strong induction the stronger statement $T(n)$ for all integers $n \geq 4$, which states that in every triangulation of a simple polygon, at least two of the triangles in the triangulation have two sides bordering the exterior of the polygon.

***24.** A stable assignment, defined in the preamble to Exercise 60 in Section 3.1, is called **optimal for suitors** if no stable assignment exists in which a suitor is paired with a suitee whom this suitor prefers to the person to whom this suitor is paired in this stable assignment. Use strong induction to show that the deferred acceptance algorithm produces a stable assignment that is optimal for suitors.

25. Suppose that $P(n)$ is a propositional function. Determine for which positive integers n the statement $P(n)$ must be true, and justify your answer, if

- a) $P(1)$ is true; for all positive integers n , if $P(n)$ is true, then $P(n + 2)$ is true.
- b) $P(1)$ and $P(2)$ are true; for all positive integers n , if $P(n)$ and $P(n + 1)$ are true, then $P(n + 2)$ is true.
- c) $P(1)$ is true; for all positive integers n , if $P(n)$ is true, then $P(2n)$ is true.
- d) $P(1)$ is true; for all positive integers n , if $P(n)$ is true, then $P(n + 1)$ is true.

26. Suppose that $P(n)$ is a propositional function. Determine for which nonnegative integers n the statement $P(n)$ must be true if

- a) $P(0)$ is true; for all nonnegative integers n , if $P(n)$ is true, then $P(n + 2)$ is true.
- b) $P(0)$ is true; for all nonnegative integers n , if $P(n)$ is true, then $P(n + 3)$ is true.
- c) $P(0)$ and $P(1)$ are true; for all nonnegative integers n , if $P(n)$ and $P(n + 1)$ are true, then $P(n + 2)$ is true.
- d) $P(0)$ is true; for all nonnegative integers n , if $P(n)$ is true, then $P(n + 2)$ and $P(n + 3)$ are true.

27. Show that if the statement $P(n)$ is true for infinitely many positive integers n and $P(n + 1) \rightarrow P(n)$ is true for all positive integers n , then $P(n)$ is true for all positive integers n .

28. Let b be a fixed integer and j a fixed positive integer. Show that if $P(b), P(b + 1), \dots, P(b + j)$ are true and $[P(b) \wedge P(b + 1) \wedge \dots \wedge P(k)] \rightarrow P(k + 1)$ is true for every integer $k \geq b + j$, then $P(n)$ is true for all integers n with $n \geq b$.

29. What is wrong with this “proof” by strong induction?

“Theorem” For every nonnegative integer n , $5n = 0$.

Basis Step: $5 \cdot 0 = 0$.

Inductive Step: Suppose that $5j = 0$ for all nonnegative integers j with $0 \leq j \leq k$. Write $k + 1 = i + j$, where i and j are natural numbers less than $k + 1$. By the inductive hypothesis, $5(k + 1) = 5(i + j) = 5i + 5j = 0 + 0 = 0$.

- *30. Find the flaw with the following “proof” that $a^n = 1$ for all nonnegative integers n , whenever a is a nonzero real number.

Basis Step: $a^0 = 1$ is true by the definition of a^0 .

Inductive Step: Assume that $a^j = 1$ for all nonnegative integers j with $j \leq k$. Then note that

$$a^{k+1} = \frac{a^k \cdot a^k}{a^{k-1}} = \frac{1 \cdot 1}{1} = 1.$$

- *31. Show that strong induction is a valid method of proof by showing that it follows from the well-ordering property.

32. Find the flaw with the following “proof” that every postage of three cents or more can be formed using just three-cent and four-cent stamps.

Basis Step: We can form postage of three cents with a single three-cent stamp and we can form postage of four cents using a single four-cent stamp.

Inductive Step: Assume that we can form postage of j cents for all nonnegative integers j with $j \leq k$ using just three-cent and four-cent stamps. We can then form postage of $k + 1$ cents by replacing one three-cent stamp with a four-cent stamp or by replacing two four-cent stamps by three three-cent stamps.

33. Show that we can prove that $P(n, k)$ is true for all pairs of positive integers n and k if we show

- a) $P(1, 1)$ is true and $P(n, k) \rightarrow [P(n + 1, k) \wedge P(n, k + 1)]$ is true for all positive integers n and k .
- b) $P(1, k)$ is true for all positive integers k , and $P(n, k) \rightarrow P(n + 1, k)$ is true for all positive integers n and k .
- c) $P(n, 1)$ is true for all positive integers n , and $P(n, k) \rightarrow P(n, k + 1)$ is true for all positive integers n and k .

34. Prove that $\sum_{j=1}^n j(j + 1)(j + 2) \cdots (j + k - 1) = n(n + 1)(n + 2) \cdots (n + k)/(k + 1)$ for all positive integers k and n . [Hint: Use a technique from Exercise 33.]

- *35. Show that if a_1, a_2, \dots, a_n are n distinct real numbers, exactly $n - 1$ multiplications are used to compute the product of these n numbers no matter how parentheses are inserted into their product. [Hint: Use strong induction and consider the last multiplication.]

- *36. The well-ordering property can be used to show that there is a unique greatest common divisor of two positive integers. Let a and b be positive integers, and let S be

the set of positive integers of the form $as + bt$, where s and t are integers.

- a) Show that S is nonempty.
- b) Use the well-ordering property to show that S has a smallest element c .
- c) Show that if d is a common divisor of a and b , then d is a divisor of c .
- d) Show that $c \mid a$ and $c \mid b$. [Hint: First, assume that $c \nmid a$. Then $a = qc + r$, where $0 < r < c$. Show that $r \in S$, contradicting the choice of c .]
- e) Conclude from (c) and (d) that the greatest common divisor of a and b exists. Finish the proof by showing that this greatest common divisor is unique.
- 37. Let a be an integer and d be a positive integer. Show that the integers q and r with $a = dq + r$ and $0 \leq r < d$, which were shown to exist in Example 5, are unique.
- 38. Use mathematical induction to show that a rectangular checkerboard with an even number of cells and two squares missing, one white and one black, can be covered by dominoes.
- **39. Can you use the well-ordering property to prove the statement: “Every positive integer can be described using no more than fifteen English words”? Assume the words come from a particular dictionary of English. [Hint: Suppose that there are positive integers that cannot be described using no more than fifteen English words. By well ordering, the smallest positive integer that cannot be described using no more than fifteen English words would then exist.]
- 40. Use the well-ordering principle to show that if x and y are real numbers with $x < y$, then there is a rational number r with $x < r < y$. [Hint: Use the Archimedean property, given in Appendix 1, to find a positive integer A with $A > 1/(y - x)$. Then show that there is a rational number r with denominator A between x and y by looking at the numbers $\lfloor x \rfloor + j/A$, where j is a positive integer.]
- *41. Show that the well-ordering property can be proved when the principle of mathematical induction is taken as an axiom.
- *42. Show that the principle of mathematical induction and strong induction are equivalent; that is, each can be shown to be valid from the other.
- *43. Show that we can prove the well-ordering property when we take strong induction as an axiom instead of taking the well-ordering property as an axiom.

5.3 Recursive Definitions and Structural Induction

Introduction

Sometimes it is difficult to define an object explicitly. However, it may be easy to define this object in terms of itself. This process is called **recursion**. For instance, the picture shown in Figure 1 is produced recursively. First, an original picture is given. Then a process of successively superimposing centered smaller pictures on top of the previous pictures is carried out.

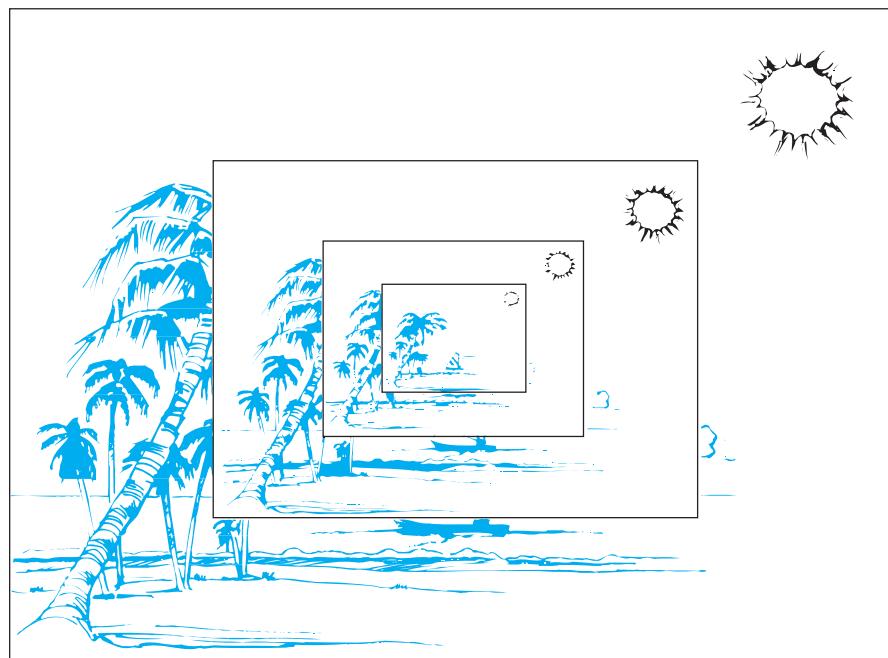


FIGURE 1 A Recursively Defined Picture.

We can use recursion to define sequences, functions, and sets. In Section 2.4, and in most beginning mathematics courses, the terms of a sequence are specified using an explicit formula. For instance, the sequence of powers of 2 is given by $a_n = 2^n$ for $n = 0, 1, 2, \dots$. Recall from Section 2.4 that we can also define a sequence recursively by specifying how terms of the sequence are found from previous terms. The sequence of powers of 2 can also be defined by giving the first term of the sequence, namely, $a_0 = 1$, and a rule for finding a term of the sequence from the previous one, namely, $a_{n+1} = 2a_n$ for $n = 0, 1, 2, \dots$. When we define a sequence recursively by specifying how terms of the sequence are found from previous terms, we can use induction to prove results about the sequence.

When we define a set recursively, we specify some initial elements in a basis step and provide a rule for constructing new elements from those we already have in the recursive step. To prove results about recursively defined sets we use a method called *structural induction*.

Recursively Defined Functions

We use two steps to define a function with the set of nonnegative integers as its domain:

BASIS STEP: Specify the value of the function at zero.



RECURSIVE STEP: Give a rule for finding its value at an integer from its values at smaller integers.

Such a definition is called a **recursive or inductive definition**. Note that a function $f(n)$ from the set of nonnegative integers to the set of real numbers is the same as a sequence a_0, a_1, \dots where a_i is a real number for every nonnegative integer i . So, defining a real-valued sequence a_0, a_1, \dots using a recurrence relation, as was done in Section 2.4, is the same as defining a function from the set of nonnegative integers to the set of real numbers.

EXAMPLE 1 Suppose that f is defined recursively by



$$\begin{aligned}f(0) &= 3, \\f(n+1) &= 2f(n) + 3.\end{aligned}$$

Find $f(1)$, $f(2)$, $f(3)$, and $f(4)$.

Solution: From the recursive definition it follows that

$$\begin{aligned}f(1) &= 2f(0) + 3 = 2 \cdot 3 + 3 = 9, \\f(2) &= 2f(1) + 3 = 2 \cdot 9 + 3 = 21, \\f(3) &= 2f(2) + 3 = 2 \cdot 21 + 3 = 45, \\f(4) &= 2f(3) + 3 = 2 \cdot 45 + 3 = 93.\end{aligned}$$



Recursively defined functions are **well defined**. That is, for every positive integer, the value of the function at this integer is determined in an unambiguous way. This means that given any positive integer, we can use the two parts of the definition to find the value of the function at that integer, and that we obtain the same value no matter how we apply the two parts of the definition. This is a consequence of the principle of mathematical induction. (See Exercise 56.) Additional examples of recursive definitions are given in Examples 2 and 3.

EXAMPLE 2 Give a recursive definition of a^n , where a is a nonzero real number and n is a nonnegative integer.

Solution: The recursive definition contains two parts. First a^0 is specified, namely, $a^0 = 1$. Then the rule for finding a^{n+1} from a^n , namely, $a^{n+1} = a \cdot a^n$, for $n = 0, 1, 2, 3, \dots$, is given. These two equations uniquely define a^n for all nonnegative integers n .



EXAMPLE 3 Give a recursive definition of

$$\sum_{k=0}^n a_k.$$

Solution: The first part of the recursive definition is

$$\sum_{k=0}^0 a_k = a_0.$$

The second part is

$$\sum_{k=0}^{n+1} a_k = \left(\sum_{k=0}^n a_k \right) + a_{n+1}.$$



In some recursive definitions of functions, the values of the function at the first k positive integers are specified, and a rule is given for determining the value of the function at larger integers from its values at some or all of the preceding k integers. That recursive definitions defined in this way produce well-defined functions follows from strong induction (see Exercise 57).

Recall from Section 2.4 that the Fibonacci numbers, f_0, f_1, f_2, \dots , are defined by the equations $f_0 = 0$, $f_1 = 1$, and



$$f_n = f_{n-1} + f_{n-2}$$

for $n = 2, 3, 4, \dots$ [We can think of the Fibonacci number f_n either as the n th term of the sequence of Fibonacci numbers f_0, f_1, \dots or as the value at the integer n of a function $f(n)$.]

We can use the recursive definition of the Fibonacci numbers to prove many properties of these numbers. We give one such property in Example 4.

EXAMPLE 4 Show that whenever $n \geq 3$, $f_n > \alpha^{n-2}$, where $\alpha = (1 + \sqrt{5})/2$.



Solution: We can use strong induction to prove this inequality. Let $P(n)$ be the statement $f_n > \alpha^{n-2}$. We want to show that $P(n)$ is true whenever n is an integer greater than or equal to 3.

BASIS STEP: First, note that

$$\alpha < 2 = f_3, \quad \alpha^2 = (3 + \sqrt{5})/2 < 3 = f_4,$$

so $P(3)$ and $P(4)$ are true.

INDUCTIVE STEP: Assume that $P(j)$ is true, namely, that $f_j > \alpha^{j-2}$, for all integers j with $3 \leq j \leq k$, where $k \geq 4$. We must show that $P(k+1)$ is true, that is, that $f_{k+1} > \alpha^{k-1}$. Because α is a solution of $x^2 - x - 1 = 0$ (as the quadratic formula shows), it follows that $\alpha^2 = \alpha + 1$. Therefore,

$$\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha + 1)\alpha^{k-3} = \alpha \cdot \alpha^{k-3} + 1 \cdot \alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}.$$

By the inductive hypothesis, because $k \geq 4$, we have

$$f_{k-1} > \alpha^{k-3}, \quad f_k > \alpha^{k-2}.$$

Therefore, it follows that

$$f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}.$$

Hence, $P(k+1)$ is true. This completes the proof.

Remark: The inductive step shows that whenever $k \geq 4$, $P(k+1)$ follows from the assumption that $P(j)$ is true for $3 \leq j \leq k$. Hence, the inductive step does *not* show that $P(3) \rightarrow P(4)$. Therefore, we had to show that $P(4)$ is true separately.

We can now show that the Euclidean algorithm, introduced in Section 4.3, uses $O(\log b)$ divisions to find the greatest common divisor of the positive integers a and b , where $a \geq b$.

THEOREM 1

LAMÉ'S THEOREM Let a and b be positive integers with $a \geq b$. Then the number of divisions used by the Euclidean algorithm to find $\gcd(a, b)$ is less than or equal to five times the number of decimal digits in b .

Proof: Recall that when the Euclidean algorithm is applied to find $\gcd(a, b)$ with $a \geq b$, this sequence of equations (where $a = r_0$ and $b = r_1$) is obtained.

$$\begin{aligned} r_0 &= r_1 q_1 + r_2 & 0 \leq r_2 < r_1, \\ r_1 &= r_2 q_2 + r_3 & 0 \leq r_3 < r_2, \\ &\vdots & \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n & 0 \leq r_n < r_{n-1}, \\ r_{n-1} &= r_n q_n. \end{aligned}$$

Here n divisions have been used to find $r_n = \gcd(a, b)$. Note that the quotients q_1, q_2, \dots, q_{n-1} are all at least 1. Moreover, $q_n \geq 2$, because $r_n < r_{n-1}$. This implies that

$$\begin{aligned} r_n &\geq 1 = f_2, \\ r_{n-1} &\geq 2r_n \geq 2f_2 = f_3, \\ r_{n-2} &\geq r_{n-1} + r_n \geq f_3 + f_2 = f_4, \\ &\vdots \\ &\vdots \\ r_2 &\geq r_3 + r_4 \geq f_{n-1} + f_{n-2} = f_n, \\ b = r_1 &\geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1}. \end{aligned}$$

It follows that if n divisions are used by the Euclidean algorithm to find $\gcd(a, b)$ with $a \geq b$, then $b \geq f_{n+1}$. By Example 4 we know that $f_{n+1} > \alpha^{n-1}$ for $n > 2$, where $\alpha = (1 + \sqrt{5})/2$. Therefore, it follows that $b > \alpha^{n-1}$. Furthermore, because $\log_{10} \alpha \approx 0.208 > 1/5$, we see that

$$\log_{10} b > (n - 1) \log_{10} \alpha > (n - 1)/5.$$

Hence, $n - 1 < 5 \cdot \log_{10} b$. Now suppose that b has k decimal digits. Then $b < 10^k$ and $\log_{10} b < k$. It follows that $n - 1 < 5k$, and because k is an integer, it follows that $n \leq 5k$. This finishes the proof. \triangleleft

Because the number of decimal digits in b , which equals $\lfloor \log_{10} b \rfloor + 1$, is less than or equal to $\log_{10} b + 1$, Theorem 1 tells us that the number of divisions required to find $\gcd(a, b)$ with



FIBONACCI (1170–1250) Fibonacci (short for *filius Bonacci*, or “son of Bonacci”) was also known as Leonardo of Pisa. He was born in the Italian commercial center of Pisa. Fibonacci was a merchant who traveled extensively throughout the Mideast, where he came into contact with Arabian mathematics. In his book *Liber Abaci*, Fibonacci introduced the European world to Arabic notation for numerals and algorithms for arithmetic. It was in this book that his famous rabbit problem (described in Section 8.1) appeared. Fibonacci also wrote books on geometry and trigonometry and on Diophantine equations, which involve finding integer solutions to equations.

$a > b$ is less than or equal to $5(\log_{10} b + 1)$. Because $5(\log_{10} b + 1)$ is $O(\log b)$, we see that $O(\log b)$ divisions are used by the Euclidean algorithm to find $\gcd(a, b)$ whenever $a > b$.

Recursively Defined Sets and Structures



We have explored how functions can be defined recursively. We now turn our attention to how sets can be defined recursively. Just as in the recursive definition of functions, recursive definitions of sets have two parts, a **basis step** and a **recursive step**. In the basis step, an initial collection of elements is specified. In the recursive step, rules for forming new elements in the set from those already known to be in the set are provided. Recursive definitions may also include an **exclusion rule**, which specifies that a recursively defined set contains nothing other than those elements specified in the basis step or generated by applications of the recursive step. In our discussions, we will always tacitly assume that the exclusion rule holds and no element belongs to a recursively defined set unless it is in the initial collection specified in the basis step or can be generated using the recursive step one or more times. Later we will see how we can use a technique known as structural induction to prove results about recursively defined sets.

Examples 5, 6, 8, and 9 illustrate the recursive definition of sets. In each example, we show those elements generated by the first few applications of the recursive step.

EXAMPLE 5 Consider the subset S of the set of integers recursively defined by

BASIS STEP: $3 \in S$.

RECURSIVE STEP: If $x \in S$ and $y \in S$, then $x + y \in S$.



The new elements found to be in S are 3 by the basis step, $3 + 3 = 6$ at the first application of the recursive step, $3 + 6 = 6 + 3 = 9$ and $6 + 6 = 12$ at the second application of the recursive step, and so on. We will show in Example 10 that S is the set of all positive multiples of 3.

Recursive definitions play an important role in the study of strings. (See Chapter 13 for an introduction to the theory of formal languages, for example.) Recall from Section 2.4 that a string over an alphabet Σ is a finite sequence of symbols from Σ . We can define Σ^* , the set of strings over Σ , recursively, as Definition 1 shows.

DEFINITION 1

The set Σ^* of *strings* over the alphabet Σ is defined recursively by

BASIS STEP: $\lambda \in \Sigma^*$ (where λ is the empty string containing no symbols).

RECURSIVE STEP: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$.



GABRIEL LAMÉ (1795–1870) Gabriel Lamé entered the École Polytechnique in 1813, graduating in 1817. He continued his education at the École des Mines, graduating in 1820.

In 1820 Lamé went to Russia, where he was appointed director of the Schools of Highways and Transportation in St. Petersburg. Not only did he teach, but he also planned roads and bridges while in Russia. He returned to Paris in 1832, where he helped found an engineering firm. However, he soon left the firm, accepting the chair of physics at the École Polytechnique, which he held until 1844. While holding this position, he was active outside academia as an engineering consultant, serving as chief engineer of mines and participating in the building of railways.

Lamé contributed original work to number theory, applied mathematics, and thermodynamics. His best-known work involves the introduction of curvilinear coordinates. His work on number theory includes proving Fermat's last theorem for $n = 7$, as well as providing the upper bound for the number of divisions used by the Euclidean algorithm given in this text.

In the opinion of Gauss, one of the most important mathematicians of all time, Lamé was the foremost French mathematician of his time. However, French mathematicians considered him too practical, whereas French scientists considered him too theoretical.

The basis step of the recursive definition of strings says that the empty string belongs to Σ^* . The recursive step states that new strings are produced by adding a symbol from Σ to the end of strings in Σ^* . At each application of the recursive step, strings containing one additional symbol are generated.

EXAMPLE 6 If $\Sigma = \{0, 1\}$, the strings found to be in Σ^* , the set of all bit strings, are λ , specified to be in Σ^* in the basis step, 0 and 1 formed during the first application of the recursive step, 00, 01, 10, and 11 formed during the second application of the recursive step, and so on. 

Recursive definitions can be used to define operations or functions on the elements of recursively defined sets. This is illustrated in Definition 2 of the concatenation of two strings and Example 7 concerning the length of a string.

DEFINITION 2

Two strings can be combined via the operation of *concatenation*. Let Σ be a set of symbols and Σ^* the set of strings formed from symbols in Σ . We can define the concatenation of two strings, denoted by \cdot , recursively as follows.

BASIS STEP: If $w \in \Sigma^*$, then $w \cdot \lambda = w$, where λ is the empty string.

RECURSIVE STEP: If $w_1 \in \Sigma^*$ and $w_2 \in \Sigma^*$ and $x \in \Sigma$, then $w_1 \cdot (w_2x) = (w_1 \cdot w_2)x$.

The concatenation of the strings w_1 and w_2 is often written as w_1w_2 rather than $w_1 \cdot w_2$. By repeated application of the recursive definition, it follows that the concatenation of two strings w_1 and w_2 consists of the symbols in w_1 followed by the symbols in w_2 . For instance, the concatenation of $w_1 = abra$ and $w_2 = cadabra$ is $w_1w_2 = abracadabra$.

EXAMPLE 7 Length of a String Give a recursive definition of $l(w)$, the length of the string w .

Solution: The length of a string can be recursively defined by

$$\begin{aligned} l(\lambda) &= 0; \\ l(wx) &= l(w) + 1 \text{ if } w \in \Sigma^* \text{ and } x \in \Sigma. \end{aligned}$$


Another important use of recursive definitions is to define **well-formed formulae** of various types. This is illustrated in Examples 8 and 9.

EXAMPLE 8 Well-Formed Formulae in Propositional Logic We can define the set of well-formed formulae in propositional logic involving **T**, **F**, propositional variables, and operators from the set $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.

BASIS STEP: **T**, **F**, and s , where s is a propositional variable, are well-formed formulae.

RECURSIVE STEP: If E and F are well-formed formulae, then $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$, and $(E \leftrightarrow F)$ are well-formed formulae.

For example, by the basis step we know that **T**, **F**, p , and q are well-formed formulae, where p and q are propositional variables. From an initial application of the recursive step, we know that $(p \vee q)$, $(p \rightarrow F)$, $(F \rightarrow q)$, and $(q \wedge F)$ are well-formed formulae. A second application of the recursive step shows that $((p \vee q) \rightarrow (q \wedge F))$, $(q \vee (p \vee q))$, and $((p \rightarrow F) \rightarrow T)$ are well-formed formulae. We leave it to the reader to show that $p \neg \wedge q$, $p q \wedge$, and $\neg \wedge p q$ are *not* well-formed formulae, by showing that none can be obtained using the basis step and one or more applications of the recursive step. 

EXAMPLE 9 Well-Formed Formulae of Operators and Operands We can define the set of well-formed formulae consisting of variables, numerals, and operators from the set $\{+, -, *, /, \uparrow\}$ (where $*$ denotes multiplication and \uparrow denotes exponentiation) recursively.

BASIS STEP: x is a well-formed formula if x is a numeral or a variable.

RECURSIVE STEP: If F and G are well-formed formulae, then $(F + G)$, $(F - G)$, $(F * G)$, (F/G) , and $(F \uparrow G)$ are well-formed formulae.

For example, by the basis step we see that x , y , 0, and 3 are well-formed formulae (as is any variable or numeral). Well-formed formulae generated by applying the recursive step once include $(x + 3)$, $(3 + y)$, $(x - y)$, $(3 - 0)$, $(x * 3)$, $(3 * y)$, $(3/0)$, (x/y) , $(3 \uparrow x)$, and $(0 \uparrow 3)$. Applying the recursive step twice shows that formulae such as $((x + 3) + 3)$ and $(x - (3 * y))$ are well-formed formulae. [Note that $(3/0)$ is a well-formed formula because we are concerned only with syntax matters here.] We leave it to the reader to show that each of the formulae $x +$, $y * + x$, and $* x/y$ is *not* a well-formed formula by showing that none of them can be obtained from the basis step and one or more applications of the recursive step. \blacktriangleleft

We will study trees extensively in Chapter 11. A tree is a special type of a graph; a graph is made up of vertices and edges connecting some pairs of vertices. We will study graphs in Chapter 10. We will briefly introduce them here to illustrate how they can be defined recursively.

DEFINITION 3

The set of *rooted trees*, where a rooted tree consists of a set of vertices containing a distinguished vertex called the *root*, and edges connecting these vertices, can be defined recursively by these steps:

BASIS STEP: A single vertex r is a rooted tree.

RECURSIVE STEP: Suppose that T_1, T_2, \dots, T_n are disjoint rooted trees with roots r_1, r_2, \dots, r_n , respectively. Then the graph formed by starting with a root r , which is not in any of the rooted trees T_1, T_2, \dots, T_n , and adding an edge from r to each of the vertices r_1, r_2, \dots, r_n , is also a rooted tree.

In Figure 2 we illustrate some of the rooted trees formed starting with the basis step and applying the recursive step one time and two times. Note that infinitely many rooted trees are formed at each application of the recursive definition.

Basis step



Step 1



Step 2

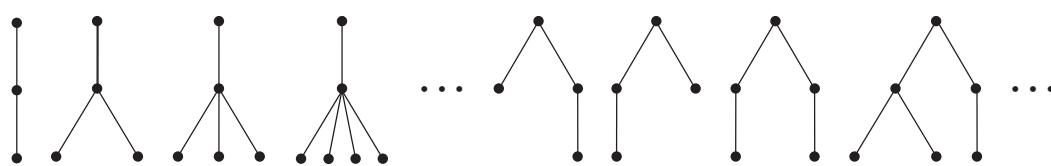


FIGURE 2 Building Up Rooted Trees.

Basis step \emptyset

Step 1 \bullet

Step 2



Step 3

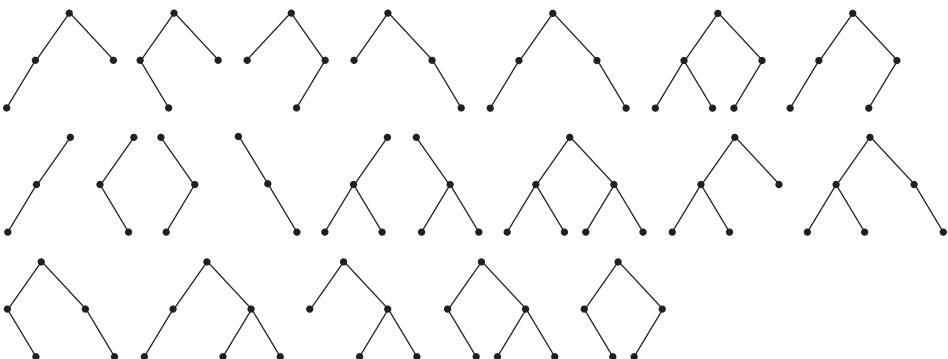


FIGURE 3 Building Up Extended Binary Trees.

Binary trees are a special type of rooted trees. We will provide recursive definitions of two types of binary trees—full binary trees and extended binary trees. In the recursive step of the definition of each type of binary tree, two binary trees are combined to form a new tree with one of these trees designated the left subtree and the other the right subtree. In extended binary trees, the left subtree or the right subtree can be empty, but in full binary trees this is not possible. Binary trees are one of the most important types of structures in computer science. In Chapter 11 we will see how they can be used in searching and sorting algorithms, in algorithms for compressing data, and in many other applications. We first define extended binary trees.

DEFINITION 4

The set of *extended binary trees* can be defined recursively by these steps:

BASIS STEP: The empty set is an extended binary tree.

RECURSIVE STEP: If T_1 and T_2 are disjoint extended binary trees, there is an extended binary tree, denoted by $T_1 \cdot T_2$, consisting of a root r together with edges connecting the root to each of the roots of the left subtree T_1 and the right subtree T_2 when these trees are nonempty.

Figure 3 shows how extended binary trees are built up by applying the recursive step from one to three times.

We now show how to define the set of full binary trees. Note that the difference between this recursive definition and that of extended binary trees lies entirely in the basis step.

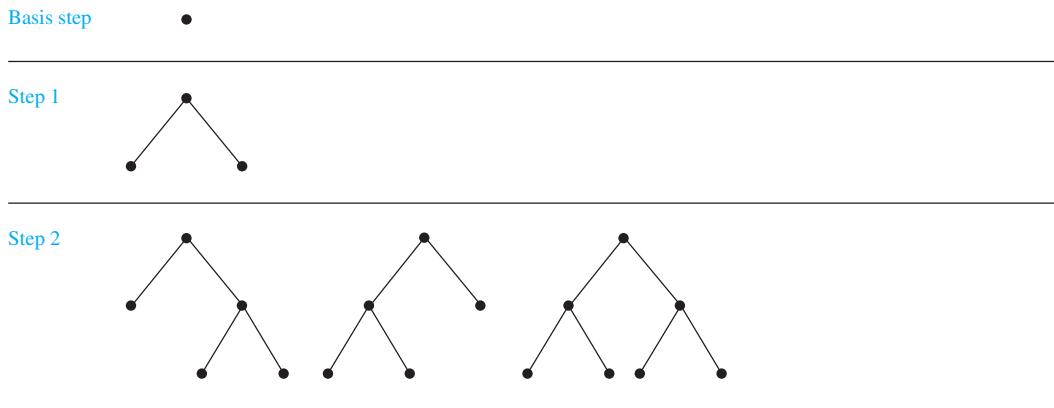


FIGURE 4 Building Up Full Binary Trees.

DEFINITION 5

The set of full binary trees can be defined recursively by these steps:

BASIS STEP: There is a full binary tree consisting only of a single vertex r .

RECURSIVE STEP: If T_1 and T_2 are disjoint full binary trees, there is a full binary tree, denoted by $T_1 \cdot T_2$, consisting of a root r together with edges connecting the root to each of the roots of the left subtree T_1 and the right subtree T_2 .

Figure 4 shows how full binary trees are built up by applying the recursive step one and two times.

Structural Induction

To prove results about recursively defined sets, we generally use some form of mathematical induction. Example 10 illustrates the connection between recursively defined sets and mathematical induction.

EXAMPLE 10

Show that the set S defined in Example 5 by specifying that $3 \in S$ and that if $x \in S$ and $y \in S$, then $x + y \in S$, is the set of all positive integers that are multiples of 3.

Solution: Let A be the set of all positive integers divisible by 3. To prove that $A = S$, we must show that A is a subset of S and that S is a subset of A . To prove that A is a subset of S , we must show that every positive integer divisible by 3 is in S . We will use mathematical induction to prove this.

Let $P(n)$ be the statement that $3n$ belongs to S . The basis step holds because by the first part of the recursive definition of S , $3 \cdot 1 = 3$ is in S . To establish the inductive step, assume that $P(k)$ is true, namely, that $3k$ is in S . Because $3k$ is in S and because 3 is in S , it follows from the second part of the recursive definition of S that $3k + 3 = 3(k + 1)$ is also in S .

To prove that S is a subset of A , we use the recursive definition of S . First, the basis step of the definition specifies that 3 is in S . Because $3 = 3 \cdot 1$, all elements specified to be in S in this step are divisible by 3 and are therefore in A . To finish the proof, we must show that all integers in S generated using the second part of the recursive definition are in A . This consists of showing that $x + y$ is in A whenever x and y are elements of S also assumed to be in A . Now if x and y are both in A , it follows that $3 \mid x$ and $3 \mid y$. By part (i) of Theorem 1 of Section 4.1, it follows that $3 \mid x + y$, completing the proof. \blacktriangleleft

In Example 10 we used mathematical induction over the set of positive integers and a recursive definition to prove a result about a recursively defined set. However, instead of using mathematical induction directly to prove results about recursively defined sets, we can use a more convenient form of induction known as **structural induction**. A proof by structural induction consists of two parts. These parts are

BASIS STEP: Show that the result holds for all elements specified in the basis step of the recursive definition to be in the set.

RECURSIVE STEP: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.

The validity of structural induction follows from the principle of mathematical induction for the nonnegative integers. To see this, let $P(n)$ state that the claim is true for all elements of the set that are generated by n or fewer applications of the rules in the recursive step of a recursive definition. We will have established that the principle of mathematical induction implies the principle of structural induction if we can show that $P(n)$ is true whenever n is a positive integer. In the basis step of a proof by structural induction we show that $P(0)$ is true. That is, we show that the result is true of all elements specified to be in the set in the basis step of the definition. A consequence of the recursive step is that if we assume $P(k)$ is true, it follows that $P(k + 1)$ is true. When we have completed a proof using structural induction, we have shown that $P(0)$ is true and that $P(k)$ implies $P(k + 1)$. By mathematical induction it follows that $P(n)$ is true for all nonnegative integers n . This also shows that the result is true for all elements generated by the recursive definition, and shows that structural induction is a valid proof technique.

EXAMPLES OF PROOFS USING STRUCTURAL INDUCTION Structural induction can be used to prove that all members of a set constructed recursively have a particular property. We will illustrate this idea by using structural induction to prove results about well-formed formulae, strings, and binary trees. For each proof, we have to carry out the appropriate basis step and the appropriate recursive step. For example, to use structural induction to prove a result about the set of well-formed formulae defined in Example 8, where we specify that **T**, **F**, and every propositional variable s are well-formed formulae and where we specify that if E and F are well-formed formulae, then $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \rightarrow F)$, and $(E \leftrightarrow F)$ are well-formed formulae, we need to complete this basis step and this recursive step.

BASIS STEP: Show that the result is true for **T**, **F**, and s whenever s is a propositional variable.

RECURSIVE STEP: Show that if the result is true for the compound propositions p and q , it is also true for $(\neg p)$, $(p \vee q)$, $(p \wedge q)$, $(p \rightarrow q)$, and $(p \leftrightarrow q)$.

Example 11 illustrates how we can prove results about well-formed formulae using structural induction.

EXAMPLE 11 Show that every well-formed formula for compound propositions, as defined in Example 8, contains an equal number of left and right parentheses.

Solution:

BASIS STEP: Each of the formula **T**, **F**, and s contains no parentheses, so clearly they contain an equal number of left and right parentheses.

RECURSIVE STEP: Assume p and q are well-formed formulae each containing an equal number of left and right parentheses. That is, if l_p and l_q are the number of left parentheses in p and q , respectively, and r_p and r_q are the number of right parentheses in p and q , respectively, then $l_p = r_p$ and $l_q = r_q$. To complete the inductive step, we need to show that each of

$(\neg p), (p \vee q), (p \wedge q), (p \rightarrow q)$, and $(p \leftrightarrow q)$ also contains an equal number of left and right parentheses. The number of left parentheses in the first of these compound propositions equals $l_p + 1$ and in each of the other compound propositions equals $l_p + l_q + 1$. Similarly, the number of right parentheses in the first of these compound propositions equals $r_p + 1$ and in each of the other compound propositions equals $r_p + r_q + 1$. Because $l_p = r_p$ and $l_q = r_q$, it follows that each of these compound expressions contains the same number of left and right parentheses. This completes the proof by structural induction. \blacktriangleleft

Suppose that $P(w)$ is a propositional function over the set of strings $w \in \Sigma^*$. To use structural induction to prove that $P(w)$ holds for all strings $w \in \Sigma^*$, we need to complete both a basis step and a recursive step. These steps are:

BASIS STEP: Show that $P(\lambda)$ is true.

RECURSIVE STEP: Assume that $P(w)$ is true, where $w \in \Sigma^*$. Show that if $x \in \Sigma$, then $P(wx)$ must also be true.

Example 12 illustrates how structural induction can be used in proofs about strings.

EXAMPLE 12 Use structural induction to prove that $l(xy) = l(x) + l(y)$, where x and y belong to Σ^* , the set of strings over the alphabet Σ .

Solution: We will base our proof on the recursive definition of the set Σ^* given in Definition 1 and the definition of the length of a string in Example 7, which specifies that $l(\lambda) = 0$ and $l(wx) = l(w) + 1$ when $w \in \Sigma^*$ and $x \in \Sigma$. Let $P(y)$ be the statement that $l(xy) = l(x) + l(y)$ whenever x belongs to Σ^* .

BASIS STEP: To complete the basis step, we must show that $P(\lambda)$ is true. That is, we must show that $l(x\lambda) = l(x) + l(\lambda)$ for all $x \in \Sigma^*$. Because $l(x\lambda) = l(x) = l(x) + 0 = l(x) + l(\lambda)$ for every string x , it follows that $P(\lambda)$ is true.

RECURSIVE STEP: To complete the inductive step, we assume that $P(y)$ is true and show that this implies that $P(ya)$ is true whenever $a \in \Sigma$. What we need to show is that $l(xya) = l(x) + l(ya)$ for every $a \in \Sigma$. To show this, note that by the recursive definition of $l(w)$ (given in Example 7), we have $l(xya) = l(xy) + 1$ and $l(ya) = l(y) + 1$. And, by the inductive hypothesis, $l(xy) = l(x) + l(y)$. We conclude that $l(xya) = l(x) + l(y) + 1 = l(x) + l(ya)$. \blacktriangleleft

We can prove results about trees or special classes of trees using structural induction. For example, to prove a result about full binary trees using structural induction we need to complete this basis step and this recursive step.

BASIS STEP: Show that the result is true for the tree consisting of a single vertex.

RECURSIVE STEP: Show that if the result is true for the trees T_1 and T_2 , then it is true for tree $T_1 \cdot T_2$ consisting of a root r , which has T_1 as its left subtree and T_2 as its right subtree.

Before we provide an example showing how structural induction can be used to prove a result about full binary trees, we need some definitions. We will recursively define the height $h(T)$ and the number of vertices $n(T)$ of a full binary tree T . We begin by defining the height of a full binary tree.

DEFINITION 6

We define the height $h(T)$ of a full binary tree T recursively.

BASIS STEP: The height of the full binary tree T consisting of only a root r is $h(T) = 0$.

RECURSIVE STEP: If T_1 and T_2 are full binary trees, then the full binary tree $T = T_1 \cdot T_2$ has height $h(T) = 1 + \max(h(T_1), h(T_2))$.

If we let $n(T)$ denote the number of vertices in a full binary tree, we observe that $n(T)$ satisfies the following recursive formula:

BASIS STEP: The number of vertices $n(T)$ of the full binary tree T consisting of only a root r is $n(T) = 1$.

RECURSIVE STEP: If T_1 and T_2 are full binary trees, then the number of vertices of the full binary tree $T = T_1 \cdot T_2$ is $n(T) = 1 + n(T_1) + n(T_2)$.

We now show how structural induction can be used to prove a result about full binary trees.

THEOREM 2

If T is a full binary tree T , then $n(T) \leq 2^{h(T)+1} - 1$.

Proof: We prove this inequality using structural induction.

BASIS STEP: For the full binary tree consisting of just the root r the result is true because $n(T) = 1$ and $h(T) = 0$, so that $n(T) = 1 \leq 2^{0+1} - 1 = 1$.

RECURSIVE STEP: For the inductive hypothesis we assume that $n(T_1) \leq 2^{h(T_1)+1} - 1$ and $n(T_2) \leq 2^{h(T_2)+1} - 1$ whenever T_1 and T_2 are full binary trees. By the recursive formulae for $n(T)$ and $h(T)$ we have $n(T) = 1 + n(T_1) + n(T_2)$ and $h(T) = 1 + \max(h(T_1), h(T_2))$.

We find that

$$\begin{aligned}
 n(T) &= 1 + n(T_1) + n(T_2) && \text{by the recursive formula for } n(T) \\
 &\leq 1 + (2^{h(T_1)+1} - 1) + (2^{h(T_2)+1} - 1) && \text{by the inductive hypothesis} \\
 &\leq 2 \cdot \max(2^{h(T_1)+1}, 2^{h(T_2)+1}) - 1 && \text{because the sum of two terms is at most 2 times the larger} \\
 &= 2 \cdot 2^{\max(h(T_1), h(T_2))+1} - 1 && \text{because } \max(2^x, 2^y) = 2^{\max(x, y)} \\
 &= 2 \cdot 2^{h(T)} - 1 && \text{by the recursive definition of } h(T) \\
 &= 2^{h(T)+1} - 1.
 \end{aligned}$$

This completes the recursive step. □

Generalized Induction

We can extend mathematical induction to prove results about other sets that have the well-ordering property besides the set of integers. Although we will discuss this concept in detail in Section 9.6, we provide an example here to illustrate the usefulness of such an approach.

As an example, note that we can define an ordering on $\mathbf{N} \times \mathbf{N}$, the ordered pairs of non-negative integers, by specifying that (x_1, y_1) is less than or equal to (x_2, y_2) if either $x_1 < x_2$, or $x_1 = x_2$ and $y_1 < y_2$; this is called the **lexicographic ordering**. The set $\mathbf{N} \times \mathbf{N}$ with this ordering has the property that every subset of $\mathbf{N} \times \mathbf{N}$ has a least element (see Exercise 53 in Section 9.6). This implies that we can recursively define the terms $a_{m,n}$, with $m \in \mathbf{N}$ and $n \in \mathbf{N}$, and prove results about them using a variant of mathematical induction, as illustrated in Example 13.

EXAMPLE 13 Suppose that $a_{m,n}$ is defined recursively for $(m, n) \in \mathbf{N} \times \mathbf{N}$ by $a_{0,0} = 0$ and

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{if } n = 0 \text{ and } m > 0 \\ a_{m,n-1} + n & \text{if } n > 0. \end{cases}$$

Show that $a_{m,n} = m + n(n + 1)/2$ for all $(m, n) \in \mathbf{N} \times \mathbf{N}$, that is, for all pairs of nonnegative integers.

Solution: We can prove that $a_{m,n} = m + n(n + 1)/2$ using a generalized version of mathematical induction. The basis step requires that we show that this formula is valid when $(m, n) = (0, 0)$. The induction step requires that we show that if the formula holds for all pairs smaller than (m, n) in the lexicographic ordering of $\mathbf{N} \times \mathbf{N}$, then it also holds for (m, n) .

BASIS STEP: Let $(m, n) = (0, 0)$. Then by the basis case of the recursive definition of $a_{m,n}$ we have $a_{0,0} = 0$. Furthermore, when $m = n = 0$, $m + n(n + 1)/2 = 0 + (0 \cdot 1)/2 = 0$. This completes the basis step.

INDUCTIVE STEP: Suppose that $a_{m',n'} = m' + n'(n' + 1)/2$ whenever (m', n') is less than (m, n) in the lexicographic ordering of $\mathbf{N} \times \mathbf{N}$. By the recursive definition, if $n = 0$, then $a_{m,n} = a_{m-1,n} + 1$. Because $(m - 1, n)$ is smaller than (m, n) , the inductive hypothesis tells us that $a_{m-1,n} = m - 1 + n(n + 1)/2$, so that $a_{m,n} = m - 1 + n(n + 1)/2 + 1 = m + n(n + 1)/2$, giving us the desired equality. Now suppose that $n > 0$, so $a_{m,n} = a_{m,n-1} + n$. Because $(m, n - 1)$ is smaller than (m, n) , the inductive hypothesis tells us that $a_{m,n-1} = m + (n - 1)n/2$, so $a_{m,n} = m + (n - 1)n/2 + n = m + (n^2 - n + 2n)/2 = m + n(n + 1)/2$. This finishes the inductive step. 

As mentioned, we will justify this proof technique in Section 9.6.

Exercises

1. Find $f(1), f(2), f(3)$, and $f(4)$ if $f(n)$ is defined recursively by $f(0) = 1$ and for $n = 0, 1, 2, \dots$
 - a) $f(n + 1) = f(n) + 2$.
 - b) $f(n + 1) = 3f(n)$.
 - c) $f(n + 1) = 2^{f(n)}$.
 - d) $f(n + 1) = f(n)^2 + f(n) + 1$.
2. Find $f(1), f(2), f(3), f(4)$, and $f(5)$ if $f(n)$ is defined recursively by $f(0) = 3$ and for $n = 0, 1, 2, \dots$
 - a) $f(n + 1) = -2f(n)$.
 - b) $f(n + 1) = 3f(n) + 7$.
 - c) $f(n + 1) = f(n)^2 - 2f(n) - 2$.
 - d) $f(n + 1) = 3^{f(n)/3}$.
3. Find $f(2), f(3), f(4)$, and $f(5)$ if f is defined recursively by $f(0) = -1, f(1) = 2$, and for $n = 1, 2, \dots$
 - a) $f(n + 1) = f(n) + 3f(n - 1)$.
 - b) $f(n + 1) = f(n)^2 f(n - 1)$.
 - c) $f(n + 1) = 3f(n)^2 - 4f(n - 1)^2$.
 - d) $f(n + 1) = f(n - 1)/f(n)$.
4. Find $f(2), f(3), f(4)$, and $f(5)$ if f is defined recursively by $f(0) = f(1) = 1$ and for $n = 1, 2, \dots$
 - a) $f(n + 1) = f(n) - f(n - 1)$.
 - b) $f(n + 1) = f(n)f(n - 1)$.
 - c) $f(n + 1) = f(n)^2 + f(n - 1)^3$.
 - d) $f(n + 1) = f(n)/f(n - 1)$.
5. Determine whether each of these proposed definitions is a valid recursive definition of a function f from the set of nonnegative integers to the set of integers. If f is well defined, find a formula for $f(n)$ when n is a nonnegative integer and prove that your formula is valid.
 - a) $f(0) = 0, f(n) = 2f(n - 2)$ for $n \geq 1$
 - b) $f(0) = 1, f(n) = f(n - 1) - 1$ for $n \geq 1$
 - c) $f(0) = 2, f(1) = 3, f(n) = f(n - 1) - 1$ for $n \geq 2$
 - d) $f(0) = 1, f(1) = 2, f(n) = 2f(n - 2)$ for $n \geq 2$
 - e) $f(0) = 1, f(n) = 3f(n - 1)$ if n is odd and $n \geq 1$ and $f(n) = 9f(n - 2)$ if n is even and $n \geq 2$
6. Determine whether each of these proposed definitions is a valid recursive definition of a function f from the set of nonnegative integers to the set of integers. If f is well defined, find a formula for $f(n)$ when n is a nonnegative integer and prove that your formula is valid.
 - a) $f(0) = 1, f(n) = -f(n - 1)$ for $n \geq 1$
 - b) $f(0) = 1, f(1) = 0, f(2) = 2, f(n) = 2f(n - 3)$ for $n \geq 3$
 - c) $f(0) = 0, f(1) = 1, f(n) = 2f(n + 1)$ for $n \geq 2$
 - d) $f(0) = 0, f(1) = 1, f(n) = 2f(n - 1)$ for $n \geq 1$
 - e) $f(0) = 2, f(n) = f(n - 1)$ if n is odd and $n \geq 1$ and $f(n) = 2f(n - 2)$ if $n \geq 2$

7. Give a recursive definition of the sequence $\{a_n\}$, $n = 1, 2, 3, \dots$ if
- $a_n = 6n$.
 - $a_n = 2n + 1$.
 - $a_n = 10^n$.
 - $a_n = 5$.
8. Give a recursive definition of the sequence $\{a_n\}$, $n = 1, 2, 3, \dots$ if
- $a_n = 4n - 2$.
 - $a_n = 1 + (-1)^n$.
 - $a_n = n(n + 1)$.
 - $a_n = n^2$.
9. Let F be the function such that $F(n)$ is the sum of the first n positive integers. Give a recursive definition of $F(n)$.
10. Give a recursive definition of $S_m(n)$, the sum of the integer m and the nonnegative integer n .
11. Give a recursive definition of $P_m(n)$, the product of the integer m and the nonnegative integer n .

In Exercises 12–19 f_n is the n th Fibonacci number.

12. Prove that $f_1^2 + f_2^2 + \dots + f_n^2 = f_n f_{n+1}$ when n is a positive integer.
13. Prove that $f_1 + f_3 + \dots + f_{2n-1} = f_{2n}$ when n is a positive integer.
- *14. Show that $f_{n+1}f_{n-1} - f_n^2 = (-1)^n$ when n is a positive integer.
- *15. Show that $f_0f_1 + f_1f_2 + \dots + f_{2n-1}f_{2n} = f_{2n}^2$ when n is a positive integer.
- *16. Show that $f_0 - f_1 + f_2 - \dots - f_{2n-1} + f_{2n} = f_{2n-1} - 1$ when n is a positive integer.
17. Determine the number of divisions used by the Euclidean algorithm to find the greatest common divisor of the Fibonacci numbers f_n and f_{n+1} , where n is a nonnegative integer. Verify your answer using mathematical induction.

18. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

Show that

$$\mathbf{A}^n = \begin{bmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{bmatrix}$$

when n is a positive integer.

19. By taking determinants of both sides of the equation in Exercise 18, prove the identity given in Exercise 14. (Recall that the determinant of the matrix $\begin{vmatrix} a & b \\ c & d \end{vmatrix}$ is $ad - bc$.)
- *20. Give a recursive definition of the functions max and min so that $\max(a_1, a_2, \dots, a_n)$ and $\min(a_1, a_2, \dots, a_n)$ are the maximum and minimum of the n numbers a_1, a_2, \dots, a_n , respectively.
- *21. Let a_1, a_2, \dots, a_n , and b_1, b_2, \dots, b_n be real numbers. Use the recursive definitions that you gave in Exercise 20 to prove these.
- $\max(-a_1, -a_2, \dots, -a_n) = -\min(a_1, a_2, \dots, a_n)$
 - $\max(a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$
 $\leq \max(a_1, a_2, \dots, a_n) + \max(b_1, b_2, \dots, b_n)$
 - $\min(a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$
 $\geq \min(a_1, a_2, \dots, a_n) + \min(b_1, b_2, \dots, b_n)$
22. Show that the set S defined by $1 \in S$ and $s + t \in S$ whenever $s \in S$ and $t \in S$ is the set of positive integers.

23. Give a recursive definition of the set of positive integers that are multiples of 5.
24. Give a recursive definition of
- the set of odd positive integers.
 - the set of positive integer powers of 3.
 - the set of polynomials with integer coefficients.
25. Give a recursive definition of
- the set of even integers.
 - the set of positive integers congruent to 2 modulo 3.
 - the set of positive integers not divisible by 5.
26. Let S be the subset of the set of ordered pairs of integers defined recursively by
- Basis step:* $(0, 0) \in S$.
- Recursive step:* If $(a, b) \in S$, then $(a + 2, b + 3) \in S$ and $(a + 3, b + 2) \in S$.
- List the elements of S produced by the first five applications of the recursive definition.
 - Use strong induction on the number of applications of the recursive step of the definition to show that $5 \mid a + b$ when $(a, b) \in S$.
 - Use structural induction to show that $5 \mid a + b$ when $(a, b) \in S$.
27. Let S be the subset of the set of ordered pairs of integers defined recursively by
- Basis step:* $(0, 0) \in S$.
- Recursive step:* If $(a, b) \in S$, then $(a, b + 1) \in S$, $(a + 1, b + 1) \in S$, and $(a + 2, b + 1) \in S$.
- List the elements of S produced by the first four applications of the recursive definition.
 - Use strong induction on the number of applications of the recursive step of the definition to show that $a \leq 2b$ whenever $(a, b) \in S$.
 - Use structural induction to show that $a \leq 2b$ whenever $(a, b) \in S$.
28. Give a recursive definition of each of these sets of ordered pairs of positive integers. [Hint: Plot the points in the set in the plane and look for lines containing points in the set.]
- $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, \text{ and } a + b \text{ is odd}\}$
 - $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, \text{ and } a \mid b\}$
 - $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, \text{ and } 3 \mid a + b\}$
29. Give a recursive definition of each of these sets of ordered pairs of positive integers. Use structural induction to prove that the recursive definition you found is correct. [Hint: To find a recursive definition, plot the points in the set in the plane and look for patterns.]
- $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, \text{ and } a + b \text{ is even}\}$
 - $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, \text{ and } a \text{ or } b \text{ is odd}\}$
 - $S = \{(a, b) \mid a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, a + b \text{ is odd, and } 3 \mid b\}$
30. Prove that in a bit string, the string 01 occurs at most one more time than the string 10.
31. Define well-formed formulae of sets, variables representing sets, and operators from $\{\setminus, \cup, \cap, -\}$.

- 32.** a) Give a recursive definition of the function $\text{ones}(s)$, which counts the number of ones in a bit string s .
 b) Use structural induction to prove that $\text{ones}(st) = \text{ones}(s) + \text{ones}(t)$.
- 33.** a) Give a recursive definition of the function $m(s)$, which equals the smallest digit in a nonempty string of decimal digits.
 b) Use structural induction to prove that $m(st) = \min(m(s), m(t))$.

The **reversal** of a string is the string consisting of the symbols of the string in reverse order. The reversal of the string w is denoted by w^R .

- 34.** Find the reversal of the following bit strings.

a) 0101 b) 11011 c) 100010010111

- 35.** Give a recursive definition of the reversal of a string. [Hint: First define the reversal of the empty string. Then write a string w of length $n + 1$ as xy , where x is a string of length n , and express the reversal of w in terms of x^R and y .]

- ***36.** Use structural induction to prove that $(w_1 w_2)^R = w_2^R w_1^R$.
37. Give a recursive definition of w^i , where w is a string and i is a nonnegative integer. (Here w^i represents the concatenation of i copies of the string w).
***38.** Give a recursive definition of the set of bit strings that are palindromes.
39. When does a string belong to the set A of bit strings defined recursively by

$$\begin{aligned} \lambda &\in A \\ 0x1 &\in A \text{ if } x \in A, \end{aligned}$$

where λ is the empty string?

- ***40.** Recursively define the set of bit strings that have more zeros than ones.
41. Use Exercise 37 and mathematical induction to show that $l(w^i) = i \cdot l(w)$, where w is a string and i is a nonnegative integer.
***42.** Show that $(w^R)^i = (w^i)^R$ whenever w is a string and i is a nonnegative integer; that is, show that the i th power of the reversal of a string is the reversal of the i th power of the string.
43. Use structural induction to show that $n(T) \geq 2h(T) + 1$, where T is a full binary tree, $n(T)$ equals the number of vertices of T , and $h(T)$ is the height of T .

The set of leaves and the set of internal vertices of a full binary tree can be defined recursively.

Basis step: The root r is a leaf of the full binary tree with exactly one vertex r . This tree has no internal vertices.

Recursive step: The set of leaves of the tree $T = T_1 \cdot T_2$ is the union of the sets of leaves of T_1 and of T_2 . The internal vertices of T are the root r of T and the union of the set of internal vertices of T_1 and the set of internal vertices of T_2 .

- 44.** Use structural induction to show that $l(T)$, the number of leaves of a full binary tree T , is 1 more than $i(T)$, the number of internal vertices of T .

- 45.** Use generalized induction as was done in Example 13 to show that if $a_{m,n}$ is defined recursively by $a_{0,0} = 0$ and

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1 & \text{if } n = 0 \text{ and } m > 0 \\ a_{m,n-1} + 1 & \text{if } n > 0, \end{cases}$$

then $a_{m,n} = m + n$ for all $(m, n) \in \mathbb{N} \times \mathbb{N}$.

- 46.** Use generalized induction as was done in Example 13 to show that if $a_{m,n}$ is defined recursively by $a_{1,1} = 5$ and

$$a_{m,n} = \begin{cases} a_{m-1,n} + 2 & \text{if } n = 1 \text{ and } m > 1 \\ a_{m,n-1} + 2 & \text{if } n > 1, \end{cases}$$

then $a_{m,n} = 2(m + n) + 1$ for all $(m, n) \in \mathbb{Z}^+ \times \mathbb{Z}^+$.

- ***47.** A **partition** of a positive integer n is a way to write n as a sum of positive integers where the order of terms in the sum does not matter. For instance, $7 = 3 + 2 + 1 + 1$ is a partition of 7. Let P_m equal the number of different partitions of m , and let $P_{m,n}$ be the number of different ways to express m as the sum of positive integers not exceeding n .

- a) Show that $P_{m,m} = P_m$.

- b) Show that the following recursive definition for $P_{m,n}$ is correct:

$$P_{m,n} = \begin{cases} 1 & \text{if } m = 1 \\ 1 & \text{if } n = 1 \\ P_{m,m} & \text{if } m < n \\ 1 + P_{m,m-1} & \text{if } m = n > 1 \\ P_{m,n-1} + P_{m-n,n} & \text{if } m > n > 1. \end{cases}$$

- c) Find the number of partitions of 5 and of 6 using this recursive definition.



Consider an inductive definition of a version of **Ackermann's function**. This function was named after Wilhelm Ackermann, a German mathematician who was a student of the great mathematician David Hilbert. Ackermann's function plays an important role in the theory of recursive functions and in the study of the complexity of certain algorithms involving set unions. (There are several different variants of this function. All are called Ackermann's function and have similar properties even though their values do not always agree.)

$$A(m, n) = \begin{cases} 2n & \text{if } m = 0 \\ 0 & \text{if } m \geq 1 \text{ and } n = 0 \\ 2 & \text{if } m \geq 1 \text{ and } n = 1 \\ A(m - 1, A(m, n - 1)) & \text{if } m \geq 1 \text{ and } n \geq 2 \end{cases}$$

Exercises 48–55 involve this version of Ackermann's function.

- 48.** Find these values of Ackermann's function.

a) $A(1, 0)$	b) $A(0, 1)$
c) $A(1, 1)$	d) $A(2, 2)$

- 49.** Show that $A(m, 2) = 4$ whenever $m \geq 1$.

- 50.** Show that $A(1, n) = 2^n$ whenever $n \geq 1$.

- 51.** Find these values of Ackermann's function.

a) $A(2, 3)$	*b) $A(3, 3)$
--------------	---------------

- ***52.** Find $A(3, 4)$.

- **53.** Prove that $A(m, n + 1) > A(m, n)$ whenever m and n are nonnegative integers.
- *54.** Prove that $A(m + 1, n) \geq A(m, n)$ whenever m and n are nonnegative integers.
- 55.** Prove that $A(i, j) \geq j$ whenever i and j are nonnegative integers.
- 56.** Use mathematical induction to prove that a function F defined by specifying $F(0)$ and a rule for obtaining $F(n + 1)$ from $F(n)$ is well defined.
- 57.** Use strong induction to prove that a function F defined by specifying $F(0)$ and a rule for obtaining $F(n + 1)$ from the values $F(k)$ for $k = 0, 1, 2, \dots, n$ is well defined.
- 58.** Show that each of these proposed recursive definitions of a function on the set of positive integers does not produce a well-defined function.
- $F(n) = 1 + F(\lfloor n/2 \rfloor)$ for $n \geq 1$ and $F(1) = 1$.
 - $F(n) = 1 + F(n - 3)$ for $n \geq 2$, $F(1) = 2$, and $F(2) = 3$.
 - $F(n) = 1 + F(n/2)$ for $n \geq 2$, $F(1) = 1$, and $F(2) = 2$.
 - $F(n) = 1 + F(n/2)$ if n is even and $n \geq 2$, $F(n) = 1 - F(n - 1)$ if n is odd, and $F(1) = 1$.
 - $F(n) = 1 + F(n/2)$ if n is even and $n \geq 2$, $F(n) = F(3n - 1)$ if n is odd and $n \geq 3$, and $F(1) = 1$.
- 59.** Show that each of these proposed recursive definitions of a function on the set of positive integers does not produce a well-defined function.
- $F(n) = 1 + F(\lfloor (n + 1)/2 \rfloor)$ for $n \geq 1$ and $F(1) = 1$.
 - $F(n) = 1 + F(n - 2)$ for $n \geq 2$ and $F(1) = 0$.
 - $F(n) = 1 + F(n/3)$ for $n \geq 3$, $F(1) = 1$, $F(2) = 2$, and $F(3) = 3$.
 - $F(n) = 1 + F(n/2)$ if n is even and $n \geq 2$, $F(n) = 1 + F(n - 2)$ if n is odd, and $F(1) = 1$.
 - $F(n) = 1 + F(F(n - 1))$ if $n \geq 2$ and $F(1) = 2$.

Exercises 60–62 deal with iterations of the logarithm function. Let $\log n$ denote the logarithm of n to the base 2, as usual. The function $\log^{(k)} n$ is defined recursively by

$$\log^{(k)} n = \begin{cases} n & \text{if } k = 0 \\ \log(\log^{(k-1)} n) & \text{if } \log^{(k-1)} n \text{ is defined and positive} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The **iterated logarithm** is the function $\log^* n$ whose value at n is the smallest nonnegative integer k such that $\log^{(k)} n \leq 1$.

- 60.** Find these values.

- a) $\log^{(2)} 16$ b) $\log^{(3)} 256$
c) $\log^{(3)} 2^{65536}$ d) $\log^{(4)} 2^{2^{65536}}$

- 61.** Find the value of $\log^* n$ for these values of n .

- a) 2 b) 4 c) 8 d) 16
e) 256 f) 65536 g) 2^{2048}

- 62.** Find the largest integer n such that $\log^* n = 5$. Determine the number of decimal digits in this number.

Exercises 63–65 deal with values of iterated functions. Suppose that $f(n)$ is a function from the set of real numbers, or positive real numbers, or some other set of real numbers, to the set of real numbers such that $f(n)$ is monotonically increasing [that is, $f(n) < f(m)$ when $n < m$] and $f(n) < n$ for all n in the domain of f .] The function $f^{(k)}(n)$ is defined recursively by

$$f^{(k)}(n) = \begin{cases} n & \text{if } k = 0 \\ f(f^{(k-1)}(n)) & \text{if } k > 0. \end{cases}$$

Furthermore, let c be a positive real number. The **iterated function** f_c^* is the number of iterations of f required to reduce its argument to c or less, so $f_c^*(n)$ is the smallest nonnegative integer k such that $f^k(n) \leq c$.

- 63.** Let $f(n) = n - a$, where a is a positive integer. Find a formula for $f^{(k)}(n)$. What is the value of $f_0^*(n)$ when n is a positive integer?
- 64.** Let $f(n) = n/2$. Find a formula for $f^{(k)}(n)$. What is the value of $f_1^*(n)$ when n is a positive integer?
- 65.** Let $f(n) = \sqrt{n}$. Find a formula for $f^{(k)}(n)$. What is the value of $f_2^*(n)$ when n is a positive integer?

5.4 Recursive Algorithms

Introduction

Sometimes we can reduce the solution to a problem with a particular set of input values to the solution of the same problem with smaller input values. For instance, the problem of finding the greatest common divisor of two positive integers a and b , where $b > a$, can be reduced to finding the greatest common divisor of a pair of smaller integers, namely, $b \bmod a$ and a , because $\gcd(b \bmod a, a) = \gcd(a, b)$. When such a reduction can be done, the solution to the original problem can be found with a sequence of reductions, until the problem has been reduced to some initial case for which the solution is known. For instance, for finding the greatest common divisor, the reduction continues until the smaller of the two numbers is zero, because $\gcd(a, 0) = a$ when $a > 0$.

We will see that algorithms that successively reduce a problem to the same problem with smaller input are used to solve a wide variety of problems.

Here's a famous humorous quote: "To understand recursion, you must first understand recursion."

DEFINITION 1

An algorithm is called *recursive* if it solves a problem by reducing it to an instance of the same problem with smaller input.



We will describe a variety of different recursive algorithms in this section.

EXAMPLE 1

Give a recursive algorithm for computing $n!$, where n is a nonnegative integer.



Solution: We can build a recursive algorithm that finds $n!$, where n is a nonnegative integer, based on the recursive definition of $n!$, which specifies that $n! = n \cdot (n - 1)!$ when n is a positive integer, and that $0! = 1$. To find $n!$ for a particular integer, we use the recursive step n times, each time replacing a value of the factorial function with the value of the factorial function at the next smaller integer. At this last step, we insert the value of $0!$. The recursive algorithm we obtain is displayed as Algorithm 1.

To help understand how this algorithm works, we trace the steps used by the algorithm to compute $4!$. First, we use the recursive step to write $4! = 4 \cdot 3!$. We then use the recursive step repeatedly to write $3! = 3 \cdot 2!$, $2! = 2 \cdot 1!$, and $1! = 1 \cdot 0!$. Inserting the value of $0! = 1$, and working back through the steps, we see that $1! = 1 \cdot 1 = 1$, $2! = 2 \cdot 1! = 2$, $3! = 3 \cdot 2! = 3 \cdot 2 = 6$, and $4! = 4 \cdot 3! = 4 \cdot 6 = 24$. 

ALGORITHM 1 A Recursive Algorithm for Computing $n!$.

```

procedure factorial( $n$ : nonnegative integer)
if  $n = 0$  then return 1
else return  $n \cdot \text{factorial}(n - 1)$ 
{output is  $n!$ }
```

Example 2 shows how a recursive algorithm can be constructed to evaluate a function from its recursive definition.

EXAMPLE 2

Give a recursive algorithm for computing a^n , where a is a nonzero real number and n is a nonnegative integer.

Solution: We can base a recursive algorithm on the recursive definition of a^n . This definition states that $a^{n+1} = a \cdot a^n$ for $n > 0$ and the initial condition $a^0 = 1$. To find a^n , successively use the recursive step to reduce the exponent until it becomes zero. We give this procedure in Algorithm 2. 

ALGORITHM 2 A Recursive Algorithm for Computing a^n .

```

procedure power( $a$ : nonzero real number,  $n$ : nonnegative integer)
if  $n = 0$  then return 1
else return  $a \cdot \text{power}(a, n - 1)$ 
{output is  $a^n$ }
```

Next we give a recursive algorithm for finding greatest common divisors.

EXAMPLE 3 Give a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b with $a < b$.

Solution: We can base a recursive algorithm on the reduction $\gcd(a, b) = \gcd(b \bmod a, a)$ and the condition $\gcd(0, b) = b$ when $b > 0$. This produces the procedure in Algorithm 3, which is a recursive version of the Euclidean algorithm.

We illustrate the workings of Algorithm 3 with a trace when the input is $a = 5, b = 8$. With this input, the algorithm uses the “else” clause to find that $\gcd(5, 8) = \gcd(8 \bmod 5, 5) = \gcd(3, 5)$. It uses this clause again to find that $\gcd(3, 5) = \gcd(5 \bmod 3, 3) = \gcd(2, 3)$, then to get $\gcd(2, 3) = \gcd(3 \bmod 2, 2) = \gcd(1, 2)$, then to get $\gcd(1, 2) = \gcd(2 \bmod 1, 1) = \gcd(0, 1)$. Finally, to find $\gcd(0, 1)$ it uses the first step with $a = 0$ to find that $\gcd(0, 1) = 1$. Consequently, the algorithm finds that $\gcd(5, 8) = 1$. 

ALGORITHM 3 A Recursive Algorithm for Computing $\gcd(a, b)$.

```

procedure gcd(a, b: nonnegative integers with  $a < b$ )
if  $a = 0$  then return  $b$ 
else return  $\gcd(b \bmod a, a)$ 
{output is  $\gcd(a, b)$ }
```

EXAMPLE 4 Devise a recursive algorithm for computing $b^n \bmod m$, where b, n , and m are integers with $m \geq 2, n \geq 0$, and $1 \leq b < m$.

Solution: We can base a recursive algorithm on the fact that

$$b^n \bmod m = (b \cdot (b^{n-1} \bmod m)) \bmod m,$$

which follows by Corollary 2 in Section 4.1, and the initial condition $b^0 \bmod m = 1$. We leave this as Exercise 12 for the reader.

However, we can devise a much more efficient recursive algorithm based on the observation that

$$b^n \bmod m = (b^{n/2} \bmod m)^2 \bmod m$$

when n is even and

$$b^n \bmod m = ((b^{\lfloor n/2 \rfloor} \bmod m)^2 \bmod m \cdot b \bmod m) \bmod m$$

when n is odd, which we describe in pseudocode as Algorithm 4.

We trace the execution of Algorithm 4 with input $b = 2, n = 5$, and $m = 3$ to illustrate how it works. First, because $n = 5$ is odd we use the “else” clause to see that $mpower(2, 5, 3) = (mpower(2, 2, 3)^2 \bmod 3 \cdot 2 \bmod 3) \bmod 3$. We next use the “else if” clause to see that $mpower(2, 2, 3) = mpower(2, 1, 3)^2 \bmod 3$. Using the “else” clause again, we see that $mpower(2, 1, 3) = (mpower(2, 0, 3)^2 \bmod 3 \cdot 2 \bmod 3) \bmod 3$. Finally, using the “if” clause, we see that $mpower(2, 0, 3) = 1$. Working backwards, it follows that $mpower(2, 1, 3) = (1^2 \bmod 3 \cdot 2 \bmod 3) \bmod 3 = 2$, so $mpower(2, 2, 3) = 2^2 \bmod 3 = 1$, and finally $mpower(2, 5, 3) = (1^2 \bmod 3 \cdot 2 \bmod 3) \bmod 3 = 2$. 

ALGORITHM 4 Recursive Modular Exponentiation.

```

procedure mpower( $b, n, m$ : integers with  $b > 0$  and  $m \geq 2, n \geq 0$ )
if  $n = 0$  then
    return 1
else if  $n$  is even then
    return mpower( $b, n/2, m$ ) $^2 \bmod m$ 
else
    return ( $\text{mpower}(b, \lfloor n/2 \rfloor, m)^2 \bmod m \cdot b \bmod m$ )  $\bmod m$ 
{output is  $b^n \bmod m$ }

```

We will now give recursive versions of searching algorithms that were introduced in Section 3.1.

EXAMPLE 5 Express the linear search algorithm as a recursive procedure.

Solution: To search for the first occurrence of x in the sequence a_1, a_2, \dots, a_n , at the i th step of the algorithm, x and a_i are compared. If x equals a_i , then the algorithm returns i , the location of x in the sequence. Otherwise, the search for the first occurrence of x is reduced to a search in a sequence with one fewer element, namely, the sequence a_{i+1}, \dots, a_n . The algorithm returns 0 when x is never found in the sequence after all terms have been examined. We can now give a recursive procedure, which is displayed as pseudocode in Algorithm 5.

Let $\text{search}(i, j, x)$ be the procedure that searches for the first occurrence of x in the sequence a_i, a_{i+1}, \dots, a_j . The input to the procedure consists of the triple (i, n, x) . The algorithm terminates at a step if the first term of the remaining sequence is x or if there is only one term of the sequence and this is not x . If x is not the first term and there are additional terms, the same procedure is carried out but with a search sequence of one fewer term, obtained by deleting the first term of the search sequence. If the algorithm terminates without x having been found, the algorithm returns the value 0. 

ALGORITHM 5 A Recursive Linear Search Algorithm.

```

procedure search( $i, j, x$ :  $i, j, x$  integers,  $1 \leq i \leq j \leq n$ )
if  $a_i = x$  then
    return  $i$ 
else if  $i = j$  then
    return 0
else
    return search( $i + 1, j, x$ )
{output is the location of  $x$  in  $a_1, a_2, \dots, a_n$  if it appears; otherwise it is 0}

```

EXAMPLE 6 Construct a recursive version of a binary search algorithm.

Solution: Suppose we want to locate x in the sequence a_1, a_2, \dots, a_n of integers in increasing order. To perform a binary search, we begin by comparing x with the middle term, $a_{\lfloor(n+1)/2\rfloor}$. Our algorithm will terminate if x equals this term and return the location of this term in the sequence. Otherwise, we reduce the search to a smaller search sequence, namely, the first half of the sequence if x is smaller than the middle term of the original sequence, and the second half otherwise. We have reduced the solution of the search problem to the solution of the same

problem with a sequence at most half as long. If we have never encountered the search term x , our algorithm returns the value 0. We express this recursive version of a binary search algorithm as Algorithm 6.

ALGORITHM 6 A Recursive Binary Search Algorithm.

```

procedure binary search(i, j, x: i, j, x integers,  $1 \leq i \leq j \leq n$ )
  m :=  $\lfloor (i + j)/2 \rfloor$ 
  if x =  $a_m$  then
    return m
  else if (x <  $a_m$  and i < m) then
    return binary search(i, m - 1, x)
  else if (x >  $a_m$  and j > m) then
    return binary search(m + 1, j, x)
  else return 0
  {output is location of x in  $a_1, a_2, \dots, a_n$  if it appears; otherwise it is 0}

```

Proving Recursive Algorithms Correct

Mathematical induction, and its variant strong induction, can be used to prove that a recursive algorithm is correct, that is, that it produces the desired output for all possible input values. Examples 7 and 8 illustrate how mathematical induction or strong induction can be used to prove that recursive algorithms are correct. First, we will show that Algorithm 2 is correct.

EXAMPLE 7 Prove that Algorithm 2, which computes powers of real numbers, is correct.

Solution: We use mathematical induction on the exponent n .

BASIS STEP: If $n = 0$, the first step of the algorithm tells us that $\text{power}(a, 0) = 1$. This is correct because $a^0 = 1$ for every nonzero real number a . This completes the basis step.

INDUCTIVE STEP: The inductive hypothesis is the statement that $\text{power}(a, k) = a^k$ for all $a \neq 0$ for an arbitrary nonnegative integer k . That is, the inductive hypothesis is the statement that the algorithm correctly computes a^k . To complete the inductive step, we show that if the inductive hypothesis is true, then the algorithm correctly computes a^{k+1} . Because $k + 1$ is a positive integer, when the algorithm computes a^{k+1} , the algorithm sets $\text{power}(a, k + 1) = a \cdot \text{power}(a, k)$. By the inductive hypothesis, we have $\text{power}(a, k) = a^k$, so $\text{power}(a, k + 1) = a \cdot \text{power}(a, k) = a \cdot a^k = a^{k+1}$. This completes the inductive step.

We have completed the basis step and the inductive step, so we can conclude that Algorithm 2 always computes a^n correctly when $a \neq 0$ and n is a nonnegative integer.

Generally, we need to use strong induction to prove that recursive algorithms are correct, rather than just mathematical induction. Example 8 illustrates this; it shows how strong induction can be used to prove that Algorithm 4 is correct.

EXAMPLE 8 Prove that Algorithm 4, which computes modular powers, is correct.

Solution: We use strong induction on the exponent n .

BASIS STEP: Let b be an integer and m an integer with $m \geq 2$. When $n = 0$, the algorithm sets $\text{mpower}(b, n, m)$ equal to 1. This is correct because $b^0 \bmod m = 1$. The basis step is complete.

INDUCTIVE STEP: For the inductive hypothesis we assume that $\text{mpower}(b, j, m) = b^j \bmod m$ for all integers $0 \leq j < k$ whenever b is a positive integer and m is an integer with $m \geq 2$. To complete the inductive step, we show that if the inductive hypothesis is correct, then $\text{mpower}(b, k, m) = b^k \bmod m$. Because the recursive algorithm handles odd and even values of k differently, we split the inductive step into two cases.

When k is even, we have

$$\text{mpower}(b, k, m) = (\text{mpower}(b, k/2, m))^2 \bmod m = (b^{k/2} \bmod m)^2 \bmod m = b^k \bmod m,$$

where we have used the inductive hypothesis to replace $\text{mpower}(b, k/2, m)$ by $b^{k/2} \bmod m$.

When k is odd, we have

$$\begin{aligned}\text{mpower}(b, k, m) &= ((\text{mpower}(b, \lfloor k/2 \rfloor, m))^2 \bmod m \cdot b \bmod m) \bmod m \\ &= ((b^{\lfloor k/2 \rfloor} \bmod m)^2 \bmod m \cdot b \bmod m) \bmod m \\ &= b^{2\lfloor k/2 \rfloor + 1} \bmod m = b^k \bmod m,\end{aligned}$$

using Corollary 2 in Section 4.1, because $2\lfloor k/2 \rfloor + 1 = 2(k - 1)/2 + 1 = k$ when k is odd. Here we have used the inductive hypothesis to replace $\text{mpower}(b, \lfloor k/2 \rfloor, m)$ by $b^{\lfloor k/2 \rfloor} \bmod m$. This completes the inductive step.

We have completed the basis step and the inductive step, so by strong induction we know that Algorithm 4 is correct. 

Recursion and Iteration

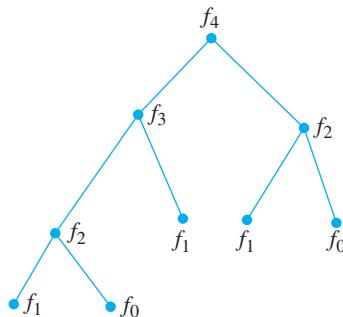
A recursive definition expresses the value of a function at a positive integer in terms of the values of the function at smaller integers. This means that we can devise a recursive algorithm to evaluate a recursively defined function at a positive integer. Instead of successively reducing the computation to the evaluation of the function at smaller integers, we can start with the value of the function at one or more integers, the base cases, and successively apply the recursive definition to find the values of the function at successive larger integers. Such a procedure is called **iterative**. Often an iterative approach for the evaluation of a recursively defined sequence requires much less computation than a procedure using recursion (unless special-purpose recursive machines are used). This is illustrated by the iterative and recursive procedures for finding the n th Fibonacci number. The recursive procedure is given first.

ALGORITHM 7 A Recursive Algorithm for Fibonacci Numbers.

```
procedure fibonacci(n: nonnegative integer)
if n = 0 then return 0
else if n = 1 then return 1
else return fibonacci(n - 1) + fibonacci(n - 2)
{output is fibonacci(n)}
```

When we use a recursive procedure to find f_n , we first express f_n as $f_{n-1} + f_{n-2}$. Then we replace both of these Fibonacci numbers by the sum of two previous Fibonacci numbers, and so on. When f_1 or f_0 arises, it is replaced by its value.

Note that at each stage of the recursion, until f_1 or f_0 is obtained, the number of Fibonacci numbers to be evaluated has doubled. For instance, when we find f_4 using this recursive algorithm, we must carry out all the computations illustrated in the tree diagram in Figure 1. This

**FIGURE 1** Evaluating f_4 Recursively.

tree consists of a root labeled with f_4 , and branches from the root to vertices labeled with the two Fibonacci numbers f_3 and f_2 that occur in the reduction of the computation of f_4 . Each subsequent reduction produces two branches in the tree. This branching ends when f_0 and f_1 are reached. The reader can verify that this algorithm requires $f_{n+1} - 1$ additions to find f_n .

Now consider the amount of computation required to find f_n using the iterative approach in Algorithm 8.

ALGORITHM 8 An Iterative Algorithm for Computing Fibonacci Numbers.

```

procedure iterative_fibonacci(n: nonnegative integer)
if n = 0 then return 0
else
  x := 0
  y := 1
  for i := 1 to n - 1
    z := x + y
    x := y
    y := z
  return y
{output is the nth Fibonacci number}
  
```

This procedure initializes x as $f_0 = 0$ and y as $f_1 = 1$. When the loop is traversed, the sum of x and y is assigned to the auxiliary variable z . Then x is assigned the value of y and y is assigned the value of the auxiliary variable z . Therefore, after going through the loop the first time, it follows that x equals f_1 and y equals $f_0 + f_1 = f_2$. Furthermore, after going through the loop $n - 1$ times, x equals f_{n-1} and y equals f_n (the reader should verify this statement). Only $n - 1$ additions have been used to find f_n with this iterative approach when $n > 1$. Consequently, this algorithm requires far less computation than does the recursive algorithm.

We have shown that a recursive algorithm may require far more computation than an iterative one when a recursively defined function is evaluated. It is sometimes preferable to use a recursive procedure even if it is less efficient than the iterative procedure. In particular, this is true when the recursive approach is easily implemented and the iterative approach is not. (Also, machines designed to handle recursion may be available that eliminate the advantage of using iteration.)

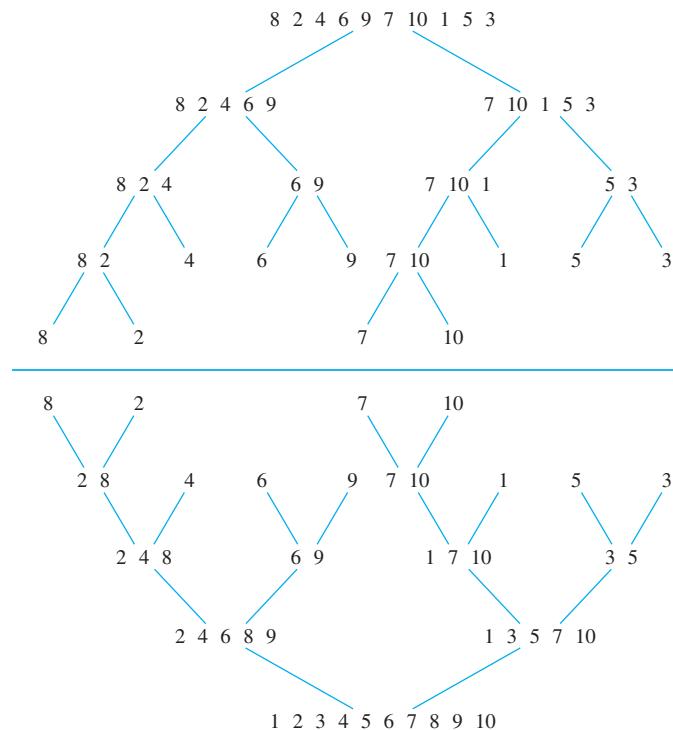


FIGURE 2 The Merge Sort of 8, 2, 4, 6, 9, 7, 10, 1, 5, 3.

The Merge Sort



We now describe a recursive sorting algorithm called the **merge sort** algorithm. We will demonstrate how the merge sort algorithm works with an example before describing it in generality.

EXAMPLE 9

Use the merge sort to put the terms of the list 8, 2, 4, 6, 9, 7, 10, 1, 5, 3 in increasing order.

Solution: A merge sort begins by splitting the list into individual elements by successively splitting lists in two. The progression of sublists for this example is represented with the balanced binary tree of height 4 shown in the upper half of Figure 2.

Sorting is done by successively merging pairs of lists. At the first stage, pairs of individual elements are merged into lists of length two in increasing order. Then successive merges of pairs of lists are performed until the entire list is put into increasing order. The succession of merged lists in increasing order is represented by the balanced binary tree of height 4 shown in the lower half of Figure 2 (note that this tree is displayed “upside down”).



In general, a merge sort proceeds by iteratively splitting lists into two sublists of equal length (or where one sublist has one more element than the other) until each sublist contains one element. This succession of sublists can be represented by a balanced binary tree. The procedure continues by successively merging pairs of lists, where both lists are in increasing order, into a larger list with elements in increasing order, until the original list is put into increasing order. The succession of merged lists can be represented by a balanced binary tree.

We can also describe the merge sort recursively. To do a merge sort, we split a list into two sublists of equal, or approximately equal, size, sorting each sublist using the merge sort

algorithm, and then merging the two lists. The recursive version of the merge sort is given in Algorithm 9. This algorithm uses the subroutine *merge*, which is described in Algorithm 10.

ALGORITHM 9 A Recursive Merge Sort.

```

procedure mergesort( $L = a_1, \dots, a_n$ )
if  $n > 1$  then
     $m := \lfloor n/2 \rfloor$ 
     $L_1 := a_1, a_2, \dots, a_m$ 
     $L_2 := a_{m+1}, a_{m+2}, \dots, a_n$ 
     $L := \text{merge}(\text{mergesort}(L_1), \text{mergesort}(L_2))$ 
    { $L$  is now sorted into elements in nondecreasing order}

```

An efficient algorithm for merging two ordered lists into a larger ordered list is needed to implement the merge sort. We will now describe such a procedure.

EXAMPLE 10 Merge the two lists 2, 3, 5, 6 and 1, 4.

Solution: Table 1 illustrates the steps we use. First, compare the smallest elements in the two lists, 2 and 1, respectively. Because 1 is the smaller, put it at the beginning of the merged list and remove it from the second list. At this stage, the first list is 2, 3, 5, 6, the second is 4, and the combined list is 1.

Next, compare 2 and 4, the smallest elements of the two lists. Because 2 is the smaller, add it to the combined list and remove it from the first list. At this stage the first list is 3, 5, 6, the second is 4, and the combined list is 1, 2.

Continue by comparing 3 and 4, the smallest elements of their respective lists. Because 3 is the smaller of these two elements, add it to the combined list and remove it from the first list. At this stage the first list is 5, 6, and the second is 4. The combined list is 1, 2, 3.

Then compare 5 and 4, the smallest elements in the two lists. Because 4 is the smaller of these two elements, add it to the combined list and remove it from the second list. At this stage the first list is 5, 6, the second list is empty, and the combined list is 1, 2, 3, 4.

Finally, because the second list is empty, all elements of the first list can be appended to the end of the combined list in the order they occur in the first list. This produces the ordered list 1, 2, 3, 4, 5, 6. 

We will now consider the general problem of merging two ordered lists L_1 and L_2 into an ordered list L . We will describe an algorithm for solving this problem. Start with an empty list L . Compare the smallest elements of the two lists. Put the smaller of these two elements at the right end of L , and remove it from the list it was in. Next, if one of L_1 and L_2 is empty, append the other (nonempty) list to L , which completes the merging. If neither L_1 nor L_2 is empty, repeat this process. Algorithm 10 gives a pseudocode description of this procedure.

TABLE 1 Merging the Two Sorted Lists 2, 3, 5, 6 and 1, 4.

First List	Second List	Merged List	Comparison
2 3 5 6	1 4		1 < 2
2 3 5 6	4	1	2 < 4
3 5 6	4	1 2	3 < 4
5 6	4	1 2 3	4 < 5
5 6		1 2 3 4	
		1 2 3 4 5 6	

We will need estimates for the number of comparisons used to merge two ordered lists in the analysis of the merge sort. We can easily obtain such an estimate for Algorithm 10. Each time a comparison of an element from L_1 and an element from L_2 is made, an additional element is added to the merged list L . However, when either L_1 or L_2 is empty, no more comparisons are needed. Hence, Algorithm 10 is least efficient when $m + n - 2$ comparisons are carried out, where m and n are the number of elements in L_1 and L_2 , respectively, leaving one element in each of L_1 and L_2 . The next comparison will be the last one needed, because it will make one of these lists empty. Hence, Algorithm 10 uses no more than $m + n - 1$ comparisons. Lemma 1 summarizes this estimate.

ALGORITHM 10 Merging Two Lists.

```

procedure merge( $L_1, L_2$ : sorted lists)
   $L :=$  empty list
  while  $L_1$  and  $L_2$  are both nonempty
    remove smaller of first elements of  $L_1$  and  $L_2$  from its list; put it at the right end of  $L$ 
    if this removal makes one list empty then remove all elements from the other list and
      append them to  $L$ 
  return  $L$  { $L$  is the merged list with elements in increasing order}

```

LEMMA 1

Two sorted lists with m elements and n elements can be merged into a sorted list using no more than $m + n - 1$ comparisons.

Sometimes two sorted lists of length m and n can be merged using far fewer than $m + n - 1$ comparisons. For instance, when $m = 1$, a binary search procedure can be applied to put the one element in the first list into the second list. This requires only $\lceil \log n \rceil$ comparisons, which is much smaller than $m + n - 1 = n$, for $m = 1$. On the other hand, for some values of m and n , Lemma 1 gives the best possible bound. That is, there are lists with m and n elements that cannot be merged using fewer than $m + n - 1$ comparisons. (See Exercise 47.)

We can now analyze the complexity of the merge sort. Instead of studying the general problem, we will assume that n , the number of elements in the list, is a power of 2, say 2^m . This will make the analysis less complicated, but when this is not the case, various modifications can be applied that will yield the same estimate.

At the first stage of the splitting procedure, the list is split into two sublists, of 2^{m-1} elements each, at level 1 of the tree generated by the splitting. This process continues, splitting the two sublists with 2^{m-1} elements into four sublists of 2^{m-2} elements each at level 2, and so on. In general, there are 2^{k-1} lists at level $k - 1$, each with 2^{m-k+1} elements. These lists at level $k - 1$ are split into 2^k lists at level k , each with 2^{m-k} elements. At the end of this process, we have 2^m lists each with one element at level m .

We start merging by combining pairs of the 2^m lists of one element into 2^{m-1} lists, at level $m - 1$, each with two elements. To do this, 2^{m-1} pairs of lists with one element each are merged. The merger of each pair requires exactly one comparison.

The procedure continues, so that at level k ($k = m, m - 1, m - 2, \dots, 3, 2, 1$), 2^k lists each with 2^{m-k} elements are merged into 2^{k-1} lists, each with 2^{m-k+1} elements, at level $k - 1$. To do this a total of 2^{k-1} mergers of two lists, each with 2^{m-k} elements, are needed. But,

by Lemma 1, each of these mergers can be carried out using at most $2^{m-k} + 2^{m-k} - 1 = 2^{m-k+1} - 1$ comparisons. Hence, going from level k to $k - 1$ can be accomplished using at most $2^{k-1}(2^{m-k+1} - 1)$ comparisons.

Summing all these estimates shows that the number of comparisons required for the merge sort is at most

$$\sum_{k=1}^m 2^{k-1}(2^{m-k+1} - 1) = \sum_{k=1}^m 2^m - \sum_{k=1}^m 2^{k-1} = m2^m - (2^m - 1) = n \log n - n + 1,$$

because $m = \log n$ and $n = 2^m$. (We evaluated $\sum_{k=1}^m 2^m$ by noting that it is the sum of m identical terms, each equal to 2^m . We evaluated $\sum_{k=1}^m 2^{k-1}$ using the formula for the sum of the terms of a geometric progression from Theorem 1 of Section 2.4.)

Theorem 1 summarizes what we have discovered about the worst-case complexity of the merge sort algorithm.

THEOREM 1

The number of comparisons needed to merge sort a list with n elements is $O(n \log n)$.

In Chapter 11 we will show that the fastest comparison-based sorting algorithm have $O(n \log n)$ time complexity. (A comparison-based sorting algorithm has the comparison of two elements as its basic operation.) Theorem 1 tells us that the merge sort achieves this best possible big- O estimate for the complexity of a sorting algorithm. We describe another efficient algorithm, the quick sort, in the preamble to Exercise 50.

Exercises

1. Trace Algorithm 1 when it is given $n = 5$ as input. That is, show all steps used by Algorithm 1 to find $5!$, as is done in Example 1 to find $4!$.
2. Trace Algorithm 1 when it is given $n = 6$ as input. That is, show all steps used by Algorithm 1 to find $6!$, as is done in Example 1 to find $4!$.
3. Trace Algorithm 3 when it finds $\gcd(8, 13)$. That is, show all the steps used by Algorithm 3 to find $\gcd(8, 13)$.
4. Trace Algorithm 3 when it finds $\gcd(12, 17)$. That is, show all the steps used by Algorithm 3 to find $\gcd(12, 17)$.
5. Trace Algorithm 4 when it is given $m = 5$, $n = 11$, and $b = 3$ as input. That is, show all the steps Algorithm 4 uses to find $3^{11} \bmod 5$.
6. Trace Algorithm 4 when it is given $m = 7$, $n = 10$, and $b = 2$ as input. That is, show all the steps Algorithm 4 uses to find $2^{10} \bmod 7$.
7. Give a recursive algorithm for computing nx whenever n is a positive integer and x is an integer, using just addition.
8. Give a recursive algorithm for finding the sum of the first n positive integers.
9. Give a recursive algorithm for finding the sum of the first n odd positive integers.
10. Give a recursive algorithm for finding the maximum of a finite set of integers, making use of the fact that the maximum of n integers is the larger of the last integer in the list and the maximum of the first $n - 1$ integers in the list.
11. Give a recursive algorithm for finding the minimum of a finite set of integers, making use of the fact that the minimum of n integers is the smaller of the last integer in the list and the minimum of the first $n - 1$ integers in the list.
12. Devise a recursive algorithm for finding $x^n \bmod m$ whenever n , x , and m are positive integers based on the fact that $x^n \bmod m = (x^{n-1} \bmod m \cdot x \bmod m) \bmod m$.
13. Give a recursive algorithm for finding $n! \bmod m$ whenever n and m are positive integers.
14. Give a recursive algorithm for finding a **mode** of a list of integers. (A **mode** is an element in the list that occurs at least as often as every other element.)
15. Devise a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b with $a < b$ using the fact that $\gcd(a, b) = \gcd(a, b - a)$.
16. Prove that the recursive algorithm for finding the sum of the first n positive integers you found in Exercise 8 is correct.

17. Describe a recursive algorithm for multiplying two non-negative integers x and y based on the fact that $xy = 2(x \cdot (y/2))$ when y is even and $xy = 2(x \cdot \lfloor y/2 \rfloor) + x$ when y is odd, together with the initial condition $xy = 0$ when $y = 0$.
18. Prove that Algorithm 1 for computing $n!$ when n is a non-negative integer is correct.
19. Prove that Algorithm 3 for computing $\gcd(a, b)$ when a and b are positive integers with $a < b$ is correct.
20. Prove that the algorithm you devised in Exercise 17 is correct.
21. Prove that the recursive algorithm that you found in Exercise 7 is correct.
22. Prove that the recursive algorithm that you found in Exercise 10 is correct.
23. Devise a recursive algorithm for computing n^2 where n is a nonnegative integer, using the fact that $(n+1)^2 = n^2 + 2n + 1$. Then prove that this algorithm is correct.
24. Devise a recursive algorithm to find a^{2^n} , where a is a real number and n is a positive integer. [Hint: Use the equality $a^{2^{n+1}} = (a^{2^n})^2$.]
25. How does the number of multiplications used by the algorithm in Exercise 24 compare to the number of multiplications used by Algorithm 2 to evaluate a^{2^n} ?
- *26. Use the algorithm in Exercise 24 to devise an algorithm for evaluating a^n when n is a nonnegative integer. [Hint: Use the binary expansion of n .]
- *27. How does the number of multiplications used by the algorithm in Exercise 26 compare to the number of multiplications used by Algorithm 2 to evaluate a^n ?
28. How many additions are used by the recursive and iterative algorithms given in Algorithms 7 and 8, respectively, to find the Fibonacci number f_7 ?
29. Devise a recursive algorithm to find the n th term of the sequence defined by $a_0 = 1$, $a_1 = 2$, and $a_n = a_{n-1} \cdot a_{n-2}$, for $n = 2, 3, 4, \dots$
30. Devise an iterative algorithm to find the n th term of the sequence defined in Exercise 29.
31. Is the recursive or the iterative algorithm for finding the sequence in Exercise 29 more efficient?
32. Devise a recursive algorithm to find the n th term of the sequence defined by $a_0 = 1$, $a_1 = 2$, $a_2 = 3$, and $a_n = a_{n-1} + a_{n-2} + a_{n-3}$, for $n = 3, 4, 5, \dots$
33. Devise an iterative algorithm to find the n th term of the sequence defined in Exercise 32.
34. Is the recursive or the iterative algorithm for finding the sequence in Exercise 32 more efficient?
35. Give iterative and recursive algorithms for finding the n th term of the sequence defined by $a_0 = 1$, $a_1 = 3$, $a_2 = 5$, and $a_n = a_{n-1} \cdot a_{n-2}^2 \cdot a_{n-3}^3$. Which is more efficient?
36. Give a recursive algorithm to find the number of partitions of a positive integer based on the recursive definition given in Exercise 47 in Section 5.3.
37. Give a recursive algorithm for finding the reversal of a bit string. (See the definition of the reversal of a bit string in the preamble of Exercise 34 in Section 5.3.)
38. Give a recursive algorithm for finding the string w^i , the concatenation of i copies of w , when w is a bit string.
39. Prove that the recursive algorithm for finding the reversal of a bit string that you gave in Exercise 37 is correct.
40. Prove that the recursive algorithm for finding the concatenation of i copies of a bit string that you gave in Exercise 38 is correct.
- *41. Give a recursive algorithm for tiling a $2^n \times 2^n$ checkerboard with one square missing using right triominoes.
42. Give a recursive algorithm for triangulating a simple polygon with n sides, using Lemma 1 in Section 5.2.
43. Give a recursive algorithm for computing values of the Ackermann function. [Hint: See the preamble to Exercise 48 in Section 5.3.]
44. Use a merge sort to sort 4, 3, 2, 5, 1, 8, 7, 6 into increasing order. Show all the steps used by the algorithm.
45. Use a merge sort to sort $b, d, a, f, g, h, z, p, o, k$ into alphabetic order. Show all the steps used by the algorithm.
46. How many comparisons are required to merge these pairs of lists using Algorithm 10?
- a) 1, 3, 5, 7, 9; 2, 4, 6, 8, 10
 - b) 1, 2, 3, 4, 5; 6, 7, 8, 9, 10
 - c) 1, 5, 6, 7, 8; 2, 3, 4, 9, 10
47. Show that for all positive integers m and n there are sorted lists with m elements and n elements, respectively, such that Algorithm 10 uses $m + n - 1$ comparisons to merge them into one sorted list.
- *48. What is the least number of comparisons needed to merge any two lists in increasing order into one list in increasing order when the number of elements in the two lists are
 a) 1, 4? b) 2, 4? c) 3, 4? d) 4, 4?
- *49. Prove that the merge sort algorithm is correct.
- The **quick sort** is an efficient algorithm. To sort a_1, a_2, \dots, a_n , this algorithm begins by taking the first element a_1 and forming two sublists, the first containing those elements that are less than a_1 , in the order they arise, and the second containing those elements greater than a_1 , in the order they arise. Then a_1 is put at the end of the first sublist. This procedure is repeated recursively for each sublist, until all sublists contain one item. The ordered list of n items is obtained by combining the sublists of one item in the order they occur.
50. Sort 3, 5, 7, 8, 1, 9, 2, 4, 6 using the quick sort.
51. Let a_1, a_2, \dots, a_n be a list of n distinct real numbers. How many comparisons are needed to form two sublists from this list, the first containing elements less than a_1 and the second containing elements greater than a_1 ?
52. Describe the quick sort algorithm using pseudocode.
53. What is the largest number of comparisons needed to order a list of four elements using the quick sort algorithm?
54. What is the least number of comparisons needed to order a list of four elements using the quick sort algorithm?
55. Determine the worst-case complexity of the quick sort algorithm in terms of the number of comparisons used.

5.5 Program Correctness

Introduction

Suppose that we have designed an algorithm to solve a problem and have written a program to implement it. How can we be sure that the program always produces the correct answer? After all the bugs have been removed so that the syntax is correct, we can test the program with sample input. It is not correct if an incorrect result is produced for any sample input. But even if the program gives the correct answer for all sample input, it may not always produce the correct answer (unless all possible input has been tested). We need a proof to show that the program *always* gives the correct output.

Program verification, the proof of correctness of programs, uses the rules of inference and proof techniques described in this chapter, including mathematical induction. Because an incorrect program can lead to disastrous results, a large amount of methodology has been constructed for verifying programs. Efforts have been devoted to automating program verification so that it can be carried out using a computer. However, only limited progress has been made toward this goal. Indeed, some mathematicians and theoretical computer scientists argue that it will never be realistic to mechanize the proof of correctness of complex programs.

Some of the concepts and methods used to prove that programs are correct will be introduced in this section. Many different methods have been devised for proving that programs are correct. We will discuss a widely used method for program verification introduced by Tony Hoare in this section; several other methods are also commonly used. Furthermore, we will not develop a complete methodology for program verification in this book. This section is meant to be a brief introduction to the area of program verification, which ties together the rules of logic, proof techniques, and the concept of an algorithm.

Program Verification

A program is said to be **correct** if it produces the correct output for every possible input. A proof that a program is correct consists of two parts. The first part shows that the correct answer is obtained if the program terminates. This part of the proof establishes the **partial correctness** of the program. The second part of the proof shows that the program always terminates.

To specify what it means for a program to produce the correct output, two propositions are used. The first is the **initial assertion**, which gives the properties that the input values must have. The second is the **final assertion**, which gives the properties that the output of the program should have, if the program did what was intended. The appropriate initial and final assertions must be provided when a program is checked.

DEFINITION 1

A program, or program segment, S is said to be *partially correct with respect to* the initial assertion p and the final assertion q if whenever p is true for the input values of S and S terminates, then q is true for the output values of S . The notation $p\{S\}q$ indicates that the program, or program segment, S is partially correct with respect to the initial assertion p and the final assertion q .

Note: The notation $p\{S\}q$ is known as a *Hoare triple*. Tony Hoare introduced the concept of partial correctness.



Note that the notion of partial correctness has nothing to do with whether a program terminates; it focuses only on whether the program does what it is expected to do if it terminates.

A simple example illustrates the concepts of initial and final assertions.

EXAMPLE 1 Show that the program segment

$$\begin{aligned}y &:= 2 \\z &:= x + y\end{aligned}$$



is correct with respect to the initial assertion $p: x = 1$ and the final assertion $q: z = 3$.

Solution: Suppose that p is true, so that $x = 1$ as the program begins. Then y is assigned the value 2, and z is assigned the sum of the values of x and y , which is 3. Hence, S is correct with respect to the initial assertion p and the final assertion q . Thus, $p\{S\}q$ is true. \blacktriangleleft

Rules of Inference

A useful rule of inference proves that a program is correct by splitting the program into a sequence of subprograms and then showing that each subprogram is correct.

Suppose that the program S is split into subprograms S_1 and S_2 . Write $S = S_1; S_2$ to indicate that S is made up of S_1 followed by S_2 . Suppose that the correctness of S_1 with respect to the initial assertion p and final assertion q , and the correctness of S_2 with respect to the initial assertion q and the final assertion r , have been established. It follows that if p is true and S_1 is executed and terminates, then q is true; and if q is true, and S_2 executes and terminates, then r is true. Thus, if p is true and $S = S_1; S_2$ is executed and terminates, then r is true. This rule of inference, called the **composition rule**, can be stated as

$$\frac{\begin{array}{c} p\{S_1\}q \\ q\{S_2\}r \end{array}}{\therefore p\{S_1; S_2\}r}.$$

This rule of inference will be used later in this section.

Next, some rules of inference for program segments involving conditional statements and loops will be given. Because programs can be split into segments for proofs of correctness, this will let us verify many different programs.

Conditional Statements

First, rules of inference for conditional statements will be given. Suppose that a program segment has the form

```
if condition then
    S
```

where S is a block of statements. Then S is executed if *condition* is true, and it is not executed when *condition* is false. To verify that this segment is correct with respect to the initial assertion p and final assertion q , two things must be done. First, it must be shown that when p is true and *condition* is also true, then q is true after S terminates. Second, it must be shown that when p is true and *condition* is false, then q is true (because in this case S does not execute).

This leads to the following rule of inference:

$$\frac{\begin{array}{c} (p \wedge \text{condition})\{S\}q \\ (p \wedge \neg\text{condition}) \rightarrow q \end{array}}{\therefore p\{\text{if condition then } S\}q}.$$

Example 2 illustrates how this rule of inference is used.

EXAMPLE 2 Verify that the program segment

```
if  $x > y$  then
     $y := x$ 
```

is correct with respect to the initial assertion **T** and the final assertion $y \geq x$.

Solution: When the initial assertion is true and $x > y$, the assignment $y := x$ is carried out. Hence, the final assertion, which asserts that $y \geq x$, is true in this case. Moreover, when the initial assertion is true and $x > y$ is false, so that $x \leq y$, the final assertion is again true. Hence, using the rule of inference for program segments of this type, this program is correct with respect to the given initial and final assertions. \blacktriangleleft

Similarly, suppose that a program has a statement of the form

```
if condition then
     $S_1$ 
else
     $S_2$ 
```

If *condition* is true, then S_1 executes; if *condition* is false, then S_2 executes. To verify that this program segment is correct with respect to the initial assertion p and the final assertion q , two things must be done. First, it must be shown that when p is true and *condition* is true, then q is true after S_1 terminates. Second, it must be shown that when p is true and *condition* is false, then q is true after S_2 terminates. This leads to the following rule of inference:

$$\frac{(p \wedge \text{condition})\{S_1\}q \quad (p \wedge \neg\text{condition})\{S_2\}q}{\therefore p\{\text{if condition then } S_1 \text{ else } S_2\}q.}$$



C. ANTHONY R. HOARE (BORN 1934) Tony Hoare was born in Colombo, Ceylon (now known as Sri Lanka), where his father was a civil servant of the British Empire and his mother's father owned a plantation. He spent his early childhood in Ceylon, moving to England in 1945. Hoare studied philosophy, together with the classics, at the University of Oxford, where he became interested in computing as a result of his fascination with the power of mathematical logic and the certainty of mathematical truth. He received his bachelors degree from Oxford in 1956.

Hoare learned Russian during his service in the Royal Navy, and latter studied the computer translation of natural languages at Moscow State University. He returned to England in 1960, taking a job at a small computer manufacturer, where he wrote a compiler for the programming language Algol. In 1968, he became Professor of Computing Science at the Queen's University, Belfast; in 1977, he moved to the University of Oxford as Professor of Computing; he is now Professor Emeritus. He is a Fellow of the Royal Society and also holds a position at Microsoft Research in Cambridge.

Hoare has made many contributions to the theory of programming languages and to programming methodology. He was first to define a programming language based on how programs could be proved correct with respect to their specifications. Hoare also invented quick sort, one of the most commonly used sorting algorithms (see the preamble to Exercise 50 in Section 5.4). He received the ACM Turing Award in 1980 and in 2000 he was knighted for services to education and computer science. Hoare is a noted writer in the technical and social aspects of computer science.

Example 3 illustrates how this rule of inference is used.

EXAMPLE 3 Verify that the program segment

```
if  $x < 0$  then
     $abs := -x$ 
else
     $abs := x$ 
```

is correct with respect to the initial assertion \mathbf{T} and the final assertion $abs = |x|$.

Solution: Two things must be demonstrated. First, it must be shown that if the initial assertion is true and $x < 0$, then $abs = |x|$. This is correct, because when $x < 0$ the assignment statement $abs := -x$ sets $abs = -x$, which is $|x|$ by definition when $x < 0$. Second, it must be shown that if the initial assertion is true and $x < 0$ is false, so that $x \geq 0$, then $abs = |x|$. This is also correct, because in this case the program uses the assignment statement $abs := x$, and x is $|x|$ by definition when $x \geq 0$, so $abs := x$. Hence, using the rule of inference for program segments of this type, this segment is correct with respect to the given initial and final assertions. 

Loop Invariants



Next, proofs of correctness of **while** loops will be described. To develop a rule of inference for program segments of the type

```
while condition
    S
```

note that S is repeatedly executed until $condition$ becomes false. An assertion that remains true each time S is executed must be chosen. Such an assertion is called a **loop invariant**. In other words, p is a loop invariant if $(p \wedge condition)\{S\}p$ is true.

Suppose that p is a loop invariant. It follows that if p is true before the program segment is executed, p and $\neg condition$ are true after termination, if it occurs. This rule of inference is

$$\frac{(p \wedge condition)\{S\}p}{\therefore p\{\text{while } condition \text{ } S\}(\neg condition \wedge p)}.$$

The use of a loop invariant is illustrated in Example 4.

EXAMPLE 4 A loop invariant is needed to verify that the program segment



```
i := 1
factorial := 1
while i < n
    i := i + 1
    factorial := factorial · i
```

terminates with $factorial = n!$ when n is a positive integer.

Let p be the assertion “ $\text{factorial} = i!$ and $i \leq n$.” We first prove that p is a loop invariant. Suppose that, at the beginning of one execution of the **while** loop, p is true and the condition of the **while** loop holds; in other words, assume that $\text{factorial} = i!$ and that $i < n$. The new values i_{new} and $\text{factorial}_{\text{new}}$ of i and factorial are $i_{\text{new}} = i + 1$ and $\text{factorial}_{\text{new}} = \text{factorial} \cdot (i + 1) = (i + 1)! = i_{\text{new}}!$. Because $i < n$, we also have $i_{\text{new}} = i + 1 \leq n$. Thus, p is true at the end of the execution of the loop. This shows that p is a loop invariant.

Now we consider the program segment. Just before entering the loop, $i = 1 \leq n$ and $\text{factorial} = 1 = 1! = i!$ both hold, so p is true. Because p is a loop invariant, the rule of inference just introduced implied that if the **while** loop terminates, it terminates with p true and with $i < n$ false. In this case, at the end, $\text{factorial} = i!$ and $i \leq n$ are true, but $i < n$ is false; in other words, $i = n$ and $\text{factorial} = i! = n!$, as desired.

Finally, we need to check that the **while** loop actually terminates. At the beginning of the program i is assigned the value 1, so after $n - 1$ traversals of the loop, the new value of i will be n , and the loop terminates at that point. 

A final example will be given to show how the various rules of inference can be used to verify the correctness of a longer program.

EXAMPLE 5 We will outline how to verify the correctness of the program S for computing the product of two integers.

```

procedure multiply( $m, n$ : integers)

     $S_1 \left\{ \begin{array}{l} \text{if } n < 0 \text{ then } a := -n \\ \text{else } a := n \end{array} \right.$ 
     $S_2 \left\{ \begin{array}{l} k := 0 \\ x := 0 \end{array} \right.$ 
     $S_3 \left\{ \begin{array}{l} \text{while } k < a \\ \quad x := x + m \\ \quad k := k + 1 \end{array} \right.$ 
     $S_4 \left\{ \begin{array}{l} \text{if } n < 0 \text{ then } product := -x \\ \text{else } product := x \end{array} \right.$ 

return product
{product equals  $mn$ }

```

The goal is to prove that after S is executed, $product$ has the value mn . The proof of correctness can be carried out by splitting S into four segments, with $S = S_1; S_2; S_3; S_4$, as shown in the listing of S . The rule of composition can be used to build the correctness proof. Here is how the argument proceeds. The details will be left as an exercise for the reader.

Let p be the initial assertion “ m and n are integers.” Then, it can be shown that $p\{S_1\}q$ is true, when q is the proposition $p \wedge (a = |n|)$. Next, let r be the proposition $q \wedge (k = 0) \wedge (x = 0)$. It is easily verified that $q\{S_2\}r$ is true. It can be shown that “ $x = mk$ and $k \leq a$ ” is an invariant for the loop in S_3 . Furthermore, it is easy to see that the loop terminates after a iterations, with $k = a$, so $x = ma$ at this point. Because r implies that $x = m \cdot 0$ and $0 \leq a$, the loop invariant is true before the loop is entered. Because the loop terminates with $k = a$, it follows that $r\{S_3\}s$ is true where s is the proposition “ $x = ma$ and $a = |n|$.” Finally, it can be shown that S_4 is correct with respect to the initial assertion s and final assertion t , where t is the proposition “ $product = mn$.”

Putting all this together, because $p\{S_1\}q$, $q\{S_2\}r$, $r\{S_3\}s$, and $s\{S_4\}t$ are all true, it follows from the rule of composition that $p\{S\}t$ is true. Furthermore, because all four segments terminate, S does terminate. This verifies the correctness of the program. 

Exercises

1. Prove that the program segment

```
y := 1
z := x + y
```

is correct with respect to the initial assertion $x = 0$ and the final assertion $z = 1$.

2. Verify that the program segment

```
if x < 0 then x := 0
```

is correct with respect to the initial assertion T and the final assertion $x \geq 0$.

3. Verify that the program segment

```
x := 2
z := x + y
if y > 0 then
    z := z + 1
else
    z := 0
```

is correct with respect to the initial assertion $y = 3$ and the final assertion $z = 6$.

4. Verify that the program segment

```
if x < y then
    min := x
else
    min := y
```

is correct with respect to the initial assertion T and the final assertion $(x \leq y \wedge \min = x) \vee (x > y \wedge \min = y)$.

- *5. Devise a rule of inference for verification of partial correctness of statements of the form

```
if condition 1 then
    S1
else if condition 2 then
    S2
    :
else
    Sn
```

where S_1, S_2, \dots, S_n are blocks.

6. Use the rule of inference developed in Exercise 5 to verify that the program

if $x < 0$ **then**

```
y := -2|x|/x
else if  $x > 0$  then
```

```
y := 2|x|/x
else if  $x = 0$  then
```

```
y := 2
```

is correct with respect to the initial assertion T and the final assertion $y = 2$.

7. Use a loop invariant to prove that the following program segment for computing the n th power, where n is a positive integer, of a real number x is correct.

```
power := 1
i := 1
while i ≤ n
    power := power * x
    i := i + 1
```

- *8. Prove that the iterative program for finding f_n given in Section 5.4 is correct.

9. Provide all the details in the proof of correctness given in Example 5.

10. Suppose that both the conditional statement $p_0 \rightarrow p_1$ and the program assertion $p_1\{S\}q$ are true. Show that $p_0\{S\}q$ also must be true.

11. Suppose that both the program assertion $p\{S\}q_0$ and the conditional statement $q_0 \rightarrow q_1$ are true. Show that $p\{S\}q_1$ also must be true.

12. This program computes quotients and remainders.

```
r := a
q := 0
while r ≥ d
    r := r - d
    q := q + 1
```

Verify that it is partially correct with respect to the initial assertion “ a and d are positive integers” and the final assertion “ q and r are integers such that $a = dq + r$ and $0 \leq r < d$.”

13. Use a loop invariant to verify that the Euclidean algorithm (Algorithm 1 in Section 4.3) is partially correct with respect to the initial assertion “ a and b are positive integers” and the final assertion “ $x = \gcd(a, b)$.”

Key Terms and Results

TERMS

sequence: a function with domain that is a subset of the set of integers

geometric progression: a sequence of the form a, ar, ar^2, \dots , where a and r are real numbers

arithmetic progression: a sequence of the form $a, a + d, a + 2d, \dots$, where a and d are real numbers

the principle of mathematical induction: the statement $\forall n P(n)$ is true if $P(1)$ is true and $\forall k[P(k) \rightarrow P(k + 1)]$ is true.

basis step: the proof of $P(1)$ in a proof by mathematical induction of $\forall n P(n)$

inductive step: the proof of $P(k) \rightarrow P(k + 1)$ for all positive integers k in a proof by mathematical induction of $\forall n P(n)$

strong induction: the statement $\forall n P(n)$ is true if $P(1)$ is true and $\forall k[(P(1) \wedge \dots \wedge P(k)) \rightarrow P(k+1)]$ is true

well-ordering property: Every nonempty set of nonnegative integers has a least element.

recursive definition of a function: a definition of a function that specifies an initial set of values and a rule for obtaining values of this function at integers from its values at smaller integers

recursive definition of a set: a definition of a set that specifies an initial set of elements in the set and a rule for obtaining other elements from those in the set

structural induction: a technique for proving results about recursively defined sets

recursive algorithm: an algorithm that proceeds by reducing a problem to the same problem with smaller input

merge sort: a sorting algorithm that sorts a list by splitting it in two, sorting each of the two resulting lists, and merging the results into a sorted list

iteration: a procedure based on the repeated use of operations in a loop

program correctness: verification that a procedure always produces the correct result

loop invariant: a property that remains true during every traversal of a loop

initial assertion: the statement specifying the properties of the input values of a program

final assertion: the statement specifying the properties the output values should have if the program worked correctly

Review Questions

1. a) Can you use the principle of mathematical induction to find a formula for the sum of the first n terms of a sequence?
b) Can you use the principle of mathematical induction to determine whether a given formula for the sum of the first n terms of a sequence is correct?
c) Find a formula for the sum of the first n even positive integers, and prove it using mathematical induction.
2. a) For which positive integers n is $11n + 17 \leq 2^n$?
b) Prove the conjecture you made in part (a) using mathematical induction.
3. a) Which amounts of postage can be formed using only 5-cent and 9-cent stamps?
b) Prove the conjecture you made using mathematical induction.
c) Prove the conjecture you made using strong induction.
d) Find a proof of your conjecture different from the ones you gave in (b) and (c).
4. Give two different examples of proofs that use strong induction.
5. a) State the well-ordering property for the set of positive integers.
b) Use this property to show that every positive integer greater than one can be written as the product of primes.
6. a) Explain why a function f from the set of positive integers to the set of real numbers is well-defined if it is defined recursively by specifying $f(1)$ and a rule for finding $f(n)$ from $f(n - 1)$.
b) Provide a recursive definition of the function $f(n) = (n + 1)!$.
7. a) Give a recursive definition of the Fibonacci numbers.
b) Show that $f_n > \alpha^{n-2}$ whenever $n \geq 3$, where f_n is the n th term of the Fibonacci sequence and $\alpha = (1 + \sqrt{5})/2$.
8. a) Explain why a sequence a_n is well defined if it is defined recursively by specifying a_1 and a_2 and a rule for finding a_n from a_1, a_2, \dots, a_{n-1} for $n = 3, 4, 5, \dots$.
b) Find the value of a_n if $a_1 = 1$, $a_2 = 2$, and $a_n = a_{n-1} + a_{n-2} + \dots + a_1$, for $n = 3, 4, 5, \dots$.
9. Give two examples of how well-formed formulae are defined recursively for different sets of elements and operators.
10. a) Give a recursive definition of the length of a string.
b) Use the recursive definition from part (a) and structural induction to prove that $l(xy) = l(x) + l(y)$.
11. a) What is a recursive algorithm?
b) Describe a recursive algorithm for computing the sum of n numbers in a sequence.
12. Describe a recursive algorithm for computing the greatest common divisor of two positive integers.
13. a) Describe the merge sort algorithm.
b) Use the merge sort algorithm to put the list 4, 10, 1, 5, 3, 8, 7, 2, 6, 9 in increasing order.
c) Give a big- O estimate for the number of comparisons used by the merge sort.
14. a) Does testing a computer program to see whether it produces the correct output for certain input values verify that the program always produces the correct output?
b) Does showing that a computer program is partially correct with respect to an initial assertion and a final assertion verify that the program always produces the correct output? If not, what else is needed?
15. What techniques can you use to show that a long computer program is partially correct with respect to an initial assertion and a final assertion?
16. What is a loop invariant? How is a loop invariant used?

Supplementary Exercises

1. Use mathematical induction to show that $\frac{2}{3} + \frac{2}{9} + \frac{2}{27} + \cdots + \frac{2}{3^n} = 1 - \frac{1}{3^n}$ whenever n is a positive integer.
2. Use mathematical induction to show that $1^3 + 3^3 + 5^3 + \cdots + (2n+1)^3 = (n+1)^2(2n^2+4n+1)$ whenever n is a positive integer.
3. Use mathematical induction to show that $1 \cdot 2^0 + 2 \cdot 2^1 + 3 \cdot 2^2 + \cdots + n \cdot 2^{n-1} = (n-1) \cdot 2^n + 1$ whenever n is a positive integer.
4. Use mathematical induction to show that

$$\frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \cdots + \frac{1}{(2n-1)(2n+1)} = \frac{n}{2n+1}$$

whenever n is a positive integer.

5. Show that

$$\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \cdots + \frac{1}{(3n-2)(3n+1)} = \frac{n}{3n+1}$$

whenever n is a positive integer.

6. Use mathematical induction to show that $2^n > n^2 + n$ whenever n is an integer greater than 4.
7. Use mathematical induction to show that $2^n > n^3$ whenever n is an integer greater than 9.
8. Find an integer N such that $2^n > n^4$ whenever n is greater than N . Prove that your result is correct using mathematical induction.
9. Use mathematical induction to prove that $a - b$ is a factor of $a^n - b^n$ whenever n is a positive integer.
10. Use mathematical induction to prove that 9 divides $n^3 + (n+1)^3 + (n+2)^3$ whenever n is a nonnegative integer.
11. Use mathematical induction to prove that 43 divides $6^{n+1} + 7^{2n-1}$ for every positive integer n .
12. Use mathematical induction to prove that 64 divides $3^{2n+2} + 56n + 55$ for every positive integer n .
13. Use mathematical induction to prove this formula for the sum of the terms of an arithmetic progression.

$$a + (a+d) + \cdots + (a+nd) = (n+1)(2a+nd)/2$$

14. Suppose that $a_j \equiv b_j \pmod{m}$ for $j = 1, 2, \dots, n$. Use mathematical induction to prove that

a) $\sum_{j=1}^n a_j \equiv \sum_{j=1}^n b_j \pmod{m}$.

b) $\prod_{j=1}^n a_j \equiv \prod_{j=1}^n b_j \pmod{m}$.

15. Show that if n is a positive integer, then

$$\sum_{k=1}^n \frac{k+4}{k(k+1)(k+2)} = \frac{n(3n+7)}{2(n+1)(n+2)}.$$

16. For which positive integers n is $n+6 < (n^2 - 8n)/16$? Prove your answer using mathematical induction.

17. (Requires calculus) Suppose that $f(x) = e^x$ and $g(x) = xe^x$. Use mathematical induction together with the prod-

uct rule and the fact that $f'(x) = e^x$ to prove that $g^{(n)}(x) = (x+n)e^x$ whenever n is a positive integer.

18. (Requires calculus) Suppose that $f(x) = e^x$ and $g(x) = e^{cx}$, where c is a constant. Use mathematical induction together with the chain rule and the fact that $f'(x) = e^x$ to prove that $g^{(n)} = c^n e^{cx}$ whenever n is a positive integer.

- *19. Formulate a conjecture about which Fibonacci numbers are even, and use a form of mathematical induction to prove your conjecture.

- *20. Determine which Fibonacci numbers are divisible by 3. Use a form of mathematical induction to prove your conjecture.

- *21. Prove that $f_k f_n + f_{k+1} f_{n+1} = f_{n+k+1}$ for all nonnegative integers n and k , where f_i denotes the i th Fibonacci number.

Recall from Example 15 of Section 2.4 that the sequence of **Lucas numbers** is defined by $l_0 = 2$, $l_1 = 1$, and $l_n = l_{n-1} + l_{n-2}$ for $n = 2, 3, 4, \dots$

22. Show that $f_n + f_{n+2} = l_{n+1}$ whenever n is a positive integer, where f_i and l_i are the i th Fibonacci number and i th Lucas number, respectively.

23. Show that $l_0^2 + l_1^2 + \cdots + l_n^2 = l_n l_{n+1} + 2$ whenever n is a nonnegative integer and l_i is the i th Lucas number.

- *24. Use mathematical induction to show that the product of any n consecutive positive integers is divisible by $n!$. [Hint: Use the identity $m(m+1)\cdots(m+n-1)/n! = (m-1)m(m+1)\cdots(m+n-2)/n! + m(m+1)\cdots(m+n-2)/(n-1)!$]

25. Use mathematical induction to show that $(\cos x + i \sin x)^n = \cos nx + i \sin nx$ whenever n is a positive integer. (Here i is the square root of -1 .) [Hint: Use the identities $\cos(a+b) = \cos a \cos b - \sin a \sin b$ and $\sin(a+b) = \sin a \cos b + \cos a \sin b$.]

- *26. Use mathematical induction to show that $\sum_{j=1}^n \cos jx = \cos[(n+1)x/2] \sin(nx/2)/\sin(x/2)$ whenever n is a positive integer and $\sin(x/2) \neq 0$.

27. Use mathematical induction to prove that $\sum_{j=1}^n j^{22j} = n^2 2^{n+1} - n 2^{n+2} + 3 \cdot 2^{n+1} - 6$ for every positive integer n .

28. (Requires calculus) Suppose that the sequence $x_1, x_2, \dots, x_n, \dots$ is recursively defined by $x_1 = 0$ and $x_{n+1} = \sqrt{x_n + 6}$.

- a) Use mathematical induction to show that $x_1 < x_2 < \cdots < x_n < \cdots$, that is, the sequence $\{x_n\}$ is monotonically increasing.

- b) Use mathematical induction to prove that $x_n < 3$ for $n = 1, 2, \dots$

- c) Show that $\lim_{n \rightarrow \infty} x_n = 3$.

29. Show if n is a positive integer with $n \geq 2$, then

$$\sum_{j=2}^n \frac{1}{j^2 - 1} = \frac{(n-1)(3n+2)}{4n(n+1)}.$$

- 30.** Use mathematical induction to prove Theorem 1 in Section 4.2, that is, show if b is an integer, where $b > 1$, and n is a positive integer, then n can be expressed uniquely in the form $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$.
- *31.** A **lattice point** in the plane is a point (x, y) where both x and y are integers. Use mathematical induction to show that at least $n + 1$ straight lines are needed to ensure that every lattice point (x, y) with $x \geq 0$, $y \geq 0$, and $x + y \leq n$ lies on one of these lines.
- 32.** (*Requires calculus*) Use mathematical induction and the product rule to show that if n is a positive integer and $f_1(x), f_2(x), \dots, f_n(x)$, are all differentiable functions, then

$$\begin{aligned} & \frac{(f_1(x)f_2(x) \cdots f_n(x))'}{f_1(x)f_2(x) \cdots f_n(x)} \\ &= \frac{f'_1(x)}{f_1(x)} + \frac{f'_2(x)}{f_2(x)} + \cdots + \frac{f'_n(x)}{f_n(x)}. \end{aligned}$$

- 33.** (*Requires material in Section 2.6*) Suppose that $\mathbf{B} = \mathbf{M}\mathbf{A}\mathbf{M}^{-1}$, where \mathbf{A} and \mathbf{B} are $n \times n$ matrices and \mathbf{M} is invertible. Show that $\mathbf{B}^k = \mathbf{M}\mathbf{A}^k\mathbf{M}^{-1}$ for all positive integers k . (Consult both the text of Section 2.6 and the preamble to Exercise 18 of Section 2.6.)
- 34.** Use mathematical induction to show that if you draw lines in the plane you only need two colors to color the regions formed so that no two regions that have an edge in common have a common color.
- 35.** Show that $n!$ can be represented as the sum of n of its distinct positive divisors whenever $n \geq 3$. [Hint: Use inductive loading. First try to prove this result using mathematical induction. By examining where your proof fails, find a stronger statement that you can easily prove using mathematical induction.]
- *36.** Use mathematical induction to prove that if x_1, x_2, \dots, x_n are positive real numbers with $n \geq 2$, then
- $$\begin{aligned} & \left(x_1 + \frac{1}{x_1} \right) \left(x_2 + \frac{1}{x_2} \right) \cdots \left(x_n + \frac{1}{x_n} \right) \geq \\ & \left(x_1 + \frac{1}{x_2} \right) \left(x_2 + \frac{1}{x_3} \right) \cdots \left(x_{n-1} + \frac{1}{x_n} \right) \left(x_n + \frac{1}{x_1} \right) \end{aligned}$$
- 37.** Use mathematical induction to prove that if n people stand in a line, where n is a positive integer, and if the first person in the line is a woman and the last person in line is a man, then somewhere in the line there is a woman directly in front of a man.
- *38.** Suppose that for every pair of cities in a country there is a direct one-way road connecting them in one direction or the other. Use mathematical induction to show that there is a city that can be reached from every other city either directly or via exactly one other city.
- 39.** Use mathematical induction to show that when n circles divide the plane into regions, these regions can be colored with two different colors such that no regions with a common boundary are colored the same.
- *40.** Suppose that among a group of cars on a circular track there is enough fuel for one car to complete a lap. Use mathematical induction to show that there is a car in the

group that can complete a lap by obtaining gas from other cars as it travels around the track.

- 41.** Show that if n is a positive integer, then

$$\sum_{j=1}^n (2j-1) \left(\sum_{k=j}^n \frac{1}{k} \right) = n(n+1)/2.$$

- 42.** Use mathematical induction to show that if a , b , and c are the lengths of the sides of a right triangle, where c is the length of the hypotenuse, then $a^n + b^n < c^n$ for all integers n with $n \geq 3$.
- *43.** Use mathematical induction to show that if n is a positive integers, the sequence $2 \bmod n, 2^2 \bmod n, 2^{2^2} \bmod n, 2^{2^{2^2}} \bmod n, \dots$ is eventually constant (that is, all terms after a finite number of terms are all the same).

- 44.** A **unit** or **Egyptian fraction** is a fraction of the form $1/n$, where n is a positive integer. In this exercise, we will use strong induction to show that a greedy algorithm can be used to express every rational number p/q with $0 < p/q < 1$ as the sum of distinct unit fractions. At each step of the algorithm, we find the smallest positive integer n such that $1/n$ can be added to the sum without exceeding p/q . For example, to express $5/7$ we first start the sum with $1/2$. Because $5/7 - 1/2 = 3/14$ we add $1/5$ to the sum because 5 is the smallest positive integer k such that $1/k < 3/14$. Because $3/14 - 1/5 = 1/70$, the algorithm terminates, showing that $5/7 = 1/2 + 1/5 + 1/70$. Let $T(p)$ be the statement that this algorithm terminates for all rational numbers p/q with $0 < p/q < 1$. We will prove that the algorithm always terminates by showing that $T(p)$ holds for all positive integers p .

- a) Show that the basis step $T(1)$ holds.
b) Suppose that $T(k)$ holds for positive integers k with $k < p$. That is, assume that the algorithm terminates for all rational numbers k/r , where $1 \leq k < p$. Show that if we start with p/q and the fraction $1/n$ is selected in the first step of the algorithm, then $p/q = p'/q' + 1/n$, where $p' = np - q$ and $q' = nq$. After considering the case where $p/q = 1/n$, use the inductive hypothesis to show that the greedy algorithm terminates when it begins with p'/q' and complete the inductive step.

The **McCarthy 91 function** (defined by John McCarthy, one of the founders of artificial intelligence) is defined using the rule

$$M(n) = \begin{cases} n - 10 & \text{if } n > 100 \\ M(M(n + 11)) & \text{if } n \leq 100 \end{cases}$$

for all positive integers n .

- 45.** By successively using the defining rule for $M(n)$, find

- a) $M(102)$. b) $M(101)$. c) $M(99)$.
d) $M(97)$. e) $M(87)$. f) $M(76)$.

- **46.** Show that the function $M(n)$ is a well-defined function from the set of positive integers to the set of positive integers. [Hint: Prove that $M(n) = 91$ for all positive integers n with $n \leq 101$.]

47. Is this proof that

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \cdots + \frac{1}{(n-1)n} = \frac{3}{2} - \frac{1}{n},$$

whenever n is a positive integer, correct? Justify your answer.

Basis step: The result is true when $n = 1$ because

$$\frac{1}{1 \cdot 2} = \frac{3}{2} - \frac{1}{1}.$$

Inductive step: Assume that the result is true for n . Then

$$\begin{aligned} \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \cdots + \frac{1}{(n-1)n} + \frac{1}{n(n+1)} \\ = \frac{3}{2} - \frac{1}{n} + \left(\frac{1}{n} - \frac{1}{n+1} \right) \\ = \frac{3}{2} - \frac{1}{n+1}. \end{aligned}$$

Hence, the result is true for $n + 1$ if it is true for n . This completes the proof.

48. Suppose that A_1, A_2, \dots, A_n are a collection of sets. Suppose that $R_2 = A_1 \oplus A_2$ and $R_k = R_{k-1} \oplus A_k$ for $k = 3, 4, \dots, n$. Use mathematical induction to prove that $x \in R_n$ if and only if x belongs to an odd number of the sets A_1, A_2, \dots, A_n . (Recall that $S \oplus T$ is the symmetric difference of the sets S and T defined in the preamble to Exercise 32 of Section 2.2.)
- *49. Show that n circles divide the plane into $n^2 - n + 2$ regions if every two circles intersect in exactly two points and no three circles contain a common point.
- *50. Show that n planes divide three-dimensional space into $(n^3 + 5n + 6)/6$ regions if any three of these planes have exactly one point in common and no four contain a common point.
- *51. Use the well-ordering property to show that $\sqrt{2}$ is irrational. [Hint: Assume that $\sqrt{2}$ is rational. Show that the set of positive integers of the form $b\sqrt{2}$ has a least
- element a . Then show that $a\sqrt{2} - a$ is a smaller positive integer of this form.]
52. A set is **well ordered** if every nonempty subset of this set has a least element. Determine whether each of the following sets is well ordered.
- the set of integers
 - the set of integers greater than -100
 - the set of positive rationals
 - the set of positive rationals with denominator less than 100
53. a) Show that if a_1, a_2, \dots, a_n are positive integers, then $\gcd(a_1, a_2, \dots, a_{n-1}, a_n) = \gcd(a_1, a_2, \dots, a_{n-2}, \gcd(a_{n-1}, a_n))$.
- b) Use part (a), together with the Euclidean algorithm, to develop a recursive algorithm for computing the greatest common divisor of a set of n positive integers.
- *54. Describe a recursive algorithm for writing the greatest common divisor of n positive integers as a linear combination of these integers.
55. Find an explicit formula for $f(n)$ if $f(1) = 1$ and $f(n) = f(n-1) + 2n - 1$ for $n \geq 2$. Prove your result using mathematical induction.
- **56. Give a recursive definition of the set of bit strings that contain twice as many 0s as 1s.
57. Let S be the set of bit strings defined recursively by $\lambda \in S$ and $0x \in S, x1 \in S$ if $x \in S$, where λ is the empty string.
- Find all strings in S of length not exceeding five.
 - Give an explicit description of the elements of S .
58. Let S be the set of strings defined recursively by $abc \in S$, $bac \in S$, and $acb \in S$, where a, b , and c are fixed letters; and for all $x \in S$, $abcx \in S$; $abxc \in S$, $axbc \in S$, and $xabc \in S$, where x is a variable representing a string of letters.
- Find all elements of S of length eight or less.
 - Show that every element of S has a length divisible by three.



JOHN McCARTHY (BORN 1927) John McCarthy was born in Boston. He grew up in Boston and in Los Angeles. He studied mathematics as both an undergraduate and a graduate student, receiving his B.S. in 1948 from the California Institute of Technology and his Ph.D. in 1951 from Princeton. After graduating from Princeton, McCarthy held positions at Princeton, Stanford, Dartmouth, and M.I.T. He held a position at Stanford from 1962 until 1994, and is now an emeritus professor there. At Stanford, he was the director of the Artificial Intelligence Laboratory, held a named chair in the School of Engineering, and was a senior fellow in the Hoover Institution.

McCarthy was a pioneer in the study of artificial intelligence, a term he coined in 1955. He worked on problems related to the reasoning and information needs required for intelligent computer behavior. McCarthy was among the first computer scientists to design time-sharing computer systems. He developed LISP, a programming language for computing using symbolic expressions. He played an important role in using logic to verify the correctness of computer programs. McCarthy has also worked on the social implications of computer technology. He is currently working on the problem of how people and computers make conjectures through assumptions that complications are absent from situations. McCarthy is an advocate of the sustainability of human progress and is an optimist about the future of humanity. He has also begun writing science fiction stories. Some of his recent writing explores the possibility that the world is a computer program written by some higher force.

Among the awards McCarthy has won are the Turing Award from the Association for Computing Machinery, the Research Excellence Award of the International Conference on Artificial Intelligence, the Kyoto Prize, and the National Medal of Science.

The set B of all **balanced strings of parentheses** is defined recursively by $\lambda \in B$, where λ is the empty string; $(x) \in B$, $xy \in B$ if $x, y \in B$.

59. Show that $((())$) is a balanced string of parentheses and $(())$ is not a balanced string of parentheses.
60. Find all balanced strings of parentheses with exactly six symbols.
61. Find all balanced strings of parentheses with four or fewer symbols.
62. Use induction to show that if x is a balanced string of parentheses, then the number of left parentheses equals the number of right parentheses in x .

Define the function N on the set of strings of parentheses by

$$\begin{aligned} N(\lambda) &= 0, N(() = 1, N()) = -1, \\ N(uv) &= N(u) + N(v), \end{aligned}$$

where λ is the empty string, and u and v are strings. It can be shown that N is well defined.

63. Find
 - a) $N(())$.
 - b) $N(()))()()$.
 - c) $N(((())$.
 - d) $N()((())())$.
- **64. Show that a string w of parentheses is balanced if and only if $N(w) = 0$ and $N(u) \geq 0$ whenever u is a prefix of w , that is, $w = uv$.
- *65. Give a recursive algorithm for finding all balanced strings of parentheses containing n or fewer symbols.
66. Give a recursive algorithm for finding $\gcd(a, b)$, where a and b are nonnegative integers not both zero, based on these facts: $\gcd(a, b) = \gcd(b, a)$ if $a > b$, $\gcd(0, b) = b$, $\gcd(a, b) = 2\gcd(a/2, b/2)$ if a and b are even, $\gcd(a, b) = \gcd(a/2, b)$ if a is even and b is odd, and $\gcd(a, b) = \gcd(a, b - a)$.
67. Verify the program segment

```
if x > y then
  x := y
```

with respect to the initial assertion \mathbf{T} and the final assertion $x \leq y$.

- *68. Develop a rule of inference for verifying recursive programs and use it to verify the recursive algorithm for computing factorials given as Algorithm 1 in Section 5.4.

Computer Projects

Write programs with these input and output.

- **1. Given a $2^n \times 2^n$ checkerboard with one square missing, construct a tiling of this checkerboard using right triominoes.
- **2. Generate all well-formed formulae for expressions involving the variables x , y , and z and the operators $\{+, *, /, -\}$ with n or fewer symbols.
- **3. Generate all well-formed formulae for propositions with n or fewer symbols where each symbol is \mathbf{T} , \mathbf{F} , one of

69. Devise a recursive algorithm that counts the number of times the integer 0 occurs in a list of integers.

Exercises 70–77 deal with some unusual sequences, informally called **self-generating sequences**, produced by simple recurrence relations or rules. In particular, Exercises 70–75 deal with the sequence $\{a(n)\}$ defined by $a(n) = n - a(a(n - 1))$ for $n \geq 1$ and $a(0) = 0$. (This sequence, as well as those in Exercises 74 and 75, are defined in Douglas Hofstadter's fascinating book *Gödel, Escher, Bach* ([Ho99]).

70. Find the first 10 terms of the sequence $\{a(n)\}$ defined in the preamble to this exercise.
- *71. Prove that this sequence is well defined. That is, show that $a(n)$ is uniquely defined for all nonnegative integers n .
- **72. Prove that $a(n) = \lfloor (n+1)\mu \rfloor$ where $\mu = (-1 + \sqrt{5})/2$. [Hint: First show for all $n > 0$ that $(\mu n - \lfloor \mu n \rfloor) + (\mu^2 n - \lfloor \mu^2 n \rfloor) = 1$. Then show for all real numbers α with $0 \leq \alpha < 1$ and $\alpha \neq 1 - \mu$ that $\lfloor (1+\mu)(1-\alpha) \rfloor + \lfloor \alpha + \mu \rfloor = 1$, considering the cases $0 \leq \alpha < 1 - \mu$ and $1 - \mu < \alpha < 1$ separately.]
- *73. Use the formula from Exercise 72 to show that $a(n) = a(n-1)$ if $\mu n - \lfloor \mu n \rfloor < 1 - \mu$ and $a(n) = a(n-1) + 1$ otherwise.
74. Find the first 10 terms of each of the following self-generating sequences:
 - a) $a(n) = n - a(a(a(n-1)))$ for $n \geq 1$, $a(0) = 0$
 - b) $a(n) = n - a(a(a(a(n-1))))$ for $n \geq 1$, $a(0) = 0$
 - c) $a(n) = a(n - a(n-1)) + a(n - a(n-2))$ for $n \geq 3$, $a(1) = 1$ and $a(2) = 1$
75. Find the first 10 terms of both the sequences $m(n)$ and $f(n)$ defined by the following pair of interwoven recurrence relations: $m(n) = n - f(m(n-1))$, $f(n) = n - m(f(n-1))$ for $n \geq 1$, $f(0) = 1$ and $m(0) = 0$.

Golomb's self-generating sequence is the unique nondecreasing sequence of positive integers a_1, a_2, a_3, \dots that has the property that it contains exactly a_k occurrences of k for each positive integer k .

76. Find the first 20 terms of Golomb's self-generating sequence.
- *77. Show that if $f(n)$ is the largest integer m such that $a_m = n$, where a_m is the m th term of Golomb's self-generating sequence, then $f(n) = \sum_{k=1}^n a_k$ and $f(f(n)) = \sum_{k=1}^n kak$.

the propositional variables p and q , or an operator from $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$.

4. Given a string, find its reversal.
5. Given a real number a and a nonnegative integer n , find a^n using recursion.
6. Given a real number a and a nonnegative integer n , find a^{2^n} using recursion.

- *7. Given a real number a and a nonnegative integer n , find a^n using the binary expansion of n and a recursive algorithm for computing a^{2^k} .
- 8. Given two integers not both zero, find their greatest common divisor using recursion.
- 9. Given a list of integers and an element x , locate x in this list using a recursive implementation of a linear search.
- 10. Given a list of integers and an element x , locate x in this list using a recursive implementation of a binary search.
- 11. Given a nonnegative integer n , find the n th Fibonacci number using iteration.
- 12. Given a nonnegative integer n , find the n th Fibonacci number using recursion.
- 13. Given a positive integer, find the number of partitions of this integer. (See Exercise 47 of Section 5.3.)
- 14. Given positive integers m and n , find $A(m, n)$, the value of Ackermann's function at the pair (m, n) . (See the preamble to Exercise 48 of Section 5.3.)
- 15. Given a list of n integers, sort these integers using the merge sort.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

- 1. What are the largest values of n for which $n!$ has fewer than 100 decimal digits and fewer than 1000 decimal digits?
- 2. Determine which Fibonacci numbers are divisible by 5, which are divisible by 7, and which are divisible by 11. Prove that your conjectures are correct.
- 3. Construct tilings using right triominoes of various 16×16 , 32×32 , and 64×64 checkerboards with one square missing.
- 4. Explore which $m \times n$ checkerboards can be completely covered by right triominoes. Can you make a conjecture that answers this question?
- **5. Implement an algorithm for determining whether a point is in the interior or exterior of a simple polygon.
- **6. Implement an algorithm for triangulating a simple polygon.
- 7. Which values of Ackermann's function are small enough that you are able to compute them?
- 8. Compare either the number of operations or the time needed to compute Fibonacci numbers recursively versus that needed to compute them iteratively.

Writing Projects

Respond to these with essays using outside sources.

- 1. Describe the origins of mathematical induction. Who were the first people to use it and to which problems did they apply it?
- 2. Explain how to prove the Jordan curve theorem for simple polygons and describe an algorithm for determining whether a point is in the interior or exterior of a simple polygon.
- 3. Describe how the triangulation of simple polygons is used in some key algorithms in computational geometry.
- 4. Describe a variety of different applications of the Fibonacci numbers to the biological and the physical sciences.
- 5. Discuss the uses of Ackermann's function both in the theory of recursive definitions and in the analysis of the complexity of algorithms for set unions.
- 6. Discuss some of the various methodologies used to establish the correctness of programs and compare them to Hoare's methods described in Section 5.5.
- 7. Explain how the ideas and concepts of program correctness can be extended to prove that operating systems are secure.

6

Counting

- 6.1** The Basics of Counting
- 6.2** The Pigeonhole Principle
- 6.3** Permutations and Combinations
- 6.4** Binomial Coefficients and Identities
- 6.5** Generalized Permutations and Combinations
- 6.6** Generating Permutations and Combinations

Combinatorics, the study of arrangements of objects, is an important part of discrete mathematics. This subject was studied as long ago as the seventeenth century, when combinatorial questions arose in the study of gambling games. Enumeration, the counting of objects with certain properties, is an important part of combinatorics. We must count objects to solve many different types of problems. For instance, counting is used to determine the complexity of algorithms. Counting is also required to determine whether there are enough telephone numbers or Internet protocol addresses to meet demand. Recently, it has played a key role in mathematical biology, especially in sequencing DNA. Furthermore, counting techniques are used extensively when probabilities of events are computed.

The basic rules of counting, which we will study in Section 6.1, can solve a tremendous variety of problems. For instance, we can use these rules to enumerate the different telephone numbers possible in the United States, the allowable passwords on a computer system, and the different orders in which the runners in a race can finish. Another important combinatorial tool is the pigeonhole principle, which we will study in Section 6.2. This states that when objects are placed in boxes and there are more objects than boxes, then there is a box containing at least two objects. For instance, we can use this principle to show that among a set of 15 or more students, at least 3 were born on the same day of the week.

We can phrase many counting problems in terms of ordered or unordered arrangements of the objects of a set with or without repetitions. These arrangements, called permutations and combinations, are used in many counting problems. For instance, suppose the 100 top finishers on a competitive exam taken by 2000 students are invited to a banquet. We can count the possible sets of 100 students that will be invited, as well as the ways in which the top 10 prizes can be awarded.

Another problem in combinatorics involves generating all the arrangements of a specified kind. This is often important in computer simulations. We will devise algorithms to generate arrangements of various types.

6.1 The Basics of Counting

Introduction

Suppose that a password on a computer system consists of six, seven, or eight characters. Each of these characters must be a digit or a letter of the alphabet. Each password must contain at least one digit. How many such passwords are there? The techniques needed to answer this question and a wide variety of other counting problems will be introduced in this section.

Counting problems arise throughout mathematics and computer science. For example, we must count the successful outcomes of experiments and all the possible outcomes of these experiments to determine probabilities of discrete events. We need to count the number of operations used by an algorithm to study its time complexity.

We will introduce the basic techniques of counting in this section. These methods serve as the foundation for almost all counting techniques.

Basic Counting Principles



We first present two basic counting principles, the **product rule** and the **sum rule**. Then we will show how they can be used to solve many different counting problems.

The product rule applies when a procedure is made up of separate tasks.

THE PRODUCT RULE Suppose that a procedure can be broken down into a sequence of two tasks. If there are n_1 ways to do the first task and for each of these ways of doing the first task, there are n_2 ways to do the second task, then there are $n_1 n_2$ ways to do the procedure.



Examples 1–10 show how the product rule is used.

EXAMPLE 1 A new company with just two employees, Sanchez and Patel, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

Solution: The procedure of assigning offices to these two employees consists of assigning an office to Sanchez, which can be done in 12 ways, then assigning an office to Patel different from the office assigned to Sanchez, which can be done in 11 ways. By the product rule, there are $12 \cdot 11 = 132$ ways to assign offices to these two employees.

EXAMPLE 2 The chairs of an auditorium are to be labeled with an uppercase English letter followed by a positive integer not exceeding 100. What is the largest number of chairs that can be labeled differently?

Solution: The procedure of labeling a chair consists of two tasks, namely, assigning to the seat one of the 26 uppercase English letters, and then assigning to it one of the 100 possible integers. The product rule shows that there are $26 \cdot 100 = 2600$ different ways that a chair can be labeled. Therefore, the largest number of chairs that can be labeled differently is 2600.

EXAMPLE 3 There are 32 microcomputers in a computer center. Each microcomputer has 24 ports. How many different ports to a microcomputer in the center are there?

Solution: The procedure of choosing a port consists of two tasks, first picking a microcomputer and then picking a port on this microcomputer. Because there are 32 ways to choose the microcomputer and 24 ways to choose the port no matter which microcomputer has been selected, the product rule shows that there are $32 \cdot 24 = 768$ ports.

An extended version of the product rule is often useful. Suppose that a procedure is carried out by performing the tasks T_1, T_2, \dots, T_m in sequence. If each task $T_i, i = 1, 2, \dots, n$, can be done in n_i ways, regardless of how the previous tasks were done, then there are $n_1 \cdot n_2 \cdot \dots \cdot n_m$ ways to carry out the procedure. This version of the product rule can be proved by mathematical induction from the product rule for two tasks (see Exercise 72).

EXAMPLE 4 How many different bit strings of length seven are there?

Solution: Each of the seven bits can be chosen in two ways, because each bit is either 0 or 1. Therefore, the product rule shows there are a total of $2^7 = 128$ different bit strings of length seven.

EXAMPLE 5

How many different license plates can be made if each plate contains a sequence of three uppercase English letters followed by three digits (and no sequences of letters are prohibited, even if they are obscene)?

 26 choices for each letter 10 choices for each digit

Solution: There are 26 choices for each of the three uppercase English letters and ten choices for each of the three digits. Hence, by the product rule there are a total of $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17,576,000$ possible license plates. 

EXAMPLE 6

Counting Functions How many functions are there from a set with m elements to a set with n elements?

Solution: A function corresponds to a choice of one of the n elements in the codomain for each of the m elements in the domain. Hence, by the product rule there are $n \cdot n \cdot \dots \cdot n = n^m$ functions from a set with m elements to one with n elements. For example, there are $5^3 = 125$ different functions from a set with three elements to a set with five elements. 

EXAMPLE 7

Counting One-to-One Functions How many one-to-one functions are there from a set with m elements to one with n elements?

Counting the number of onto functions is harder. We'll do this in Chapter 8.

Solution: First note that when $m > n$ there are no one-to-one functions from a set with m elements to a set with n elements.

Now let $m \leq n$. Suppose the elements in the domain are a_1, a_2, \dots, a_m . There are n ways to choose the value of the function at a_1 . Because the function is one-to-one, the value of the function at a_2 can be picked in $n - 1$ ways (because the value used for a_1 cannot be used again). In general, the value of the function at a_k can be chosen in $n - k + 1$ ways. By the product rule, there are $n(n - 1)(n - 2) \cdots (n - m + 1)$ one-to-one functions from a set with m elements to one with n elements.

For example, there are $5 \cdot 4 \cdot 3 = 60$ one-to-one functions from a set with three elements to a set with five elements. 

EXAMPLE 8

The Telephone Numbering Plan The *North American numbering plan (NANP)* specifies the format of telephone numbers in the U.S., Canada, and many other parts of North America. A telephone number in this plan consists of 10 digits, which are split into a three-digit area code, a three-digit office code, and a four-digit station code. Because of signaling considerations, there are certain restrictions on some of these digits. To specify the allowable format, let X denote a digit that can take any of the values 0 through 9, let N denote a digit that can take any of the values 2 through 9, and let Y denote a digit that must be a 0 or a 1. Two numbering plans, which will be called the old plan, and the new plan, will be discussed. (The old plan, in use in the 1960s, has been replaced by the new plan, but the recent rapid growth in demand for new numbers for mobile phones and devices will eventually make even this new plan obsolete. In this example, the letters used to represent digits follow the conventions of the *North American Numbering Plan*.) As will be shown, the new plan allows the use of more numbers.



Current projections are that by 2038, it will be necessary to add one or more digits to North American telephone numbers.

In the old plan, the formats of the area code, office code, and station code are NYX , NNX , and $XXXX$, respectively, so that telephone numbers had the form $NYX-NNX-XXXX$. In the new plan, the formats of these codes are NXX , NXX , and $XXXX$, respectively, so that telephone numbers have the form $NXX-NXX-XXXX$. How many different North American telephone numbers are possible under the old plan and under the new plan?

Solution: By the product rule, there are $8 \cdot 2 \cdot 10 = 160$ area codes with format NYX and $8 \cdot 10 \cdot 10 = 800$ area codes with format NXX . Similarly, by the product rule, there are $8 \cdot 8 \cdot 10 = 640$ office codes with format NNX . The product rule also shows that there are $10 \cdot 10 \cdot 10 \cdot 10 = 10,000$ station codes with format $XXXX$.

Note that we have ignored restrictions that rule out N11 station codes for most area codes.

Consequently, applying the product rule again, it follows that under the old plan there are

$$160 \cdot 640 \cdot 10,000 = 1,024,000,000$$

different numbers available in North America. Under the new plan, there are

$$800 \cdot 800 \cdot 10,000 = 6,400,000,000$$

different numbers available. 

EXAMPLE 9 What is the value of k after the following code, where n_1, n_2, \dots, n_m are positive integers, has been executed?

```

k := 0
for i1 := 1 to n1
    for i2 := 1 to n2
        .
        .
        .
    for im := 1 to nm
        k := k + 1
    
```

Solution: The initial value of k is zero. Each time the nested loop is traversed, 1 is added to k . Let T_i be the task of traversing the i th loop. Then the number of times the loop is traversed is the number of ways to do the tasks T_1, T_2, \dots, T_m . The number of ways to carry out the task T_j , $j = 1, 2, \dots, m$, is n_j , because the j th loop is traversed once for each integer i_j with $1 \leq i_j \leq n_j$. By the product rule, it follows that the nested loop is traversed $n_1 n_2 \cdots n_m$ times. Hence, the final value of k is $n_1 n_2 \cdots n_m$. 

EXAMPLE 10 **Counting Subsets of a Finite Set** Use the product rule to show that the number of different subsets of a finite set S is $2^{|S|}$.

Solution: Let S be a finite set. List the elements of S in arbitrary order. Recall from Section 2.2 that there is a one-to-one correspondence between subsets of S and bit strings of length $|S|$. Namely, a subset of S is associated with the bit string with a 1 in the i th position if the i th element in the list is in the subset, and a 0 in this position otherwise. By the product rule, there are $2^{|S|}$ bit strings of length $|S|$. Hence, $|P(S)| = 2^{|S|}$. (Recall that we used mathematical induction to prove this fact in Example 10 of Section 5.1.) 

The product rule is often phrased in terms of sets in this way: If A_1, A_2, \dots, A_m are finite sets, then the number of elements in the Cartesian product of these sets is the product of the number of elements in each set. To relate this to the product rule, note that the task of choosing an element in the Cartesian product $A_1 \times A_2 \times \cdots \times A_m$ is done by choosing an element in A_1 , an element in A_2, \dots , and an element in A_m . By the product rule it follows that

$$|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \cdots \cdot |A_m|.$$

EXAMPLE 11 **DNA and Genomes** The hereditary information of a living organism is encoded using deoxyribonucleic acid (DNA), or in certain viruses, ribonucleic acid (RNA). DNA and RNA are extremely complex molecules, with different molecules interacting in a vast variety of ways to

enable living process. For our purposes, we give only the briefest description of how DNA and RNA encode genetic information.

DNA molecules consist of two strands consisting of blocks known as nucleotides. Each nucleotide contains subcomponents called **bases**, each of which is adenine (A), cytosine (C), guanine (G), or thymine (T). The two strands of DNA are held together by hydrogen bonds connecting different bases, with A bonding only with T, and C bonding only with G. Unlike DNA, RNA is single stranded, with uracil (U) replacing thymine as a base. So, in DNA the possible base pairs are A-T and C-G, while in RNA they are A-U, and C-G. The DNA of a living creature consists of multiple pieces of DNA forming separate chromosomes. A **gene** is a segment of a DNA molecule that encodes a particular protein. The entirety of genetic information of an organism is called its **genome**.

Sequences of bases in DNA and RNA encode long chains of proteins called amino acids. There are 22 essential amino acids for human beings. We can quickly see that a sequence of at least three bases are needed to encode these 22 different amino acid. First note, that because there are four possibilities for each base in DNA, A, C, G, and T, by the product rule there are $4^2 = 16 < 22$ different sequences of two bases. However, there are $4^3 = 64$ different sequences of three bases, which provide enough different sequences to encode the 22 different amino acids (even after taking into account that several different sequences of three bases encode the same amino acid).

The DNA of simple living creatures such as algae and bacteria have between 10^5 and 10^7 links, where each link is one of the four possible bases. More complex organisms, such as insects, birds, and mammals have between 10^8 and 10^{10} links in their DNA. So, by the product rule, there are at least 4^{10^5} different sequences of bases in the DNA of simple organisms and at least 4^{10^8} different sequences of bases in the DNA of more complex organisms. These are both incredibly huge numbers, which helps explain why there is such tremendous variability among living organisms. In the past several decades techniques have been developed for determining the genome of different organisms. The first step is to locate each gene in the DNA of an organism. The next task, called **gene sequencing**, is the determination of the sequence of links on each gene. (Of course, the specific sequence of kinks on these genes depends on the particular individual representative of a species whose DNA is analyzed.) For example, the human genome includes approximately 23,000 genes, each with 1,000 or more links. Gene sequencing techniques take advantage of many recently developed algorithms and are based on numerous new ideas in combinatorics. Many mathematicians and computer scientists work on problems involving genomes, taking part in the fast moving fields of bioinformatics and computational biology. 

Soon it won't be that costly to have your own genetic code found.

We now introduce the sum rule.

THE SUM RULE If a task can be done either in one of n_1 ways or in one of n_2 ways, where none of the set of n_1 ways is the same as any of the set of n_2 ways, then there are $n_1 + n_2$ ways to do the task.

Example 12 illustrates how the sum rule is used.

EXAMPLE 12

Suppose that either a member of the mathematics faculty or a student who is a mathematics major is chosen as a representative to a university committee. How many different choices are there for this representative if there are 37 members of the mathematics faculty and 83 mathematics majors and no one is both a faculty member and a student?

Solution: There are 37 ways to choose a member of the mathematics faculty and there are 83 ways to choose a student who is a mathematics major. Choosing a member of the mathematics faculty is never the same as choosing a student who is a mathematics major because no one is

both a faculty member and a student. By the sum rule it follows that there are $37 + 83 = 120$ possible ways to pick this representative. 

We can extend the sum rule to more than two tasks. Suppose that a task can be done in one of n_1 ways, in one of n_2 ways, \dots , or in one of n_m ways, where none of the set of n_i ways of doing the task is the same as any of the set of n_j ways, for all pairs i and j with $1 \leq i < j \leq m$. Then the number of ways to do the task is $n_1 + n_2 + \dots + n_m$. This extended version of the sum rule is often useful in counting problems, as Examples 13 and 14 show. This version of the sum rule can be proved using mathematical induction from the sum rule for two sets. (This is Exercise 71.)

EXAMPLE 13 A student can choose a computer project from one of three lists. The three lists contain 23, 15, and 19 possible projects, respectively. No project is on more than one list. How many possible projects are there to choose from?

Solution: The student can choose a project by selecting a project from the first list, the second list, or the third list. Because no project is on more than one list, by the sum rule there are $23 + 15 + 19 = 57$ ways to choose a project. 

EXAMPLE 14 What is the value of k after the following code, where n_1, n_2, \dots, n_m are positive integers, has been executed?

```

k := 0
for i1 := 1 to n1
    k := k + 1
for i2 := 1 to n2
    k := k + 1
.
.
.
for im := 1 to nm
    k := k + 1

```

Solution: The initial value of k is zero. This block of code is made up of m different loops. Each time a loop is traversed, 1 is added to k . To determine the value of k after this code has been executed, we need to determine how many times we traverse a loop. Note that there are n_i ways to traverse the i th loop. Because we only traverse one loop at a time, the sum rule shows that the final value of k , which is the number of ways to traverse one of the m loops is $n_1 + n_2 + \dots + n_m$. 

The sum rule can be phrased in terms of sets as: If A_1, A_2, \dots, A_m are pairwise disjoint finite sets, then the number of elements in the union of these sets is the sum of the numbers of elements in the sets. To relate this to our statement of the sum rule, note there are $|A_i|$ ways to choose an element from A_i for $i = 1, 2, \dots, m$. Because the sets are pairwise disjoint, when we select an element from one of the sets A_i , we do not also select an element from a different set A_j . Consequently, by the sum rule, because we cannot select an element from two of these sets at the same time, the number of ways to choose an element from one of the sets, which is the number of elements in the union, is

$$|A_1 \cup A_2 \cup \dots \cup A_m| = |A_1| + |A_2| + \dots + |A_m| \text{ when } A_i \cap A_j = \emptyset \text{ for all } i, j.$$

This equality applies only when the sets in question are pairwise disjoint. The situation is much more complicated when these sets have elements in common. That situation will be briefly discussed later in this section and discussed in more depth in Chapter 8.

More Complex Counting Problems

Many counting problems cannot be solved using just the sum rule or just the product rule. However, many complicated counting problems can be solved using both of these rules in combination. We begin by counting the number of variable names in the programming language BASIC. (In the exercises, we consider the number of variable names in JAVA.) Then we will count the number of valid passwords subject to a particular set of restrictions.

EXAMPLE 15



In a version of the computer language BASIC, the name of a variable is a string of one or two alphanumeric characters, where uppercase and lowercase letters are not distinguished. (An *alphanumeric* character is either one of the 26 English letters or one of the 10 digits.) Moreover, a variable name must begin with a letter and must be different from the five strings of two characters that are reserved for programming use. How many different variable names are there in this version of BASIC?

Solution: Let V equal the number of different variable names in this version of BASIC. Let V_1 be the number of these that are one character long and V_2 be the number of these that are two characters long. Then by the sum rule, $V = V_1 + V_2$. Note that $V_1 = 26$, because a one-character variable name must be a letter. Furthermore, by the product rule there are $26 \cdot 36$ strings of length two that begin with a letter and end with an alphanumeric character. However, five of these are excluded, so $V_2 = 26 \cdot 36 - 5 = 931$. Hence, there are $V = V_1 + V_2 = 26 + 931 = 957$ different names for variables in this version of BASIC. 

EXAMPLE 16

Each user on a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?

Solution: Let P be the total number of possible passwords, and let P_6 , P_7 , and P_8 denote the number of possible passwords of length 6, 7, and 8, respectively. By the sum rule, $P = P_6 + P_7 + P_8$. We will now find P_6 , P_7 , and P_8 . Finding P_6 directly is difficult. To find P_6 it is easier to find the number of strings of uppercase letters and digits that are six characters long, including those with no digits, and subtract from this the number of strings with no digits. By the product rule, the number of strings of six characters is 36^6 , and the number of strings with no digits is 26^6 . Hence,

$$P_6 = 36^6 - 26^6 = 2,176,782,336 - 308,915,776 = 1,867,866,560.$$

Similarly, we have

$$P_7 = 36^7 - 26^7 = 78,364,164,096 - 8,031,810,176 = 70,332,353,920$$

and

$$\begin{aligned} P_8 &= 36^8 - 26^8 = 2,821,109,907,456 - 208,827,064,576 \\ &= 2,612,282,842,880. \end{aligned}$$

Consequently,

$$P = P_6 + P_7 + P_8 = 2,684,483,063,360. \quad \blacktriangleleft$$

EXAMPLE 17



Counting Internet Addresses In the Internet, which is made up of interconnected physical networks of computers, each computer (or more precisely, each network connection of a computer) is assigned an *Internet address*. In Version 4 of the Internet Protocol (IPv4), now in use,

Bit Number	0	1	2	3	4	8	16	24	31	
Class A	0	netid				hostid				
Class B	1	0	netid				hostid			
Class C	1	1	0	netid					hostid	
Class D	1	1	1	0	Multicast Address					
Class E	1	1	1	1	0	Address				

FIGURE 1 Internet Addresses (IPv4).

an address is a string of 32 bits. It begins with a *network number (netid)*. The netid is followed by a *host number (hostid)*, which identifies a computer as a member of a particular network.

Three forms of addresses are used, with different numbers of bits used for netids and hostids. **Class A addresses**, used for the largest networks, consist of 0, followed by a 7-bit netid and a 24-bit hostid. **Class B addresses**, used for medium-sized networks, consist of 10, followed by a 14-bit netid and a 16-bit hostid. **Class C addresses**, used for the smallest networks, consist of 110, followed by a 21-bit netid and an 8-bit hostid. There are several restrictions on addresses because of special uses: 1111111 is not available as the netid of a Class A network, and the hostids consisting of all 0s and all 1s are not available for use in any network. A computer on the Internet has either a Class A, a Class B, or a Class C address. (Besides Class A, B, and C addresses, there are also Class D addresses, reserved for use in multicasting when multiple computers are addressed at a single time, consisting of 1110 followed by 28 bits, and Class E addresses, reserved for future use, consisting of 11110 followed by 27 bits. Neither Class D nor Class E addresses are assigned as the IPv4 address of a computer on the Internet.) Figure 1 illustrates IPv4 addressing. (Limitations on the number of Class A and Class B netids have made IPv4 addressing inadequate; IPv6, a new version of IP, uses 128-bit addresses to solve this problem.)

How many different IPv4 addresses are available for computers on the Internet?

Solution: Let x be the number of available addresses for computers on the Internet, and let x_A , x_B , and x_C denote the number of Class A, Class B, and Class C addresses available, respectively. By the sum rule, $x = x_A + x_B + x_C$.

To find x_A , note that there are $2^7 - 1 = 127$ Class A netids, recalling that the netid 1111111 is unavailable. For each netid, there are $2^{24} - 2 = 16,777,214$ hostids, recalling that the hostids consisting of all 0s and all 1s are unavailable. Consequently, $x_A = 127 \cdot 16,777,214 = 2,130,706,178$.

To find x_B and x_C , note that there are $2^{14} = 16,384$ Class B netids and $2^{21} = 2,097,152$ Class C netids. For each Class B netid, there are $2^{16} - 2 = 65,534$ hostids, and for each Class C netid, there are $2^8 - 2 = 254$ hostids, recalling that in each network the hostids consisting of all 0s and all 1s are unavailable. Consequently, $x_B = 1,073,709,056$ and $x_C = 532,676,608$.

We conclude that the total number of IPv4 addresses available is $x = x_A + x_B + x_C = 2,130,706,178 + 1,073,709,056 + 532,676,608 = 3,737,091,842$. 

The Subtraction Rule (Inclusion–Exclusion for Two Sets)

Suppose that a task can be done in one of two ways, but some of the ways to do it are common to both ways. In this situation, we cannot use the sum rule to count the number of ways to do the task. If we add the number of ways to do the tasks in these two ways, we get an overcount of the total number of ways to do it, because the ways to do the task that are common to the two ways are counted twice. To correctly count the number of ways to do the two tasks, we must subtract the number of ways that are counted twice. This leads us to an important counting rule.

The lack of available
IPv4 address has
become a crisis!

Overcounting is perhaps
the most common
enumeration error.

THE SUBTRACTION RULE If a task can be done in either n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common to the two different ways.

The subtraction rule is also known as the **principle of inclusion–exclusion**, especially when it is used to count the number of elements in the union of two sets. Suppose that A_1 and A_2 are sets. Then, there are $|A_1|$ ways to select an element from A_1 and $|A_2|$ ways to select an element from A_2 . The number of ways to select an element from A_1 or from A_2 , that is, the number of ways to select an element from their union, is the sum of the number of ways to select an element from A_1 and the number of ways to select an element from A_2 , minus the number of ways to select an element that is in both A_1 and A_2 . Because there are $|A_1 \cup A_2|$ ways to select an element in either A_1 or in A_2 , and $|A_1 \cap A_2|$ ways to select an element common to both sets, we have

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

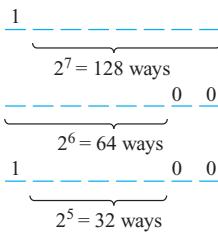
This is the formula given in Section 2.2 for the number of elements in the union of two sets.

Example 18 illustrates how we can solve counting problems using the subtraction principle.

EXAMPLE 18 How many bit strings of length eight either start with a 1 bit or end with the two bits 00?



Solution: We can construct a bit string of length eight that either starts with a 1 bit or ends with the two bits 00, by constructing a bit string of length eight beginning with a 1 bit or by constructing a bit string of length eight that ends with the two bits 00. We can construct a bit string of length eight that begins with a 1 in $2^7 = 128$ ways. This follows by the product rule, because the first bit can be chosen in only one way and each of the other seven bits can be chosen in two ways. Similarly, we can construct a bit string of length eight ending with the two bits 00, in $2^6 = 64$ ways. This follows by the product rule, because each of the first six bits can be chosen in two ways and the last two bits can be chosen in only one way.



Some of the ways to construct a bit string of length eight starting with a 1 are the same as the ways to construct a bit string of length eight that ends with the two bits 00. There are $2^5 = 32$ ways to construct such a string. This follows by the product rule, because the first bit can be chosen in only one way, each of the second through the sixth bits can be chosen in two ways, and the last two bits can be chosen in one way. Consequently, the number of bit strings of length eight that begin with a 1 or end with a 00, which equals the number of ways to construct a bit string of length eight that begins with a 1 or that ends with 00, equals $128 + 64 - 32 = 160$.

We present an example that illustrates how the formulation of the principle of inclusion–exclusion can be used to solve counting problems.

EXAMPLE 19

A computer company receives 350 applications from computer graduates for a job planning a line of new Web servers. Suppose that 220 of these applicants majored in computer science, 147 majored in business, and 51 majored both in computer science and in business. How many of these applicants majored neither in computer science nor in business?

Solution: To find the number of these applicants who majored neither in computer science nor in business, we can subtract the number of students who majored either in computer science or in business (or both) from the total number of applicants. Let A_1 be the set of students who majored in computer science and A_2 the set of students who majored in business. Then $A_1 \cup A_2$ is the set of students who majored in computer science or business (or both), and $A_1 \cap A_2$ is the

set of students who majored both in computer science and in business. By the subtraction rule the number of students who majored either in computer science or in business (or both) equals

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2| = 220 + 147 - 51 = 316.$$

We conclude that $350 - 316 = 34$ of the applicants majored neither in computer science nor in business. ◀

The subtraction rule, or the principle of inclusion–exclusion, can be generalized to find the number of ways to do one of n different tasks or, equivalently, to find the number of elements in the union of n sets, whenever n is a positive integer. We will study the inclusion–exclusion principle and some of its many applications in Chapter 8.

The Division Rule

We have introduced the product, sum, and subtraction rules for counting. You may wonder whether there is also a division rule for counting. In fact, there is such a rule, which can be useful when solving certain types of enumeration problems.

THE DIVISION RULE There are n/d ways to do a task if it can be done using a procedure that can be carried out in n ways, and for every way w , exactly d of the n ways correspond to way w .

We can restate the division rule in terms of sets: “If the finite set A is the union of n pairwise disjoint subsets each with d elements, then $n = |A|/d$. ”

We can also formulate the division rule in terms of functions: “If f is a function from A to B where A and B are finite sets, and that for every value $y \in B$ there are exactly d values $x \in A$ such that $f(x) = y$ (in which case, we say that f is d -to-one), then $|B| = |A|/d$. ”

We illustrate the use of the division rule for counting with an example.

EXAMPLE 20

How many different ways are there to seat four people around a circular table, where two seatings are considered the same when each person has the same left neighbor and the same right neighbor?

Solution: We arbitrarily select a seat at the table and label it seat 1. We number the rest of the seats in numerical order, proceeding clockwise around the table. Note that are four ways to select the person for seat 1, three ways to select the person for seat 2, two ways to select the person for seat 3, and one way to select the person for seat 4. Thus, there are $4! = 24$ ways to order the given four people for these seats. However, each of the four choices for seat 1 leads to the same arrangement, as we distinguish two arrangements only when one of the people has a different immediate left or immediate right neighbor. Because there are four ways to choose the person for seat 1, by the division rule there are $24/4 = 6$ different seating arrangements of four people around the circular table. ◀

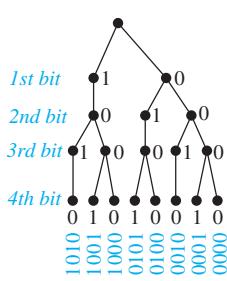


FIGURE 2 Bit Strings of Length Four without Consecutive 1s.

Tree Diagrams

Counting problems can be solved using **tree diagrams**. A tree consists of a root, a number of branches leaving the root, and possible additional branches leaving the endpoints of other branches. (We will study trees in detail in Chapter 11.) To use trees in counting, we use a branch to represent each possible choice. We represent the possible outcomes by the leaves, which are the endpoints of branches not having other branches starting at them.

Note that when a tree diagram is used to solve a counting problem, the number of choices of which branch to follow to reach a leaf can vary (see Example 21, for example).

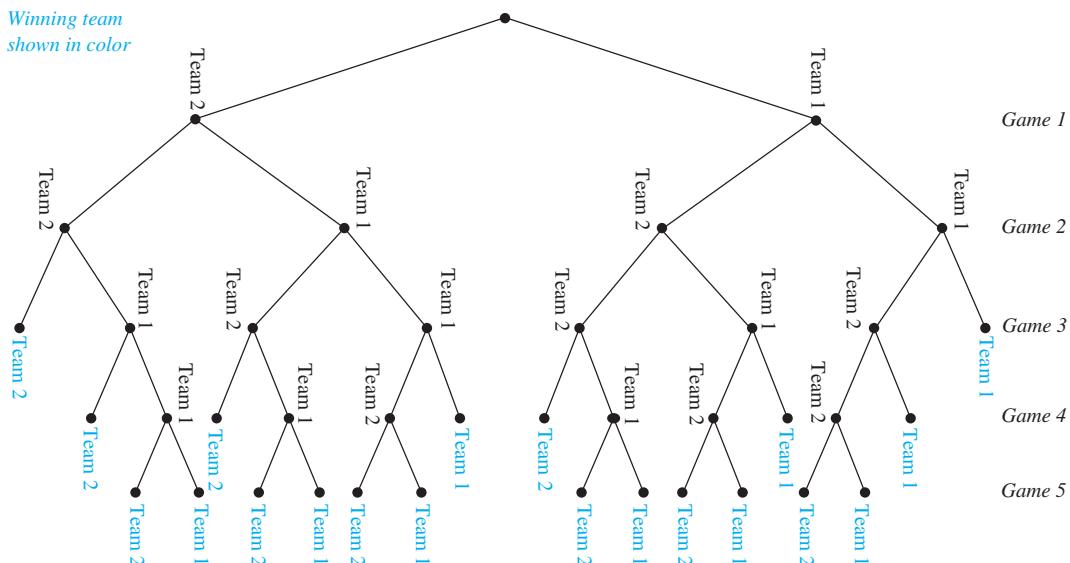


FIGURE 3 Best Three Games Out of Five Playoffs.

EXAMPLE 21 How many bit strings of length four do not have two consecutive 1s?

Solution: The tree diagram in Figure 2 displays all bit strings of length four without two consecutive 1s. We see that there are eight bit strings of length four without two consecutive 1s. \blacktriangleleft

EXAMPLE 22 A playoff between two teams consists of at most five games. The first team that wins three games wins the playoff. In how many different ways can the playoff occur?

Solution: The tree diagram in Figure 3 displays all the ways the playoff can proceed, with the winner of each game shown. We see that there are 20 different ways for the playoff to occur. \blacktriangleleft

EXAMPLE 23 Suppose that “I Love New Jersey” T-shirts come in five different sizes: S, M, L, XL, and XXL. Further suppose that each size comes in four colors, white, red, green, and black, except for XL, which comes only in red, green, and black, and XXL, which comes only in green and black. How many different shirts does a souvenir shop have to stock to have at least one of each available size and color of the T-shirt?

Solution: The tree diagram in Figure 4 displays all possible size and color pairs. It follows that the souvenir shop owner needs to stock 17 different T-shirts. \blacktriangleleft

W = white, R = red, G = green, B = black

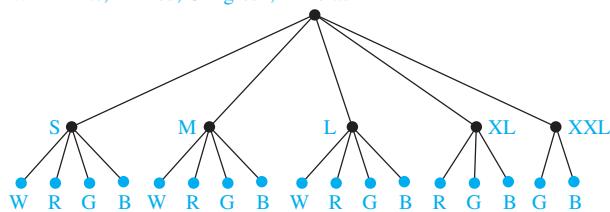


FIGURE 4 Counting Varieties of T-Shirts.

Exercises

1. There are 18 mathematics majors and 325 computer science majors at a college.
 - a) In how many ways can two representatives be picked so that one is a mathematics major and the other is a computer science major?
 - b) In how many ways can one representative be picked who is either a mathematics major or a computer science major?
2. An office building contains 27 floors and has 37 offices on each floor. How many offices are in the building?
3. A multiple-choice test contains 10 questions. There are four possible answers for each question.
 - a) In how many ways can a student answer the questions on the test if the student answers every question?
 - b) In how many ways can a student answer the questions on the test if the student can leave answers blank?
4. A particular brand of shirt comes in 12 colors, has a male version and a female version, and comes in three sizes for each sex. How many different types of this shirt are made?
5. Six different airlines fly from New York to Denver and seven fly from Denver to San Francisco. How many different pairs of airlines can you choose on which to book a trip from New York to San Francisco via Denver, when you pick an airline for the flight to Denver and an airline for the continuation flight to San Francisco?
6. There are four major auto routes from Boston to Detroit and six from Detroit to Los Angeles. How many major auto routes are there from Boston to Los Angeles via Detroit?
7. How many different three-letter initials can people have?
8. How many different three-letter initials with none of the letters repeated can people have?
9. How many different three-letter initials are there that begin with an *A*?
10. How many bit strings are there of length eight?
11. How many bit strings of length ten both begin and end with a 1?
12. How many bit strings are there of length six or less, not counting the empty string?
13. How many bit strings with length not exceeding n , where n is a positive integer, consist entirely of 1s, not counting the empty string?
14. How many bit strings of length n , where n is a positive integer, start and end with 1s?
15. How many strings are there of lowercase letters of length four or less, not counting the empty string?
16. How many strings are there of four lowercase letters that have the letter *x* in them?
17. How many strings of five ASCII characters contain the character @ ("at" sign) at least once? [Note: There are 128 different ASCII characters.]
18. How many 5-element DNA sequences
 - a) end with A?
 - b) start with T and end with G?
 - c) contain only A and T?
 - d) do not contain C?
19. How many 6-element RNA sequences
 - a) do not contain U?
 - b) end with GU?
 - c) start with C?
 - d) contain only A or U?
20. How many positive integers between 5 and 31
 - a) are divisible by 3? Which integers are these?
 - b) are divisible by 4? Which integers are these?
 - c) are divisible by 3 and by 4? Which integers are these?
21. How many positive integers between 50 and 100
 - a) are divisible by 7? Which integers are these?
 - b) are divisible by 11? Which integers are these?
 - c) are divisible by both 7 and 11? Which integers are these?
22. How many positive integers less than 1000
 - a) are divisible by 7?
 - b) are divisible by 7 but not by 11?
 - c) are divisible by both 7 and 11?
 - d) are divisible by either 7 or 11?
 - e) are divisible by exactly one of 7 and 11?
 - f) are divisible by neither 7 nor 11?
 - g) have distinct digits?
 - h) have distinct digits and are even?
23. How many positive integers between 100 and 999 inclusive
 - a) are divisible by 7?
 - b) are odd?
 - c) have the same three decimal digits?
 - d) are not divisible by 4?
 - e) are divisible by 3 or 4?
 - f) are not divisible by either 3 or 4?
 - g) are divisible by 3 but not by 4?
 - h) are divisible by 3 and 4?
24. How many positive integers between 1000 and 9999 inclusive
 - a) are divisible by 9?
 - b) are even?
 - c) have distinct digits?
 - d) are not divisible by 3?
 - e) are divisible by 5 or 7?
 - f) are not divisible by either 5 or 7?
 - g) are divisible by 5 but not by 7?
 - h) are divisible by 5 and 7?

- 25.** How many strings of three decimal digits
- do not contain the same digit three times?
 - begin with an odd digit?
 - have exactly two digits that are 4s?
- 26.** How many strings of four decimal digits
- do not contain the same digit twice?
 - end with an even digit?
 - have exactly three digits that are 9s?
- 27.** A committee is formed consisting of one representative from each of the 50 states in the United States, where the representative from a state is either the governor or one of the two senators from that state. How many ways are there to form this committee?
- 28.** How many license plates can be made using either three digits followed by three uppercase English letters or three uppercase English letters followed by three digits?
- 29.** How many license plates can be made using either two uppercase English letters followed by four digits or two digits followed by four uppercase English letters?
- 30.** How many license plates can be made using either three uppercase English letters followed by three digits or four uppercase English letters followed by two digits?
- 31.** How many license plates can be made using either two or three uppercase English letters followed by either two or three digits?
- 32.** How many strings of eight uppercase English letters are there
- if letters can be repeated?
 - if no letter can be repeated?
 - that start with X, if letters can be repeated?
 - that start with X, if no letter can be repeated?
 - that start and end with X, if letters can be repeated?
 - that start with the letters BO (in that order), if letters can be repeated?
 - that start and end with the letters BO (in that order), if letters can be repeated?
 - that start or end with the letters BO (in that order), if letters can be repeated?
- 33.** How many strings of eight English letters are there
- that contain no vowels, if letters can be repeated?
 - that contain no vowels, if letters cannot be repeated?
 - that start with a vowel, if letters can be repeated?
 - that start with a vowel, if letters cannot be repeated?
 - that contain at least one vowel, if letters can be repeated?
 - that contain exactly one vowel, if letters can be repeated?
 - that start with X and contain at least one vowel, if letters can be repeated?
 - that start and end with X and contain at least one vowel, if letters can be repeated?
- 34.** How many different functions are there from a set with 10 elements to sets with the following numbers of elements?
- 2
 - 3
 - 4
 - 5
- 35.** How many one-to-one functions are there from a set with five elements to sets with the following number of elements?
- 4
 - 5
 - 6
 - 7
- 36.** How many functions are there from the set $\{1, 2, \dots, n\}$, where n is a positive integer, to the set $\{0, 1\}$?
- 37.** How many functions are there from the set $\{1, 2, \dots, n\}$, where n is a positive integer, to the set $\{0, 1\}$
- that are one-to-one?
 - that assign 0 to both 1 and n ?
 - that assign 1 to exactly one of the positive integers less than n ?
- 38.** How many partial functions (see Section 2.3) are there from a set with five elements to sets with each of these number of elements?
- 1
 - 2
 - 5
 - 9
- 39.** How many partial functions (see Definition 13 of Section 2.3) are there from a set with m elements to a set with n elements, where m and n are positive integers?
- 40.** How many subsets of a set with 100 elements have more than one element?
- 41.** A **palindrome** is a string whose reversal is identical to the string. How many bit strings of length n are palindromes?
- 42.** How many 4-element DNA sequences
- do not contain the base T?
 - contain the sequence ACG?
 - contain all four bases A, T, C, and G?
 - contain exactly three of the four bases A, T, C, and G?
- 43.** How many 4-element RNA sequences
- contain the base U?
 - do not contain the sequence CUG?
 - do not contain all four bases A, U, C, and G?
 - contain exactly two of the four bases A, U, C, and G?
- 44.** How many ways are there to seat four of a group of ten people around a circular table where two seatings are considered the same when everyone has the same immediate left and immediate right neighbor?
- 45.** How many ways are there to seat six people around a circular table where two seatings are considered the same when everyone has the same two neighbors without regard to whether they are right or left neighbors?
- 46.** In how many ways can a photographer at a wedding arrange 6 people in a row from a group of 10 people, where the bride and the groom are among these 10 people, if
- the bride must be in the picture?
 - both the bride and groom must be in the picture?
 - exactly one of the bride and the groom is in the picture?
- 47.** In how many ways can a photographer at a wedding arrange six people in a row, including the bride and groom, if
- the bride must be next to the groom?
 - the bride is not next to the groom?
 - the bride is positioned somewhere to the left of the groom?

- 48.** How many bit strings of length seven either begin with two 0s or end with three 1s?
- 49.** How many bit strings of length 10 either begin with three 0s or end with two 0s?
- *50.** How many bit strings of length 10 contain either five consecutive 0s or five consecutive 1s?
- **51.** How many bit strings of length eight contain either three consecutive 0s or four consecutive 1s?
- 52.** Every student in a discrete mathematics class is either a computer science or a mathematics major or is a joint major in these two subjects. How many students are in the class if there are 38 computer science majors (including joint majors), 23 mathematics majors (including joint majors), and 7 joint majors?
- 53.** How many positive integers not exceeding 100 are divisible either by 4 or by 6?
- 54.** How many different initials can someone have if a person has at least two, but no more than five, different initials? Assume that each initial is one of the 26 uppercase letters of the English language.
- 55.** Suppose that a password for a computer system must have at least 8, but no more than 12, characters, where each character in the password is a lowercase English letter, an uppercase English letter, a digit, or one of the six special characters *, >, <, !, +, and =.
- How many different passwords are available for this computer system?
 - How many of these passwords contain at least one occurrence of at least one of the six special characters?
 - Using your answer to part (a), determine how long it takes a hacker to try every possible password, assuming that it takes one nanosecond for a hacker to check each possible password.
- 56.** The name of a variable in the C programming language is a string that can contain uppercase letters, lowercase letters, digits, or underscores. Further, the first character in the string must be a letter, either uppercase or lowercase, or an underscore. If the name of a variable is determined by its first eight characters, how many different variables can be named in C? (Note that the name of a variable may contain fewer than eight characters.)
- 57.** The name of a variable in the JAVA programming language is a string of between 1 and 65,535 characters, inclusive, where each character can be an uppercase or a lowercase letter, a dollar sign, an underscore, or a digit, except that the first character must not be a digit. Determine the number of different variable names in JAVA.
- 58.** The International Telecommunications Union (ITU) specifies that a telephone number must consist of a country code with between 1 and 3 digits, except that the code 0 is not available for use as a country code, followed by a number with at most 15 digits. How many available possible telephone numbers are there that satisfy these restrictions?
- 59.** Suppose that at some future time every telephone in the world is assigned a number that contains a country code 1 to 3 digits long, that is, of the form X , XX , or XXX , followed by a 10-digit telephone number of the form $XXX-XXX-XXXX$ (as described in Example 8). How many different telephone numbers would be available worldwide under this numbering plan?
- 60.** A key in the Vigenère cryptosystem is a string of English letters, where the case of the letters does not matter. How many different keys for this cryptosystem are there with three, four, five, or six letters?
- 61.** A wired equivalent privacy (WEP) key for a wireless fidelity (WiFi) network is a string of either 10, 26, or 58 hexadecimal digits. How many different WEP keys are there?
- 62.** Suppose that p and q are prime numbers and that $n = pq$. Use the principle of inclusion–exclusion to find the number of positive integers not exceeding n that are relatively prime to n .
- 63.** Use the principle of inclusion–exclusion to find the number of positive integers less than 1,000,000 that are not divisible by either 4 or by 6.
- 64.** Use a tree diagram to find the number of bit strings of length four with no three consecutive 0s.
- 65.** How many ways are there to arrange the letters a , b , c , and d such that a is not followed immediately by b ?
- 66.** Use a tree diagram to find the number of ways that the World Series can occur, where the first team that wins four games out of seven wins the series.
- 67.** Use a tree diagram to determine the number of subsets of $\{3, 7, 9, 11, 24\}$ with the property that the sum of the elements in the subset is less than 28.
- 68.**
 - Suppose that a store sells six varieties of soft drinks: cola, ginger ale, orange, root beer, lemonade, and cream soda. Use a tree diagram to determine the number of different types of bottles the store must stock to have all varieties available in all size bottles if all varieties are available in 12-ounce bottles, all but lemonade are available in 20-ounce bottles, only cola and ginger ale are available in 32-ounce bottles, and all but lemonade and cream soda are available in 64-ounce bottles?
 - Answer the question in part (a) using counting rules.
- 69.**
 - Suppose that a popular style of running shoe is available for both men and women. The woman's shoe comes in sizes 6, 7, 8, and 9, and the man's shoe comes in sizes 8, 9, 10, 11, and 12. The man's shoe comes in white and black, while the woman's shoe comes in white, red, and black. Use a tree diagram to determine the number of different shoes that a store has to stock to have at least one pair of this type of running shoe for all available sizes and colors for both men and women.
 - Answer the question in part (a) using counting rules.
- *70.** Use the product rule to show that there are 2^{2^n} different truth tables for propositions in n variables.

71. Use mathematical induction to prove the sum rule for m tasks from the sum rule for two tasks.
72. Use mathematical induction to prove the product rule for m tasks from the product rule for two tasks.
73. How many diagonals does a convex polygon with n sides have? (Recall that a polygon is convex if every line segment connecting two points in the interior or boundary of the polygon lies entirely within this set and that a diagonal of a polygon is a line segment connecting two vertices that are not adjacent.)
74. Data are transmitted over the Internet in **datagrams**, which are structured blocks of bits. Each datagram contains header information organized into a maximum of 14 different fields (specifying many things, including the source and destination addresses) and a data area that contains the actual data that are transmitted. One of the 14 header fields is the **header length field** (denoted by HLEN), which is specified by the protocol to be 4 bits long and that specifies the header length in terms of 32-bit blocks of bits. For example, if HLEN = 0110, the header

is made up of six 32-bit blocks. Another of the 14 header fields is the 16-bit-long **total length field** (denoted by TOTAL LENGTH), which specifies the length in bits of the entire datagram, including both the header fields and the data area. The length of the data area is the total length of the datagram minus the length of the header.

- a) The largest possible value of TOTAL LENGTH (which is 16 bits long) determines the maximum total length in octets (blocks of 8 bits) of an Internet datagram. What is this value?
- b) The largest possible value of HLEN (which is 4 bits long) determines the maximum total header length in 32-bit blocks. What is this value? What is the maximum total header length in octets?
- c) The minimum (and most common) header length is 20 octets. What is the maximum total length in octets of the data area of an Internet datagram?
- d) How many different strings of octets in the data area can be transmitted if the header length is 20 octets and the total length is as long as possible?

6.2 The Pigeonhole Principle

Introduction



Suppose that a flock of 20 pigeons flies into a set of 19 pigeonholes to roost. Because there are 20 pigeons but only 19 pigeonholes, at least one of these 19 pigeonholes must have at least two pigeons in it. To see why this is true, note that if each pigeonhole had at most one pigeon in it, at most 19 pigeons, one per hole, could be accommodated. This illustrates a general principle called the **pigeonhole principle**, which states that if there are more pigeons than pigeonholes, then there must be at least one pigeonhole with at least two pigeons in it (see Figure 1). Of course, this principle applies to other objects besides pigeons and pigeonholes.

THEOREM 1

THE PIGEONHOLE PRINCIPLE If k is a positive integer and $k + 1$ or more objects are placed into k boxes, then there is at least one box containing two or more of the objects.

(a)

(b)

(c)

FIGURE 1 There Are More Pigeons Than Pigeonholes.

Proof: We prove the pigeonhole principle using a proof by contraposition. Suppose that none of the k boxes contains more than one object. Then the total number of objects would be at most k . This is a contradiction, because there are at least $k + 1$ objects. \triangleleft

The pigeonhole principle is also called the **Dirichlet drawer principle**, after the nineteenth-century German mathematician G. Lejeune Dirichlet, who often used this principle in his work. (Dirichlet was not the first person to use this principle; a demonstration that there were at least two Parisians with the same number of hairs on their heads dates back to the 17th century—see Exercise 33.) It is an important additional proof technique supplementing those we have developed in earlier chapters. We introduce it in this chapter because of its many important applications to combinatorics.

We will illustrate the usefulness of the pigeonhole principle. We first show that it can be used to prove a useful corollary about functions.

COROLLARY 1

A function f from a set with $k + 1$ or more elements to a set with k elements is not one-to-one.

Proof: Suppose that for each element y in the codomain of f we have a box that contains all elements x of the domain of f such that $f(x) = y$. Because the domain contains $k + 1$ or more elements and the codomain contains only k elements, the pigeonhole principle tells us that one of these boxes contains two or more elements x of the domain. This means that f cannot be one-to-one. \triangleleft

Examples 1–3 show how the pigeonhole principle is used.

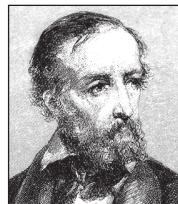
EXAMPLE 1 Among any group of 367 people, there must be at least two with the same birthday, because there are only 366 possible birthdays. \triangleleft

EXAMPLE 2 In any group of 27 English words, there must be at least two that begin with the same letter, because there are 26 letters in the English alphabet. \triangleleft

EXAMPLE 3 How many students must be in a class to guarantee that at least two students receive the same score on the final exam, if the exam is graded on a scale from 0 to 100 points?

Solution: There are 101 possible scores on the final. The pigeonhole principle shows that among any 102 students there must be at least 2 students with the same score. \triangleleft

Links



G. LEJEUNE DIRICHLET (1805–1859) G. Lejeune Dirichlet was born into a Belgian family living near Cologne, Germany. His father was a postmaster. He became passionate about mathematics at a young age. He was spending all his spare money on mathematics books by the time he entered secondary school in Bonn at the age of 12. At 14 he entered the Jesuit College in Cologne, and at 16 he began his studies at the University of Paris. In 1825 he returned to Germany and was appointed to a position at the University of Breslau. In 1828 he moved to the University of Berlin. In 1855 he was chosen to succeed Gauss at the University of Göttingen. Dirichlet is said to be the first person to master Gauss's *Disquisitiones Arithmeticae*, which appeared 20 years earlier. He is said to have kept a copy at his side even when he traveled. Dirichlet made many important discoveries in number theory, including the theorem that there are infinitely many primes in arithmetical progressions $an + b$ when a and b are relatively prime. He proved the $n = 5$ case of Fermat's last theorem, that there are no nontrivial solutions in integers to $x^5 + y^5 = z^5$. Dirichlet also made many contributions to analysis. Dirichlet was considered to be an excellent teacher who could explain ideas with great clarity. He was married to Rebecca Mendelssohn, one of the sisters of the composer Frederick Mendelssohn.

The pigeonhole principle is a useful tool in many proofs, including proofs of surprising results, such as that given in Example 4.

EXAMPLE 4



Show that for every integer n there is a multiple of n that has only 0s and 1s in its decimal expansion.

Solution: Let n be a positive integer. Consider the $n + 1$ integers 1, 11, 111, ..., 11...1 (where the last integer in this list is the integer with $n + 1$ 1s in its decimal expansion). Note that there are n possible remainders when an integer is divided by n . Because there are $n + 1$ integers in this list, by the pigeonhole principle there must be two with the same remainder when divided by n . The larger of these integers less the smaller one is a multiple of n , which has a decimal expansion consisting entirely of 0s and 1s. \blacktriangleleft

The Generalized Pigeonhole Principle

The pigeonhole principle states that there must be at least two objects in the same box when there are more objects than boxes. However, even more can be said when the number of objects exceeds a multiple of the number of boxes. For instance, among any set of 21 decimal digits there must be 3 that are the same. This follows because when 21 objects are distributed into 10 boxes, one box must have more than 2 objects.

THEOREM 2

THE GENERALIZED PIGEONHOLE PRINCIPLE If N objects are placed into k boxes, then there is at least one box containing at least $\lceil N/k \rceil$ objects.

Proof: We will use a proof by contraposition. Suppose that none of the boxes contains more than $\lceil N/k \rceil - 1$ objects. Then, the total number of objects is at most

$$k \left(\left\lceil \frac{N}{k} \right\rceil - 1 \right) < k \left(\left(\frac{N}{k} + 1 \right) - 1 \right) = N,$$

where the inequality $\lceil N/k \rceil < (N/k) + 1$ has been used. This is a contradiction because there are a total of N objects. \blacktriangleleft

A common type of problem asks for the minimum number of objects such that at least r of these objects must be in one of k boxes when these objects are distributed among the boxes. When we have N objects, the generalized pigeonhole principle tells us there must be at least r objects in one of the boxes as long as $\lceil N/k \rceil \geq r$. The smallest integer N with $N/k > r - 1$, namely, $N = k(r - 1) + 1$, is the smallest integer satisfying the inequality $\lceil N/k \rceil \geq r$. Could a smaller value of N suffice? The answer is no, because if we had $k(r - 1)$ objects, we could put $r - 1$ of them in each of the k boxes and no box would have at least r objects.

When thinking about problems of this type, it is useful to consider how you can avoid having at least r objects in one of the boxes as you add successive objects. To avoid adding a r th object to any box, you eventually end up with $r - 1$ objects in each box. There is no way to add the next object without putting an r th object in that box.

Examples 5–8 illustrate how the generalized pigeonhole principle is applied.

EXAMPLE 5

Among 100 people there are at least $\lceil 100/12 \rceil = 9$ who were born in the same month. \blacktriangleleft

EXAMPLE 6

What is the minimum number of students required in a discrete mathematics class to be sure that at least six will receive the same grade, if there are five possible grades, A, B, C, D, and F?

Solution: The minimum number of students needed to ensure that at least six students receive the same grade is the smallest integer N such that $\lceil N/5 \rceil = 6$. The smallest such integer is $N = 5 \cdot 5 + 1 = 26$. If you have only 25 students, it is possible for there to be five who have received each grade so that no six students have received the same grade. Thus, 26 is the minimum number of students needed to ensure that at least six students will receive the same grade. \blacktriangleleft

EXAMPLE 7

- How many cards must be selected from a standard deck of 52 cards to guarantee that at least three cards of the same suit are chosen?
- How many must be selected to guarantee that at least three hearts are selected?

A standard deck of 52 cards has 13 kinds of cards, with four cards of each kind, one in each of the four suits, hearts, diamonds, spades, and clubs.

Solution: a) Suppose there are four boxes, one for each suit, and as cards are selected they are placed in the box reserved for cards of that suit. Using the generalized pigeonhole principle, we see that if N cards are selected, there is at least one box containing at least $\lceil N/4 \rceil$ cards. Consequently, we know that at least three cards of one suit are selected if $\lceil N/4 \rceil \geq 3$. The smallest integer N such that $\lceil N/4 \rceil \geq 3$ is $N = 2 \cdot 4 + 1 = 9$, so nine cards suffice. Note that if eight cards are selected, it is possible to have two cards of each suit, so more than eight cards are needed. Consequently, nine cards must be selected to guarantee that at least three cards of one suit are chosen. One good way to think about this is to note that after the eighth card is chosen, there is no way to avoid having a third card of some suit.

b) We do not use the generalized pigeonhole principle to answer this question, because we want to make sure that there are three hearts, not just three cards of one suit. Note that in the worst case, we can select all the clubs, diamonds, and spades, 39 cards in all, before we select a single heart. The next three cards will be all hearts, so we may need to select 42 cards to get three hearts. \blacktriangleleft

EXAMPLE 8

What is the least number of area codes needed to guarantee that the 25 million phones in a state can be assigned distinct 10-digit telephone numbers? (Assume that telephone numbers are of the form $XXX-XXX-XXXX$, where the first three digits form the area code, N represents a digit from 2 to 9 inclusive, and X represents any digit.)

Solution: There are eight million different phone numbers of the form $XXX-XXXX$ (as shown in Example 8 of Section 6.1). Hence, by the generalized pigeonhole principle, among 25 million telephones, at least $\lceil 25,000,000/8,000,000 \rceil = 4$ of them must have identical phone numbers. Hence, at least four area codes are required to ensure that all 10-digit numbers are different. \blacktriangleleft

Example 9, although not an application of the generalized pigeonhole principle, makes use of similar principles.

EXAMPLE 9

Suppose that a computer science laboratory has 15 workstations and 10 servers. A cable can be used to directly connect a workstation to a server. For each server, only one direct connection to that server can be active at any time. We want to guarantee that at any time any set of 10 or fewer workstations can simultaneously access different servers via direct connections. Although we could do this by connecting every workstation directly to every server (using 150 connections), what is the minimum number of direct connections needed to achieve this goal?

Solution: Suppose that we label the workstations W_1, W_2, \dots, W_{15} and the servers S_1, S_2, \dots, S_{10} . Furthermore, suppose that we connect W_k to S_k for $k = 1, 2, \dots, 10$ and each of $W_{11}, W_{12}, W_{13}, W_{14}$, and W_{15} to all 10 servers. We have a total of 60 direct connections. Clearly any set of 10 or fewer workstations can simultaneously access different servers. We see this by noting that if workstation W_j is included with $1 \leq j \leq 10$, it can access server S_j , and for each workstation W_k with $k \geq 11$ included, there must be a corresponding workstation W_j .

with $1 \leq j \leq 10$ not included, so W_k can access server S_j . (This follows because there are at least as many available servers S_j as there are workstations W_j with $1 \leq j \leq 10$ not included.)

Now suppose there are fewer than 60 direct connections between workstations and servers. Then some server would be connected to at most $\lfloor 59/10 \rfloor = 5$ workstations. (If all servers were connected to at least six workstations, there would be at least $6 \cdot 10 = 60$ direct connections.) This means that the remaining nine servers are not enough to allow the other 10 workstations to simultaneously access different servers. Consequently, at least 60 direct connections are needed. It follows that 60 is the answer. 

Some Elegant Applications of the Pigeonhole Principle

In many interesting applications of the pigeonhole principle, the objects to be placed in boxes must be chosen in a clever way. A few such applications will be described here.

EXAMPLE 10 During a month with 30 days, a baseball team plays at least one game a day, but no more than 45 games. Show that there must be a period of some number of consecutive days during which the team must play exactly 14 games.

Solution: Let a_j be the number of games played on or before the j th day of the month. Then a_1, a_2, \dots, a_{30} is an increasing sequence of distinct positive integers, with $1 \leq a_j \leq 45$. Moreover, $a_1 + 14, a_2 + 14, \dots, a_{30} + 14$ is also an increasing sequence of distinct positive integers, with $15 \leq a_j + 14 \leq 59$.

The 60 positive integers $a_1, a_2, \dots, a_{30}, a_1 + 14, a_2 + 14, \dots, a_{30} + 14$ are all less than or equal to 59. Hence, by the pigeonhole principle two of these integers are equal. Because the integers a_j , $j = 1, 2, \dots, 30$ are all distinct and the integers $a_j + 14$, $j = 1, 2, \dots, 30$ are all distinct, there must be indices i and j with $a_i = a_j + 14$. This means that exactly 14 games were played from day $j + 1$ to day i . 

EXAMPLE 11 Show that among any $n + 1$ positive integers not exceeding $2n$ there must be an integer that divides one of the other integers.

Solution: Write each of the $n + 1$ integers a_1, a_2, \dots, a_{n+1} as a power of 2 times an odd integer. In other words, let $a_j = 2^{k_j} q_j$ for $j = 1, 2, \dots, n + 1$, where k_j is a nonnegative integer and q_j is odd. The integers q_1, q_2, \dots, q_{n+1} are all odd positive integers less than $2n$. Because there are only n odd positive integers less than $2n$, it follows from the pigeonhole principle that two of the integers q_1, q_2, \dots, q_{n+1} must be equal. Therefore, there are distinct integers i and j such that $q_i = q_j$. Let q be the common value of q_i and q_j . Then, $a_i = 2^{k_i} q$ and $a_j = 2^{k_j} q$. It follows that if $k_i < k_j$, then a_i divides a_j ; while if $k_i > k_j$, then a_j divides a_i . 

A clever application of the pigeonhole principle shows the existence of an increasing or a decreasing subsequence of a certain length in a sequence of distinct integers. We review some definitions before this application is presented. Suppose that a_1, a_2, \dots, a_N is a sequence of real numbers. A **subsequence** of this sequence is a sequence of the form $a_{i_1}, a_{i_2}, \dots, a_{i_m}$, where $1 \leq i_1 < i_2 < \dots < i_m \leq N$. Hence, a subsequence is a sequence obtained from the original sequence by including some of the terms of the original sequence in their original order, and perhaps not including other terms. A sequence is called **strictly increasing** if each term is larger than the one that precedes it, and it is called **strictly decreasing** if each term is smaller than the one that precedes it.

THEOREM 3

Every sequence of $n^2 + 1$ distinct real numbers contains a subsequence of length $n + 1$ that is either strictly increasing or strictly decreasing.

We give an example before presenting the proof of Theorem 3.

EXAMPLE 12

The sequence 8, 11, 9, 1, 4, 6, 12, 10, 5, 7 contains 10 terms. Note that $10 = 3^2 + 1$. There are four strictly increasing subsequences of length four, namely, 1, 4, 6, 12; 1, 4, 6, 7; 1, 4, 6, 10; and 1, 4, 5, 7. There is also a strictly decreasing subsequence of length four, namely, 11, 9, 6, 5. 

The proof of the theorem will now be given.

Proof: Let $a_1, a_2, \dots, a_{n^2+1}$ be a sequence of $n^2 + 1$ distinct real numbers. Associate an ordered pair with each term of the sequence, namely, associate (i_k, d_k) to the term a_k , where i_k is the length of the longest increasing subsequence starting at a_k , and d_k is the length of the longest decreasing subsequence starting at a_k .



Suppose that there are no increasing or decreasing subsequences of length $n + 1$. Then i_k and d_k are both positive integers less than or equal to n , for $k = 1, 2, \dots, n^2 + 1$. Hence, by the product rule there are n^2 possible ordered pairs for (i_k, d_k) . By the pigeonhole principle, two of these $n^2 + 1$ ordered pairs are equal. In other words, there exist terms a_s and a_t , with $s < t$ such that $i_s = i_t$ and $d_s = d_t$. We will show that this is impossible. Because the terms of the sequence are distinct, either $a_s < a_t$ or $a_s > a_t$. If $a_s < a_t$, then, because $i_s = i_t$, an increasing subsequence of length $i_t + 1$ can be built starting at a_s , by taking a_s followed by an increasing subsequence of length i_t beginning at a_t . This is a contradiction. Similarly, if $a_s > a_t$, the same reasoning shows that d_s must be greater than d_t , which is a contradiction. 



The final example shows how the generalized pigeonhole principle can be applied to an important part of combinatorics called **Ramsey theory**, after the English mathematician F. P. Ramsey. In general, Ramsey theory deals with the distribution of subsets of elements of sets.

EXAMPLE 13

Assume that in a group of six people, each pair of individuals consists of two friends or two enemies. Show that there are either three mutual friends or three mutual enemies in the group.

Solution: Let A be one of the six people. Of the five other people in the group, there are either three or more who are friends of A , or three or more who are enemies of A . This follows from the generalized pigeonhole principle, because when five objects are divided into two sets, one of the sets has at least $\lceil 5/2 \rceil = 3$ elements. In the former case, suppose that B , C , and D are friends of A . If any two of these three individuals are friends, then these two and A form a group of three mutual friends. Otherwise, B , C , and D form a set of three mutual enemies. The proof in the latter case, when there are three or more enemies of A , proceeds in a similar manner. 

The **Ramsey number** $R(m, n)$, where m and n are positive integers greater than or equal to 2, denotes the minimum number of people at a party such that there are either m mutual friends or n mutual enemies, assuming that every pair of people at the party are friends or enemies. Example 13 shows that $R(3, 3) \leq 6$. We conclude that $R(3, 3) = 6$ because in a group of five



FRANK PLUMPTON RAMSEY (1903–1930) Frank Plumpton Ramsey, son of the president of Magdalene College, Cambridge, was educated at Winchester and Trinity Colleges. After graduating in 1923, he was elected a fellow of King's College, Cambridge, where he spent the remainder of his life. Ramsey made important contributions to mathematical logic. What we now call Ramsey theory began with his clever combinatorial arguments, published in the paper "On a Problem of Formal Logic." Ramsey also made contributions to the mathematical theory of economics. He was noted as an excellent lecturer on the foundations of mathematics. According to one of his brothers, he was interested in almost everything, including English literature and politics. Ramsey was married and had two daughters. His death at the age of 26 resulting from chronic liver problems deprived the mathematical community and Cambridge University of a brilliant young scholar.

people where every two people are friends or enemies, there may not be three mutual friends or three mutual enemies (see Exercise 26).

It is possible to prove some useful properties about Ramsey numbers, but for the most part it is difficult to find their exact values. Note that by symmetry it can be shown that $R(m, n) = R(n, m)$ (see Exercise 30). We also have $R(2, n) = n$ for every positive integer $n \geq 2$ (see Exercise 29). The exact values of only nine Ramsey numbers $R(m, n)$ with $3 \leq m \leq n$ are known, including $R(4, 4) = 18$. Only bounds are known for many other Ramsey numbers, including $R(5, 5)$, which is known to satisfy $43 \leq R(5, 5) \leq 49$. The reader interested in learning more about Ramsey numbers should consult [MiRo91] or [GrRoSp90].

Exercises

1. Show that in any set of six classes, each meeting regularly once a week on a particular day of the week, there must be two that meet on the same day, assuming that no classes are held on weekends.
2. Show that if there are 30 students in a class, then at least two have last names that begin with the same letter.
3. A drawer contains a dozen brown socks and a dozen black socks, all unmatched. A man takes socks out at random in the dark.
 - a) How many socks must he take out to be sure that he has at least two socks of the same color?
 - b) How many socks must he take out to be sure that he has at least two black socks?
4. A bowl contains 10 red balls and 10 blue balls. A woman selects balls at random without looking at them.
 - a) How many balls must she select to be sure of having at least three balls of the same color?
 - b) How many balls must she select to be sure of having at least three blue balls?
5. Show that among any group of five (not necessarily consecutive) integers, there are two with the same remainder when divided by 4.
6. Let d be a positive integer. Show that among any group of $d + 1$ (not necessarily consecutive) integers there are two with exactly the same remainder when they are divided by d .
7. Let n be a positive integer. Show that in any set of n consecutive integers there is exactly one divisible by n .
8. Show that if f is a function from S to T , where S and T are finite sets with $|S| > |T|$, then there are elements s_1 and s_2 in S such that $f(s_1) = f(s_2)$, or in other words, f is not one-to-one.
9. What is the minimum number of students, each of whom comes from one of the 50 states, who must be enrolled in a university to guarantee that there are at least 100 who come from the same state?
- *10. Let $(x_i, y_i), i = 1, 2, 3, 4, 5$, be a set of five distinct points with integer coordinates in the xy plane. Show that the midpoint of the line joining at least one pair of these points has integer coordinates.
- *11. Let $(x_i, y_i, z_i), i = 1, 2, 3, 4, 5, 6, 7, 8, 9$, be a set of nine distinct points with integer coordinates in xyz space. Show that the midpoint of at least one pair of these points has integer coordinates.
12. How many ordered pairs of integers (a, b) are needed to guarantee that there are two ordered pairs (a_1, b_1) and (a_2, b_2) such that $a_1 \bmod 5 = a_2 \bmod 5$ and $b_1 \bmod 5 = b_2 \bmod 5$?
13. a) Show that if five integers are selected from the first eight positive integers, there must be a pair of these integers with a sum equal to 9.
b) Is the conclusion in part (a) true if four integers are selected rather than five?
14. a) Show that if seven integers are selected from the first 10 positive integers, there must be at least two pairs of these integers with the sum 11.
b) Is the conclusion in part (a) true if six integers are selected rather than seven?
15. How many numbers must be selected from the set $\{1, 2, 3, 4, 5, 6\}$ to guarantee that at least one pair of these numbers add up to 7?
16. How many numbers must be selected from the set $\{1, 3, 5, 7, 9, 11, 13, 15\}$ to guarantee that at least one pair of these numbers add up to 16?
17. A company stores products in a warehouse. Storage bins in this warehouse are specified by their aisle, location in the aisle, and shelf. There are 50 aisles, 85 horizontal locations in each aisle, and 5 shelves throughout the warehouse. What is the least number of products the company can have so that at least two products must be stored in the same bin?
18. Suppose that there are nine students in a discrete mathematics class at a small college.
 - a) Show that the class must have at least five male students or at least five female students.
 - b) Show that the class must have at least three male students or at least seven female students.
19. Suppose that every student in a discrete mathematics class of 25 students is a freshman, a sophomore, or a junior.
 - a) Show that there are at least nine freshmen, at least nine sophomores, or at least nine juniors in the class.

- b)** Show that there are either at least three freshmen, at least 19 sophomores, or at least five juniors in the class.
- 20.** Find an increasing subsequence of maximal length and a decreasing subsequence of maximal length in the sequence 22, 5, 7, 2, 23, 10, 15, 21, 3, 17.
- 21.** Construct a sequence of 16 positive integers that has no increasing or decreasing subsequence of five terms.
- 22.** Show that if there are 101 people of different heights standing in a line, it is possible to find 11 people in the order they are standing in the line with heights that are either increasing or decreasing.
- *23.** Show that whenever 25 girls and 25 boys are seated around a circular table there is always a person both of whose neighbors are boys.
- **24.** Suppose that 21 girls and 21 boys enter a mathematics competition. Furthermore, suppose that each entrant solves at most six questions, and for every boy-girl pair, there is at least one question that they both solved. Show that there is a question that was solved by at least three girls and at least three boys.
- *25.** Describe an algorithm in pseudocode for producing the largest increasing or decreasing subsequence of a sequence of distinct integers.
- 26.** Show that in a group of five people (where any two people are either friends or enemies), there are not necessarily three mutual friends or three mutual enemies.
- 27.** Show that in a group of 10 people (where any two people are either friends or enemies), there are either three mutual friends or four mutual enemies, and there are either three mutual enemies or four mutual friends.
- 28.** Use Exercise 27 to show that among any group of 20 people (where any two people are either friends or enemies), there are either four mutual friends or four mutual enemies.
- 29.** Show that if n is an integer with $n \geq 2$, then the Ramsey number $R(2, n)$ equals n . (Recall that Ramsey numbers were discussed after Example 13 in Section 6.2.)
- 30.** Show that if m and n are integers with $m \geq 2$ and $n \geq 2$, then the Ramsey numbers $R(m, n)$ and $R(n, m)$ are equal. (Recall that Ramsey numbers were discussed after Example 13 in Section 6.2.)
- 31.** Show that there are at least six people in California (population: 37 million) with the same three initials who were born on the same day of the year (but not necessarily in the same year). Assume that everyone has three initials.
- 32.** Show that if there are 100,000,000 wage earners in the United States who earn less than 1,000,000 dollars (but at least a penny), then there are two who earned exactly the same amount of money, to the penny, last year.
- 33.** In the 17th century, there were more than 800,000 inhabitants of Paris. At the time, it was believed that no one had more than 200,000 hairs on their head. Assuming these numbers are correct and that everyone has at least one hair on their head (that is, no one is completely bald), use the pigeonhole principle to show, as the French writer Pierre

Nicole did, that there had to be two Parisians with the same number of hairs on their heads. Then use the generalized pigeonhole principle to show that there had to be at least five Parisians at that time with the same number of hairs on their heads.

- 34.** Assuming that no one has more than 1,000,000 hairs on the head of any person and that the population of New York City was 8,008,278 in 2010, show there had to be at least nine people in New York City in 2010 with the same number of hairs on their heads.
- 35.** There are 38 different time periods during which classes at a university can be scheduled. If there are 677 different classes, how many different rooms will be needed?
- 36.** A computer network consists of six computers. Each computer is directly connected to at least one of the other computers. Show that there are at least two computers in the network that are directly connected to the same number of other computers.
- 37.** A computer network consists of six computers. Each computer is directly connected to zero or more of the other computers. Show that there are at least two computers in the network that are directly connected to the same number of other computers. [*Hint:* It is impossible to have a computer linked to none of the others and a computer linked to all the others.]
- 38.** Find the least number of cables required to connect eight computers to four printers to guarantee that for every choice of four of the eight computers, these four computers can directly access four different printers. Justify your answer.
- 39.** Find the least number of cables required to connect 100 computers to 20 printers to guarantee that every subset of 20 computers can directly access 20 different printers. (Here, the assumptions about cables and computers are the same as in Example 9.) Justify your answer.
- *40.** Prove that at a party where there are at least two people, there are two people who know the same number of other people there.
- 41.** An arm wrestler is the champion for a period of 75 hours. (Here, by an hour, we mean a period starting from an exact hour, such as 1 P.M., until the next hour.) The arm wrestler had at least one match an hour, but no more than 125 total matches. Show that there is a period of consecutive hours during which the arm wrestler had exactly 24 matches.
- *42.** Is the statement in Exercise 41 true if 24 is replaced by
a) 2? **b)** 23? **c)** 25? **d)** 30?
- 43.** Show that if f is a function from S to T , where S and T are nonempty finite sets and $m = \lceil |S| / |T| \rceil$, then there are at least m elements of S mapped to the same value of T . That is, show that there are distinct elements s_1, s_2, \dots, s_m of S such that $f(s_1) = f(s_2) = \dots = f(s_m)$.
- 44.** There are 51 houses on a street. Each house has an address between 1000 and 1099, inclusive. Show that at least two houses have addresses that are consecutive integers.

- *45. Let x be an irrational number. Show that for some positive integer j not exceeding the positive integer n , the absolute value of the difference between jx and the nearest integer to jx is less than $1/n$.
46. Let n_1, n_2, \dots, n_t be positive integers. Show that if $n_1 + n_2 + \dots + n_t - t + 1$ objects are placed into t boxes, then for some i , $i = 1, 2, \dots, t$, the i th box contains at least n_i objects.
- *47. An alternative proof of Theorem 3 based on the generalized pigeonhole principle is outlined in this exercise. The notation used is the same as that used in the proof in the text.
- a) Assume that $i_k \leq n$ for $k = 1, 2, \dots, n^2 + 1$. Use the generalized pigeonhole principle to show that there are $n + 1$ terms $a_{k_1}, a_{k_2}, \dots, a_{k_{n+1}}$ with $i_{k_1} = i_{k_2} = \dots = i_{k_{n+1}}$, where $1 \leq k_1 < k_2 < \dots < k_{n+1}$.
- b) Show that $a_{k_j} > a_{k_{j+1}}$ for $j = 1, 2, \dots, n$. [Hint: Assume that $a_{k_j} < a_{k_{j+1}}$, and show that this implies that $i_{k_j} > i_{k_{j+1}}$, which is a contradiction.]
- c) Use parts (a) and (b) to show that if there is no increasing subsequence of length $n + 1$, then there must be a decreasing subsequence of this length.

6.3 Permutations and Combinations

Introduction

Many counting problems can be solved by finding the number of ways to arrange a specified number of distinct elements of a set of a particular size, where the order of these elements matters. Many other counting problems can be solved by finding the number of ways to select a particular number of elements from a set of a particular size, where the order of the elements selected does not matter. For example, in how many ways can we select three students from a group of five students to stand in line for a picture? How many different committees of three students can be formed from a group of four students? In this section we will develop methods to answer questions such as these.

Permutations

We begin by solving the first question posed in the introduction to this section, as well as related questions.

EXAMPLE 1

In how many ways can we select three students from a group of five students to stand in line for a picture? In how many ways can we arrange all five of these students in a line for a picture?



Solution: First, note that the order in which we select the students matters. There are five ways to select the first student to stand at the start of the line. Once this student has been selected, there are four ways to select the second student in the line. After the first and second students have been selected, there are three ways to select the third student in the line. By the product rule, there are $5 \cdot 4 \cdot 3 = 60$ ways to select three students from a group of five students to stand in line for a picture.

To arrange all five students in a line for a picture, we select the first student in five ways, the second in four ways, the third in three ways, the fourth in two ways, and the fifth in one way. Consequently, there are $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$ ways to arrange all five students in a line for a picture. 

Example 1 illustrates how ordered arrangements of distinct objects can be counted. This leads to some terminology.



A **permutation** of a set of distinct objects is an ordered arrangement of these objects. We also are interested in ordered arrangements of some of the elements of a set. An ordered arrangement of r elements of a set is called an **r -permutation**.

EXAMPLE 2 Let $S = \{1, 2, 3\}$. The ordered arrangement $3, 1, 2$ is a permutation of S . The ordered arrangement $3, 2$ is a 2-permutation of S .

The number of r -permutations of a set with n elements is denoted by $P(n, r)$. We can find $P(n, r)$ using the product rule.

EXAMPLE 3 Let $S = \{a, b, c\}$. The 2-permutations of S are the ordered arrangements $a, b; a, c; b, a; b, c; c, a;$ and c, b . Consequently, there are six 2-permutations of this set with three elements. There are always six 2-permutations of a set with three elements. There are three ways to choose the first element of the arrangement. There are two ways to choose the second element of the arrangement, because it must be different from the first element. Hence, by the product rule, we see that $P(3, 2) = 3 \cdot 2 = 6$. By the product rule, it follows that $P(3, 2) = 3 \cdot 2 = 6$.

We now use the product rule to find a formula for $P(n, r)$ whenever n and r are positive integers with $1 \leq r \leq n$.

THEOREM 1

If n is a positive integer and r is an integer with $1 \leq r \leq n$, then there are

$$P(n, r) = n(n - 1)(n - 2) \cdots (n - r + 1)$$

r -permutations of a set with n distinct elements.

Proof: We will use the product rule to prove that this formula is correct. The first element of the permutation can be chosen in n ways because there are n elements in the set. There are $n - 1$ ways to choose the second element of the permutation, because there are $n - 1$ elements left in the set after using the element picked for the first position. Similarly, there are $n - 2$ ways to choose the third element, and so on, until there are exactly $n - (r - 1) = n - r + 1$ ways to choose the r th element. Consequently, by the product rule, there are

$$n(n - 1)(n - 2) \cdots (n - r + 1)$$

r -permutations of the set.

Note that $P(n, 0) = 1$ whenever n is a nonnegative integer because there is exactly one way to order zero elements. That is, there is exactly one list with no elements in it, namely the empty list.

We now state a useful corollary of Theorem 1.

COROLLARY 1

If n and r are integers with $0 \leq r \leq n$, then $P(n, r) = \frac{n!}{(n - r)!}$.

Proof: When n and r are integers with $1 \leq r \leq n$, by Theorem 1 we have

$$P(n, r) = n(n - 1)(n - 2) \cdots (n - r + 1) = \frac{n!}{(n - r)!}$$

Because $\frac{n!}{(n - 0)!} = \frac{n!}{n!} = 1$ whenever n is a nonnegative integer, we see that the formula $P(n, r) = \frac{n!}{(n - r)!}$ also holds when $r = 0$.

By Theorem 1 we know that if n is a positive integer, then $P(n, n) = n!$. We will illustrate this result with some examples.

EXAMPLE 4 How many ways are there to select a first-prize winner, a second-prize winner, and a third-prize winner from 100 different people who have entered a contest?

Solution: Because it matters which person wins which prize, the number of ways to pick the three prize winners is the number of ordered selections of three elements from a set of 100 elements, that is, the number of 3-permutations of a set of 100 elements. Consequently, the answer is

$$P(100, 3) = 100 \cdot 99 \cdot 98 = 970,200.$$

EXAMPLE 5 Suppose that there are eight runners in a race. The winner receives a gold medal, the second-place finisher receives a silver medal, and the third-place finisher receives a bronze medal. How many different ways are there to award these medals, if all possible outcomes of the race can occur and there are no ties?

Solution: The number of different ways to award the medals is the number of 3-permutations of a set with eight elements. Hence, there are $P(8, 3) = 8 \cdot 7 \cdot 6 = 336$ possible ways to award the medals.

EXAMPLE 6 Suppose that a saleswoman has to visit eight different cities. She must begin her trip in a specified city, but she can visit the other seven cities in any order she wishes. How many possible orders can the saleswoman use when visiting these cities?

Solution: The number of possible paths between the cities is the number of permutations of seven elements, because the first city is determined, but the remaining seven can be ordered arbitrarily. Consequently, there are $7! = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 5040$ ways for the saleswoman to choose her tour. If, for instance, the saleswoman wishes to find the path between the cities with minimum distance, and she computes the total distance for each possible path, she must consider a total of 5040 paths!

EXAMPLE 7 How many permutations of the letters $ABCDEFGH$ contain the string ABC ?

Solution: Because the letters ABC must occur as a block, we can find the answer by finding the number of permutations of six objects, namely, the block ABC and the individual letters D , E , F , G , and H . Because these six objects can occur in any order, there are $6! = 720$ permutations of the letters $ABCDEFGH$ in which ABC occurs as a block.

Combinations

We now turn our attention to counting unordered selections of objects. We begin by solving a question posed in the introduction to this section of the chapter.

EXAMPLE 8 How many different committees of three students can be formed from a group of four students?

Solution: To answer this question, we need only find the number of subsets with three elements from the set containing the four students. We see that there are four such subsets, one for each of the four students, because choosing three students is the same as choosing one of the four students to leave out of the group. This means that there are four ways to choose the three students for the committee, where the order in which these students are chosen does not matter.



Example 8 illustrates that many counting problems can be solved by finding the number of subsets of a particular size of a set with n elements, where n is a positive integer.

An **r -combination** of elements of a set is an unordered selection of r elements from the set. Thus, an r -combination is simply a subset of the set with r elements.

EXAMPLE 9

Let S be the set $\{1, 2, 3, 4\}$. Then $\{1, 3, 4\}$ is a 3-combination from S . (Note that $\{4, 1, 3\}$ is the same 3-combination as $\{1, 3, 4\}$, because the order in which the elements of a set are listed does not matter.)

The number of r -combinations of a set with n distinct elements is denoted by $C(n, r)$. Note that $C(n, r)$ is also denoted by $\binom{n}{r}$ and is called a **binomial coefficient**. We will learn where this terminology comes from in Section 6.4.

EXAMPLE 10

We see that $C(4, 2) = 6$, because the 2-combinations of $\{a, b, c, d\}$ are the six subsets $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, and $\{c, d\}$.

We can determine the number of r -combinations of a set with n elements using the formula for the number of r -permutations of a set. To do this, note that the r -permutations of a set can be obtained by first forming r -combinations and then ordering the elements in these combinations. The proof of Theorem 2, which gives the value of $C(n, r)$, is based on this observation.

THEOREM 2

The number of r -combinations of a set with n elements, where n is a nonnegative integer and r is an integer with $0 \leq r \leq n$, equals

$$C(n, r) = \frac{n!}{r!(n-r)!}.$$

Proof: The $P(n, r)$ r -permutations of the set can be obtained by forming the $C(n, r)$ r -combinations of the set, and then ordering the elements in each r -combination, which can be done in $P(r, r)$ ways. Consequently, by the product rule,

$$P(n, r) = C(n, r) \cdot P(r, r).$$

This implies that

$$C(n, r) = \frac{P(n, r)}{P(r, r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{r!(n-r)!}.$$

We can also use the division rule for counting to construct a proof of this theorem. Because the order of elements in a combination does not matter and there are $P(r, r)$ ways to order r elements in an r -combination of n elements, each of the $C(n, r)$ r -combinations of a set with n elements corresponds to exactly $P(r, r)$ r -permutations. Hence, by the division rule, $C(n, r) = \frac{P(n, r)}{P(r, r)}$, which implies as before that $C(n, r) = \frac{n!}{r!(n-r)!}$.

The formula in Theorem 2, although explicit, is not helpful when $C(n, r)$ is computed for large values of n and r . The reasons are that it is practical to compute exact values of factorials exactly only for small integer values, and when floating point arithmetic is used, the formula in Theorem 2 may produce a value that is not an integer. When computing $C(n, r)$, first note that when we cancel out $(n-r)!$ from the numerator and denominator of the expression for $C(n, r)$ in Theorem 2, we obtain

$$C(n, r) = \frac{n!}{r!(n-r)!} = \frac{n(n-1)\cdots(n-r+1)}{r!}.$$

Consequently, to compute $C(n, r)$ you can cancel out all the terms in the larger factorial in the denominator from the numerator and denominator, then multiply all the terms that do not cancel in the numerator and finally divide by the smaller factorial in the denominator. [When doing this calculation by hand, instead of by machine, it is also worthwhile to factor out common factors in the numerator $n(n - 1) \cdots (n - r + 1)$ and in the denominator $r!$.] Note that many calculators have a built-in function for $C(n, r)$ that can be used for relatively small values of n and r and many computational programs can be used to find $C(n, r)$. [Such functions may be called *choose*(n, k) or *binom*(n, k)].

Example 11 illustrates how $C(n, k)$ is computed when k is relatively small compared to n and when k is close to n . It also illustrates a key identity enjoyed by the numbers $C(n, k)$.

EXAMPLE 11 How many poker hands of five cards can be dealt from a standard deck of 52 cards? Also, how many ways are there to select 47 cards from a standard deck of 52 cards?

Solution: Because the order in which the five cards are dealt from a deck of 52 cards does not matter, there are

$$C(52, 5) = \frac{52!}{5!47!}$$

different hands of five cards that can be dealt. To compute the value of $C(52, 5)$, first divide the numerator and denominator by $47!$ to obtain

$$C(52, 5) = \frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}.$$

This expression can be simplified by first dividing the factor 5 in the denominator into the factor 50 in the numerator to obtain a factor 10 in the numerator, then dividing the factor 4 in the denominator into the factor 48 in the numerator to obtain a factor of 12 in the numerator, then dividing the factor 3 in the denominator into the factor 51 in the numerator to obtain a factor of 17 in the numerator, and finally, dividing the factor 2 in the denominator into the factor 52 in the numerator to obtain a factor of 26 in the numerator. We find that

$$C(52, 5) = 26 \cdot 17 \cdot 10 \cdot 49 \cdot 12 = 2,598,960.$$

Consequently, there are 2,598,960 different poker hands of five cards that can be dealt from a standard deck of 52 cards.

Note that there are

$$C(52, 47) = \frac{52!}{47!5!}$$

different ways to select 47 cards from a standard deck of 52 cards. We do not need to compute this value because $C(52, 47) = C(52, 5)$. (Only the order of the factors $5!$ and $47!$ is different in the denominators in the formulae for these quantities.) It follows that there are also 2,598,960 different ways to select 47 cards from a standard deck of 52 cards. 

In Example 11 we observed that $C(52, 5) = C(52, 47)$. This is a special case of the useful identity for the number of r -combinations of a set given in Corollary 2.

COROLLARY 2

Let n and r be nonnegative integers with $r \leq n$. Then $C(n, r) = C(n, n - r)$.

Proof: From Theorem 2 it follows that

$$C(n, r) = \frac{n!}{r!(n - r)!}$$

and

$$C(n, n - r) = \frac{n!}{(n - r)! [n - (n - r)]!} = \frac{n!}{(n - r)! r!}.$$

Hence, $C(n, r) = C(n, n - r)$. 

We can also prove Corollary 2 without relying on algebraic manipulation. Instead, we can use a combinatorial proof. We describe this important type of proof in Definition 1.

DEFINITION 1

A *combinatorial proof* of an identity is a proof that uses counting arguments to prove that both sides of the identity count the same objects but in different ways or a proof that is based on showing that there is a bijection between the sets of objects counted by the two sides of the identity. These two types of proofs are called *double counting proofs* and *bijective proofs*, respectively.

Combinatorial proofs are almost always much shorter and provide more insights than proofs based on algebraic manipulation.

Many identities involving binomial coefficients can be proved using combinatorial proofs. We now show how to prove Corollary 2 using a combinatorial proof. We will provide both a double counting proof and a bijective proof, both based on the same basic idea.

Proof: We will use a bijective proof to show that $C(n, r) = C(n, n - r)$ for all integers n and r with $0 \leq r \leq n$. Suppose that S is a set with n elements. The function that maps a subset A of S to \bar{A} is a bijection between subsets of S with r elements and subsets with $n - r$ elements (as the reader should verify). The identity $C(n, r) = C(n, n - r)$ follows because when there is a bijection between two finite sets, the two sets must have the same number of elements.

Alternatively, we can reformulate this argument as a double counting proof. By definition, the number of subsets of S with r elements equals $C(n, r)$. But each subset A of S is also determined by specifying which elements are not in A , and so are in \bar{A} . Because the complement of a subset of S with r elements has $n - r$ elements, there are also $C(n, n - r)$ subsets of S with r elements. It follows that $C(n, r) = C(n, n - r)$. 

EXAMPLE 12



How many ways are there to select five players from a 10-member tennis team to make a trip to a match at another school?

Solution: The answer is given by the number of 5-combinations of a set with 10 elements. By Theorem 2, the number of such combinations is

$$C(10, 5) = \frac{10!}{5! 5!} = 252. \quad \blacktriangleleft$$

EXAMPLE 13

A group of 30 people have been trained as astronauts to go on the first mission to Mars. How many ways are there to select a crew of six people to go on this mission (assuming that all crew members have the same job)?

Solution: The number of ways to select a crew of six from the pool of 30 people is the number of 6-combinations of a set with 30 elements, because the order in which these people are chosen does not matter. By Theorem 2, the number of such combinations is

$$C(30, 6) = \frac{30!}{6! 24!} = \frac{30 \cdot 29 \cdot 28 \cdot 27 \cdot 26 \cdot 25}{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 593,775. \quad \blacktriangleleft$$

EXAMPLE 14 How many bit strings of length n contain exactly r 1s?

Solution: The positions of r 1s in a bit string of length n form an r -combination of the set $\{1, 2, 3, \dots, n\}$. Hence, there are $C(n, r)$ bit strings of length n that contain exactly r 1s. 

EXAMPLE 15 Suppose that there are 9 faculty members in the mathematics department and 11 in the computer science department. How many ways are there to select a committee to develop a discrete mathematics course at a school if the committee is to consist of three faculty members from the mathematics department and four from the computer science department?

Solution: By the product rule, the answer is the product of the number of 3-combinations of a set with nine elements and the number of 4-combinations of a set with 11 elements. By Theorem 2, the number of ways to select the committee is

$$C(9, 3) \cdot C(11, 4) = \frac{9!}{3!6!} \cdot \frac{11!}{4!7!} = 84 \cdot 330 = 27,720.$$


Exercises

1. List all the permutations of $\{a, b, c\}$.
2. How many different permutations are there of the set $\{a, b, c, d, e, f, g\}$?
3. How many permutations of $\{a, b, c, d, e, f, g\}$ end with a ?
4. Let $S = \{1, 2, 3, 4, 5\}$.
 - a) List all the 3-permutations of S .
 - b) List all the 3-combinations of S .
5. Find the value of each of these quantities.

a) $P(6, 3)$	b) $P(6, 5)$
c) $P(8, 1)$	d) $P(8, 5)$
e) $P(8, 8)$	f) $P(10, 9)$
6. Find the value of each of these quantities.

a) $C(5, 1)$	b) $C(5, 3)$
c) $C(8, 4)$	d) $C(8, 8)$
e) $C(8, 0)$	f) $C(12, 6)$
7. Find the number of 5-permutations of a set with nine elements.
8. In how many different orders can five runners finish a race if no ties are allowed?
9. How many possibilities are there for the win, place, and show (first, second, and third) positions in a horse race with 12 horses if all orders of finish are possible?
10. There are six different candidates for governor of a state. In how many different orders can the names of the candidates be printed on a ballot?
11. How many bit strings of length 10 contain
 - a) exactly four 1s?
 - b) at most four 1s?
 - c) at least four 1s?
 - d) an equal number of 0s and 1s?
12. How many bit strings of length 12 contain
 - a) exactly three 1s?
 - b) at most three 1s?
 - c) at least three 1s?
 - d) an equal number of 0s and 1s?
13. A group contains n men and n women. How many ways are there to arrange these people in a row if the men and women alternate?
14. In how many ways can a set of two positive integers less than 100 be chosen?
15. In how many ways can a set of five letters be selected from the English alphabet?
16. How many subsets with an odd number of elements does a set with 10 elements have?
17. How many subsets with more than two elements does a set with 100 elements have?
18. A coin is flipped eight times where each flip comes up either heads or tails. How many possible outcomes
 - a) are there in total?
 - b) contain exactly three heads?
 - c) contain at least three heads?
 - d) contain the same number of heads and tails?
19. A coin is flipped 10 times where each flip comes up either heads or tails. How many possible outcomes
 - a) are there in total?
 - b) contain exactly two heads?
 - c) contain at most three tails?
 - d) contain the same number of heads and tails?
20. How many bit strings of length 10 have
 - a) exactly three 0s?
 - b) more 0s than 1s?
 - c) at least seven 1s?
 - d) at least three 1s?

- 21.** How many permutations of the letters $ABCDEFG$ contain
- the string BCD ?
 - the string $CFG A$?
 - the strings BA and GF ?
 - the strings ABC and DE ?
 - the strings ABC and CDE ?
 - the strings CBA and BED ?
- 22.** How many permutations of the letters $ABCDEFGHI$ contain
- the string ED ?
 - the string CDE ?
 - the strings BA and FGH ?
 - the strings AB , DE , and GH ?
 - the strings CAB and BED ?
 - the strings BCA and ABF ?
- 23.** How many ways are there for eight men and five women to stand in a line so that no two women stand next to each other? [Hint: First position the men and then consider possible positions for the women.]
- 24.** How many ways are there for 10 women and six men to stand in a line so that no two men stand next to each other? [Hint: First position the women and then consider possible positions for the men.]
- 25.** One hundred tickets, numbered 1, 2, 3, ..., 100, are sold to 100 different people for a drawing. Four different prizes are awarded, including a grand prize (a trip to Tahiti). How many ways are there to award the prizes if
- there are no restrictions?
 - the person holding ticket 47 wins the grand prize?
 - the person holding ticket 47 wins one of the prizes?
 - the person holding ticket 47 does not win a prize?
 - the people holding tickets 19 and 47 both win prizes?
 - the people holding tickets 19, 47, and 73 all win prizes?
 - the people holding tickets 19, 47, 73, and 97 all win prizes?
 - none of the people holding tickets 19, 47, 73, and 97 wins a prize?
 - the grand prize winner is a person holding ticket 19, 47, 73, or 97?
 - the people holding tickets 19 and 47 win prizes, but the people holding tickets 73 and 97 do not win prizes?
- 26.** Thirteen people on a softball team show up for a game.
- How many ways are there to choose 10 players to take the field?
 - How many ways are there to assign the 10 positions by selecting players from the 13 people who show up?
 - Of the 13 people who show up, three are women. How many ways are there to choose 10 players to take the field if at least one of these players must be a woman?
- 27.** A club has 25 members.
- How many ways are there to choose four members of the club to serve on an executive committee?
 - How many ways are there to choose a president, vice president, secretary, and treasurer of the club, where no person can hold more than one office?
- 28.** A professor writes 40 discrete mathematics true/false questions. Of the statements in these questions, 17 are true. If the questions can be positioned in any order, how many different answer keys are possible?
- *29.** How many 4-permutations of the positive integers not exceeding 100 contain three consecutive integers k , $k + 1$, $k + 2$, in the correct order
- where these consecutive integers can perhaps be separated by other integers in the permutation?
 - where they are in consecutive positions in the permutation?
- 30.** Seven women and nine men are on the faculty in the mathematics department at a school.
- How many ways are there to select a committee of five members of the department if at least one woman must be on the committee?
 - How many ways are there to select a committee of five members of the department if at least one woman and at least one man must be on the committee?
- 31.** The English alphabet contains 21 consonants and five vowels. How many strings of six lowercase letters of the English alphabet contain
- exactly one vowel?
 - exactly two vowels?
 - at least one vowel?
 - at least two vowels?
- 32.** How many strings of six lowercase letters from the English alphabet contain
- the letter a ?
 - the letters a and b ?
 - the letters a and b in consecutive positions with a preceding b , with all the letters distinct?
 - the letters a and b , where a is somewhere to the left of b in the string, with all the letters distinct?
- 33.** Suppose that a department contains 10 men and 15 women. How many ways are there to form a committee with six members if it must have the same number of men and women?
- 34.** Suppose that a department contains 10 men and 15 women. How many ways are there to form a committee with six members if it must have more women than men?
- 35.** How many bit strings contain exactly eight 0s and 10 1s if every 0 must be immediately followed by a 1?
- 36.** How many bit strings contain exactly five 0s and 14 1s if every 0 must be immediately followed by two 1s?
- 37.** How many bit strings of length 10 contain at least three 1s and at least three 0s?
- 38.** How many ways are there to select 12 countries in the United Nations to serve on a council if 3 are selected from a block of 45, 4 are selected from a block of 57, and the others are selected from the remaining 69 countries?

- 39.** How many license plates consisting of three letters followed by three digits contain no letter or digit twice?

A **circular r -permutation of n** people is a seating of r of these n people around a circular table, where seatings are considered to be the same if they can be obtained from each other by rotating the table.

- 40.** Find the number of circular 3-permutations of 5 people.
41. Find a formula for the number of circular r -permutations of n people.
42. Find a formula for the number of ways to seat r of n people around a circular table, where seatings are considered the same if every person has the same two neighbors without regard to which side these neighbors are sitting on.
43. How many ways are there for a horse race with three horses to finish if ties are possible? [Note: Two or three horses may tie.]
***44.** How many ways are there for a horse race with four horses to finish if ties are possible? [Note: Any number of the four horses may tie.]
***45.** There are six runners in the 100-yard dash. How many ways are there for three medals to be awarded if ties are possible? (The runner or runners who finish with the fastest time receive gold medals, the runner or runners who finish with exactly one runner ahead receive silver

medals, and the runner or runners who finish with exactly two runners ahead receive bronze medals.)

- *46.** This procedure is used to break ties in games in the championship round of the World Cup soccer tournament. Each team selects five players in a prescribed order. Each of these players takes a penalty kick, with a player from the first team followed by a player from the second team and so on, following the order of players specified. If the score is still tied at the end of the 10 penalty kicks, this procedure is repeated. If the score is still tied after 20 penalty kicks, a sudden-death shootout occurs, with the first team scoring an unanswered goal victorious.
- a) How many different scoring scenarios are possible if the game is settled in the first round of 10 penalty kicks, where the round ends once it is impossible for a team to equal the number of goals scored by the other team?
 - b) How many different scoring scenarios for the first and second groups of penalty kicks are possible if the game is settled in the second round of 10 penalty kicks?
 - c) How many scoring scenarios are possible for the full set of penalty kicks if the game is settled with no more than 10 total additional kicks after the two rounds of five kicks for each team?

6.4 Binomial Coefficients and Identities

As we remarked in Section 6.3, the number of r -combinations from a set with n elements is often denoted by $\binom{n}{r}$. This number is also called a **binomial coefficient** because these numbers occur as coefficients in the expansion of powers of binomial expressions such as $(a + b)^n$. We will discuss the **binomial theorem**, which gives a power of a binomial expression as a sum of terms involving binomial coefficients. We will prove this theorem using a combinatorial proof. We will also show how combinatorial proofs can be used to establish some of the many different identities that express relationships among binomial coefficients.

The Binomial Theorem



The binomial theorem gives the coefficients of the expansion of powers of binomial expressions. A **binomial expression** is simply the sum of two terms, such as $x + y$. (The terms can be products of constants and variables, but that does not concern us here.)

Example 1 illustrates how the coefficients in a typical expansion can be found and prepares us for the statement of the binomial theorem.

EXAMPLE 1

The expansion of $(x + y)^3$ can be found using combinatorial reasoning instead of multiplying the three terms out. When $(x + y)^3 = (x + y)(x + y)(x + y)$ is expanded, all products of a term in the first sum, a term in the second sum, and a term in the third sum are added. Terms of the form x^3 , x^2y , xy^2 , and y^3 arise. To obtain a term of the form x^3 , an x must be chosen in each of the sums, and this can be done in only one way. Thus, the x^3 term in the product has a coefficient of 1. To obtain a term of the form x^2y , an x must be chosen in two of the three sums (and consequently a y in the other sum). Hence, the number of such terms is the number of 2-combinations of three objects, namely, $\binom{3}{2}$. Similarly, the number of terms of the form xy^2 is the number of ways to pick one of the three sums to obtain an x (and consequently take a y

from each of the other two sums). This can be done in $\binom{3}{1}$ ways. Finally, the only way to obtain a y^3 term is to choose the y for each of the three sums in the product, and this can be done in exactly one way. Consequently, it follows that

$$\begin{aligned}(x+y)^3 &= (x+y)(x+y)(x+y) = (xx+xy+yx+yy)(x+y) \\ &= xxx+xxy+xyx+xyy+yxx+yxy+yyx+yyy \\ &= x^3 + 3x^2y + 3xy^2 + y^3.\end{aligned}$$



We now state the binomial theorem.

THEOREM 1

THE BINOMIAL THEOREM Let x and y be variables, and let n be a nonnegative integer. Then

$$(x+y)^n = \sum_{j=0}^n \binom{n}{j} x^{n-j} y^j = \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \cdots + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n.$$

Proof: We use a combinatorial proof. The terms in the product when it is expanded are of the form $x^{n-j} y^j$ for $j = 0, 1, 2, \dots, n$. To count the number of terms of the form $x^{n-j} y^j$, note that to obtain such a term it is necessary to choose $n - j$ x s from the n sums (so that the other j terms in the product are y s). Therefore, the coefficient of $x^{n-j} y^j$ is $\binom{n}{n-j}$, which is equal to $\binom{n}{j}$. This proves the theorem. ◀

Some computational uses of the binomial theorem are illustrated in Examples 2–4.

EXAMPLE 2

What is the expansion of $(x+y)^4$?



Solution: From the binomial theorem it follows that

$$\begin{aligned}(x+y)^4 &= \sum_{j=0}^4 \binom{4}{j} x^{4-j} y^j \\ &= \binom{4}{0} x^4 + \binom{4}{1} x^3 y + \binom{4}{2} x^2 y^2 + \binom{4}{3} x y^3 + \binom{4}{4} y^4 \\ &= x^4 + 4x^3 y + 6x^2 y^2 + 4x y^3 + y^4.\end{aligned}$$



EXAMPLE 3

What is the coefficient of $x^{12} y^{13}$ in the expansion of $(x+y)^{25}$?

Solution: From the binomial theorem it follows that this coefficient is

$$\binom{25}{13} = \frac{25!}{13! 12!} = 5,200,300.$$



EXAMPLE 4

What is the coefficient of $x^{12} y^{13}$ in the expansion of $(2x-3y)^{25}$?

Solution: First, note that this expression equals $(2x+(-3y))^{25}$. By the binomial theorem, we have

$$(2x+(-3y))^{25} = \sum_{j=0}^{25} \binom{25}{j} (2x)^{25-j} (-3y)^j.$$

Consequently, the coefficient of $x^{12}y^{13}$ in the expansion is obtained when $j = 13$, namely,

$$\binom{25}{13}2^{12}(-3)^{13} = -\frac{25!}{13!12!}2^{12}3^{13}. \quad \blacktriangleleft$$

We can prove some useful identities using the binomial theorem, as Corollaries 1, 2, and 3 demonstrate.

COROLLARY 1

Let n be a nonnegative integer. Then

$$\sum_{k=0}^n \binom{n}{k} = 2^n.$$

Proof: Using the binomial theorem with $x = 1$ and $y = 1$, we see that

$$2^n = (1+1)^n = \sum_{k=0}^n \binom{n}{k} 1^k 1^{n-k} = \sum_{k=0}^n \binom{n}{k}.$$

This is the desired result. \blacktriangleleft

There is also a nice combinatorial proof of Corollary 1, which we now present.

Proof: A set with n elements has a total of 2^n different subsets. Each subset has zero elements, one element, two elements, \dots , or n elements in it. There are $\binom{n}{0}$ subsets with zero elements, $\binom{n}{1}$ subsets with one element, $\binom{n}{2}$ subsets with two elements, \dots , and $\binom{n}{n}$ subsets with n elements. Therefore,

$$\sum_{k=0}^n \binom{n}{k}$$

counts the total number of subsets of a set with n elements. By equating the two formulas we have for the number of subsets of a set with n elements, we see that

$$\sum_{k=0}^n \binom{n}{k} = 2^n. \quad \blacktriangleleft$$

COROLLARY 2

Let n be a positive integer. Then

$$\sum_{k=0}^n (-1)^k \binom{n}{k} = 0.$$

Proof: When we use the binomial theorem with $x = -1$ and $y = 1$, we see that

$$0 = 0^n = ((-1) + 1)^n = \sum_{k=0}^n \binom{n}{k} (-1)^k 1^{n-k} = \sum_{k=0}^n \binom{n}{k} (-1)^k.$$

This proves the corollary. \blacktriangleleft

Remark: Corollary 2 implies that

$$\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \cdots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots.$$

COROLLARY 3

Let n be a nonnegative integer. Then

$$\sum_{k=0}^n 2^k \binom{n}{k} = 3^n.$$

Proof: We recognize that the left-hand side of this formula is the expansion of $(1 + 2)^n$ provided by the binomial theorem. Therefore, by the binomial theorem, we see that

$$(1 + 2)^n = \sum_{k=0}^n \binom{n}{k} 1^{n-k} 2^k = \sum_{k=0}^n \binom{n}{k} 2^k.$$

Hence

$$\sum_{k=0}^n 2^k \binom{n}{k} = 3^n.$$



Pascal's Identity and Triangle

The binomial coefficients satisfy many different identities. We introduce one of the most important of these now.

THEOREM 2

PASCAL'S IDENTITY Let n and k be positive integers with $n \geq k$. Then

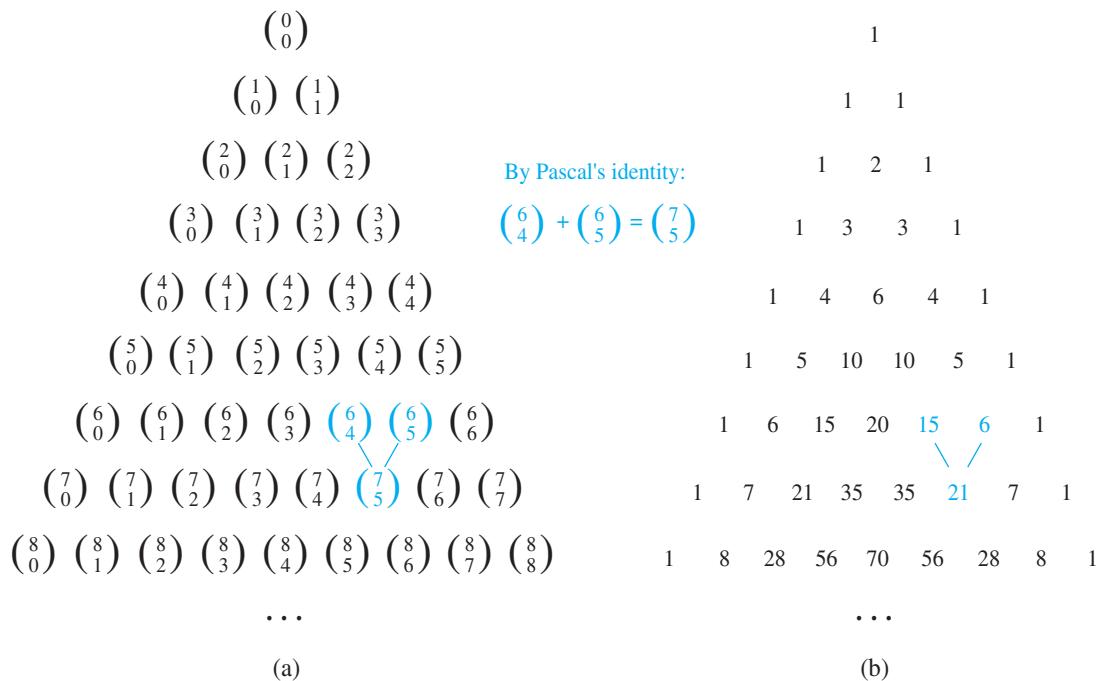
$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}.$$

Proof: We will use a combinatorial proof. Suppose that T is a set containing $n + 1$ elements. Let a be an element in T , and let $S = T - \{a\}$. Note that there are $\binom{n+1}{k}$ subsets of T containing k elements. However, a subset of T with k elements either contains a together with $k - 1$ elements of S , or contains k elements of S and does not contain a . Because there are $\binom{n}{k-1}$ subsets of $k - 1$ elements of S , there are $\binom{n}{k-1}$ subsets of k elements of T that contain a . And there are $\binom{n}{k}$ subsets of k elements of T that do not contain a , because there are $\binom{n}{k}$ subsets of k elements of S . Consequently,

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}.$$



Remark: It is also possible to prove this identity by algebraic manipulation from the formula for $\binom{n}{r}$ (see Exercise 19).

**FIGURE 1** Pascal's Triangle.

Remark: Pascal's identity, together with the initial conditions $\binom{n}{0} = \binom{n}{n} = 1$ for all integers n , can be used to recursively define binomial coefficients. This recursive definition is useful in the computation of binomial coefficients because only addition, and not multiplication, of integers is needed to use this recursive definition.

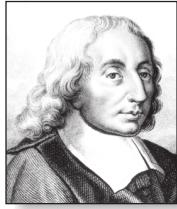
Pascal's identity is the basis for a geometric arrangement of the binomial coefficients in a triangle, as shown in Figure 1.

The n th row in the triangle consists of the binomial coefficients

$$\binom{n}{k}, \quad k = 0, 1, \dots, n.$$

This triangle is known as **Pascal's triangle**. Pascal's identity shows that when two adjacent binomial coefficients in this triangle are added, the binomial coefficient in the next row between these two coefficients is produced.

Links



BLAISE PASCAL (1623–1662) Blaise Pascal exhibited his talents at an early age, although his father, who had made discoveries in analytic geometry, kept mathematics books away from him to encourage other interests. At 16 Pascal discovered an important result concerning conic sections. At 18 he designed a calculating machine, which he built and sold. Pascal, along with Fermat, laid the foundations for the modern theory of probability. In this work, he made new discoveries concerning what is now called Pascal's triangle. In 1654, Pascal abandoned his mathematical pursuits to devote himself to theology. After this, he returned to mathematics only once. One night, distracted by a severe toothache, he sought comfort by studying the mathematical properties of the cycloid. Miraculously, his pain subsided, which he took as a sign of divine approval of the study of mathematics.

Other Identities Involving Binomial Coefficients

We conclude this section with combinatorial proofs of two of the many identities enjoyed by the binomial coefficients.

THEOREM 3

VANDERMONDE'S IDENTITY Let m , n , and r be nonnegative integers with r not exceeding either m or n . Then

$$\binom{m+n}{r} = \sum_{k=0}^r \binom{m}{r-k} \binom{n}{k}.$$



Remark: This identity was discovered by mathematician Alexandre-Théophile Vandermonde in the eighteenth century.

Proof: Suppose that there are m items in one set and n items in a second set. Then the total number of ways to pick r elements from the union of these sets is $\binom{m+n}{r}$.

Another way to pick r elements from the union is to pick k elements from the second set and then $r - k$ elements from the first set, where k is an integer with $0 \leq k \leq r$. Because there are $\binom{n}{k}$ ways to choose k elements from the second set and $\binom{m}{r-k}$ ways to choose $r - k$ elements from the first set, the product rule tells us that this can be done in $\binom{m}{r-k} \binom{n}{k}$ ways. Hence, the total number of ways to pick r elements from the union also equals $\sum_{k=0}^r \binom{m}{r-k} \binom{n}{k}$.

We have found two expressions for the number of ways to pick r elements from the union of a set with m items and a set with n items. Equating them gives us Vandermonde's identity.

Corollary 4 follows from Vandermonde's identity.

COROLLARY 4

If n is a nonnegative integer, then

$$\binom{2n}{n} = \sum_{k=0}^n \binom{n}{k}^2.$$

Proof: We use Vandermonde's identity with $m = r = n$ to obtain

$$\binom{2n}{n} = \sum_{k=0}^n \binom{n}{n-k} \binom{n}{k} = \sum_{k=0}^n \binom{n}{k}^2.$$

The last equality was obtained using the identity $\binom{n}{k} = \binom{n}{n-k}$.



ALEXANDRE-THÉOPHILE VANDERMONDE (1735–1796) Because Alexandre-Théophile Vandermonde was a sickly child, his physician father directed him to a career in music. However, he later developed an interest in mathematics. His complete mathematical work consists of four papers published in 1771–1772. These papers include fundamental contributions on the roots of equations, on the theory of determinants, and on the knight's tour problem (introduced in the exercises in Section 10.5). Vandermonde's interest in mathematics lasted for only 2 years. Afterward, he published papers on harmony, experiments with cold, and the manufacture of steel. He also became interested in politics, joining the cause of the French revolution and holding several different positions in government.

We can prove combinatorial identities by counting bit strings with different properties, as the proof of Theorem 4 will demonstrate.

THEOREM 4

Let n and r be nonnegative integers with $r \leq n$. Then

$$\binom{n+1}{r+1} = \sum_{j=r}^n \binom{j}{r}.$$

Proof: We use a combinatorial proof. By Example 14 in Section 6.3, the left-hand side, $\binom{n+1}{r+1}$, counts the bit strings of length $n+1$ containing $r+1$ ones.

We show that the right-hand side counts the same objects by considering the cases corresponding to the possible locations of the final 1 in a string with $r+1$ ones. This final one must occur at position $r+1, r+2, \dots$, or $n+1$. Furthermore, if the last one is the k th bit there must be r ones among the first $k-1$ positions. Consequently, by Example 14 in Section 6.3, there are $\binom{k-1}{r}$ such bit strings. Summing over k with $r+1 \leq k \leq n+1$, we find that there are

$$\sum_{k=r+1}^{n+1} \binom{k-1}{r} = \sum_{j=r}^n \binom{j}{r}$$

bit strings of length n containing exactly $r+1$ ones. (Note that the last step follows from the change of variables $j = k-1$.) Because the left-hand side and the right-hand side count the same objects, they are equal. This completes the proof. \triangleleft

Exercises

1. Find the expansion of $(x+y)^4$
 - using combinatorial reasoning, as in Example 1.
 - using the binomial theorem.
2. Find the expansion of $(x+y)^5$
 - using combinatorial reasoning, as in Example 1.
 - using the binomial theorem.
3. Find the expansion of $(x+y)^6$.
4. Find the coefficient of x^5y^8 in $(x+y)^{13}$.
5. How many terms are there in the expansion of $(x+y)^{100}$ after like terms are collected?
6. What is the coefficient of x^7 in $(1+x)^{11}$?
7. What is the coefficient of x^9 in $(2-x)^{19}$?
8. What is the coefficient of x^8y^9 in the expansion of $(3x+2y)^{17}$?
9. What is the coefficient of $x^{101}y^{99}$ in the expansion of $(2x-3y)^{200}$?
- *10. Give a formula for the coefficient of x^k in the expansion of $(x+1/x)^{100}$, where k is an integer.
- *11. Give a formula for the coefficient of x^k in the expansion of $(x^2-1/x)^{100}$, where k is an integer.
12. The row of Pascal's triangle containing the binomial coefficients $\binom{10}{k}$, $0 \leq k \leq 10$, is:
1 10 45 120 210 252 210 120 45 10 1
13. What is the row of Pascal's triangle containing the binomial coefficients $\binom{9}{k}$, $0 \leq k \leq 9$?
14. Show that if n is a positive integer, then $1 = \binom{n}{0} < \binom{n}{1} < \cdots < \binom{n}{\lfloor n/2 \rfloor} = \binom{n}{\lceil n/2 \rceil} > \cdots > \binom{n}{n-1} > \binom{n}{n} = 1$.
15. Show that $\binom{n}{k} \leq 2^n$ for all positive integers n and all integers k with $0 \leq k \leq n$.
16. a) Use Exercise 14 and Corollary 1 to show that if n is an integer greater than 1, then $\binom{n}{\lfloor n/2 \rfloor} \geq 2^n/n$.
b) Conclude from part (a) that if n is a positive integer, then $\binom{2n}{n} \geq 4^n/2n$.
17. Show that if n and k are integers with $1 \leq k \leq n$, then $\binom{n}{k} \leq n^k/2^{k-1}$.
18. Suppose that b is an integer with $b \geq 7$. Use the binomial theorem and the appropriate row of Pascal's triangle to find the base- b expansion of $(11)_b^4$ [that is, the fourth power of the number $(11)_b$ in base- b notation].
19. Prove Pascal's identity, using the formula for $\binom{n}{r}$.
20. Suppose that k and n are integers with $1 \leq k < n$. Prove the **hexagon identity**

$$\binom{n-1}{k-1} \binom{n}{k+1} \binom{n+1}{k} = \binom{n-1}{k} \binom{n}{k-1} \binom{n+1}{k+1},$$

which relates terms in Pascal's triangle that form a hexagon.

Use Pascal's identity to produce the row immediately following this row in Pascal's triangle.

- 21.** Prove that if n and k are integers with $1 \leq k \leq n$, then $k \binom{n}{k} = n \binom{n-1}{k-1}$.
- using a combinatorial proof. [Hint: Show that the two sides of the identity count the number of ways to select a subset with k elements from a set with n elements and then an element of this subset.]
 - using an algebraic proof based on the formula for $\binom{n}{r}$ given in Theorem 2 in Section 6.3.
- 22.** Prove the identity $\binom{n}{r} \binom{r}{k} = \binom{n}{k} \binom{n-k}{r-k}$, whenever n, r , and k are nonnegative integers with $r \leq n$ and $k \leq r$,
- using a combinatorial argument.
 - using an argument based on the formula for the number of r -combinations of a set with n elements.
- 23.** Show that if n and k are positive integers, then

$$\binom{n+1}{k} = (n+1) \binom{n}{k-1} / k.$$

Use this identity to construct an inductive definition of the binomial coefficients.

- 24.** Show that if p is a prime and k is an integer such that $1 \leq k \leq p-1$, then p divides $\binom{p}{k}$.
- 25.** Let n be a positive integer. Show that

$$\binom{2n}{n+1} + \binom{2n}{n} = \binom{2n+2}{n+1} / 2.$$

- *26.** Let n and k be integers with $1 \leq k \leq n$. Show that

$$\sum_{k=1}^n \binom{n}{k} \binom{n}{k-1} = \binom{2n+2}{n+1} / 2 - \binom{2n}{n}.$$

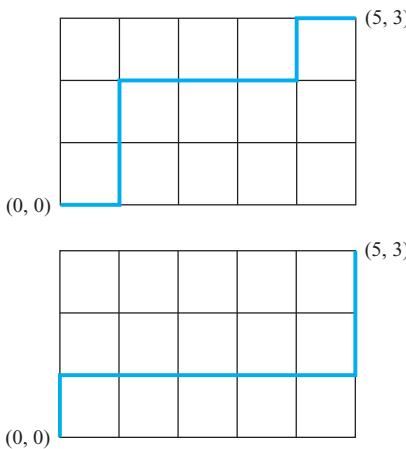
- *27.** Prove the **hockeystick identity**

$$\sum_{k=0}^r \binom{n+k}{k} = \binom{n+r+1}{r}$$

whenever n and r are positive integers,

- using a combinatorial argument.
 - using Pascal's identity.
- 28.** Show that if n is a positive integer, then $\binom{2n}{2} = 2\binom{n}{2} + n^2$
- using a combinatorial argument.
 - by algebraic manipulation.
- *29.** Give a combinatorial proof that $\sum_{k=1}^n k \binom{n}{k} = n 2^{n-1}$. [Hint: Count in two ways the number of ways to select a committee and to then select a leader of the committee.]
- *30.** Give a combinatorial proof that $\sum_{k=1}^n k \binom{n}{k}^2 = n \binom{2n-1}{n-1}$. [Hint: Count in two ways the number of ways to select a committee, with n members from a group of n mathematics professors and n computer science professors, such that the chairperson of the committee is a mathematics professor.]
- 31.** Show that a nonempty set has the same number of subsets with an odd number of elements as it does subsets with an even number of elements.
- *32.** Prove the binomial theorem using mathematical induction.

- 33.** In this exercise we will count the number of paths in the xy plane between the origin $(0, 0)$ and point (m, n) , where m and n are nonnegative integers, such that each path is made up of a series of steps, where each step is a move one unit to the right or a move one unit upward. (No moves to the left or downward are allowed.) Two such paths from $(0, 0)$ to $(5, 3)$ are illustrated here.



- Show that each path of the type described can be represented by a bit string consisting of m 0s and n 1s, where a 0 represents a move one unit to the right and a 1 represents a move one unit upward.
 - Conclude from part (a) that there are $\binom{m+n}{n}$ paths of the desired type.
- 34.** Use Exercise 33 to give an alternative proof of Corollary 2 in Section 6.3, which states that $\binom{n}{k} = \binom{n}{n-k}$ whenever k is an integer with $0 \leq k \leq n$. [Hint: Consider the number of paths of the type described in Exercise 33 from $(0, 0)$ to $(n-k, k)$ and from $(0, 0)$ to $(k, n-k)$.]
- 35.** Use Exercise 33 to prove Theorem 4. [Hint: Count the number of paths with n steps of the type described in Exercise 33. Every such path must end at one of the points $(n-k, k)$ for $k = 0, 1, 2, \dots, n$.]
- 36.** Use Exercise 33 to prove Pascal's identity. [Hint: Show that a path of the type described in Exercise 33 from $(0, 0)$ to $(n+1-k, k)$ passes through either $(n+1-k, k-1)$ or $(n-k, k)$, but not through both.]
- 37.** Use Exercise 33 to prove the hockeystick identity from Exercise 27. [Hint: First, note that the number of paths from $(0, 0)$ to $(n+1, r)$ equals $\binom{n+1+r}{r}$. Second, count the number of paths by summing the number of these paths that start by going k units upward for $k = 0, 1, 2, \dots, r$.]
- 38.** Give a combinatorial proof that if n is a positive integer then $\sum_{k=0}^n k^2 \binom{n}{k} = n(n+1)2^{n-2}$. [Hint: Show that both sides count the ways to select a subset of a set of n elements together with two not necessarily distinct elements from this subset. Furthermore, express the right-hand side as $n(n-1)2^{n-2} + n2^{n-1}$.]
- *39.** Determine a formula involving binomial coefficients for the n th term of a sequence if its initial terms are those listed. [Hint: Looking at Pascal's triangle will be helpful.]

Although infinitely many sequences start with a specified set of terms, each of the following lists is the start of a sequence of the type desired.]

- a) 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, ...
- b) 1, 4, 10, 20, 35, 56, 84, 120, 165, 220, ...

- c) 1, 2, 6, 20, 70, 252, 924, 3432, 12870, 48620, ...
- d) 1, 1, 2, 3, 6, 10, 20, 35, 70, 126, ...
- e) 1, 1, 1, 3, 1, 5, 15, 35, 1, 9, ...
- f) 1, 3, 15, 84, 495, 3003, 18564, 116280, 735471, 4686825, ...

6.5 Generalized Permutations and Combinations

Introduction



In many counting problems, elements may be used repeatedly. For instance, a letter or digit may be used more than once on a license plate. When a dozen donuts are selected, each variety can be chosen repeatedly. This contrasts with the counting problems discussed earlier in the chapter where we considered only permutations and combinations in which each item could be used at most once. In this section we will show how to solve counting problems where elements may be used more than once.

Also, some counting problems involve indistinguishable elements. For instance, to count the number of ways the letters of the word *SUCCESS* can be rearranged, the placement of identical letters must be considered. This contrasts with the counting problems discussed earlier where all elements were considered distinguishable. In this section we will describe how to solve counting problems in which some elements are indistinguishable.

Moreover, in this section we will explain how to solve another important class of counting problems, problems involving counting the ways distinguishable elements can be placed in boxes. An example of this type of problem is the number of different ways poker hands can be dealt to four players.

Taken together, the methods described earlier in this chapter and the methods introduced in this section form a useful toolbox for solving a wide range of counting problems. When the additional methods discussed in Chapter 8 are added to this arsenal, you will be able to solve a large percentage of the counting problems that arise in a wide range of areas of study.

Permutations with Repetition

Counting permutations when repetition of elements is allowed can easily be done using the product rule, as Example 1 shows.

EXAMPLE 1 How many strings of length r can be formed from the uppercase letters of the English alphabet?

Solution: By the product rule, because there are 26 uppercase English letters, and because each letter can be used repeatedly, we see that there are 26^r strings of uppercase English letters of length r . 

The number of r -permutations of a set with n elements when repetition is allowed is given in Theorem 1.

THEOREM 1

The number of r -permutations of a set of n objects with repetition allowed is n^r .

Proof: There are n ways to select an element of the set for each of the r positions in the r -permutation when repetition is allowed, because for each choice all n objects are available. Hence, by the product rule there are n^r r -permutations when repetition is allowed. 

Combinations with Repetition

Consider these examples of combinations with repetition of elements allowed.

EXAMPLE 2

How many ways are there to select four pieces of fruit from a bowl containing apples, oranges, and pears if the order in which the pieces are selected does not matter, only the type of fruit and not the individual piece matters, and there are at least four pieces of each type of fruit in the bowl?

Solution: To solve this problem we list all the ways possible to select the fruit. There are 15 ways:

4 apples	4 oranges	4 pears
3 apples, 1 orange	3 apples, 1 pear	3 oranges, 1 apple
3 oranges, 1 pear	3 pears, 1 apple	3 pears, 1 orange
2 apples, 2 oranges	2 apples, 2 pears	2 oranges, 2 pears
2 apples, 1 orange, 1 pear	2 oranges, 1 apple, 1 pear	2 pears, 1 apple, 1 orange

The solution is the number of 4-combinations with repetition allowed from a three-element set, $\{\text{apple}, \text{orange}, \text{pear}\}$. 

To solve more complex counting problems of this type, we need a general method for counting the r -combinations of an n -element set. In Example 3 we will illustrate such a method.

EXAMPLE 3

How many ways are there to select five bills from a cash box containing \$1 bills, \$2 bills, \$5 bills, \$10 bills, \$20 bills, \$50 bills, and \$100 bills? Assume that the order in which the bills are chosen does not matter, that the bills of each denomination are indistinguishable, and that there are at least five bills of each type.

Solution: Because the order in which the bills are selected does not matter and seven different types of bills can be selected as many as five times, this problem involves counting 5-combinations with repetition allowed from a set with seven elements. Listing all possibilities would be tedious, because there are a large number of solutions. Instead, we will illustrate the use of a technique for counting combinations with repetition allowed.

Suppose that a cash box has seven compartments, one to hold each type of bill, as illustrated in Figure 1. These compartments are separated by six dividers, as shown in the picture. The choice of five bills corresponds to placing five markers in the compartments holding different types of bills. Figure 2 illustrates this correspondence for three different ways to select five bills, where the six dividers are represented by bars and the five bills by stars.

The number of ways to select five bills corresponds to the number of ways to arrange six bars and five stars in a row with a total of 11 positions. Consequently, the number of ways to select the five bills is the number of ways to select the positions of the five stars from the 11 positions. This corresponds to the number of unordered selections of 5 objects from a set of 11

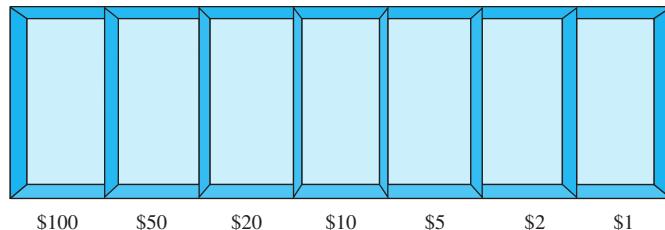
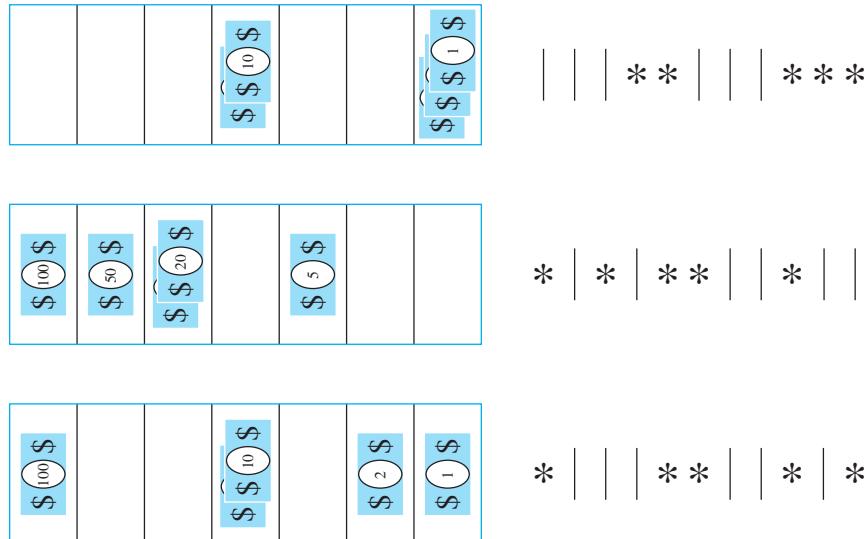


FIGURE 1 Cash Box with Seven Types of Bills.

**FIGURE 2** Examples of Ways to Select Five Bills.

objects, which can be done in $C(11, 5)$ ways. Consequently, there are

$$C(11, 5) = \frac{11!}{5! 6!} = 462$$

ways to choose five bills from the cash box with seven types of bills. ◀

Theorem 2 generalizes this discussion.

THEOREM 2

There are $C(n + r - 1, r) = C(n + r - 1, n - 1)$ r -combinations from a set with n elements when repetition of elements is allowed.

Proof: Each r -combination of a set with n elements when repetition is allowed can be represented by a list of $n - 1$ bars and r stars. The $n - 1$ bars are used to mark off n different cells, with the i th cell containing a star for each time the i th element of the set occurs in the combination. For instance, a 6-combination of a set with four elements is represented with three bars and six stars. Here

** | * | | * * *

represents the combination containing exactly two of the first element, one of the second element, none of the third element, and three of the fourth element of the set.

As we have seen, each different list containing $n - 1$ bars and r stars corresponds to an r -combination of the set with n elements, when repetition is allowed. The number of such lists is $C(n - 1 + r, r)$, because each list corresponds to a choice of the r positions to place the r stars from the $n - 1 + r$ positions that contain r stars and $n - 1$ bars. The number of such lists is also equal to $C(n - 1 + r, n - 1)$, because each list corresponds to a choice of the $n - 1$ positions to place the $n - 1$ bars. ◀

Examples 4–6 show how Theorem 2 is applied.

EXAMPLE 4 Suppose that a cookie shop has four different kinds of cookies. How many different ways can six cookies be chosen? Assume that only the type of cookie, and not the individual cookies or the order in which they are chosen, matters.



Solution: The number of ways to choose six cookies is the number of 6-combinations of a set with four elements. From Theorem 2 this equals $C(4 + 6 - 1, 6) = C(9, 6)$. Because

$$C(9, 6) = C(9, 3) = \frac{9 \cdot 8 \cdot 7}{1 \cdot 2 \cdot 3} = 84,$$

there are 84 different ways to choose the six cookies.

Theorem 2 can also be used to find the number of solutions of certain linear equations where the variables are integers subject to constraints. This is illustrated by Example 5.

EXAMPLE 5 How many solutions does the equation

$$x_1 + x_2 + x_3 = 11$$

have, where x_1 , x_2 , and x_3 are nonnegative integers?

Solution: To count the number of solutions, we note that a solution corresponds to a way of selecting 11 items from a set with three elements so that x_1 items of type one, x_2 items of type two, and x_3 items of type three are chosen. Hence, the number of solutions is equal to the number of 11-combinations with repetition allowed from a set with three elements. From Theorem 2 it follows that there are

$$C(3 + 11 - 1, 11) = C(13, 11) = C(13, 2) = \frac{13 \cdot 12}{1 \cdot 2} = 78$$

solutions.

The number of solutions of this equation can also be found when the variables are subject to constraints. For instance, we can find the number of solutions where the variables are integers with $x_1 \geq 1$, $x_2 \geq 2$, and $x_3 \geq 3$. A solution to the equation subject to these constraints corresponds to a selection of 11 items with x_1 items of type one, x_2 items of type two, and x_3 items of type three, where, in addition, there is at least one item of type one, two items of type two, and three items of type three. So, a solution corresponds to a choice of one item of type one, two of type two, and three of type three, together with a choice of five additional items of any type. By Theorem 2 this can be done in

$$C(3 + 5 - 1, 5) = C(7, 5) = C(7, 2) = \frac{7 \cdot 6}{1 \cdot 2} = 21$$

ways. Thus, there are 21 solutions of the equation subject to the given constraints.

Example 6 shows how counting the number of combinations with repetition allowed arises in determining the value of a variable that is incremented each time a certain type of nested loop is traversed.

TABLE 1 Combinations and Permutations With and Without Repetition.

Type	Repetition Allowed?	Formula
r -permutations	No	$\frac{n!}{(n - r)!}$
r -combinations	No	$\frac{n!}{r! (n - r)!}$
r -permutations	Yes	n^r
r -combinations	Yes	$\frac{(n + r - 1)!}{r! (n - 1)!}$

EXAMPLE 6 What is the value of k after the following pseudocode has been executed?

```

k := 0
for i1 := 1 to n
    for i2 := 1 to i1
        .
        .
        .
    for im := 1 to im-1
        k := k + 1

```

Solution: Note that the initial value of k is 0 and that 1 is added to k each time the nested loop is traversed with a sequence of integers i_1, i_2, \dots, i_m such that

$$1 \leq i_m \leq i_{m-1} \leq \cdots \leq i_1 \leq n.$$

The number of such sequences of integers is the number of ways to choose m integers from $\{1, 2, \dots, n\}$, with repetition allowed. (To see this, note that once such a sequence has been selected, if we order the integers in the sequence in nondecreasing order, this uniquely defines an assignment of i_m, i_{m-1}, \dots, i_1 . Conversely, every such assignment corresponds to a unique unordered set.) Hence, from Theorem 2, it follows that $k = C(n + m - 1, m)$ after this code has been executed. 

The formulae for the numbers of ordered and unordered selections of r elements, chosen with and without repetition allowed from a set with n elements, are shown in Table 1.

Permutations with Indistinguishable Objects

Some elements may be indistinguishable in counting problems. When this is the case, care must be taken to avoid counting things more than once. Consider Example 7.

EXAMPLE 7 How many different strings can be made by reordering the letters of the word *SUCCESS*?



Solution: Because some of the letters of *SUCCESS* are the same, the answer is *not* given by the number of permutations of seven letters. This word contains three Ss, two Cs, one U, and one E. To determine the number of different strings that can be made by reordering the letters, first note that the three Ss can be placed among the seven positions in $C(7, 3)$ different ways, leaving four

positions free. Then the two C s can be placed in $C(4, 2)$ ways, leaving two free positions. The U can be placed in $C(2, 1)$ ways, leaving just one position free. Hence E can be placed in $C(1, 1)$ way. Consequently, from the product rule, the number of different strings that can be made is

$$\begin{aligned} C(7, 3)C(4, 2)C(2, 1)C(1, 1) &= \frac{7!}{3!4!} \cdot \frac{4!}{2!2!} \cdot \frac{2!}{1!1!} \cdot \frac{1!}{1!0!} \\ &= \frac{7!}{3!2!1!1!} \\ &= 420. \end{aligned}$$



We can prove Theorem 3 using the same sort of reasoning as in Example 7.

THEOREM 3

The number of different permutations of n objects, where there are n_1 indistinguishable objects of type 1, n_2 indistinguishable objects of type 2, ..., and n_k indistinguishable objects of type k , is

$$\frac{n!}{n_1! n_2! \cdots n_k!}.$$

Proof: To determine the number of permutations, first note that the n_1 objects of type one can be placed among the n positions in $C(n, n_1)$ ways, leaving $n - n_1$ positions free. Then the objects of type two can be placed in $C(n - n_1, n_2)$ ways, leaving $n - n_1 - n_2$ positions free. Continue placing the objects of type three, ..., type $k - 1$, until at the last stage, n_k objects of type k can be placed in $C(n - n_1 - n_2 - \cdots - n_{k-1}, n_k)$ ways. Hence, by the product rule, the total number of different permutations is

$$C(n, n_1)C(n - n_1, n_2) \cdots C(n - n_1 - \cdots - n_{k-1}, n_k)$$

$$\begin{aligned} &= \frac{n!}{n_1!(n - n_1)!} \frac{(n - n_1)!}{n_2!(n - n_1 - n_2)!} \cdots \frac{(n - n_1 - \cdots - n_{k-1})!}{n_k!0!} \\ &= \frac{n!}{n_1! n_2! \cdots n_k!}. \end{aligned}$$



Distributing Objects into Boxes



Many counting problems can be solved by enumerating the ways objects can be placed into boxes (where the order these objects are placed into the boxes does not matter). The objects can be either *distinguishable*, that is, different from each other, or *indistinguishable*, that is, considered identical. Distinguishable objects are sometimes said to be *labeled*, whereas indistinguishable objects are said to be *unlabeled*. Similarly, boxes can be *distinguishable*, that is, different, or *indistinguishable*, that is, identical. Distinguishable boxes are often said to be *labeled*, while indistinguishable boxes are said to be *unlabeled*. When you solve a counting problem using the model of distributing objects into boxes, you need to determine whether the objects are distinguishable and whether the boxes are distinguishable. Although the context of the counting problem makes these two decisions clear, counting problems are sometimes ambiguous and it may be unclear which model applies. In such a case it is best to state whatever assumptions you are making and explain why the particular model you choose conforms to your assumptions.

Extra Examples

We will see that there are closed formulae for counting the ways to distribute objects, distinguishable or indistinguishable, into distinguishable boxes. We are not so lucky when we count the ways to distribute objects, distinguishable or indistinguishable, into indistinguishable boxes; there are no closed formulae to use in these cases.

DISTINGUISHABLE OBJECTS AND DISTINGUISHABLE BOXES We first consider the case when distinguishable objects are placed into distinguishable boxes. Consider Example 8 in which the objects are cards and the boxes are hands of players.

EXAMPLE 8 How many ways are there to distribute hands of 5 cards to each of four players from the standard deck of 52 cards?

Solution: We will use the product rule to solve this problem. To begin, note that the first player can be dealt 5 cards in $C(52, 5)$ ways. The second player can be dealt 5 cards in $C(47, 5)$ ways, because only 47 cards are left. The third player can be dealt 5 cards in $C(42, 5)$ ways. Finally, the fourth player can be dealt 5 cards in $C(37, 5)$ ways. Hence, the total number of ways to deal four players 5 cards each is

$$\begin{aligned} C(52, 5)C(47, 5)C(42, 5)C(37, 5) &= \frac{52!}{47! 5!} \cdot \frac{47!}{42! 5!} \cdot \frac{42!}{37! 5!} \cdot \frac{37!}{32! 5!} \\ &= \frac{52!}{5! 5! 5! 5! 32!}. \end{aligned}$$



Remark: The solution to Example 8 equals the number of permutations of 52 objects, with 5 indistinguishable objects of each of four different types, and 32 objects of a fifth type. This equality can be seen by defining a one-to-one correspondence between permutations of this type and distributions of cards to the players. To define this correspondence, first order the cards from 1 to 52. Then cards dealt to the first player correspond to the cards in the positions assigned to objects of the first type in the permutation. Similarly, cards dealt to the second, third, and fourth players, respectively, correspond to cards in the positions assigned to objects of the second, third, and fourth type, respectively. The cards not dealt to any player correspond to cards in the positions assigned to objects of the fifth type. The reader should verify that this is a one-to-one correspondence.

Example 8 is a typical problem that involves distributing distinguishable objects into distinguishable boxes. The distinguishable objects are the 52 cards, and the five distinguishable boxes are the hands of the four players and the rest of the deck. Counting problems that involve distributing distinguishable objects into boxes can be solved using Theorem 4.

THEOREM 4

The number of ways to distribute n distinguishable objects into k distinguishable boxes so that n_i objects are placed into box i , $i = 1, 2, \dots, k$, equals

$$\frac{n!}{n_1! n_2! \cdots n_k!}.$$

Theorem 4 can be proved using the product rule. We leave the details as Exercise 47. It can also be proved (see Exercise 48) by setting up a one-to-one correspondence between the permutations counted by Theorem 3 and the ways to distribute objects counted by Theorem 4.

INDISTINGUISHABLE OBJECTS AND DISTINGUISHABLE BOXES Counting the number of ways of placing n indistinguishable objects into k distinguishable boxes turns out to be the same as counting the number of n -combinations for a set with k elements when repetitions are allowed. The reason behind this is that there is a one-to-one correspondence between

n -combinations from a set with k elements when repetition is allowed and the ways to place n indistinguishable balls into k distinguishable boxes. To set up this correspondence, we put a ball in the i th bin each time the i th element of the set is included in the n -combination.

EXAMPLE 9 How many ways are there to place 10 indistinguishable balls into eight distinguishable bins?

Solution: The number of ways to place 10 indistinguishable balls into eight bins equals the number of 10-combinations from a set with eight elements when repetition is allowed. Consequently, there are

$$C(8 + 10 - 1, 10) = C(17, 10) = \frac{17!}{10!7!} = 19,448.$$



This means that there are $C(n + r - 1, n - 1)$ ways to place r indistinguishable objects into n distinguishable boxes.



DISTINGUISHABLE OBJECTS AND INDISTINGUISHABLE BOXES Counting the ways to place n distinguishable objects into k indistinguishable boxes is more difficult than counting the ways to place objects, distinguishable or indistinguishable objects, into distinguishable boxes. We illustrate this with an example.

EXAMPLE 10 How many ways are there to put four different employees into three indistinguishable offices, when each office can contain any number of employees?

Solution: We will solve this problem by enumerating all the ways these employees can be placed into the offices. We represent the four employees by A, B, C , and D . First, we note that we can distribute employees so that all four are put into one office, three are put into one office and a fourth is put into a second office, two employees are put into one office and two put into a second office, and finally, two are put into one office, and one each put into the other two offices. Each way to distribute these employees to these offices can be represented by a way to partition the elements A, B, C , and D into disjoint subsets.

We can put all four employees into one office in exactly one way, represented by $\{\{A, B, C, D\}\}$. We can put three employees into one office and the fourth employee into a different office in exactly four ways, represented by $\{\{A, B, C\}, \{D\}\}$, $\{\{A, B, D\}, \{C\}\}$, $\{\{A, C, D\}, \{B\}\}$, and $\{\{B, C, D\}, \{A\}\}$. We can put two employees into one office and two into a second office in exactly three ways, represented by $\{\{A, B\}, \{C, D\}\}$, $\{\{A, C\}, \{B, D\}\}$, and $\{\{A, D\}, \{B, C\}\}$. Finally, we can put two employees into one office, and one each into each of the remaining two offices in six ways, represented by $\{\{A, B\}, \{C\}, \{D\}\}$, $\{\{A, C\}, \{B\}, \{D\}\}$, $\{\{A, D\}, \{B\}, \{C\}\}$, $\{\{B, C\}, \{A\}, \{D\}\}$, $\{\{B, D\}, \{A\}, \{C\}\}$, and $\{\{C, D\}, \{A\}, \{B\}\}$.

Counting all the possibilities, we find that there are 14 ways to put four different employees into three indistinguishable offices. Another way to look at this problem is to look at the number of offices into which we put employees. Note that there are six ways to put four different employees into three indistinguishable offices so that no office is empty, seven ways to put four different employees into two indistinguishable offices so that no office is empty, and one way to put four employees into one office so that it is not empty.



There is no simple closed formula for the number of ways to distribute n distinguishable objects into j indistinguishable boxes. However, there is a formula involving a summation, which we will now describe. Let $S(n, j)$ denote the number of ways to distribute n distinguishable objects into j indistinguishable boxes so that no box is empty. The numbers $S(n, j)$ are called **Stirling numbers of the second kind**. For instance, Example 10 shows that $S(4, 3) = 6$, $S(4, 2) = 7$, and $S(4, 1) = 1$. We see that the number of ways to distribute n distinguishable objects into k indistinguishable boxes (where the number of boxes that are nonempty equals k , $k - 1, \dots, 2$, or 1) equals $\sum_{j=1}^k S(n, j)$. For instance, following the reasoning in Example 10, the number of ways to distribute four distinguishable objects into three indistinguishable boxes

equals $S(4, 1) + S(4, 2) + S(4, 3) = 1 + 7 + 6 = 14$. Using the inclusion–exclusion principle (see Section 8.6) it can be shown that

$$S(n, j) = \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j-i)^n.$$

Consequently, the number of ways to distribute n distinguishable objects into k indistinguishable boxes equals

$$\sum_{j=1}^k S(n, j) = \sum_{j=1}^k \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j-i)^n.$$

Remark: The reader may be curious about the Stirling numbers of the first kind. A combinatorial definition of the **signless Stirling numbers of the first kind**, the absolute values of the Stirling numbers of the first kind, can be found in the preamble to Exercise 47 in the Supplementary Exercises. For the definition of Stirling numbers of the first kind, for more information about Stirling numbers of the second kind, and to learn more about Stirling numbers of the first kind and the relationship between Stirling numbers of the first and second kind, see combinatorics textbooks such as [B607], [Br99], and [RoTe05], and Chapter 6 in [MiRo91].

INDISTINGUISHABLE OBJECTS AND INDISTINGUISHABLE BOXES Some counting problems can be solved by determining the number of ways to distribute indistinguishable objects into indistinguishable boxes. We illustrate this principle with an example.

EXAMPLE 11 How many ways are there to pack six copies of the same book into four identical boxes, where a box can contain as many as six books?

Solution: We will enumerate all ways to pack the books. For each way to pack the books, we will list the number of books in the box with the largest number of books, followed by the numbers of books in each box containing at least one book, in order of decreasing number of books in a box. The ways we can pack the books are

- 6
- 5, 1
- 4, 2
- 4, 1, 1
- 3, 3
- 3, 2, 1
- 3, 1, 1, 1
- 2, 2, 2
- 2, 2, 1, 1.

For example, 4, 1, 1 indicates that one box contains four books, a second box contains a single book, and a third box contains a single book (and the fourth box is empty). We conclude that there are nine allowable ways to pack the books, because we have listed them all. 

Observe that distributing n indistinguishable objects into k indistinguishable boxes is the same as writing n as the sum of at most k positive integers in nonincreasing order. If $a_1 + a_2 + \dots + a_j = n$, where a_1, a_2, \dots, a_j are positive integers with $a_1 \geq a_2 \geq \dots \geq a_j$, we say that a_1, a_2, \dots, a_j is a **partition** of the positive integer n into j positive integers. We see that if $p_k(n)$ is the number of partitions of n into at most k positive integers, then there are $p_k(n)$ ways to distribute n indistinguishable objects into k indistinguishable boxes. No simple closed formula exists for this number. For more information about partitions of positive integers, see [Ro11].

Exercises

1. In how many different ways can five elements be selected in order from a set with three elements when repetition is allowed?
2. In how many different ways can five elements be selected in order from a set with five elements when repetition is allowed?
3. How many strings of six letters are there?
4. Every day a student randomly chooses a sandwich for lunch from a pile of wrapped sandwiches. If there are six kinds of sandwiches, how many different ways are there for the student to choose sandwiches for the seven days of a week if the order in which the sandwiches are chosen matters?
5. How many ways are there to assign three jobs to five employees if each employee can be given more than one job?
6. How many ways are there to select five unordered elements from a set with three elements when repetition is allowed?
7. How many ways are there to select three unordered elements from a set with five elements when repetition is allowed?
8. How many different ways are there to choose a dozen donuts from the 21 varieties at a donut shop?
9. A bagel shop has onion bagels, poppy seed bagels, egg bagels, salty bagels, pumpernickel bagels, sesame seed bagels, raisin bagels, and plain bagels. How many ways are there to choose
 - a) six bagels?
 - b) a dozen bagels?
 - c) two dozen bagels?
 - d) a dozen bagels with at least one of each kind?
 - e) a dozen bagels with at least three egg bagels and no more than two salty bagels?
10. A croissant shop has plain croissants, cherry croissants, chocolate croissants, almond croissants, apple croissants, and broccoli croissants. How many ways are there to choose
 - a) a dozen croissants?
 - b) three dozen croissants?
 - c) two dozen croissants with at least two of each kind?
 - d) two dozen croissants with no more than two broccoli croissants?
 - e) two dozen croissants with at least five chocolate croissants and at least three almond croissants?
 - f) two dozen croissants with at least one plain croissant, at least two cherry croissants, at least three chocolate croissants, at least one almond croissant, at least two apple croissants, and no more than three broccoli croissants?
11. How many ways are there to choose eight coins from a piggy bank containing 100 identical pennies and 80 identical nickels?
12. How many different combinations of pennies, nickels, dimes, quarters, and half dollars can a piggy bank contain if it has 20 coins in it?
13. A book publisher has 3000 copies of a discrete mathematics book. How many ways are there to store these books in their three warehouses if the copies of the book are indistinguishable?
14. How many solutions are there to the equation

$$x_1 + x_2 + x_3 + x_4 = 17,$$
 where x_1, x_2, x_3 , and x_4 are nonnegative integers?
15. How many solutions are there to the equation

$$x_1 + x_2 + x_3 + x_4 + x_5 = 21,$$
 where $x_i, i = 1, 2, 3, 4, 5$, is a nonnegative integer such that
 - a) $x_1 \geq 1$?
 - b) $x_i \geq 2$ for $i = 1, 2, 3, 4, 5$?
 - c) $0 \leq x_1 \leq 10$?
 - d) $0 \leq x_1 \leq 3, 1 \leq x_2 < 4$, and $x_3 \geq 15$?
16. How many solutions are there to the equation

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 29,$$
 where $x_i, i = 1, 2, 3, 4, 5, 6$, is a nonnegative integer such that
 - a) $x_i > 1$ for $i = 1, 2, 3, 4, 5, 6$?
 - b) $x_1 \geq 1, x_2 \geq 2, x_3 \geq 3, x_4 \geq 4, x_5 > 5$, and $x_6 \geq 6$?
 - c) $x_1 \leq 5$?
 - d) $x_1 < 8$ and $x_2 > 8$?
17. How many strings of 10 ternary digits (0, 1, or 2) are there that contain exactly two 0s, three 1s, and five 2s?
18. How many strings of 20-decimal digits are there that contain two 0s, four 1s, three 2s, one 3, two 4s, three 5s, two 7s, and three 9s?
19. Suppose that a large family has 14 children, including two sets of identical triplets, three sets of identical twins, and two individual children. How many ways are there to seat these children in a row of chairs if the identical triplets or twins cannot be distinguished from one another?
20. How many solutions are there to the inequality

$$x_1 + x_2 + x_3 \leq 11,$$
 where x_1, x_2 , and x_3 are nonnegative integers? [Hint: Introduce an auxiliary variable x_4 such that $x_1 + x_2 + x_3 + x_4 = 11$.]
21. How many ways are there to distribute six indistinguishable balls into nine distinguishable bins?
22. How many ways are there to distribute 12 indistinguishable balls into six distinguishable bins?
23. How many ways are there to distribute 12 distinguishable objects into six distinguishable boxes so that two objects are placed in each box?
24. How many ways are there to distribute 15 distinguishable objects into five distinguishable boxes so that the boxes have one, two, three, four, and five objects in them, respectively.

25. How many positive integers less than 1,000,000 have the sum of their digits equal to 19?
26. How many positive integers less than 1,000,000 have exactly one digit equal to 9 and have a sum of digits equal to 13?
27. There are 10 questions on a discrete mathematics final exam. How many ways are there to assign scores to the problems if the sum of the scores is 100 and each question is worth at least 5 points?
28. Show that there are $C(n + r - q_1 - q_2 - \dots - q_r - 1, n - q_1 - q_2 - \dots - q_r)$ different unordered selections of n objects of r different types that include at least q_1 objects of type one, q_2 objects of type two, ..., and q_r objects of type r .
29. How many different bit strings can be transmitted if the string must begin with a 1 bit, must include three additional 1 bits (so that a total of four 1 bits is sent), must include a total of 12 0 bits, and must have at least two 0 bits following each 1 bit?
30. How many different strings can be made from the letters in *MISSISSIPPI*, using all the letters?
31. How many different strings can be made from the letters in *ABRACADABRA*, using all the letters?
32. How many different strings can be made from the letters in *AARDVARK*, using all the letters, if all three As must be consecutive?
33. How many different strings can be made from the letters in *ORONO*, using some or all of the letters?
34. How many strings with five or more characters can be formed from the letters in *SEERESS*?
35. How many strings with seven or more characters can be formed from the letters in *EVERGREEN*?
36. How many different bit strings can be formed using six 1s and eight 0s?
37. A student has three mangos, two papayas, and two kiwi fruits. If the student eats one piece of fruit each day, and only the type of fruit matters, in how many different ways can these fruits be consumed?
38. A professor packs her collection of 40 issues of a mathematics journal in four boxes with 10 issues per box. How many ways can she distribute the journals if
 - each box is numbered, so that they are distinguishable?
 - the boxes are identical, so that they cannot be distinguished?
39. How many ways are there to travel in xyz space from the origin $(0, 0, 0)$ to the point $(4, 3, 5)$ by taking steps one unit in the positive x direction, one unit in the positive y direction, or one unit in the positive z direction? (Moving in the negative x , y , or z direction is prohibited, so that no backtracking is allowed.)
40. How many ways are there to travel in $xyzw$ space from the origin $(0, 0, 0, 0)$ to the point $(4, 3, 5, 4)$ by taking steps one unit in the positive x , positive y , positive z , or positive w direction?
41. How many ways are there to deal hands of seven cards to each of five players from a standard deck of 52 cards?
42. In bridge, the 52 cards of a standard deck are dealt to four players. How many different ways are there to deal bridge hands to four players?
43. How many ways are there to deal hands of five cards to each of six players from a deck containing 48 different cards?
44. In how many ways can a dozen books be placed on four distinguishable shelves
 - if the books are indistinguishable copies of the same title?
 - if no two books are the same, and the positions of the books on the shelves matter? [Hint: Break this into 12 tasks, placing each book separately. Start with the sequence 1, 2, 3, 4 to represent the shelves. Represent the books by $b_i, i = 1, 2, \dots, 12$. Place b_1 to the right of one of the terms in 1, 2, 3, 4. Then successively place b_2, b_3, \dots, b_{12} .]
45. How many ways can n books be placed on k distinguishable shelves
 - if the books are indistinguishable copies of the same title?
 - if no two books are the same, and the positions of the books on the shelves matter?
46. A shelf holds 12 books in a row. How many ways are there to choose five books so that no two adjacent books are chosen? [Hint: Represent the books that are chosen by bars and the books not chosen by stars. Count the number of sequences of five bars and seven stars so that no two bars are adjacent.]
- *47. Use the product rule to prove Theorem 4, by first placing objects in the first box, then placing objects in the second box, and so on.
- *48. Prove Theorem 4 by first setting up a one-to-one correspondence between permutations of n objects with n_i indistinguishable objects of type $i, i = 1, 2, 3, \dots, k$, and the distributions of n objects in k boxes such that n_i objects are placed in box $i, i = 1, 2, 3, \dots, k$ and then applying Theorem 3.
- *49. In this exercise we will prove Theorem 2 by setting up a one-to-one correspondence between the set of r -combinations with repetition allowed of $S = \{1, 2, 3, \dots, n\}$ and the set of r -combinations of the set $T = \{1, 2, 3, \dots, n+r-1\}$.
 - Arrange the elements in an r -combination, with repetition allowed, of S into an increasing sequence $x_1 \leq x_2 \leq \dots \leq x_r$. Show that the sequence formed by adding $k-1$ to the k th term is strictly increasing. Conclude that this sequence is made up of r distinct elements from T .
 - Show that the procedure described in (a) defines a one-to-one correspondence between the set of r -combinations, with repetition allowed, of S and the r -combinations of T . [Hint: Show the correspondence can be reversed by associating to the r -combination $\{x_1, x_2, \dots, x_r\}$ of T , with $1 \leq x_1 < x_2 < \dots < x_r \leq n+r-1$, the r -combination with

- repetition allowed from S , formed by subtracting $k - 1$ from the k th element.]
- c) Conclude that there are $C(n + r - 1, r)$ r -combinations with repetition allowed from a set with n elements.
50. How many ways are there to distribute five distinguishable objects into three indistinguishable boxes?
51. How many ways are there to distribute six distinguishable objects into four indistinguishable boxes so that each of the boxes contains at least one object?
52. How many ways are there to put five temporary employees into four identical offices?
53. How many ways are there to put six temporary employees into four identical offices so that there is at least one temporary employee in each of these four offices?
54. How many ways are there to distribute five indistinguishable objects into three indistinguishable boxes?
55. How many ways are there to distribute six indistinguishable objects into four indistinguishable boxes so that each of the boxes contains at least one object?
56. How many ways are there to pack eight identical DVDs into five indistinguishable boxes so that each box contains at least one DVD?
57. How many ways are there to pack nine identical DVDs into three indistinguishable boxes so that each box contains at least two DVDs?
58. How many ways are there to distribute five balls into seven boxes if each box must have at most one ball in it if
- both the balls and boxes are labeled?
 - the balls are labeled, but the boxes are unlabeled?
 - the balls are unlabeled, but the boxes are labeled?
 - both the balls and boxes are unlabeled?
59. How many ways are there to distribute five balls into three boxes if each box must have at least one ball in it if
- both the balls and boxes are labeled?
 - the balls are labeled, but the boxes are unlabeled?
- c) the balls are unlabeled, but the boxes are labeled?
- d) both the balls and boxes are unlabeled?
60. Suppose that a basketball league has 32 teams, split into two conferences of 16 teams each. Each conference is split into three divisions. Suppose that the North Central Division has five teams. Each of the teams in the North Central Division plays four games against each of the other teams in this division, three games against each of the 11 remaining teams in the conference, and two games against each of the 16 teams in the other conference. In how many different orders can the games of one of the teams in the North Central Division be scheduled?
- *61. Suppose that a weapons inspector must inspect each of five different sites twice, visiting one site per day. The inspector is free to select the order in which to visit these sites, but cannot visit site X, the most suspicious site, on two consecutive days. In how many different orders can the inspector visit these sites?
62. How many different terms are there in the expansion of $(x_1 + x_2 + \dots + x_m)^n$ after all terms with identical sets of exponents are added?
- *63. Prove the **Multinomial Theorem:** If n is a positive integer, then
- $$(x_1 + x_2 + \dots + x_m)^n = \sum_{n_1 + n_2 + \dots + n_m = n} C(n; n_1, n_2, \dots, n_m) x_1^{n_1} x_2^{n_2} \dots x_m^{n_m},$$
- where
- $$C(n; n_1, n_2, \dots, n_m) = \frac{n!}{n_1! n_2! \dots n_m!}$$
- is a **multinomial coefficient**.
64. Find the expansion of $(x + y + z)^4$.
65. Find the coefficient of $x^3 y^2 z^5$ in $(x + y + z)^{10}$.
66. How many terms are there in the expansion of $(x + y + z)^{100}$?

6.6 Generating Permutations and Combinations

Introduction

Methods for counting various types of permutations and combinations were described in the previous sections of this chapter, but sometimes permutations or combinations need to be generated, not just counted. Consider the following three problems. First, suppose that a salesperson must visit six different cities. In which order should these cities be visited to minimize total travel time? One way to determine the best order is to determine the travel time for each of the $6! = 720$ different orders in which the cities can be visited and choose the one with the smallest travel time. Second, suppose we are given a set of six positive integers and wish to find a subset of them that has 100 as their sum, if such a subset exists. One way to find these numbers is to generate all $2^6 = 64$ subsets and check the sum of their elements. Third, suppose a laboratory has 95 employees. A group of 12 of these employees with a particular set of 25 skills is needed for a project. (Each employee can have one or more of these skills.) One way to find such a

set of employees is to generate all sets of 12 of these employees and check whether they have the desired skills. These examples show that it is often necessary to generate permutations and combinations to solve problems.

Generating Permutations



Any set with n elements can be placed in one-to-one correspondence with the set $\{1, 2, 3, \dots, n\}$. We can list the permutations of any set of n elements by generating the permutations of the n smallest positive integers and then replacing these integers with the corresponding elements.

Many different algorithms have been developed to generate the $n!$ permutations of this set. We will describe one of these that is based on the **lexicographic** (or **dictionary**) **ordering** of the set of permutations of $\{1, 2, 3, \dots, n\}$. In this ordering, the permutation $a_1a_2 \cdots a_n$ precedes the permutation of $b_1b_2 \cdots b_n$, if for some k , with $1 \leq k \leq n$, $a_1 = b_1, a_2 = b_2, \dots, a_{k-1} = b_{k-1}$, and $a_k < b_k$. In other words, a permutation of the set of the n smallest positive integers precedes (in lexicographic order) a second permutation if the number in this permutation in the first position where the two permutations disagree is smaller than the number in that position in the second permutation.

EXAMPLE 1

The permutation 23415 of the set $\{1, 2, 3, 4, 5\}$ precedes the permutation 23514, because these permutations agree in the first two positions, but the number in the third position in the first permutation, 4, is smaller than the number in the third position in the second permutation, 5. Similarly, the permutation 41532 precedes 52143. 

An algorithm for generating the permutations of $\{1, 2, \dots, n\}$ can be based on a procedure that constructs the next permutation in lexicographic order following a given permutation $a_1a_2 \cdots a_n$. We will show how this can be done. First, suppose that $a_{n-1} < a_n$. Interchange a_{n-1} and a_n to obtain a larger permutation. No other permutation is both larger than the original permutation and smaller than the permutation obtained by interchanging a_{n-1} and a_n . For instance, the next larger permutation after 234156 is 234165. On the other hand, if $a_{n-1} > a_n$, then a larger permutation cannot be obtained by interchanging these last two terms in the permutation. Look at the last three integers in the permutation. If $a_{n-2} < a_{n-1}$, then the last three integers in the permutation can be rearranged to obtain the next largest permutation. Put the smaller of the two integers a_{n-1} and a_n that is greater than a_{n-2} in position $n - 2$. Then, place the remaining integer and a_{n-2} into the last two positions in increasing order. For instance, the next larger permutation after 234165 is 234516.

On the other hand, if $a_{n-2} > a_{n-1}$ (and $a_{n-1} > a_n$), then a larger permutation cannot be obtained by permuting the last three terms in the permutation. Based on these observations, a general method can be described for producing the next larger permutation in increasing order following a given permutation $a_1a_2 \cdots a_n$. First, find the integers a_j and a_{j+1} with $a_j < a_{j+1}$ and

$$a_{j+1} > a_{j+2} > \cdots > a_n,$$

that is, the last pair of adjacent integers in the permutation where the first integer in the pair is smaller than the second. Then, the next larger permutation in lexicographic order is obtained by putting in the j th position the least integer among a_{j+1}, a_{j+2}, \dots , and a_n that is greater than a_j and listing in increasing order the rest of the integers a_j, a_{j+1}, \dots, a_n in positions $j + 1$ to n . It is easy to see that there is no other permutation larger than the permutation $a_1a_2 \cdots a_n$ but smaller than the new permutation produced. (The verification of this fact is left as an exercise for the reader.)

EXAMPLE 2 What is the next permutation in lexicographic order after 362541?

Solution: The last pair of integers a_j and a_{j+1} where $a_j < a_{j+1}$ is $a_3 = 2$ and $a_4 = 5$. The least integer to the right of 2 that is greater than 2 in the permutation is $a_5 = 4$. Hence, 4 is placed in the third position. Then the integers 2, 5, and 1 are placed in order in the last three positions, giving 125 as the last three positions of the permutation. Hence, the next permutation is 364125.

To produce the $n!$ permutations of the integers $1, 2, 3, \dots, n$, begin with the smallest permutation in lexicographic order, namely, $123 \cdots n$, and successively apply the procedure described for producing the next larger permutation of $n! - 1$ times. This yields all the permutations of the n smallest integers in lexicographic order.

EXAMPLE 3 Generate the permutations of the integers 1, 2, 3 in lexicographic order.

Solution: Begin with 123. The next permutation is obtained by interchanging 3 and 2 to obtain 132. Next, because $3 > 2$ and $1 < 3$, permute the three integers in 132. Put the smaller of 3 and 2 in the first position, and then put 1 and 3 in increasing order in positions 2 and 3 to obtain 213. This is followed by 231, obtained by interchanging 1 and 3, because $1 < 3$. The next larger permutation has 3 in the first position, followed by 1 and 2 in increasing order, namely, 312. Finally, interchange 1 and 2 to obtain the last permutation, 321. We have generated the permutations of 1, 2, 3 in lexicographic order. They are 123, 132, 213, 231, 312, and 321.

Algorithm 1 displays the procedure for finding the next permutation in lexicographic order after a permutation that is not $n \ n - 1 \ n - 2 \ \dots \ 2 \ 1$, which is the largest permutation.

ALGORITHM 1 Generating the Next Permutation in Lexicographic Order.

```

procedure next permutation( $a_1a_2\dots a_n$ : permutation of
     $\{1, 2, \dots, n\}$  not equal to  $n \ n - 1 \ \dots \ 2 \ 1$ )
     $j := n - 1$ 
    while  $a_j > a_{j+1}$ 
         $j := j - 1$ 
        {j is the largest subscript with  $a_j < a_{j+1}$ }
     $k := n$ 
    while  $a_j > a_k$ 
         $k := k - 1$ 
        { $a_k$  is the smallest integer greater than  $a_j$  to the right of  $a_j$ }
    interchange  $a_j$  and  $a_k$ 
     $r := n$ 
     $s := j + 1$ 
    while  $r > s$ 
        interchange  $a_r$  and  $a_s$ 
         $r := r - 1$ 
         $s := s + 1$ 
{this puts the tail end of the permutation after the  $j$ th position in increasing order}
{ $a_1a_2\dots a_n$  is now the next permutation}

```

Generating Combinations



How can we generate all the combinations of the elements of a finite set? Because a combination is just a subset, we can use the correspondence between subsets of $\{a_1, a_2, \dots, a_n\}$ and bit strings of length n .

Recall that the bit string corresponding to a subset has a 1 in position k if a_k is in the subset, and has a 0 in this position if a_k is not in the subset. If all the bit strings of length n can be listed, then by the correspondence between subsets and bit strings, a list of all the subsets is obtained.

Recall that a bit string of length n is also the binary expansion of an integer between 0 and $2^n - 1$. The 2^n bit strings can be listed in order of their increasing size as integers in their binary expansions. To produce all binary expansions of length n , start with the bit string 000...00, with n zeros. Then, successively find the next expansion until the bit string 111...11 is obtained. At each stage the next binary expansion is found by locating the first position from the right that is not a 1, then changing all the 1s to the right of this position to 0s and making this first 0 (from the right) a 1.

EXAMPLE 4 Find the next bit string after 10 0010 0111.

Solution: The first bit from the right that is not a 1 is the fourth bit from the right. Change this bit to a 1 and change all the following bits to 0s. This produces the next larger bit string, 10 0010 1000.

The procedure for producing the next larger bit string after $b_{n-1}b_{n-2}\dots b_1b_0$ is given as Algorithm 2.

ALGORITHM 2 Generating the Next Larger Bit String.

```

procedure next bit string( $b_{n-1} b_{n-2}\dots b_1 b_0$ : bit string not equal to 11...11)
   $i := 0$ 
  while  $b_i = 1$ 
     $b_i := 0$ 
     $i := i + 1$ 
   $b_i := 1$ 
  { $b_{n-1} b_{n-2}\dots b_1 b_0$  is now the next bit string}

```

Next, an algorithm for generating the r -combinations of the set $\{1, 2, 3, \dots, n\}$ will be given. An r -combination can be represented by a sequence containing the elements in the subset in increasing order. The r -combinations can be listed using lexicographic order on these sequences. In this lexicographic ordering, the first r -combination is $\{1, 2, \dots, r-1, r\}$ and the last r -combination is $\{n-r+1, n-r+2, \dots, n-1, n\}$. The next r -combination after $a_1 a_2 \dots a_r$ can be obtained in the following way: First, locate the last element a_i in the sequence such that $a_i \neq n-r+i$. Then, replace a_i with $a_i + 1$ and a_j with $a_i + j - i + 1$, for $j = i + 1, i + 2, \dots, r$. It is left for the reader to show that this produces the next larger r -combination in lexicographic order. This procedure is illustrated with Example 5.

EXAMPLE 5 Find the next larger 4-combination of the set $\{1, 2, 3, 4, 5, 6\}$ after $\{1, 2, 5, 6\}$.

Solution: The last term among the terms a_i with $a_1 = 1, a_2 = 2, a_3 = 5$, and $a_4 = 6$ such that $a_i \neq 6 - 4 + i$ is $a_2 = 2$. To obtain the next larger 4-combination, increment a_2 by 1 to obtain $a_2 = 3$. Then set $a_3 = 3 + 1 = 4$ and $a_4 = 3 + 2 = 5$. Hence the next larger 4-combination is $\{1, 3, 4, 5\}$.

Algorithm 3 displays pseudocode for this procedure.

ALGORITHM 3 Generating the Next r -Combination in Lexicographic Order.

```

procedure next  $r$ -combination( $\{a_1, a_2, \dots, a_r\}$ ): proper subset of
     $\{1, 2, \dots, n\}$  not equal to  $\{n - r + 1, \dots, n\}$  with
     $a_1 < a_2 < \dots < a_r$ )
     $i := r$ 
    while  $a_i = n - r + i$ 
         $i := i - 1$ 
         $a_i := a_i + 1$ 
    for  $j := i + 1$  to  $r$ 
         $a_j := a_i + j - i$ 
    {  $\{a_1, a_2, \dots, a_r\}$  is now the next combination}

```

Exercises

1. Place these permutations of $\{1, 2, 3, 4, 5\}$ in lexicographic order: 43521, 15432, 45321, 23451, 23514, 14532, 21345, 45213, 31452, 31542.
2. Place these permutations of $\{1, 2, 3, 4, 5, 6\}$ in lexicographic order: 234561, 231456, 165432, 156423, 543216, 541236, 231465, 314562, 432561, 654321, 654312, 435612.
3. The name of a file in a computer directory consists of three uppercase letters followed by a digit, where each letter is either A, B, or C, and each digit is either 1 or 2. List the name of these files in lexicographic order, where we order letters using the usual alphabetic order of letters.
4. Suppose that the name of a file in a computer directory consists of three digits followed by two lowercase letters and each digit is 0, 1, or 2, and each letter is either a or b. List the name of these files in lexicographic order, where we order letters using the usual alphabetic order of letters.
5. Find the next larger permutation in lexicographic order after each of these permutations.

a) 1432	b) 54123	c) 12453
d) 45231	e) 6714235	f) 31528764
6. Find the next larger permutation in lexicographic order after each of these permutations.

a) 1342	b) 45321	c) 13245
d) 612345	e) 1623547	f) 23587416
7. Use Algorithm 1 to generate the 24 permutations of the first four positive integers in lexicographic order.
8. Use Algorithm 2 to list all the subsets of the set $\{1, 2, 3, 4\}$.
9. Use Algorithm 3 to list all the 3-combinations of $\{1, 2, 3, 4, 5\}$.

10. Show that Algorithm 1 produces the next larger permutation in lexicographic order.
11. Show that Algorithm 3 produces the next larger r -combination in lexicographic order after a given r -combination.
12. Develop an algorithm for generating the r -permutations of a set of n elements.

13. List all 3-permutations of $\{1, 2, 3, 4, 5\}$.

The remaining exercises in this section develop another algorithm for generating the permutations of $\{1, 2, 3, \dots, n\}$. This algorithm is based on Cantor expansions of integers. Every nonnegative integer less than $n!$ has a unique Cantor expansion

$$a_1 1! + a_2 2! + \dots + a_{n-1} (n-1)!$$

where a_i is a nonnegative integer not exceeding i , for $i = 1, 2, \dots, n-1$. The integers a_1, a_2, \dots, a_{n-1} are called the **Cantor digits** of this integer.

Given a permutation of $\{1, 2, \dots, n\}$, let $a_{k-1}, k = 2, 3, \dots, n$, be the number of integers less than k that follow k in the permutation. For instance, in the permutation 43215, a_1 is the number of integers less than 2 that follow 2, so $a_1 = 1$. Similarly, for this example $a_2 = 2$, $a_3 = 3$, and $a_4 = 0$. Consider the function from the set of permutations of $\{1, 2, 3, \dots, n\}$ to the set of nonnegative integers less than $n!$ that sends a permutation to the integer that has a_1, a_2, \dots, a_{n-1} , defined in this way, as its Cantor digits.

14. Find the Cantor digits a_1, a_2, \dots, a_{n-1} that correspond to these permutations.

a) 246531	b) 12345	c) 654321
------------------	-----------------	------------------
- *15. Show that the correspondence described in the preamble is a bijection between the set of permutations of $\{1, 2, 3, \dots, n\}$ and the nonnegative integers less than $n!$.

16. Find the permutations of $\{1, 2, 3, 4, 5\}$ that correspond to these integers with respect to the correspondence between Cantor expansions and permutations as described in the preamble to Exercise 14.
- a) 3 b) 89 c) 111
17. Develop an algorithm for producing all permutations of a set of n elements based on the correspondence described in the preamble to Exercise 14.

Key Terms and Results

TERMS

combinatorics: the study of arrangements of objects

enumeration: the counting of arrangements of objects

tree diagram: a diagram made up of a root, branches leaving the root, and other branches leaving some of the endpoints of branches

permutation: an ordered arrangement of the elements of a set
 r -permutation: an ordered arrangement of r elements of a set

$P(n,r)$: the number of r -permutations of a set with n elements

r -combination: an unordered selection of r elements of a set

$C(n,r)$: the number of r -combinations of a set with n elements

binomial coefficient $\binom{n}{r}$: also the number of r -combinations of a set with n elements

combinatorial proof: a proof that uses counting arguments rather than algebraic manipulation to prove a result

Pascal's triangle: a representation of the binomial coefficients where the i th row of the triangle contains $\binom{i}{j}$ for $j = 0, 1, 2, \dots, i$

$S(n,j)$: the Stirling number of the second kind denoting the number of ways to distribute n distinguishable objects into j indistinguishable boxes so that no box is empty

RESULTS

product rule for counting: The number of ways to do a procedure that consists of two tasks is the product of the number of ways to do the first task and the number of ways to do the second task after the first task has been done.

product rule for sets: The number of elements in the Cartesian product of finite sets is the product of the number of elements in each set.

sum rule for counting: The number of ways to do a task in one of two ways is the sum of the number of ways to do these tasks if they cannot be done simultaneously.

sum rule for sets: The number of elements in the union of pairwise disjoint finite sets is the sum of the numbers of elements in these sets.

subtraction rule for counting or inclusion-exclusion for sets: If a task can be done in either n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common to the two different ways.

subtraction rule or inclusion-exclusion for sets: The number of elements in the union of two sets is the sum of the number of elements in these sets minus the number of elements in their intersection.

division rule for counting: There are n/d ways to do a task if it can be done using a procedure that can be carried out in n ways, and for every way w , exactly d of the n ways correspond to way w .

division rule for sets: Suppose that a finite set A is the union of n disjoint subsets each with d elements. Then $n = |A|/d$.

the pigeonhole principle: When more than k objects are placed in k boxes, there must be a box containing more than one object.

the generalized pigeonhole principle: When N objects are placed in k boxes, there must be a box containing at least $\lceil N/k \rceil$ objects.

$$P(n, r) = \frac{n!}{(n-r)!}$$

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Pascal's identity: $\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$

the binomial theorem: $(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$

There are n^r r -permutations of a set with n elements when repetition is allowed.

There are $C(n+r-1, r)$ r -combinations of a set with n elements when repetition is allowed.

There are $n!/(n_1!n_2!\cdots n_k!)$ permutations of n objects of k types where there are n_i indistinguishable objects of type i for $i = 1, 2, 3, \dots, k$.

the algorithm for generating the permutations of the set $\{1, 2, \dots, n\}$

Review Questions

1. Explain how the sum and product rules can be used to find the number of bit strings with a length not exceeding 10.
2. Explain how to find the number of bit strings of length not exceeding 10 that have at least one 0 bit.
3.
 - a) How can the product rule be used to find the number of functions from a set with m elements to a set with n elements?
 - b) How many functions are there from a set with five elements to a set with 10 elements?

- c) How can the product rule be used to find the number of one-to-one functions from a set with m elements to a set with n elements?
- d) How many one-to-one functions are there from a set with five elements to a set with 10 elements?
- e) How many onto functions are there from a set with five elements to a set with 10 elements?
4. How can you find the number of possible outcomes of a playoff between two teams where the first team that wins four games wins the playoff?
5. How can you find the number of bit strings of length ten that either begin with 101 or end with 010?
6. a) State the pigeonhole principle.
b) Explain how the pigeonhole principle can be used to show that among any 11 integers, at least two must have the same last digit.
7. a) State the generalized pigeonhole principle.
b) Explain how the generalized pigeonhole principle can be used to show that among any 91 integers, there are at least ten that end with the same digit.
8. a) What is the difference between an r -combination and an r -permutation of a set with n elements?
b) Derive an equation that relates the number of r -combinations and the number of r -permutations of a set with n elements.
c) How many ways are there to select six students from a class of 25 to serve on a committee?
d) How many ways are there to select six students from a class of 25 to hold six different executive positions on a committee?
9. a) What is Pascal's triangle?
b) How can a row of Pascal's triangle be produced from the one above it?
10. What is meant by a combinatorial proof of an identity? How is such a proof different from an algebraic one?
11. Explain how to prove Pascal's identity using a combinatorial argument.
12. a) State the binomial theorem.
b) Explain how to prove the binomial theorem using a combinatorial argument.
c) Find the coefficient of $x^{100}y^{101}$ in the expansion of $(2x + 5y)^{201}$.
13. a) Explain how to find a formula for the number of ways to select r objects from n objects when repetition is allowed and order does not matter.
b) How many ways are there to select a dozen objects from among objects of five different types if objects of the same type are indistinguishable?
c) How many ways are there to select a dozen objects from these five different types if there must be at least three objects of the first type?
d) How many ways are there to select a dozen objects from these five different types if there cannot be more than four objects of the first type?
e) How many ways are there to select a dozen objects from these five different types if there must be at least two objects of the first type, but no more than three objects of the second type?
14. a) Let n and r be positive integers. Explain why the number of solutions of the equation $x_1 + x_2 + \dots + x_n = r$, where x_i is a nonnegative integer for $i = 1, 2, 3, \dots, n$, equals the number of r -combinations of a set with n elements.
b) How many solutions in nonnegative integers are there to the equation $x_1 + x_2 + x_3 + x_4 = 17$?
c) How many solutions in positive integers are there to the equation in part (b)?
15. a) Derive a formula for the number of permutations of n objects of k different types, where there are n_1 indistinguishable objects of type one, n_2 indistinguishable objects of type two, ..., and n_k indistinguishable objects of type k .
b) How many ways are there to order the letters of the word *INDISCREETNESS*?
16. Describe an algorithm for generating all the permutations of the set of the n smallest positive integers.
17. a) How many ways are there to deal hands of five cards to six players from a standard 52-card deck?
b) How many ways are there to distribute n distinguishable objects into k distinguishable boxes so that n_i objects are placed in box i ?
18. Describe an algorithm for generating all the combinations of the set of the n smallest positive integers.

Supplementary Exercises

1. How many ways are there to choose 6 items from 10 distinct items when
a) the items in the choices are ordered and repetition is not allowed?
b) the items in the choices are ordered and repetition is allowed?
c) the items in the choices are unordered and repetition is not allowed?
d) the items in the choices are unordered and repetition is allowed?
2. How many ways are there to choose 10 items from 6 distinct items when
a) the items in the choices are ordered and repetition is not allowed?
b) the items in the choices are ordered and repetition is allowed?
c) the items in the choices are unordered and repetition is not allowed?
d) the items in the choices are unordered and repetition is allowed?

3. A test contains 100 true/false questions. How many different ways can a student answer the questions on the test, if answers may be left blank?
4. How many strings of length 10 either start with 000 or end with 1111?
5. How many bit strings of length 10 over the alphabet $\{a, b, c\}$ have either exactly three *as* or exactly four *bs*?
6. The internal telephone numbers in the phone system on a campus consist of five digits, with the first digit not equal to zero. How many different numbers can be assigned in this system?
7. An ice cream parlor has 28 different flavors, 8 different kinds of sauce, and 12 toppings.
- a) In how many different ways can a dish of three scoops of ice cream be made where each flavor can be used more than once and the order of the scoops does not matter?
 - b) How many different kinds of small sundaes are there if a small sundae contains one scoop of ice cream, a sauce, and a topping?
 - c) How many different kinds of large sundaes are there if a large sundae contains three scoops of ice cream, where each flavor can be used more than once and the order of the scoops does not matter; two kinds of sauce, where each sauce can be used only once and the order of the sauces does not matter; and three toppings, where each topping can be used only once and the order of the toppings does not matter?
8. How many positive integers less than 1000
- a) have exactly three decimal digits?
 - b) have an odd number of decimal digits?
 - c) have at least one decimal digit equal to 9?
 - d) have no odd decimal digits?
 - e) have two consecutive decimal digits equal to 5?
 - f) are palindromes (that is, read the same forward and backward)?
9. When the numbers from 1 to 1000 are written out in decimal notation, how many of each of these digits are used?
- a) 0 b) 1 c) 2 d) 9
10. There are 12 signs of the zodiac. How many people are needed to guarantee that at least six of these people have the same sign?
11. A fortune cookie company makes 213 different fortunes. A student eats at a restaurant that uses fortunes from this company and gives each customer one fortune cookie at the end of a meal. What is the largest possible number of times that the student can eat at the restaurant without getting the same fortune four times?
12. How many people are needed to guarantee that at least two were born on the same day of the week and in the same month (perhaps in different years)?
13. Show that given any set of 10 positive integers not exceeding 50 there exist at least two different five-element subsets of this set that have the same sum.
14. A package of baseball cards contains 20 cards. How many packages must be purchased to ensure that two cards in these packages are identical if there are a total of 550 different cards?
15. a) How many cards must be chosen from a standard deck of 52 cards to guarantee that at least two of the four aces are chosen?
b) How many cards must be chosen from a standard deck of 52 cards to guarantee that at least two of the four aces and at least two of the 13 kinds are chosen?
c) How many cards must be chosen from a standard deck of 52 cards to guarantee that there are at least two cards of the same kind?
d) How many cards must be chosen from a standard deck of 52 cards to guarantee that there are at least two cards of each of two different kinds?
- *16. Show that in any set of $n + 1$ positive integers not exceeding $2n$ there must be two that are relatively prime.
- *17. Show that in a sequence of m integers there exists one or more consecutive terms with a sum divisible by m .
18. Show that if five points are picked in the interior of a square with a side length of 2, then at least two of these points are no farther than $\sqrt{2}$ apart.
19. Show that the decimal expansion of a rational number must repeat itself from some point onward.
20. Once a computer worm infects a personal computer via an infected e-mail message, it sends a copy of itself to 100 e-mail addresses it finds in the electronic message mailbox on this personal computer. What is the maximum number of different computers this one computer can infect in the time it takes for the infected message to be forwarded five times?
21. How many ways are there to choose a dozen donuts from 20 varieties
- a) if there are no two donuts of the same variety?
 - b) if all donuts are of the same variety?
 - c) if there are no restrictions?
 - d) if there are at least two varieties among the dozen donuts chosen?
 - e) if there must be at least six blueberry-filled donuts?
 - f) if there can be no more than six blueberry-filled donuts?
22. Find n if
- a) $P(n, 2) = 110$. b) $P(n, n) = 5040$.
 - c) $P(n, 4) = 12P(n, 2)$.
23. Find n if
- a) $C(n, 2) = 45$. b) $C(n, 3) = P(n, 2)$.
 - c) $C(n, 5) = C(n, 2)$.

- 24.** Show that if n and r are nonnegative integers and $n \geq r$, then
- $$P(n+1, r) = P(n, r)(n+1)/(n+1-r).$$
- *25.** Suppose that S is a set with n elements. How many ordered pairs (A, B) are there such that A and B are subsets of S with $A \subseteq B$? [Hint: Show that each element of S belongs to A , $B - A$, or $S - B$.]
- 26.** Give a combinatorial proof of Corollary 2 of Section 6.4 by setting up a correspondence between the subsets of a set with an even number of elements and the subsets of this set with an odd number of elements. [Hint: Take an element a in the set. Set up the correspondence by putting a in the subset if it is not already in it and taking it out if it is in the subset.]
- 27.** Let n and r be integers with $1 \leq r < n$. Show that
- $$C(n, r-1) = C(n+2, r+1) - 2C(n+1, r+1) + C(n, r+1).$$
- 28.** Prove using mathematical induction that $\sum_{j=2}^n C(j, 2) = C(n+1, 3)$ whenever n is an integer greater than 1.
- 29.** Show that if n is an integer then
- $$\sum_{k=0}^n 3^k \binom{n}{k} = 4^n.$$
- 30.** Show that $\sum_{i=1}^{n-1} \sum_{j=i+1}^n 1 = \binom{n}{2}$ if n is an integer with $n \geq 2$.
- 31.** Show that $\sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n 1 = \binom{n}{3}$ if n is an integer with $n \geq 3$.
- 32.** In this exercise we will derive a formula for the sum of the squares of the n smallest positive integers. We will count the number of triples (i, j, k) where i , j , and k are integers such that $0 \leq i < k$, $0 \leq j < k$, and $1 \leq k \leq n$ in two ways.
- a)** Show that there are k^2 such triples with a fixed k . Deduce that there are $\sum_{k=1}^n k^2$ such triples.
 - b)** Show that the number of such triples with $0 \leq i < j < k$ and the number of such triples with $0 \leq j < i < k$ both equal $C(n+1, 3)$.
 - c)** Show that the number of such triples with $0 \leq i = j < k$ equals $C(n+1, 2)$.
 - d)** Combining part (a) with parts (b) and (c), conclude that
- $$\begin{aligned} \sum_{k=1}^n k^2 &= 2C(n+1, 3) + C(n+1, 2) \\ &= n(n+1)(2n+1)/6. \end{aligned}$$
- *33.** How many bit strings of length n , where $n \geq 4$, contain exactly two occurrences of 01?
- 34.** Let S be a set. We say that a collection of subsets A_1, A_2, \dots, A_n each containing d elements, where $d \geq 2$, is *2-colorable* if it is possible to assign to each element of S one of two different colors so that

in every subset A_i there are elements that have been assigned each color. Let $m(d)$ be the largest integer such that every collection of fewer than $m(d)$ sets each containing d elements is 2-colorable.

- a)** Show that the collection of all subsets with d elements of a set S with $2d-1$ elements is not 2-colorable.
- b)** Show that $m(2) = 3$.
- **c)** Show that $m(3) = 7$. [Hint: Show that the collection $\{1, 3, 5\}, \{1, 2, 6\}, \{1, 4, 7\}, \{2, 3, 4\}, \{2, 5, 7\}, \{3, 6, 7\}, \{4, 5, 6\}$ is not 2-colorable. Then show that all collections of six sets with three elements each are 2-colorable.]
- 35.** A professor writes 20 multiple-choice questions, each with the possible answer a , b , c , or d , for a discrete mathematics test. If the number of questions with a , b , c , and d as their answer is 8, 3, 4, and 5, respectively, how many different answer keys are possible, if the questions can be placed in any order?
- 36.** How many different arrangements are there of eight people seated at a round table, where two arrangements are considered the same if one can be obtained from the other by a rotation?
- 37.** How many ways are there to assign 24 students to five faculty advisors?
- 38.** How many ways are there to choose a dozen apples from a bushel containing 20 indistinguishable Delicious apples, 20 indistinguishable Macintosh apples, and 20 indistinguishable Granny Smith apples, if at least three of each kind must be chosen?
- 39.** How many solutions are there to the equation $x_1 + x_2 + x_3 = 17$, where x_1 , x_2 , and x_3 are nonnegative integers with
 - a)** $x_1 > 1$, $x_2 > 2$, and $x_3 > 3$?
 - b)** $x_1 < 6$ and $x_3 > 5$?
 - c)** $x_1 < 4$, $x_2 < 3$, and $x_3 > 5$?
- 40.**
 - a)** How many different strings can be made from the word PEPPERCORN when all the letters are used?
 - b)** How many of these strings start and end with the letter P ?
 - c)** In how many of these strings are the three letter P s consecutive?
- 41.**
 - a)** How many subsets of a set with ten elements have fewer than five elements?
 - b)** How many subsets of a set with ten elements have more than seven elements?
 - c)** How many subsets of a set with ten elements have an odd number of elements?
- 42.** A witness to a hit-and-run accident tells the police that the license plate of the car in the accident, which contains three letters followed by three digits, starts with the letters AS and contains both the digits 1 and 2. How many different license plates can fit this description?
- 43.** How many ways are there to put n identical objects into m distinct containers so that no container is empty?
- 44.** How many ways are there to seat six boys and eight girls in a row of chairs so that no two boys are seated next to each other?

- 45.** How many ways are there to distribute six objects to five boxes if
- both the objects and boxes are labeled?
 - the objects are labeled, but the boxes are unlabeled?
 - the objects are unlabeled, but the boxes are labeled?
 - both the objects and the boxes are unlabeled?
- 46.** How many ways are there to distribute five objects into six boxes if
- both the objects and boxes are labeled?
 - the objects are labeled, but the boxes are unlabeled?
 - the objects are unlabeled, but the boxes are labeled?
 - both the objects and the boxes are unlabeled?

The **signless Stirling number of the first kind** $c(n, k)$, where k and n are integers with $1 \leq k \leq n$, equals the number of ways to arrange n people around k circular tables with at least one person seated at each table, where two seatings of m people around a circular table are considered the same if everyone has the same left neighbor and the same right neighbor.

- 47.** Find these signless Stirling numbers of the first kind.
- $c(3,2)$
 - $c(4,2)$
 - $c(4,3)$
 - $c(5,4)$
- 48.** Show that if n is a positive integer, then $\sum_{j=1}^n c(n, j) = n!$.
- 49.** Show that if n is a positive integer with $n \geq 3$, then $c(n, n-2) = (3n-1)C(n, 3)/4$.
- *50.** Show that if n and k are integers with $1 \leq k < n$, then $c(n+1, k) = c(n, k-1) + nc(n, k)$.
- 51.** Give a combinatorial proof that 2^n divides $n!$ whenever n is an even positive integer. [Hint: Use Theorem 3 in Section 6.5 to count the number of permutations of $2n$ objects where there are two indistinguishable objects of n different types.

- 52.** How many 11-element RNA sequences consist of 4 As, 3Cs, 2Us, and 2Gs, and end with CAA?

Exercises 53 and 54 are based on a discussion in [RoTe09]. A method used in the 1960s for sequencing RNA chains used enzymes to break chains after certain links. Some enzymes break RNA chains after each G link, while others break them after each C or U link. Using these enzymes it is sometimes possible to correctly sequence all the bases in an RNA chain.

- *53.** Suppose that when an enzyme that breaks RNA chains after each G link is applied to a 12-link chain, the fragments obtained are G, CCG, AAAG, and UCCG, and when an enzyme that breaks RNA chains after each C or U link is applied, the fragments obtained are C, C, C, C, GGU, and GAAAG. Can you determine the entire 12-link RNA chain from these two sets of fragments? If so, what is this RNA chain?
- *54.** Suppose that when an enzyme that breaks RNA chains after each G link is applied to a 12-link chain, the fragments obtained are AC, UG, and ACG and when an enzyme that breaks RNA chains after each C or U link is applied, the fragments obtained are U, GAC, and GAC. Can you determine the entire RNA chain from these two sets of fragments? If so, what is this RNA chain?
- 55.** Devise an algorithm for generating all the r -permutations of a finite set when repetition is allowed.
- 56.** Devise an algorithm for generating all the r -combinations of a finite set when repetition is allowed.
- *57.** Show that if m and n are integers with $m \geq 3$ and $n \geq 3$, then $R(m, n) \leq R(m, n-1) + R(m-1, n)$.
- *58.** Show that $R(3, 4) \geq 7$ by showing that in a group of six people, where any two people are friends or enemies, there are not necessarily three mutual friends or four mutual enemies.

Computer Projects

Write programs with these input and output.

- Given a positive integer n and a nonnegative integer not exceeding n , find the number of r -permutations and r -combinations of a set with n elements.
- Given positive integers n and r , find the number of r -permutations when repetition is allowed and r -combinations when repetition is allowed of a set with n elements.
- Given a sequence of positive integers, find the longest increasing and the longest decreasing subsequence of the sequence.
- Given an equation $x_1 + x_2 + \dots + x_n = C$, where C is a constant, and x_1, x_2, \dots, x_n are nonnegative integers, list all the solutions.
- Given a positive integer n , list all the permutations of the set $\{1, 2, 3, \dots, n\}$ in lexicographic order.
- Given a positive integer n and a nonnegative integer r not exceeding n , list all the r -combinations of the set $\{1, 2, 3, \dots, n\}$ in lexicographic order.
- Given a positive integer n and a nonnegative integer r not exceeding n , list all the r -permutations of the set $\{1, 2, 3, \dots, n\}$ in lexicographic order.
- Given a positive integer n , list all the combinations of the set $\{1, 2, 3, \dots, n\}$.
- Given positive integers n and r , list all the r -permutations, with repetition allowed, of the set $\{1, 2, 3, \dots, n\}$.
- Given positive integers n and r , list all the r -combinations, with repetition allowed, of the set $\{1, 2, 3, \dots, n\}$.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Find the number of possible outcomes in a two-team playoff when the winner is the first team to win 5 out of 9, 6 out of 11, 7 out of 13, and 8 out of 15.
2. Which binomial coefficients are odd? Can you formulate a conjecture based on numerical evidence?
3. Verify that $C(2n, n)$ is divisible by the square of a prime, when $n \neq 1, 2$, or 4 , for as many positive integers n as you can. [The theorem that tells that $C(2n, n)$ is divisible by the square of a prime with $n \neq 1, 2$, or 4 was proved in 1996 by Andrew Granville and Olivier Ramaré. Their proof settled a conjecture made in 1980 by Paul Erdős and Ron Graham.]
4. Find as many odd integers n less than 200 as you can for which $C(n, \lfloor n/2 \rfloor)$ is not divisible by the square of a prime. Formulate a conjecture based on your evidence.
- *5. For each integer less than 100 determine whether $C(2n, n)$ is divisible by 3. Can you formulate a conjecture that tells us for which integers n the binomial coefficient $C(2n, n)$ is divisible by 3 based on the digits in the base three expansion of n ?
6. Generate all the permutations of a set with eight elements.
7. Generate all the 6-permutations of a set with nine elements.
8. Generate all combinations of a set with eight elements.
9. Generate all 5-combinations with repetition allowed of a set with seven elements.

Writing Projects

Respond to these with essays using outside sources.

1. Describe some of the earliest uses of the pigeonhole principle by Dirichlet and other mathematicians.
2. Discuss ways in which the current telephone numbering plan can be extended to accommodate the rapid demand for more telephone numbers. (See if you can find some of the proposals coming from the telecommunications industry.) For each new numbering plan you discuss, show how to find the number of different telephone numbers it supports.
3. Discuss the importance of combinatorial reasoning in gene sequencing and related problems involving genomes.
4. Many combinatorial identities are described in this book. Find some sources of such identities and describe important combinatorial identities besides those already introduced in this book. Give some representative proofs, including combinatorial ones, of some of these identities.
5. Describe the different models used to model the distribution of particles in statistical mechanics, including Maxwell–Boltzmann, Bose–Einstein, and Fermi–Dirac statistics. In each case, describe the counting techniques used in the model.
6. Define the Stirling numbers of the first kind and describe some of their properties and the identities they satisfy.
7. Describe some of the properties and the identities that Stirling numbers of the second kind satisfy, including the connection between Stirling numbers of the first and second kinds.
8. Describe the latest discoveries of values and bounds for Ramsey numbers.
9. Describe additional ways to generate all the permutations of a set with n elements besides those found in Section 6.6. Compare these algorithms and the algorithms described in the text and exercises of Section 6.6 in terms of their computational complexity.
10. Describe at least one way to generate all the partitions of a positive integer n . (See Exercise 47 in Section 5.3.)

7

Discrete Probability

- 7.1** An Introduction to Discrete Probability
- 7.2** Probability Theory
- 7.3** Bayes' Theorem
- 7.4** Expected Value and Variance

Combinatorics and probability theory share common origins. The theory of probability was first developed more than 300 years ago, when certain gambling games were analyzed. Although probability theory was originally invented to study gambling, it now plays an essential role in a wide variety of disciplines. For example, probability theory is extensively applied in the study of genetics, where it can be used to help understand the inheritance of traits. Of course, probability still remains an extremely popular part of mathematics because of its applicability to gambling, which continues to be an extremely popular human endeavor.

In computer science, probability theory plays an important role in the study of the complexity of algorithms. In particular, ideas and techniques from probability theory are used to determine the average-case complexity of algorithms. Probabilistic algorithms can be used to solve many problems that cannot be easily or practically solved by deterministic algorithms. In a probabilistic algorithm, instead of always following the same steps when given the same input, as a deterministic algorithm does, the algorithm makes one or more random choices, which may lead to different output. In combinatorics, probability theory can even be used to show that objects with certain properties exist. The probabilistic method, a technique in combinatorics introduced by Paul Erdős and Alfréd Rényi, shows that an object with a specified property exists by showing that there is a positive probability that a randomly constructed object has this property. Probability theory can help us answer questions that involve uncertainty, such as determining whether we should reject an incoming mail message as spam based on the words that appear in the message.

7.1 An Introduction to Discrete Probability

Introduction

Probability theory dates back to 1526 when the Italian mathematician, physician, and gambler Girolamo Cardano wrote the first known systematic treatment of the subject in his book *Liber de Ludo Aleae* (*Book on Games of Chance*). (This book was not published until 1663, which may have held back the development of probability theory.) In the seventeenth century the French mathematician Blaise Pascal determined the odds of winning some popular bets based on the outcome when a pair of dice is repeatedly rolled. In the eighteenth century, the French mathematician Laplace, who also studied gambling, defined the probability of an event as the number of successful outcomes divided by the number of possible outcomes. For instance, the probability that a die comes up an odd number when it is rolled is the number of successful outcomes—namely, the number of ways it can come up odd—divided by the number of possible outcomes—namely, the number of different ways the die can come up. There are a total of six possible outcomes—namely, 1, 2, 3, 4, 5, and 6—and exactly three of these are successful outcomes—namely, 1, 3, and 5. Hence, the probability that the die comes up an odd number is $3/6 = 1/2$. (Note that it has been assumed that all possible outcomes are equally likely, or, in other words, that the die is fair.)

In this section we will restrict ourselves to experiments that have finitely many, equally likely, outcomes. This permits us to use Laplace's definition of the probability of an event. We will continue our study of probability in Section 7.2, where we will study experiments with finitely many outcomes that are not necessarily equally likely. In Section 7.2 we will also introduce

some key concepts in probability theory, including conditional probability, independence of events, and random variables. In Section 7.4 we will introduce the concepts of the expectation and variance of a random variable.

Finite Probability

An **experiment** is a procedure that yields one of a given set of possible outcomes. The **sample space** of the experiment is the set of possible outcomes. An **event** is a subset of the sample space. Laplace's definition of the probability of an event with finitely many possible outcomes will now be stated.

DEFINITION 1

If S is a finite nonempty sample space of equally likely outcomes, and E is an event, that is, a subset of S , then the *probability* of E is $p(E) = \frac{|E|}{|S|}$.

The probability of an event can never be negative or more than one!

According to Laplace's definition, the probability of an event is between 0 and 1. To see this, note that if E is an event from a finite sample space S , then $0 \leq |E| \leq |S|$, because $E \subseteq S$. Thus, $0 \leq p(E) = |E|/|S| \leq 1$.

Examples 1–7 illustrate how the probability of an event is found.

EXAMPLE 1

An urn contains four blue balls and five red balls. What is the probability that a ball chosen at random from the urn is blue?



Solution: To calculate the probability, note that there are nine possible outcomes, and four of these possible outcomes produce a blue ball. Hence, the probability that a blue ball is chosen is $4/9$.

EXAMPLE 2

What is the probability that when two dice are rolled, the sum of the numbers on the two dice is 7?

Solution: There are a total of 36 equally likely possible outcomes when two dice are rolled. (The product rule can be used to see this; because each die has six possible outcomes, the total



GIROLAMO CARDANO (1501–1576) Cardano, born in Pavia, Italy, was the illegitimate child of Fazio Cardano, a lawyer, mathematician, and friend of Leonardo da Vinci, and Chiara Micheria, a young widow. In spite of illness and poverty, Cardano was able to study at the universities of Pavia and Padua, from where he received his medical degree. Cardano was not accepted into Milan's College of Physicians because of his illegitimate birth, as well as his eccentricity and confrontational style. Nevertheless, his medical skills were highly regarded. One of his main accomplishments as a physician is the first description of typhoid fever.

Cardano published more than 100 books on a diverse range of subjects, including medicine, the natural sciences, mathematics, gambling, physical inventions and experiments, and astrology. He also wrote a fascinating autobiography. In mathematics, Cardano's book *Ars Magna*, published in 1545, established the foundations of abstract algebra. This was the most comprehensive book on abstract algebra for more than a century; it presents many novel ideas of Cardano and of others, including methods for solving cubic and quartic equations from their coefficients. Cardano also made several important contributions to cryptography. Cardano was an advocate of education for the deaf, believing, unlike his contemporaries, that deaf people could learn to read and write before learning to speak, and could use their minds just as well as hearing people.

Cardano was often short of money. However, he kept himself solvent through gambling and winning money by beating others at chess. His book about games of chance, *Liber de Ludo Aleae*, written in 1526 (but published in 1663), offers the first systematic treatment of probability; it also describes effective ways to cheat. Cardano was considered to be a man of dubious moral character; he was often described as a liar, gambler, lecher, and heretic.



number of outcomes when two dice are rolled is $6^2 = 36$.) There are six successful outcomes, namely, $(1, 6)$, $(2, 5)$, $(3, 4)$, $(4, 3)$, $(5, 2)$, and $(6, 1)$, where the values of the first and second dice are represented by an ordered pair. Hence, the probability that a seven comes up when two fair dice are rolled is $6/36 = 1/6$. \blacktriangleleft



Lotteries are extremely popular throughout the world. We can easily compute the odds of winning different types of lotteries, as illustrated in Examples 3 and 4. (The odd of winning the popular Mega Millions and Powerball lotteries are studied in the supplementary exercises.)

EXAMPLE 3

In a lottery, players win a large prize when they pick four digits that match, in the correct order, four digits selected by a random mechanical process. A smaller prize is won if only three digits are matched. What is the probability that a player wins the large prize? What is the probability that a player wins the small prize?

Solution: There is only one way to choose all four digits correctly. By the product rule, there are $10^4 = 10,000$ ways to choose four digits. Hence, the probability that a player wins the large prize is $1/10,000 = 0.0001$.

Players win the smaller prize when they correctly choose exactly three of the four digits. Exactly one digit must be wrong to get three digits correct, but not all four correct. By the sum rule, to find the number of ways to choose exactly three digits correctly, we add the number of ways to choose four digits matching the digits picked in all but the i th position, for $i = 1, 2, 3, 4$.

To count the number of successes with the first digit incorrect, note that there are nine possible choices for the first digit (all but the one correct digit), and one choice for each of the other digits, namely, the correct digits for these slots. Hence, there are nine ways to choose four digits where the first digit is incorrect, but the last three are correct. Similarly, there are nine ways to choose four digits where the second digit is incorrect, nine with the third digit incorrect, and nine with the fourth digit incorrect. Hence, there is a total of 36 ways to choose four digits with exactly three of the four digits correct. Thus, the probability that a player wins the smaller prize is $36/10,000 = 9/2500 = 0.0036$. \blacktriangleleft

EXAMPLE 4

There are many lotteries now that award enormous prizes to people who correctly choose a set of six numbers out of the first n positive integers, where n is usually between 30 and 60. What is the probability that a person picks the correct six numbers out of 40?

Solution: There is only one winning combination. The total number of ways to choose six numbers out of 40 is

$$C(40, 6) = \frac{40!}{34! 6!} = 3,838,380.$$

Consequently, the probability of picking a winning combination is $1/3,838,380 \approx 0.00000026$. (Here the symbol \approx means approximately equal to.) \blacktriangleleft



PIERRE-SIMON LAPLACE (1749–1827) Pierre-Simon Laplace came from humble origins in Normandy. In his childhood he was educated in a school run by the Benedictines. At 16 he entered the University of Caen intending to study theology. However, he soon realized his true interests were in mathematics. After completing his studies, he was named a provisional professor at Caen, and in 1769 he became professor of mathematics at the Paris Military School.

Laplace is best known for his contributions to celestial mechanics, the study of the motions of heavenly bodies. His *Traité de Mécanique Céleste* is considered one of the greatest scientific works of the early nineteenth century. Laplace was one of the founders of probability theory and made many contributions to mathematical statistics. His work in this area is documented in his book *Théorie Analytique des Probabilités*, in which he defined the probability of an event as the ratio of the number of favorable outcomes to the total number of outcomes of an experiment.

Laplace was famous for his political flexibility. He was loyal, in succession, to the French Republic, Napoleon, and King Louis XVIII. This flexibility permitted him to be productive before, during, and after the French Revolution.



Poker, and other card games, are growing in popularity. To win at these games it helps to know the probability of different hands. We can find the probability of specific hands that arise in card games using the techniques developed so far. A deck of cards contains 52 cards. There are 13 different kinds of cards, with four cards of each kind. (Among the terms commonly used instead of “kind” are “rank,” “face value,” “denomination,” and “value.”) These kinds are twos, threes, fours, fives, sixes, sevens, eights, nines, tens, jacks, queens, kings, and aces. There are also four suits: spades, clubs, hearts, and diamonds, each containing 13 cards, with one card of each kind in a suit. In many poker games, a hand consists of five cards.

EXAMPLE 5 Find the probability that a hand of five cards in poker contains four cards of one kind.

Solution: By the product rule, the number of hands of five cards with four cards of one kind is the product of the number of ways to pick one kind, the number of ways to pick the four of this kind out of the four in the deck of this kind, and the number of ways to pick the fifth card. This is

$$C(13, 1)C(4, 4)C(48, 1).$$

By Example 11 in Section 6.3 there are $C(52, 5)$ different hands of five cards. Hence, the probability that a hand contains four cards of one kind is

$$\frac{C(13, 1)C(4, 4)C(48, 1)}{C(52, 5)} = \frac{13 \cdot 1 \cdot 48}{2,598,960} \approx 0.00024.$$

EXAMPLE 6 What is the probability that a poker hand contains a full house, that is, three of one kind and two of another kind?

Solution: By the product rule, the number of hands containing a full house is the product of the number of ways to pick two kinds in order, the number of ways to pick three out of four for the first kind, and the number of ways to pick two out of four for the second kind. (Note that the order of the two kinds matters, because, for instance, three queens and two aces is different from three aces and two queens.) We see that the number of hands containing a full house is

$$P(13, 2)C(4, 3)C(4, 2) = 13 \cdot 12 \cdot 4 \cdot 6 = 3744.$$

Because there are $C(52, 5) = 2,598,960$ poker hands, the probability of a full house is

$$\frac{3744}{2,598,960} \approx 0.0014.$$

EXAMPLE 7 What is the probability that the numbers 11, 4, 17, 39, and 23 are drawn in that order from a bin containing 50 balls labeled with the numbers 1, 2, ..., 50 if (a) the ball selected is not returned to the bin before the next ball is selected and (b) the ball selected is returned to the bin before the next ball is selected?

Solution: (a) By the product rule, there are $50 \cdot 49 \cdot 48 \cdot 47 \cdot 46 = 254,251,200$ ways to select the balls because each time a ball is drawn there is one fewer ball to choose from. Consequently, the probability that 11, 4, 17, 39, and 23 are drawn in that order is $1/254,251,200$. This is an example of **sampling without replacement**.

(b) By the product rule, there are $50^5 = 312,500,000$ ways to select the balls because there are 50 possible balls to choose from each time a ball is drawn. Consequently, the probability that 11, 4, 17, 39, and 23 are drawn in that order is $1/312,500,000$. This is an example of **sampling with replacement**.

Probabilities of Complements and Unions of Events

We can use counting techniques to find the probability of events derived from other events.

THEOREM 1

Let E be an event in a sample space S . The probability of the event $\bar{E} = S - E$, the complementary event of E , is given by

$$p(\bar{E}) = 1 - p(E).$$

Proof: To find the probability of the event $\bar{E} = S - E$, note that $|\bar{E}| = |S| - |E|$. Hence,

$$p(\bar{E}) = \frac{|S| - |E|}{|S|} = 1 - \frac{|E|}{|S|} = 1 - p(E).$$



There is an alternative strategy for finding the probability of an event when a direct approach does not work well. Instead of determining the probability of the event, the probability of its complement can be found. This is often easier to do, as Example 8 shows.

EXAMPLE 8

A sequence of 10 bits is randomly generated. What is the probability that at least one of these bits is 0?

Solution: Let E be the event that at least one of the 10 bits is 0. Then \bar{E} is the event that all the bits are 1s. Because the sample space S is the set of all bit strings of length 10, it follows that

$$\begin{aligned} p(E) &= 1 - p(\bar{E}) = 1 - \frac{|\bar{E}|}{|S|} = 1 - \frac{1}{2^{10}} \\ &= 1 - \frac{1}{1024} = \frac{1023}{1024}. \end{aligned}$$

Hence, the probability that the bit string will contain at least one 0 bit is 1023/1024. It is quite difficult to find this probability directly without using Theorem 1.



We can also find the probability of the union of two events.

THEOREM 2

Let E_1 and E_2 be events in the sample space S . Then

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2).$$

Proof: Using the formula given in Section 2.2 for the number of elements in the union of two sets, it follows that

$$|E_1 \cup E_2| = |E_1| + |E_2| - |E_1 \cap E_2|.$$

Hence,

$$\begin{aligned}
 p(E_1 \cup E_2) &= \frac{|E_1 \cup E_2|}{|S|} \\
 &= \frac{|E_1| + |E_2| - |E_1 \cap E_2|}{|S|} \\
 &= \frac{|E_1|}{|S|} + \frac{|E_2|}{|S|} - \frac{|E_1 \cap E_2|}{|S|} \\
 &= p(E_1) + p(E_2) - p(E_1 \cap E_2).
 \end{aligned}$$



EXAMPLE 9



What is the probability that a positive integer selected at random from the set of positive integers not exceeding 100 is divisible by either 2 or 5?

Solution: Let E_1 be the event that the integer selected at random is divisible by 2, and let E_2 be the event that it is divisible by 5. Then $E_1 \cup E_2$ is the event that it is divisible by either 2 or 5. Also, $E_1 \cap E_2$ is the event that it is divisible by both 2 and 5, or equivalently, that it is divisible by 10. Because $|E_1| = 50$, $|E_2| = 20$, and $|E_1 \cap E_2| = 10$, it follows that

$$\begin{aligned}
 p(E_1 \cup E_2) &= p(E_1) + p(E_2) - p(E_1 \cap E_2) \\
 &= \frac{50}{100} + \frac{20}{100} - \frac{10}{100} = \frac{3}{5}.
 \end{aligned}$$



Probabilistic Reasoning

A common problem is determining which of two events is more likely. Analyzing the probabilities of such events can be tricky. Example 10 describes a problem of this type. It discusses a famous problem originating with the television game show *Let's Make a Deal* and named after the host of the show, Monty Hall.

EXAMPLE 10



The Monty Hall Three-Door Puzzle Suppose you are a game show contestant. You have a chance to win a large prize. You are asked to select one of three doors to open; the large prize is behind one of the three doors and the other two doors are losers. Once you select a door, the game show host, who knows what is behind each door, does the following. First, whether or not you selected the winning door, he opens one of the other two doors that he knows is a losing door (selecting at random if both are losing doors). Then he asks you whether you would like to switch doors. Which strategy should you use? Should you change doors or keep your original selection, or does it not matter?

Solution: The probability you select the correct door (before the host opens a door and asks you whether you want to change) is $1/3$, because the three doors are equally likely to be the correct door. The probability this is the correct door does not change once the game show host opens one of the other doors, because he will always open a door that the prize is not behind.

The probability that you selected incorrectly is the probability the prize is behind one of the two doors you did not select. Consequently, the probability that you selected incorrectly is $2/3$. If you selected incorrectly, when the game show host opens a door to show you that the prize is not behind it, the prize is behind the other door. You will always win if your initial choice was incorrect and you change doors. So, by changing doors, the probability you win is $2/3$. In other words, you should always change doors when given the chance to do so by the game show host. This doubles the probability that you will win. (A more rigorous treatment of this puzzle can be found in Exercise 15 of Section 7.3. For much more on this famous puzzle and its variations, see [Ro09].)



Exercises

1. What is the probability that a card selected at random from a standard deck of 52 cards is an ace?
2. What is the probability that a fair die comes up six when it is rolled?
3. What is the probability that a randomly selected integer chosen from the first 100 positive integers is odd?
4. What is the probability that a randomly selected day of a leap year (with 366 possible days) is in April?
5. What is the probability that the sum of the numbers on two dice is even when they are rolled?
6. What is the probability that a card selected at random from a standard deck of 52 cards is an ace or a heart?
7. What is the probability that when a coin is flipped six times in a row, it lands heads up every time?
8. What is the probability that a five-card poker hand contains the ace of hearts?
9. What is the probability that a five-card poker hand does not contain the queen of hearts?
10. What is the probability that a five-card poker hand contains the two of diamonds and the three of spades?
11. What is the probability that a five-card poker hand contains the two of diamonds, the three of spades, the six of hearts, the ten of clubs, and the king of hearts?
12. What is the probability that a five-card poker hand contains exactly one ace?
13. What is the probability that a five-card poker hand contains at least one ace?
14. What is the probability that a five-card poker hand contains cards of five different kinds?
15. What is the probability that a five-card poker hand contains two pairs (that is, two of each of two different kinds and a fifth card of a third kind)?
16. What is the probability that a five-card poker hand contains a flush, that is, five cards of the same suit?
17. What is the probability that a five-card poker hand contains a straight, that is, five cards that have consecutive kinds? (Note that an ace can be considered either the lowest card of an A-2-3-4-5 straight or the highest card of a 10-J-Q-K-A straight.)
18. What is the probability that a five-card poker hand contains a straight flush, that is, five cards of the same suit of consecutive kinds?
- *19. What is the probability that a five-card poker hand contains cards of five different kinds and does not contain a flush or a straight?
20. What is the probability that a five-card poker hand contains a royal flush, that is, the 10, jack, queen, king, and ace of one suit?
21. What is the probability that a fair die never comes up an even number when it is rolled six times?
22. What is the probability that a positive integer not exceeding 100 selected at random is divisible by 3?
23. What is the probability that a positive integer not exceeding 100 selected at random is divisible by 5 or 7?
24. Find the probability of winning a lottery by selecting the correct six integers, where the order in which these integers are selected does not matter, from the positive integers not exceeding
 - a) 30.
 - b) 36.
 - c) 42.
 - d) 48.
25. Find the probability of winning a lottery by selecting the correct six integers, where the order in which these integers are selected does not matter, from the positive integers not exceeding
 - a) 50.
 - b) 52.
 - c) 56.
 - d) 60.
26. Find the probability of selecting none of the correct six integers in a lottery, where the order in which these integers are selected does not matter, from the positive integers not exceeding
 - a) 40.
 - b) 48.
 - c) 56.
 - d) 64.
27. Find the probability of selecting exactly one of the correct six integers in a lottery, where the order in which these integers are selected does not matter, from the positive integers not exceeding
 - a) 40.
 - b) 48.
 - c) 56.
 - d) 64.
28. In a superlottery, a player selects 7 numbers out of the first 80 positive integers. What is the probability that a person wins the grand prize by picking 7 numbers that are among the 11 numbers selected at random by a computer.
29. In a superlottery, players win a fortune if they choose the eight numbers selected by a computer from the positive integers not exceeding 100. What is the probability that a player wins this superlottery?
30. What is the probability that a player of a lottery wins the prize offered for correctly choosing five (but not six) numbers out of six integers chosen at random from the integers between 1 and 40, inclusive?
31. Suppose that 100 people enter a contest and that different winners are selected at random for first, second, and third prizes. What is the probability that Michelle wins one of these prizes if she is one of the contestants?
32. Suppose that 100 people enter a contest and that different winners are selected at random for first, second, and third prizes. What is the probability that Kumar, Janice, and Pedro each win a prize if each has entered the contest?
33. What is the probability that Abby, Barry, and Sylvia win the first, second, and third prizes, respectively, in a drawing if 200 people enter a contest and
 - a) no one can win more than one prize.
 - b) winning more than one prize is allowed.
34. What is the probability that Bo, Colleen, Jeff, and Rohini win the first, second, third, and fourth prizes, respectively, in a drawing if 50 people enter a contest and
 - a) no one can win more than one prize.
 - b) winning more than one prize is allowed.

- 35.** In roulette, a wheel with 38 numbers is spun. Of these, 18 are red, and 18 are black. The other two numbers, which are neither black nor red, are 0 and 00. The probability that when the wheel is spun it lands on any particular number is 1/38.
- What is the probability that the wheel lands on a red number?
 - What is the probability that the wheel lands on a black number twice in a row?
 - What is the probability that the wheel lands on 0 or 00?
 - What is the probability that in five spins the wheel never lands on either 0 or 00?
 - What is the probability that the wheel lands on one of the first six integers on one spin, but does not land on any of them on the next spin?
- 36.** Which is more likely: rolling a total of 8 when two dice are rolled or rolling a total of 8 when three dice are rolled?
- 37.** Which is more likely: rolling a total of 9 when two dice are rolled or rolling a total of 9 when three dice are rolled?
- 38.** Two events E_1 and E_2 are called **independent** if $p(E_1 \cap E_2) = p(E_1)p(E_2)$. For each of the following pairs of events, which are subsets of the set of all possible outcomes when a coin is tossed three times, determine whether or not they are independent.
- E_1 : tails comes up with the coin is tossed the first time; E_2 : heads comes up when the coin is tossed the second time.
 - E_1 : the first coin comes up tails; E_2 : two, and not three, heads come up in a row.
 - E_1 : the second coin comes up tails; E_2 : two, and not three, heads come up in a row.
- (We will study independence of events in more depth in Section 7.2.)
- 39.** Explain what is wrong with the statement that in the Monty Hall Three-Door Puzzle the probability that the prize is behind the first door you select and the probability that the prize is behind the other of the two doors that Monty does not open are both 1/2, because there are two doors left.
- 40.** Suppose that instead of three doors, there are four doors in the Monty Hall puzzle. What is the probability that you win by not changing once the host, who knows what is behind each door, opens a losing door and gives you the chance to change doors? What is the probability that you win by changing the door you select to one of the two remaining doors among the three that you did not select?
- 41.** This problem was posed by the Chevalier de Méré and was solved by Blaise Pascal and Pierre de Fermat.
- Find the probability of rolling at least one six when a fair die is rolled four times.
 - Find the probability that a double six comes up at least once when a pair of dice is rolled 24 times. Answer the query the Chevalier de Méré made to Pascal asking whether this probability was greater than 1/2.
 - Is it more likely that a six comes up at least once when a fair die is rolled four times or that a double six comes up at least once when a pair of dice is rolled 24 times?

7.2 Probability Theory

Introduction



In Section 7.1 we introduced the notion of the probability of an event. (Recall that an event is a subset of the possible outcomes of an experiment.) We defined the probability of an event E as Laplace did, that is,

$$p(E) = \frac{|E|}{|S|},$$

the number of outcomes in E divided by the total number of outcomes. This definition assumes that all outcomes are equally likely. However, many experiments have outcomes that are not equally likely. For instance, a coin may be biased so that it comes up heads twice as often as tails. Similarly, the likelihood that the input of a linear search is a particular element in a list, or is not in the list, depends on how the input is generated. How can we model the likelihood of events in such situations? In this section we will show how to define probabilities of outcomes to study probabilities of experiments where outcomes may not be equally likely.

Suppose that a fair coin is flipped four times, and the first time it comes up heads. Given this information, what is the probability that heads comes up three times? To answer this and

similar questions, we will introduce the concept of *conditional probability*. Does knowing that the first flip comes up heads change the probability that heads comes up three times? If not, these two events are called *independent*, a concept studied later in this section.

Many questions address a particular numerical value associated with the outcome of an experiment. For instance, when we flip a coin 100 times, what is the probability that exactly 40 heads appear? How many heads should we expect to appear? In this section we will introduce *random variables*, which are functions that associate numerical values to the outcomes of experiments.

Assigning Probabilities

Let S be the sample space of an experiment with a finite or countable number of outcomes. We assign a probability $p(s)$ to each outcome s . We require that two conditions be met:

$$(i) \quad 0 \leq p(s) \leq 1 \text{ for each } s \in S$$

and

$$(ii) \quad \sum_{s \in S} p(s) = 1.$$

Condition (i) states that the probability of each outcome is a nonnegative real number no greater than 1. Condition (ii) states that the sum of the probabilities of all possible outcomes should be 1; that is, when we do the experiment, it is a certainty that one of these outcomes occurs. (Note that when the sample space is infinite, $\sum_{s \in S} p(s)$ is a convergent infinite series.) This is a generalization of Laplace's definition in which each of n outcomes is assigned a probability of $1/n$. Indeed, conditions (i) and (ii) are met when Laplace's definition of probabilities of equally likely outcomes is used and S is finite. (See Exercise 4.)

Note that when there are n possible outcomes, x_1, x_2, \dots, x_n , the two conditions to be met are

$$(i) \quad 0 \leq p(x_i) \leq 1 \text{ for } i = 1, 2, \dots, n$$

and

$$(ii) \quad \sum_{i=1}^n p(x_i) = 1.$$

The function p from the set of all outcomes of the sample space S is called a **probability distribution**.

To model an experiment, the probability $p(s)$ assigned to an outcome s should equal the limit of the number of times s occurs divided by the number of times the experiment is performed, as this number grows without bound. (We will assume that all experiments discussed have outcomes that are predictable on the average, so that this limit exists. We also assume that the outcomes of successive trials of an experiment do not depend on past results.)



HISTORICAL NOTE The Chevalier de Méré was a French nobleman, a famous gambler, and a bon vivant. He was successful at making bets with odds slightly greater than 1/2 (such as having at least one six come up in four tosses of a fair die). His correspondence with Pascal asking about the probability of having at least one double six come up when a pair of dice is rolled 24 times led to the development of probability theory. According to one account, Pascal wrote to Fermat about the Chevalier saying something like "He's a good guy but, alas, he's no mathematician."

Remark: We will not discuss probabilities of events when the set of outcomes is not finite or countable, such as when the outcome of an experiment can be any real number. In such cases, integral calculus is usually required for the study of the probabilities of events.

We can model experiments in which outcomes are either equally likely or not equally likely by choosing the appropriate function $p(s)$, as Example 1 illustrates.

EXAMPLE 1 What probabilities should we assign to the outcomes H (heads) and T (tails) when a fair coin is flipped? What probabilities should be assigned to these outcomes when the coin is biased so that heads comes up twice as often as tails?

Solution: For a fair coin, the probability that heads comes up when the coin is flipped equals the probability that tails comes up, so the outcomes are equally likely. Consequently, we assign the probability $1/2$ to each of the two possible outcomes, that is, $p(H) = p(T) = 1/2$.

For the biased coin we have

$$p(H) = 2p(T).$$

Because

$$p(H) + p(T) = 1,$$

it follows that

$$2p(T) + p(T) = 3p(T) = 1.$$

We conclude that $p(T) = 1/3$ and $p(H) = 2/3$. 

DEFINITION 1

Suppose that S is a set with n elements. The *uniform distribution* assigns the probability $1/n$ to each element of S .

We now define the probability of an event as the sum of the probabilities of the outcomes in this event.

DEFINITION 2

The *probability* of the event E is the sum of the probabilities of the outcomes in E . That is,

$$p(E) = \sum_{s \in E} p(s).$$

(Note that when E is an infinite set, $\sum_{s \in E} p(s)$ is a convergent infinite series.)

Note that when there are n outcomes in the event E , that is, if $E = \{a_1, a_2, \dots, a_n\}$, then $p(E) = \sum_{i=1}^n p(a_i)$. Note also that the uniform distribution assigns the same probability to an event that Laplace's original definition of probability assigns to this event. The experiment of selecting an element from a sample space with a uniform distribution is called selecting an element of S **at random**.

EXAMPLE 2

Suppose that a die is biased (or loaded) so that 3 appears twice as often as each other number but that the other five outcomes are equally likely. What is the probability that an odd number appears when we roll this die?

Solution: We want to find the probability of the event $E = \{1, 3, 5\}$. By Exercise 2, we have

$$p(1) = p(2) = p(4) = p(5) = p(6) = 1/7; p(3) = 2/7.$$

It follows that

$$p(E) = p(1) + p(3) + p(5) = 1/7 + 2/7 + 1/7 = 4/7. \quad \blacktriangleleft$$

When possible outcomes are equally likely and there are a finite number of possible outcomes, the definition of the probability of an event given in this section (Definition 2) agrees with Laplace's definition (Definition 1 of Section 7.1). To see this, suppose that there are n equally likely outcomes; each possible outcome has probability $1/n$, because the sum of their probabilities is 1. Suppose the event E contains m outcomes. According to Definition 2,

$$p(E) = \sum_{i=1}^m \frac{1}{n} = \frac{m}{n}.$$

Because $|E| = m$ and $|S| = n$, it follows that

$$p(E) = \frac{m}{n} = \frac{|E|}{|S|}.$$

This is Laplace's definition of the probability of the event E .

Probabilities of Complements and Unions of Events

The formulae for probabilities of combinations of events in Section 7.1 continue to hold when we use Definition 2 to define the probability of an event. For example, Theorem 1 of Section 7.1 asserts that

$$p(\bar{E}) = 1 - p(E),$$

where \bar{E} is the complementary event of the event E . This equality also holds when Definition 2 is used. To see this, note that because the sum of the probabilities of the n possible outcomes is 1, and each outcome is either in E or in \bar{E} , but not in both, we have

$$\sum_{s \in S} p(s) = 1 = p(E) + p(\bar{E}).$$

Hence, $p(\bar{E}) = 1 - p(E)$.

Under Laplace's definition, by Theorem 2 in Section 7.1, we have

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2)$$

whenever E_1 and E_2 are events in a sample space S . This also holds when we define the probability of an event as we do in this section. To see this, note that $p(E_1 \cup E_2)$ is the sum of the probabilities of the outcomes in $E_1 \cup E_2$. When an outcome x is in one, but not both, of E_1 and E_2 , $p(x)$ occurs in exactly one of the sums for $p(E_1)$ and $p(E_2)$. When an outcome x is in both E_1 and E_2 , $p(x)$ occurs in the sum for $p(E_1)$, in the sum for $p(E_2)$, and in the sum for $p(E_1 \cap E_2)$, so it occurs $1 + 1 - 1 = 1$ time on the right-hand side. Consequently, the left-hand side and right-hand side are equal.

Also, note that if the events E_1 and E_2 are disjoint, then $p(E_1 \cap E_2) = 0$, which implies that

$$p(E_1 \cup E_2) = p(E_1) + p(E_2) - p(E_1 \cap E_2) = p(E_1) + p(E_2).$$

Theorem 1 generalizes this last formula by providing a formula for the probability of the union of pairwise disjoint events.

THEOREM 1

If E_1, E_2, \dots is a sequence of pairwise disjoint events in a sample space S , then

$$p\left(\bigcup_i E_i\right) = \sum_i p(E_i).$$

(Note that this theorem applies when the sequence E_1, E_2, \dots consists of a finite number or a countably infinite number of pairwise disjoint events.)

We leave the proof of Theorem 1 to the reader (see Exercises 36 and 37).

Conditional Probability



Suppose that we flip a coin three times, and all eight possibilities are equally likely. Moreover, suppose we know that the event F , that the first flip comes up tails, occurs. Given this information, what is the probability of the event E , that an odd number of tails appears? Because the first flip comes up tails, there are only four possible outcomes: TTT , TTH , THT , and THH , where H and T represent heads and tails, respectively. An odd number of tails appears only for the outcomes TTT and THH . Because the eight outcomes have equal probability, each of the four possible outcomes, given that F occurs, should also have an equal probability of $1/4$. This suggests that we should assign the probability of $2/4 = 1/2$ to E , given that F occurs. This probability is called the **conditional probability** of E given F .

In general, to find the conditional probability of E given F , we use F as the sample space. For an outcome from E to occur, this outcome must also belong to $E \cap F$. With this motivation, we make Definition 3.

DEFINITION 3

Let E and F be events with $p(F) > 0$. The *conditional probability* of E given F , denoted by $p(E | F)$, is defined as

$$p(E | F) = \frac{p(E \cap F)}{p(F)}.$$

EXAMPLE 3



A bit string of length four is generated at random so that each of the 16 bit strings of length four is equally likely. What is the probability that it contains at least two consecutive 0s, given that its first bit is a 0? (We assume that 0 bits and 1 bits are equally likely.)

Solution: Let E be the event that a bit string of length four contains at least two consecutive 0s, and let F be the event that the first bit of a bit string of length four is a 0. The probability that a bit string of length four has at least two consecutive 0s, given that its first bit is a 0, equals

$$p(E | F) = \frac{p(E \cap F)}{p(F)}.$$

Because $E \cap F = \{0000, 0001, 0010, 0011, 0100\}$, we see that $p(E \cap F) = 5/16$. Because there are eight bit strings of length four that start with a 0, we have $p(F) = 8/16 = 1/2$. Consequently,

$$p(E | F) = \frac{5/16}{1/2} = \frac{5}{8}.$$

EXAMPLE 4 What is the conditional probability that a family with two children has two boys, given they have at least one boy? Assume that each of the possibilities BB , BG , GB , and GG is equally likely, where B represents a boy and G represents a girl. (Note that BG represents a family with an older boy and a younger girl while GB represents a family with an older girl and a younger boy.)

Solution: Let E be the event that a family with two children has two boys, and let F be the event that a family with two children has at least one boy. It follows that $E = \{BB\}$, $F = \{BB, BG, GB\}$, and $E \cap F = \{BB\}$. Because the four possibilities are equally likely, it follows that $p(F) = 3/4$ and $p(E \cap F) = 1/4$. We conclude that

$$p(E | F) = \frac{p(E \cap F)}{p(F)} = \frac{1/4}{3/4} = \frac{1}{3}.$$

Independence



Suppose a coin is flipped three times, as described in the introduction to our discussion of conditional probability. Does knowing that the first flip comes up tails (event F) alter the probability that tails comes up an odd number of times (event E)? In other words, is it the case that $p(E | F) = p(E)$? This equality is valid for the events E and F , because $p(E | F) = 1/2$ and $p(E) = 1/2$. Because this equality holds, we say that E and F are **independent events**. When two events are independent, the occurrence of one of the events gives no information about the probability that the other event occurs.

Because $p(E | F) = p(E \cap F)/p(F)$, asking whether $p(E | F) = p(E)$ is the same as asking whether $p(E \cap F) = p(E)p(F)$. This leads to Definition 4.

DEFINITION 4

The events E and F are *independent* if and only if $p(E \cap F) = p(E)p(F)$.

EXAMPLE 5



Suppose E is the event that a randomly generated bit string of length four begins with a 1 and F is the event that this bit string contains an even number of 1s. Are E and F independent, if the 16 bit strings of length four are equally likely?

Solution: There are eight bit strings of length four that begin with a one: 1000, 1001, 1010, 1011, 1100, 1101, 1110, and 1111. There are also eight bit strings of length four that contain an even number of ones: 0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111. Because there are 16 bit strings of length four, it follows that

$$p(E) = p(F) = 8/16 = 1/2.$$

Because $E \cap F = \{1111, 1100, 1010, 1001\}$, we see that

$$p(E \cap F) = 4/16 = 1/4.$$

Because

$$p(E \cap F) = 1/4 = (1/2)(1/2) = p(E)p(F),$$

we conclude that E and F are independent.

Probability has many applications to genetics, as Examples 6 and 7 illustrate.

EXAMPLE 6 Assume, as in Example 4, that each of the four ways a family can have two children is equally likely. Are the events E , that a family with two children has two boys, and F , that a family with two children has at least one boy, independent?

Solution: Because $E = \{BB\}$, we have $p(E) = 1/4$. In Example 4 we showed that $p(F) = 3/4$ and that $p(E \cap F) = 1/4$. But $p(E)p(F) = \frac{1}{4} \cdot \frac{3}{4} = \frac{3}{16}$. Therefore $p(E \cap F) \neq p(E)p(F)$, so the events E and F are not independent. \blacktriangleleft

EXAMPLE 7 Are the events E , that a family with three children has children of both sexes, and F , that this family has at most one boy, independent? Assume that the eight ways a family can have three children are equally likely.

Solution: By assumption, each of the eight ways a family can have three children, BBB , BBG , BGB , BGG , GBB , GBG , GGB , and GGG , has a probability of $1/8$. Because $E = \{BBG, BGB, BGG, GBB, GBG, GGB\}$, $F = \{BGG, GBG, GGB, GGG\}$, and $E \cap F = \{BGG, GBG, GGB\}$, it follows that $p(E) = 6/8 = 3/4$, $p(F) = 4/8 = 1/2$, and $p(E \cap F) = 3/8$. Because

$$p(E)p(F) = \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8},$$

it follows that $p(E \cap F) = p(E)p(F)$, so E and F are independent. (This conclusion may seem surprising. Indeed, if we change the number of children, the conclusion may no longer hold. See Exercise 27.) \blacktriangleleft

PAIRWISE AND MUTUAL INDEPENDENCE We can also define the independence of more than two events. However, there are two different types of independence, given in Definition 5.

DEFINITION 5

The events E_1, E_2, \dots, E_n are *pairwise independent* if and only if $p(E_i \cap E_j) = p(E_i)p(E_j)$ for all pairs of integers i and j with $1 \leq i < j \leq n$. These events are *mutually independent* if $p(E_{i_1} \cap E_{i_2} \cap \dots \cap E_{i_m}) = p(E_{i_1})p(E_{i_2}) \cdots p(E_{i_m})$ whenever $i_j, j = 1, 2, \dots, m$, are integers with $1 \leq i_1 < i_2 < \dots < i_m \leq n$ and $m \geq 2$.

From Definition 5, we see that every set of n mutually independent events is also pairwise independent. However, n pairwise independent events are not necessarily mutually independent, as we see in Exercise 25 in the Supplementary Exercises. Many theorems about n events include the hypothesis that these events are mutually independent, and not just pairwise independent. We will introduce several such theorems later in this chapter.

Bernoulli Trials and the Binomial Distribution

Suppose that an experiment can have only two possible outcomes. For instance, when a bit is generated at random, the possible outcomes are 0 and 1. When a coin is flipped, the possible outcomes are heads and tails. Each performance of an experiment with two possible outcomes is called a **Bernoulli trial**, after James Bernoulli, who made important contributions to probability theory. In general, a possible outcome of a Bernoulli trial is called a **success** or a **failure**. If p is the probability of a success and q is the probability of a failure, it follows that $p + q = 1$.



Many problems can be solved by determining the probability of k successes when an experiment consists of n mutually independent Bernoulli trials. (Bernoulli trials are **mutually independent** if the conditional probability of success on any given trial is p , given any information whatsoever about the outcomes of the other trials.) Consider Example 8.

EXAMPLE 8 A coin is biased so that the probability of heads is $2/3$. What is the probability that exactly four heads come up when the coin is flipped seven times, assuming that the flips are independent?

Solution: There are $2^7 = 128$ possible outcomes when a coin is flipped seven times. The number of ways four of the seven flips can be heads is $C(7, 4)$. Because the seven flips are independent, the probability of each of these outcomes (four heads and three tails) is $(2/3)^4(1/3)^3$. Consequently, the probability that exactly four heads appear is

$$C(7, 4)(2/3)^4(1/3)^3 = \frac{35 \cdot 16}{3^7} = \frac{560}{2187}.$$

Following the same reasoning as was used in Example 8, we can find the probability of k successes in n independent Bernoulli trials.

THEOREM 2

The probability of exactly k successes in n independent Bernoulli trials, with probability of success p and probability of failure $q = 1 - p$, is

$$C(n, k)p^k q^{n-k}.$$

Proof: When n Bernoulli trials are carried out, the outcome is an n -tuple (t_1, t_2, \dots, t_n) , where $t_i = S$ (for success) or $t_i = F$ (for failure) for $i = 1, 2, \dots, n$. Because the n trials are independent, the probability of each outcome of n trials consisting of k successes and $n - k$ failures (in any order) is $p^k q^{n-k}$. Because there are $C(n, k)$ n -tuples of S 's and F 's that contain exactly k S 's, the probability of exactly k successes is

$$C(n, k)p^k q^{n-k}.$$

We denote by $b(k; n, p)$ the probability of k successes in n independent Bernoulli trials with probability of success p and probability of failure $q = 1 - p$. Considered as a function of k , we call this function the **binomial distribution**. Theorem 2 tells us that $b(k; n, p) = C(n, k)p^k q^{n-k}$.

EXAMPLE 9



Suppose that the probability that a 0 bit is generated is 0.9, that the probability that a 1 bit is generated is 0.1, and that bits are generated independently. What is the probability that exactly eight 0 bits are generated when 10 bits are generated?

Solution: By Theorem 2, the probability that exactly eight 0 bits are generated is

$$b(8; 10, 0.9) = C(10, 8)(0.9)^8(0.1)^2 = 0.1937102445.$$



JAMES BERNOULLI (1654–1705) James Bernoulli (also known as Jacob I), was born in Basel, Switzerland. He is one of the eight prominent mathematicians in the Bernoulli family (see Section 10.1 for the Bernoulli family tree of mathematicians). Following his father's wish, James studied theology and entered the ministry. But contrary to the desires of his parents, he also studied mathematics and astronomy. He traveled throughout Europe from 1676 to 1682, learning about the latest discoveries in mathematics and the sciences. Upon returning to Basel in 1682, he founded a school for mathematics and the sciences. He was appointed professor of mathematics at the University of Basel in 1687, remaining in this position for the rest of his life.

James Bernoulli is best known for the work *Ars Conjectandi*, published eight years after his death. In this work, he described the known results in probability theory and in enumeration, often providing alternative proofs of known results. This work also includes the application of probability theory to games of chance and his introduction of the theorem known as the **law of large numbers**. This law states that if $\epsilon > 0$, as n becomes arbitrarily large the probability approaches 1 that the fraction of times an event E occurs during n trials is within ϵ of $p(E)$.

Note that the sum of the probabilities that there are k successes when n independent Bernoulli trials are carried out, for $k = 0, 1, 2, \dots, n$, equals

$$\sum_{k=0}^n C(n, k) p^k q^{n-k} = (p + q)^n = 1,$$

as should be the case. The first equality in this string of equalities is a consequence of the binomial theorem (see Section 6.4). The second equality follows because $q = 1 - p$.

Random Variables

Many problems are concerned with a numerical value associated with the outcome of an experiment. For instance, we may be interested in the total number of one bits in a randomly generated string of 10 bits; or in the number of times tails come up when a coin is flipped 20 times. To study problems of this type we introduce the concept of a random variable.

DEFINITION 6

A *random variable* is a function from the sample space of an experiment to the set of real numbers. That is, a random variable assigns a real number to each possible outcome.

Remark: Note that a random variable is a function. It is not a variable, and it is not random! The name *random variable* (the translation of *variabile casuale*) was introduced by the Italian mathematician F. P. Cantelli in 1916. In the late 1940s, the mathematicians, W. Feller and J. L. Doob flipped a coin to see whether both would use “random variable” or the more fitting term “chance variable.” Feller won; unfortunately “random variable” was used in both books and ever since.

EXAMPLE 10

Suppose that a coin is flipped three times. Let $X(t)$ be the random variable that equals the number of heads that appear when t is the outcome. Then $X(t)$ takes on the following values:

$$\begin{aligned} X(HHH) &= 3, \\ X(HHT) &= X(HTH) = X(THH) = 2, \\ X(TTH) &= X(THT) = X(HTT) = 1, \\ X(TTT) &= 0. \end{aligned}$$

DEFINITION 7

The *distribution* of a random variable X on a sample space S is the set of pairs $(r, p(X = r))$ for all $r \in X(S)$, where $p(X = r)$ is the probability that X takes the value r . (The set of pairs in this distribution is determined by the probabilities $p(X = r)$ for $r \in X(S)$.)

EXAMPLE 11

Each of the eight possible outcomes when a fair coin is flipped three times has probability $1/8$. So, the distribution of the random variable $X(t)$ in Example 10 is determined by the probabilities $P(X = 3) = 1/8$, $P(X = 2) = 3/8$, $P(X = 1) = 3/8$, and $P(X = 0) = 1/8$. Consequently, the distribution of $X(t)$ in Example 10 is the set of pairs $(3, 1/8)$, $(2, 3/8)$, $(1, 3/8)$, and $(0, 1/8)$.

EXAMPLE 12

Let X be the sum of the numbers that appear when a pair of dice is rolled. What are the values of this random variable for the 36 possible outcomes (i, j) , where i and j are the numbers that appear on the first die and the second die, respectively, when these two dice are rolled?

Solution: The random variable X takes on the following values:

$$\begin{aligned}
 X((1, 1)) &= 2, \\
 X((1, 2)) = X((2, 1)) &= 3, \\
 X((1, 3)) = X((2, 2)) = X((3, 1)) &= 4, \\
 X((1, 4)) = X((2, 3)) = X((3, 2)) = X((4, 1)) &= 5, \\
 X((1, 5)) = X((2, 4)) = X((3, 3)) = X((4, 2)) = X((5, 1)) &= 6, \\
 X((1, 6)) = X((2, 5)) = X((3, 4)) = X((4, 3)) = X((5, 2)) = X((6, 1)) &= 7, \\
 X((2, 6)) = X((3, 5)) = X((4, 4)) = X((5, 3)) = X((6, 2)) &= 8, \\
 X((3, 6)) = X((4, 5)) = X((5, 4)) = X((6, 3)) &= 9, \\
 X((4, 6)) = X((5, 5)) = X((6, 4)) &= 10, \\
 X((5, 6)) = X((6, 5)) &= 11, \\
 X((6, 6)) &= 12. \quad \blacktriangleleft
 \end{aligned}$$

We will continue our study of random variables in Section 7.4, where we will show how they can be used in a variety of applications.

The Birthday Problem

A famous puzzle asks for the smallest number of people needed in a room so that it is more likely than not that at least two of them have the same day of the year as their birthday. Most people find the answer, which we determine in Example 13, to be surprisingly small. After we solve this famous problem, we will show how similar reasoning can be adapted to solve a question about hashing functions.

EXAMPLE 13



The Birthday Problem What is the minimum number of people who need to be in a room so that the probability that at least two of them have the same birthday is greater than $1/2$?

Solution: First, we state some assumptions. We assume that the birthdays of the people in the room are independent. Furthermore, we assume that each birthday is equally likely and that there are 366 days in the year. (In reality, more people are born on some days of the year than others, such as days nine months after some holidays including New Year's Eve, and only leap years have 366 days.)

To find the probability that at least two of n people in a room have the same birthday, we first calculate the probability p_n that these people all have different birthdays. Then, the probability that at least two people have the same birthday is $1 - p_n$. To compute p_n , we consider the birthdays of the n people in some fixed order. Imagine them entering the room one at a time; we will compute the probability that each successive person entering the room has a birthday different from those of the people already in the room.

The birthday of the first person certainly does not match the birthday of someone already in the room. The probability that the birthday of the second person is different from that of the first person is $365/366$ because the second person has a different birthday when he or she was born on one of the 365 days of the year other than the day the first person was born. (The assumption that it is equally likely for someone to be born on any of the 366 days of the year enters into this and subsequent steps.)

The probability that the third person has a birthday different from both the birthdays of the first and second people given that these two people have different birthdays is $364/366$. In general, the probability that the j th person, with $2 \leq j \leq 366$, has a birthday different from the

birthdays of the $j - 1$ people already in the room given that these $j - 1$ people have different birthdays is

$$\frac{366 - (j - 1)}{366} = \frac{367 - j}{366}.$$

Because we have assumed that the birthdays of the people in the room are independent, we can conclude that the probability that the n people in the room have different birthdays is

$$p_n = \frac{365}{366} \frac{364}{366} \frac{363}{366} \cdots \frac{367 - n}{366}.$$

It follows that the probability that among n people there are at least two people with the same birthday is

$$1 - p_n = 1 - \frac{365}{366} \frac{364}{366} \frac{363}{366} \cdots \frac{367 - n}{366}.$$

To determine the minimum number of people in the room so that the probability that at least two of them have the same birthday is greater than $1/2$, we use the formula we have found for $1 - p_n$ to compute it for increasing values of n until it becomes greater than $1/2$. (There are more sophisticated approaches using calculus that can eliminate this computation, but we will not use them here.) After considerable computation we find that for $n = 22$, $1 - p_n \approx 0.475$, while for $n = 23$, $1 - p_n \approx 0.506$. Consequently, the minimum number of people needed so that the probability that at least two people have the same birthday is greater than $1/2$ is 23. 

The solution to the birthday problem leads to the solution of the question in Example 14 about hashing functions.

EXAMPLE 14

Probability of a Collision in Hashing Functions Recall from Section 4.5 that a hashing function $h(k)$ is a mapping of the keys (of the records that are to be stored in a database) to storage locations. Hashing functions map a large universe of keys (such as the approximately 300 million Social Security numbers in the United States) to a much smaller set of storage locations. A good hashing function yields few **collisions**, which are mappings of two different keys to the same memory location, when relatively few of the records are in play in a given application. What is the probability that no two keys are mapped to the same location by a hashing function, or, in other words, that there are no collisions?

Solution: To calculate this probability, we assume that the probability that a randomly selected key is mapped to a location is $1/m$, where m is the number of available locations, that is, the hashing function distributes keys uniformly. (In practice, hashing functions may not satisfy this assumption. However, for a good hashing function, this assumption should be close to correct.) Furthermore, we assume that the keys of the records selected have an equal probability to be any of the elements of the key universe and that these keys are independently selected.

Suppose that the keys are k_1, k_2, \dots, k_n . When we add the second record, the probability that it is mapped to a location different from the location of the first record, that $h(k_2) \neq h(k_1)$, is $(m - 1)/m$ because there are $m - 1$ free locations after the first record has been placed. The probability that the third record is mapped to a free location after the first and second records have been placed without a collision is $(m - 2)/m$. In general, the probability that the j th record is mapped to a free location after the first $j - 1$ records have been mapped to locations $h(k_1), h(k_2), \dots, h(k_{j-1})$ without collisions is $(m - (j - 1))/m$ because $j - 1$ of the m locations are taken.

Because the keys are independent, the probability that all n keys are mapped to different locations is

$$p_n = \frac{m - 1}{m} \cdot \frac{m - 2}{m} \cdot \dots \cdot \frac{m - n + 1}{m}.$$

It follows that the probability that there is at least one collision, that is, at least two keys are mapped to the same location, is

$$1 - p_n = 1 - \frac{m-1}{m} \cdot \frac{m-2}{m} \cdot \dots \cdot \frac{m-n+1}{m}.$$

Techniques from calculus can be used to find the smallest value of n given a value of m such that the probability of a collision is greater than a particular threshold. It can be shown that the smallest integer n such that the probability of a collision is greater than $1/2$ is approximately $n = 1.177\sqrt{m}$. For example, when $m = 1,000,000$, the smallest integer n such that the probability of a collision is greater than $1/2$ is 1178. 

Monte Carlo Algorithms

The algorithms discussed so far in this book are all deterministic. That is, each algorithm always proceeds in the same way whenever given the same input. However, there are many situations where we would like an algorithm to make a random choice at one or more steps. Such a situation arises when a deterministic algorithm would have to go through a huge number, or even an unknown number, of possible cases. Algorithms that make random choices at one or more steps are called **probabilistic algorithms**. We will discuss a particular class of probabilistic algorithms in this section, namely, **Monte Carlo algorithms**, for decision problems. Monte Carlo algorithms always produce answers to problems, but a small probability remains that these answers may be incorrect. However, the probability that the answer is incorrect decreases rapidly when the algorithm carries out sufficient computation. Decision problems have either “true” or “false” as their answer. The designation “Monte Carlo” is a reference to the famous casino in Monaco; the use of randomness and the repetitive processes in these algorithms make them similar to some gambling games. This name was introduced by the inventors of Monte Carlo methods, including Stan Ulam, Enrico Fermi, and John von Neumann.

Monte Carlo methods were invented to help develop the first nuclear weapons.

A Monte Carlo algorithm for a decision problem uses a sequence of tests. The probability that the algorithm answers the decision problem correctly increases as more tests are carried out. At each step of the algorithm, possible responses are “true,” which means that the answer is “true” and no additional iterations are needed, or “unknown,” which means that the answer could be either “true” or “false.” After running all the iterations in such an algorithm, the final answer produced is “true” if at least one iteration yields the answer “true,” and the answer is “false” if every iteration yields the answer “unknown.” If the correct answer is “false,” then the algorithm answers “false,” because every iteration will yield “unknown.” However, if the correct answer is “true,” then the algorithm could answer either “true” or “false,” because it may be possible that each iteration produced the response “unknown” even though the correct response was “true.” We will show that this possibility becomes extremely unlikely as the number of tests increases.

Suppose that p is the probability that the response of a test is “true,” given that the answer is “true.” It follows that $1-p$ is the probability that the response is “unknown,” given that the answer is “true.” Because the algorithm answers “false” when all n iterations yield the answer “unknown” and the iterations perform independent tests, the probability of error is $(1-p)^n$. When $p \neq 0$, this probability approaches 0 as the number of tests increases. Consequently, the probability that the algorithm answers “true” when the answer is “true” approaches 1.

EXAMPLE 15

Quality Control (This example is adapted from [AhU195].) Suppose that a manufacturer orders processor chips in batches of size n , where n is a positive integer. The chip maker has tested only some of these batches to make sure that all the chips in the batch are good (replacing any bad chips found during testing with good ones). In previously untested batches, the probability that a particular chip is bad has been observed to be 0.1 when random testing is done. The PC manufacturer wants to decide whether all the chips in a batch are good. To

do this, the PC manufacturer can test each chip in a batch to see whether it is good. However, this requires n tests. Assuming that each test can be carried out in constant time, these tests require $O(n)$ seconds. Can the PC manufacturer determine whether a batch of chips has been tested by the chip maker using less time?

Solution: We can use a Monte Carlo algorithm to determine whether a batch of chips has been tested by the chip maker as long as we are willing to accept some probability of error. The algorithm is set up to answer the question: “Has this batch of chips not been tested by the chip maker?” It proceeds by successively selecting chips at random from the batch and testing them one by one. When a bad chip is encountered, the algorithm answers “true” and stops. If a tested chip is good, the algorithm answers “unknown” and goes on to the next chip. After the algorithm has tested a specified number of chips, say k chips, without getting an answer of “true,” the algorithm terminates with the answer “false”; that is, the algorithm concludes that the batch is good, that is, that the chip maker has tested all the chips in the batch.

The only way for this algorithm to answer incorrectly is for it to conclude that an untested batch of chips has been tested by the chip maker. The probability that a chip is good, but that it came from an untested batch, is $1 - 0.1 = 0.9$. Because the events of testing different chips from a batch are independent, the probability that all k steps of the algorithm produce the answer “unknown,” given that the batch of chips is untested, is 0.9^k .

By taking k large enough, we can make this probability as small as we like. For example, by testing 66 chips, the probability that the algorithm decides a batch has been tested by the chip maker is 0.9^{66} , which is less than 0.001. That is, the probability is less than 1 in 1000 that the algorithm has answered incorrectly. Note that this probability is independent of n , the number of chips in a batch. That is, the Monte Carlo algorithm uses a constant number, or $O(1)$, tests and requires $O(1)$ seconds, no matter how many chips are in a batch. As long as the PC manufacturer can live with an error rate of less than 1 in 1000, the Monte Carlo algorithm will save the PC manufacturer a lot of testing. If a smaller error rate is needed, the PC manufacturer can test more chips in each batch; the reader can verify that 132 tests lower the error rate to less than 1 in 1,000,000. ◀

EXAMPLE 16

Probabilistic Primality Testing In Chapter 4 we remarked that a composite integer, that is, an integer greater than one that is not prime, passes Miller’s test (see the preamble to Exercise 44 in Section 4.4) for fewer than $n/4$ bases b with $1 < b < n$. This observation is the basis for a Monte Carlo algorithm to determine whether an integer greater than one is prime. Because large primes play an essential role in public-key cryptography (see Section 4.6), being able to generate large primes quickly has become extremely important.

The goal of the algorithm is to decide the question “Is n composite?” Given an integer n greater than one, we select an integer b at random with $1 < b < n$ and determine whether n passes Miller’s test to the base b . If n fails the test, the answer is “true” because n must be composite, and the algorithm ends. Otherwise, we perform the test k times, where k is a positive integer. Each time we select a random integer b and determine whether n passes Miller’s test to the base b . If the answer is “unknown” at each step, the algorithm answers “false,” that is, it says that n is not composite, so that it is prime. The only possibility for the algorithm to return an incorrect answer occurs when n is composite, and the answer “unknown” is the output at each of the k iterations. The probability that a composite integer n passes Miller’s test for a randomly selected base b is less than $1/4$. Because the integer b with $1 < b < n$ is selected at random at each iteration and these iterations are independent, the probability that n is composite but the algorithm responds that n is prime is less than $(1/4)^k$. By taking k to be sufficiently large, we can make this probability extremely small. For example, with 10 iterations, the probability that the algorithm decides that n is prime when it really is composite is less than 1 in 1,000,000. With 30 iterations, this probability drops to less than 1 in 10^{18} , an extremely unlikely event.

To generate large primes, say with 200 digits, we randomly choose an integer n with 200 digits and run this algorithm, with 30 iterations. If the algorithm decides that n is prime, we

A number that passes many iterations of a probabilistic primality test is called an *industrial strength prime*, even though it may be composite.

can use it as one of the two primes used in an encryption key for the RSA cryptosystem. If n is actually composite and is used as part of the key, the procedures used to decrypt messages will not produce the original encrypted message. The key is then discarded and two new possible primes are used. 

The Probabilistic Method

We discussed existence proofs in Chapter 1 and illustrated the difference between constructive existence proofs and nonconstructive existence proofs. The probabilistic method, introduced by Paul Erdős and Alfréd Rényi, is a powerful technique that can be used to create nonconstructive existence proofs. To use the probabilistic method to prove results about a set S , such as the existence of an element in S with a specified property, we assign probabilities to the elements of S . We then use methods from probability theory to prove results about the elements of S . In particular, we can show that an element with a specified property exists by showing that the probability an element $x \in S$ has this property is positive. The probabilistic method is based on the equivalent statement in Theorem 3.

THEOREM 3

THE PROBABILISTIC METHOD If the probability that an element chosen at random from a S does not have a particular property is less than 1, there exists an element in S with this property.

An existence proof based on the probabilistic method is nonconstructive because it does not find a particular element with the desired property.

We illustrate the power of the probabilistic method by finding a lower bound for the Ramsey number $R(k, k)$. Recall from Section 6.2 that $R(k, k)$ equals the minimum number of people at a party needed to ensure that there are at least k mutual friends or k mutual enemies (assuming that any two people are friends or enemies).

THEOREM 4

If k is an integer with $k \geq 2$, then $R(k, k) \geq 2^{k/2}$.

Proof: We note that the theorem holds for $k = 2$ and $k = 3$ because $R(2, 2) = 2$ and $R(3, 3) = 6$, as was shown in Section 6.2. Now suppose that $k \geq 4$. We will use the probabilistic method to show that if there are fewer than $2^{k/2}$ people at a party, it is possible that no k of them are mutual friends or mutual enemies. This will show that $R(k, k)$ is at least $2^{k/2}$.

To use the probabilistic method, we assume that it is equally likely for two people to be friends or enemies. (Note that this assumption does not have to be realistic.) Suppose there are n people at the party. It follows that there are $\binom{n}{k}$ different sets of k people at this party, which we list as $S_1, S_2, \dots, S_{\binom{n}{k}}$. Let E_i be the event that all k people in S_i are either mutual friends or mutual enemies. The probability that there are either k mutual friends or k mutual enemies among the n people equals $p(\bigcup_{i=1}^{\binom{n}{k}} E_i)$.

According to our assumption it is equally likely for two people to be friends or enemies. The probability that two people are friends equals the probability that they are enemies; both probabilities equal $1/2$. Furthermore, there are $\binom{k}{2} = k(k-1)/2$ pairs of people in S_i because there are k people in S_i . Hence, the probability that all k people in S_i are mutual friends and the probability that all k people in S_i are mutual enemies both equal $(1/2)^{k(k-1)/2}$. It follows that $p(E_i) = 2(1/2)^{k(k-1)/2}$.

The probability that there are either k mutual friends or k mutual enemies in the group of n people equals $p(\bigcup_{i=1}^{(n)} E_i)$. Using Boole's inequality (Exercise 15), it follows that



$$p\left(\bigcup_{i=1}^{(n)} E_i\right) \leq \sum_{i=1}^{(n)} p(E_i) = \binom{n}{k} \cdot 2\left(\frac{1}{2}\right)^{k(k-1)/2}.$$

By Exercise 17 in Section 6.4, we have $\binom{n}{k} \leq n^k/2^{k-1}$. Hence,

$$\binom{n}{k} 2\left(\frac{1}{2}\right)^{k(k-1)/2} \leq \frac{n^k}{2^{k-1}} 2\left(\frac{1}{2}\right)^{k(k-1)/2}.$$

Now if $n < 2^{k/2}$, we have

$$\frac{n^k}{2^{k-1}} 2\left(\frac{1}{2}\right)^{k(k-1)/2} < \frac{2^{k(k/2)}}{2^{k-1}} 2\left(\frac{1}{2}\right)^{k(k-1)/2} = 2^{2-(k/2)} \leq 1,$$

where the last step follows because $k \geq 4$.

We can now conclude that $p(\bigcup_{i=1}^{(n)} E_i) < 1$ when $k \geq 4$. Hence, the probability of the complementary event, that there is no set of either k mutual friends or mutual enemies at the party, is greater than 0. It follows that if $n < 2^{k/2}$, there is at least one set such that no subset of k people are mutual friends or mutual enemies.

Exercises

1. What probability should be assigned to the outcome of heads when a biased coin is tossed, if heads is three times as likely to come up as tails? What probability should be assigned to the outcome of tails?
2. Find the probability of each outcome when a loaded die is rolled, if a 3 is twice as likely to appear as each of the other five numbers on the die.
3. Find the probability of each outcome when a biased die is rolled, if rolling a 2 or rolling a 4 is three times as likely as rolling each of the other four numbers on the die and it is equally likely to roll a 2 or a 4.
4. Show that conditions (i) and (ii) are met under Laplace's definition of probability, when outcomes are equally likely.
5. A pair of dice is loaded. The probability that a 4 appears on the first die is $2/7$, and the probability that a 3 appears on the second die is $2/7$. Other outcomes for each die appear with probability $1/7$. What is the probability of 7 appearing as the sum of the numbers when the two dice are rolled?
6. What is the probability of these events when we randomly select a permutation of $\{1, 2, 3\}$?
 - a) 1 precedes 3.
 - b) 3 precedes 1.
 - c) 3 precedes 1 and 3 precedes 2.
7. What is the probability of these events when we randomly select a permutation of $\{1, 2, 3, 4\}$?
 - a) 1 precedes 4.
 - b) 4 precedes 1.
 - c) 4 precedes 1 and 4 precedes 2.
 - d) 4 precedes 1, 4 precedes 2, and 4 precedes 3.
 - e) 4 precedes 3 and 2 precedes 1.
8. What is the probability of these events when we randomly select a permutation of $\{1, 2, \dots, n\}$ where $n \geq 4$?
 - a) 1 precedes 2.
 - b) 2 precedes 1.
 - c) 1 immediately precedes 2.
 - d) n precedes 1 and $n - 1$ precedes 2.
 - e) n precedes 1 and n precedes 2.
9. What is the probability of these events when we randomly select a permutation of the 26 lowercase letters of the English alphabet?
 - a) The permutation consists of the letters in reverse alphabetical order.
 - b) z is the first letter of the permutation.
 - c) z precedes a in the permutation.
 - d) a immediately precedes z in the permutation.
 - e) a immediately precedes m , which immediately precedes z in the permutation.
 - f) m, n , and o are in their original places in the permutation.

- 10.** What is the probability of these events when we randomly select a permutation of the 26 lowercase letters of the English alphabet?
- The first 13 letters of the permutation are in alphabetical order.
 - a is the first letter of the permutation and z is the last letter.
 - a and z are next to each other in the permutation.
 - a and b are not next to each other in the permutation.
 - a and z are separated by at least 23 letters in the permutation.
 - z precedes both a and b in the permutation.
- 11.** Suppose that E and F are events such that $p(E) = 0.7$ and $p(F) = 0.5$. Show that $p(E \cup F) \geq 0.7$ and $p(E \cap F) \geq 0.2$.
- 12.** Suppose that E and F are events such that $p(E) = 0.8$ and $p(F) = 0.6$. Show that $p(E \cup F) \geq 0.8$ and $p(E \cap F) \geq 0.4$.
- 13.** Show that if E and F are events, then $p(E \cap F) \geq p(E) + p(F) - 1$. This is known as **Bonferroni's inequality**.
- 14.** Use mathematical induction to prove the following generalization of Bonferroni's inequality:

$$\begin{aligned} p(E_1 \cap E_2 \cap \dots \cap E_n) \\ \geq p(E_1) + p(E_2) + \dots + p(E_n) - (n-1), \end{aligned}$$

where E_1, E_2, \dots, E_n are n events.

- 15.** Show that if E_1, E_2, \dots, E_n are events from a finite sample space, then

$$\begin{aligned} p(E_1 \cup E_2 \cup \dots \cup E_n) \\ \leq p(E_1) + p(E_2) + \dots + p(E_n). \end{aligned}$$

This is known as **Boole's inequality**.

- 16.** Show that if E and F are independent events, then \bar{E} and \bar{F} are also independent events.
- 17.** If E and F are independent events, prove or disprove that \bar{E} and F are necessarily independent events.

In Exercises 18, 20, and 21 assume that the year has 366 days and all birthdays are equally likely. In Exercise 19 assume it is equally likely that a person is born in any given month of the year.

- 18. a)** What is the probability that two people chosen at random were born on the same day of the week?
b) What is the probability that in a group of n people chosen at random, there are at least two born on the same day of the week?
c) How many people chosen at random are needed to make the probability greater than $1/2$ that there are at least two people born on the same day of the week?
- 19. a)** What is the probability that two people chosen at random were born during the same month of the year?
b) What is the probability that in a group of n people chosen at random, there are at least two born in the same month of the year?
c) How many people chosen at random are needed to make the probability greater than $1/2$ that there are at least two people born in the same month of the year?

- 20.** Find the smallest number of people you need to choose at random so that the probability that at least one of them has a birthday today exceeds $1/2$.
- 21.** Find the smallest number of people you need to choose at random so that the probability that at least two of them were both born on April 1 exceeds $1/2$.
- *22.** February 29 occurs only in leap years. Years divisible by 4, but not by 100, are always leap years. Years divisible by 100, but not by 400, are not leap years, but years divisible by 400 are leap years.
- What probability distribution for birthdays should be used to reflect how often February 29 occurs?
 - Using the probability distribution from part (a), what is the probability that in a group of n people at least two have the same birthday?
- 23.** What is the conditional probability that exactly four heads appear when a fair coin is flipped five times, given that the first flip came up heads?
- 24.** What is the conditional probability that exactly four heads appear when a fair coin is flipped five times, given that the first flip came up tails?
- 25.** What is the conditional probability that a randomly generated bit string of length four contains at least two consecutive 0s, given that the first bit is a 1? (Assume the probabilities of a 0 and a 1 are the same.)
- 26.** Let E be the event that a randomly generated bit string of length three contains an odd number of 1s, and let F be the event that the string starts with 1. Are E and F independent?
- 27.** Let E and F be the events that a family of n children has children of both sexes and has at most one boy, respectively. Are E and F independent if
- $n = 2?$
 - $n = 4?$
 - $n = 5?$
- 28.** Assume that the probability a child is a boy is 0.51 and that the sexes of children born into a family are independent. What is the probability that a family of five children has
- exactly three boys?
 - at least one boy?
 - at least one girl?
 - all children of the same sex?
- 29.** A group of six people play the game of “odd person out” to determine who will buy refreshments. Each person flips a fair coin. If there is a person whose outcome is not the same as that of any other member of the group, this person has to buy the refreshments. What is the probability that there is an odd person out after the coins are flipped once?
- 30.** Find the probability that a randomly generated bit string of length 10 does not contain a 0 if bits are independent and if
- a 0 bit and a 1 bit are equally likely.
 - the probability that a bit is a 1 is 0.6.
 - the probability that the i th bit is a 1 is $1/2^i$ for $i = 1, 2, 3, \dots, 10$.

- 31.** Find the probability that a family with five children does not have a boy, if the sexes of children are independent and if
- a boy and a girl are equally likely.
 - the probability of a boy is 0.51.
 - the probability that the i th child is a boy is $0.51 - (i/100)$.
- 32.** Find the probability that a randomly generated bit string of length 10 begins with a 1 or ends with a 00 for the same conditions as in parts (a), (b), and (c) of Exercise 30, if bits are generated independently.
- 33.** Find the probability that the first child of a family with five children is a boy or that the last two children of the family are girls, for the same conditions as in parts (a), (b), and (c) of Exercise 31.
- 34.** Find each of the following probabilities when n independent Bernoulli trials are carried out with probability of success p .
- the probability of no successes
 - the probability of at least one success
 - the probability of at most one success
 - the probability of at least two successes
- 35.** Find each of the following probabilities when n independent Bernoulli trials are carried out with probability of success p .
- the probability of no failures
 - the probability of at least one failure
 - the probability of at most one failure
 - the probability of at least two failures
- 36.** Use mathematical induction to prove that if E_1, E_2, \dots, E_n is a sequence of n pairwise disjoint events in a sample space S , where n is a positive integer, then $p(\bigcup_{i=1}^n E_i) = \sum_{i=1}^n p(E_i)$.
- *37.** (Requires calculus) Show that if E_1, E_2, \dots is an infinite sequence of pairwise disjoint events in a sample space S , then $p(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} p(E_i)$. [Hint: Use Exercise 36 and take limits.]
- 38.** A pair of dice is rolled in a remote location and when you ask an honest observer whether at least one die came up six, this honest observer answers in the affirmative.
- What is the probability that the sum of the numbers that came up on the two dice is seven, given the information provided by the honest observer?

- b)** Suppose that the honest observer tells us that at least one die came up five. What is the probability the sum of the numbers that came up on the dice is seven, given this information?

- **39.** This exercise employs the probabilistic method to prove a result about round-robin tournaments. In a **round-robin tournament** with m players, every two players play one game in which one player wins and the other loses.

We want to find conditions on positive integers m and k with $k < m$ such that it is possible for the outcomes of the tournament to have the property that for every set of k players, there is a player who beats every member in this set. So that we can use probabilistic reasoning to draw conclusions about round-robin tournaments, we assume that when two players compete it is equally likely that either player wins the game and we assume that the outcomes of different games are independent. Let E be the event that for every set S with k players, where k is a positive integer less than m , there is a player who has beaten all k players in S .

- Show that $p(\bar{E}) \leq \sum_{j=1}^{\binom{m}{k}} p(F_j)$, where F_j is the event that there is no player who beats all k players from the j th set in a list of the $\binom{m}{k}$ sets of k players.
- Show that the probability of F_j is $(1 - 2^{-k})^{m-k}$.
- Conclude from parts (a) and (b) that $p(\bar{E}) \leq \binom{m}{k}(1 - 2^{-k})^{m-k}$ and, therefore, that there must be a tournament with the described property if $\binom{m}{k}(1 - 2^{-k})^{m-k} < 1$.
- Use part (c) to find values of m such that there is a tournament with m players such that for every set S of two players, there is a player who has beaten both players in S . Repeat for sets of three players.

- *40.** Devise a Monte Carlo algorithm that determines whether a permutation of the integers 1 through n has already been sorted (that is, it is in increasing order), or instead, is a random permutation. A step of the algorithm should answer “true” if it determines the list is not sorted and “unknown” otherwise. After k steps, the algorithm decides that the integers are sorted if the answer is “unknown” in each step. Show that as the number of steps increases, the probability that the algorithm produces an incorrect answer is extremely small. [Hint: For each step, test whether certain elements are in the correct order. Make sure these tests are independent.]

- 41.** Use pseudocode to write out the probabilistic primality test described in Example 16.

7.3 Bayes' Theorem

Introduction

There are many times when we want to assess the probability that a particular event occurs on the basis of partial evidence. For example, suppose we know the percentage of people who have a particular disease for which there is a very accurate diagnostic test. People who test positive for

this disease would like to know the likelihood that they actually have the disease. In this section we introduce a result that can be used to determine this probability, namely, the probability that a person has the disease given that this person tests positive for it. To use this result, we will need to know the percentage of people who do not have the disease but test positive for it and the percentage of people who have the disease but test negative for it.

Similarly, suppose we know the percentage of incoming e-mail messages that are spam. We will see that we can determine the likelihood that an incoming e-mail message is spam using the occurrence of words in the message. To determine this likelihood, we need to know the percentage of incoming messages that are spam, the percentage of spam messages in which each of these words occurs, and the percentage of messages that are not spam in which each of these words occurs.

The result that we can use to answer questions such as these is called Bayes' theorem and dates back to the eighteenth century. In the past two decades, Bayes' theorem has been extensively applied to estimate probabilities based on partial evidence in areas as diverse as medicine, law, machine learning, engineering, and software development.

Bayes' Theorem

We illustrate the idea behind Bayes' theorem with an example that shows that when extra information is available, we can derive a more realistic estimate that a particular event occurs. That is, suppose we know $p(F)$, the probability that an event F occurs, but we have knowledge that an event E occurs. Then the conditional probability that F occurs given that E occurs, $p(F | E)$, is a more realistic estimate than $p(F)$ that F occurs. In Example 1 we will see that we can find $p(F | E)$ when we know $p(F)$, $p(E | F)$, and $p(E | \bar{F})$.

EXAMPLE 1



We have two boxes. The first contains two green balls and seven red balls; the second contains four green balls and three red balls. Bob selects a ball by first choosing one of the two boxes at random. He then selects one of the balls in this box at random. If Bob has selected a red ball, what is the probability that he selected a ball from the first box?

Solution: Let E be the event that Bob has chosen a red ball; \bar{E} is the event that Bob has chosen a green ball. Let F be the event that Bob has chosen a ball from the first box; \bar{F} is the event that Bob has chosen a ball from the second box. We want to find $p(F | E)$, the probability that the ball Bob selected came from the first box, given that it is red. By the definition of conditional probability, we have $p(F | E) = p(F \cap E)/p(E)$. Can we use the information provided to determine both $p(F \cap E)$ and $p(E)$ so that we can find $p(F | E)$?

First, note that because the first box contains seven red balls out of a total of nine balls, we know that $p(E | F) = 7/9$. Similarly, because the second box contains three red balls out of a total of seven balls, we know that $p(E | \bar{F}) = 3/7$. We assumed that Bob selects a box at random, so $p(F) = p(\bar{F}) = 1/2$. Because $p(E | F) = p(E \cap F)/p(F)$, it follows that $p(E \cap F) = p(E | F)p(F) = \frac{7}{9} \cdot \frac{1}{2} = \frac{7}{18}$ [as we remarked earlier, this is one of the quantities we need to find to determine $p(F | E)$]. Similarly, because $p(E | \bar{F}) = p(E \cap \bar{F})/p(\bar{F})$, it follows that $p(E \cap \bar{F}) = p(E | \bar{F})p(\bar{F}) = \frac{3}{7} \cdot \frac{1}{2} = \frac{3}{14}$.

We can now find $p(E)$. Note that $E = (E \cap F) \cup (E \cap \bar{F})$, where $E \cap F$ and $E \cap \bar{F}$ are disjoint sets. (If x belongs to both $E \cap F$ and $E \cap \bar{F}$, then x belongs to both F and \bar{F} , which is impossible.) It follows that

$$p(E) = p(E \cap F) + p(E \cap \bar{F}) = \frac{7}{18} + \frac{3}{14} = \frac{49}{126} + \frac{27}{126} = \frac{76}{126} = \frac{38}{63}.$$

We have now found both $p(F \cap E) = 7/18$ and $p(E) = 38/63$. We conclude that

$$p(F | E) = \frac{p(F \cap E)}{p(E)} = \frac{7/18}{38/63} = \frac{49}{76} \approx 0.645.$$

Before we had any extra information, we assumed that the probability that Bob selected the first box was $1/2$. However, with the extra information that the ball selected at random is red, this probability has increased to approximately 0.645 . That is, the probability that Bob selected a ball from the first box increased from $1/2$, when no extra information was available, to 0.645 once we knew that the ball selected was red. 

Using the same type of reasoning as in Example 1, we can find the conditional probability that an event F occurs, given that an event E has occurred, when we know $p(E | F)$, $p(E | \bar{F})$, and $p(F)$. The result we can obtain is called **Bayes' theorem**; it is named after Thomas Bayes, an eighteenth-century British mathematician and minister who introduced this result.

THEOREM 1

BAYES' THEOREM Suppose that E and F are events from a sample space S such that $p(E) \neq 0$ and $p(F) \neq 0$. Then

$$p(F | E) = \frac{p(E | F)p(F)}{p(E | F)p(F) + p(E | \bar{F})p(\bar{F})}.$$

Proof: The definition of conditional probability tells us that $p(F | E) = p(E \cap F)/p(E)$ and $p(E | F) = p(E \cap F)/p(F)$. Therefore, $p(E \cap F) = p(F | E)p(E)$ and $p(E \cap F) = p(E | F)p(F)$. Equating these two expressions for $p(E \cap F)$ shows that

$$p(F | E)p(E) = p(E | F)p(F).$$

Dividing both sides by $p(E)$, we find that

$$p(F | E) = \frac{p(E | F)p(F)}{p(E)}.$$

Next, we show that $p(E) = p(E | F)p(F) + p(E | \bar{F})p(\bar{F})$. To see this, first note that $E = E \cap S = E \cap (F \cup \bar{F}) = (E \cap F) \cup (E \cap \bar{F})$. Furthermore, $E \cap F$ and $E \cap \bar{F}$ are disjoint, because if $x \in E \cap F$ and $x \in E \cap \bar{F}$, then $x \in F \cap \bar{F} = \emptyset$. Consequently, $p(E) = p(E \cap F) + p(E \cap \bar{F})$. We have already shown that $p(E \cap F) = p(E | F)p(F)$. Moreover, we have $p(E | \bar{F}) = p(E \cap \bar{F})/p(\bar{F})$, which shows that $p(E \cap \bar{F}) = p(E | \bar{F})p(\bar{F})$. It now follows that

$$p(E) = p(E \cap F) + p(E \cap \bar{F}) = p(E | F)p(F) + p(E | \bar{F})p(\bar{F}).$$

To complete the proof we insert this expression for $p(E)$ into the equation $p(F | E) = p(E | F)p(F)/p(E)$. We have proved that



$$p(F | E) = \frac{p(E | F)p(F)}{p(E | F)p(F) + p(E | \bar{F})p(\bar{F})}. \quad \triangle$$

APPLYING BAYES' THEOREM Bayes' theorem can be used to solve problems that arise in many disciplines. Next, we will discuss an application of Bayes' theorem to medicine. In particular, we will illustrate how Bayes' theorem can be used to assess the probability that someone testing positive for a disease actually has this disease. The results obtained from Bayes' theorem are often somewhat surprising, as Example 2 shows.

EXAMPLE 2 Suppose that one person in 100,000 has a particular rare disease for which there is a fairly accurate diagnostic test. This test is correct 99.0% of the time when given to a person selected at random who has the disease; it is correct 99.5% of the time when given to a person selected at random who does not have the disease. Given this information can we find

- the probability that a person who tests positive for the disease has the disease?
- the probability that a person who tests negative for the disease does not have the disease?

Should a person who tests positive be very concerned that he or she has the disease?

Solution: (a) Let F be the event that a person selected at random has the disease, and let E be the event that a person selected at random tests positive for the disease. We want to compute $p(F | E)$. To use Bayes' theorem to compute $p(F | E)$ we need to find $p(E | F)$, $p(E | \bar{F})$, $p(F)$, and $p(\bar{F})$.

We know that one person in 100,000 has this disease, so $p(F) = 1/100,000 = 0.00001$ and $p(\bar{F}) = 1 - 0.00001 = 0.99999$. Because a person who has the disease tests positive 99% of the time, we know that $p(E | F) = 0.99$; this is the probability of a true positive, that a person with the disease tests positive. It follows that $p(\bar{E} | F) = 1 - p(E | F) = 1 - 0.99 = 0.01$; this is the probability of a false negative, that a person who has the disease tests negative.

Furthermore, because a person who does not have the disease tests negative 99.5% of the time, we know that $p(\bar{E} | \bar{F}) = 0.995$. This is the probability of a true negative, that a person without the disease tests negative. Finally, we see that $p(E | \bar{F}) = 1 - p(\bar{E} | \bar{F}) = 1 - 0.995 = 0.005$; this is the probability of a false positive, that a person without the disease tests positive.

The probability that a person who tests positive for the disease actually has the disease is $p(F | E)$. By Bayes' theorem, we know that

$$\begin{aligned} p(F | E) &= \frac{p(E | F)p(F)}{p(E | F)p(F) + p(E | \bar{F})p(\bar{F})} \\ &= \frac{(0.99)(0.00001)}{(0.99)(0.00001) + (0.005)(0.99999)} \approx 0.002. \end{aligned}$$

(b) The probability that someone who tests negative for the disease does not have the disease is $p(\bar{F} | \bar{E})$. By Bayes' theorem, we know that

$$\begin{aligned} p(\bar{F} | \bar{E}) &= \frac{p(\bar{E} | \bar{F})p(\bar{F})}{p(\bar{E} | \bar{F})p(\bar{F}) + p(\bar{E} | F)p(F)} \\ &= \frac{(0.995)(0.99999)}{(0.995)(0.99999) + (0.01)(0.00001)} \approx 0.9999999. \end{aligned}$$

Consequently, 99.99999% of the people who test negative really do not have the disease.

In part (a) we showed that only 0.2% of people who test positive for the disease actually have the disease. Because the disease is extremely rare, the number of false positives on the diagnostic test is far greater than the number of true positives, making the percentage of people who test positive who actually have the disease extremely small. People who test positive for the disease should not be overly concerned that they actually have the disease. 

GENERALIZING BAYES' THEOREM Note that in the statement of Bayes' theorem, the events F and \bar{F} are mutually exclusive and cover the entire sample space S (that is, $F \cup \bar{F} = S$). We can extend Bayes' theorem to any collection of mutually exclusive events that cover the entire sample space S , in the following way.

THEOREM 2

GENERALIZED BAYES' THEOREM Suppose that E is an event from a sample space S and that F_1, F_2, \dots, F_n are mutually exclusive events such that $\bigcup_{i=1}^n F_i = S$. Assume that $p(E) \neq 0$ and $p(F_i) \neq 0$ for $i = 1, 2, \dots, n$. Then

$$p(F_j | E) = \frac{p(E | F_j)p(F_j)}{\sum_{i=1}^n p(E | F_i)p(F_i)}.$$

We leave the proof of this generalized version of Bayes' theorem as Exercise 17.

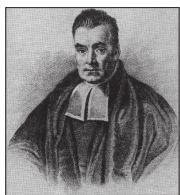
Bayesian Spam Filters

Most electronic mailboxes receive a flood of unwanted and unsolicited messages, known as **spam**. Because spam threatens to overwhelm electronic mail systems, a tremendous amount of work has been devoted to filtering it out. Some of the first tools developed for eliminating spam were based on Bayes' theorem, such as **Bayesian spam filters**.



The use of the word *spam* for unsolicited e-mail comes from a Monty Python comedy sketch about a cafe where the food product Spam comes with everything regardless of whether customers want it.

A Bayesian spam filter uses information about previously seen e-mail messages to guess whether an incoming e-mail message is spam. Bayesian spam filters look for occurrences of particular words in messages. For a particular word w , the probability that w appears in a spam e-mail message is estimated by determining the number of times w appears in a message from a large set of messages known to be spam and the number of times it appears in a large set of messages known not to be spam. When we examine e-mail messages to determine whether they might be spam, we look at words that might be indicators of spam, such as “offer,” “special,” or “opportunity,” as well as words that might indicate that a message is not spam, such as “mom,” “lunch,” or “Jan” (where Jan is one of your friends). Unfortunately, spam filters sometimes fail to identify a spam message as spam; this is called a false negative. And they sometimes identify a message that is not spam as spam; this is called a false positive. When testing for spam, it is important to minimize false positives, because filtering out wanted e-mail is much worse than letting some spam through.



THOMAS BAYES (1702–1761) Thomas Bayes was the son of a minister in a religious sect known as the Nonconformists. This sect was considered heretical in eighteenth-century Great Britain. Because of the secrecy of the Nonconformists, little is known of Thomas Bayes' life. When Thomas was young, his family moved to London. Thomas was likely educated privately; Nonconformist children generally did not attend school. In 1719 Bayes entered the University of Edinburgh, where he studied logic and theology. He was ordained as a Nonconformist minister like his father and began his work as a minister assisting his father. In 1733 he became minister of the Presbyterian Chapel in Tunbridge Wells, southeast of London, where he remained minister until 1752.

Bayes is best known for his essay on probability published in 1764, three years after his death. This essay was sent to the Royal Society by a friend who found it in the papers left behind when Bayes died. In the introduction to this essay, Bayes stated that his goal was to find a method that could measure the probability that an event happens, assuming that we know nothing about it, but that, under the same circumstances, it has happened a certain proportion of times. Bayes' conclusions were accepted by the great French mathematician Laplace but were later challenged by Boole, who questioned them in his book *Laws of Thought*. Since then Bayes' techniques have been subject to controversy.

Bayes also wrote an article that was published posthumously: “An Introduction to the Doctrine of Fluxions, and a Defense of the Mathematicians Against the Objections of the Author of The Analyst,” which supported the logical foundations of calculus. Bayes was elected a Fellow of the Royal Society in 1742, with the support of important members of the Society, even though at that time he had no published mathematical works. Bayes' sole known publication during his lifetime was allegedly a mystical book entitled *Divine Benevolence*, discussing the original causation and ultimate purpose of the universe. Although the book is commonly attributed to Bayes, no author's name appeared on the title page, and the entire work is thought to be of dubious provenance. Evidence for Bayes' mathematical talents comes from a notebook that was almost certainly written by Bayes, which contains much mathematical work, including discussions of probability, trigonometry, geometry, solutions of equations, series, and differential calculus. There are also sections on natural philosophy, in which Bayes looks at topics that include electricity, optics, and celestial mechanics. Bayes is also the author of a mathematical publication on asymptotic series, which appeared after his death.

We will develop some basic Bayesian spam filters. First, suppose we have a set B of messages known to be spam and a set G of messages known not to be spam. (For example, users could classify messages as spam when they examine them in their inboxes.) We next identify the words that occur in B and in G . We count the number of messages in the set containing each word to find $n_B(w)$ and $n_G(w)$, the number of messages containing the word w in the sets B and G , respectively. Then, the empirical probability that a spam message contains the word w is $p(w) = n_B(w)/|B|$, and the empirical probability that a message that is not spam contains the word w is $q(w) = n_G(w)/|G|$. We note that $p(w)$ and $q(w)$ estimate the probabilities that an incoming spam message, and an incoming message that is not spam, contain the word w , respectively.

Now suppose we receive a new e-mail message containing the word w . Let S be the event that the message is spam. Let E be the event that the message contains the word w . The events S , that the message is spam, and \bar{S} , that the message is not spam, partition the set of all messages. Hence, by Bayes' theorem, the probability that the message is spam, given that it contains the word w , is

$$p(S | E) = \frac{p(E | S)p(S)}{p(E | S)p(S) + p(E | \bar{S})p(\bar{S})}.$$

To apply this formula, we first estimate $p(S)$, the probability that an incoming message is spam, as well as $p(\bar{S})$, the probability that the incoming message is not spam. Without prior knowledge about the likelihood that an incoming message is spam, for simplicity we assume that the message is equally likely to be spam as it is not to be spam. That is, we assume that $p(S) = p(\bar{S}) = 1/2$. Using this assumption, we find that the probability that a message is spam, given that it contains the word w , is

$$p(S | E) = \frac{p(E | S)}{p(E | S) + p(E | \bar{S})}.$$

(Note that if we have some empirical data about the ratio of spam messages to messages that are not spam, we can change this assumption to produce a better estimate for $p(S)$ and for $p(\bar{S})$; see Exercise 22.)

Next, we estimate $p(E | S)$, the conditional probability that the message contains the word w given that the message is spam, by $p(w)$. Similarly, we estimate $p(E | \bar{S})$, the conditional probability that the message contains the word w , given that the message is not spam, by $q(w)$. Inserting these estimates for $p(E | S)$ and $p(E | \bar{S})$ tells us that $p(S | E)$ can be estimated by

$$r(w) = \frac{p(w)}{p(w) + q(w)};$$

that is, $r(w)$ estimates the probability that the message is spam, given that it contains the word w . If $r(w)$ is greater than a threshold that we set, such as 0.9, then we classify the message as spam.

EXAMPLE 3

Suppose that we have found that the word “Rolex” occurs in 250 of 2000 messages known to be spam and in 5 of 1000 messages known not to be spam. Estimate the probability that an incoming message containing the word “Rolex” is spam, assuming that it is equally likely that an incoming message is spam or not spam. If our threshold for rejecting a message as spam is 0.9, will we reject such messages?

Solution: We use the counts that the word “Rolex” appears in spam messages and messages that are not spam to find that $p(\text{Rolex}) = 250/2000 = 0.125$ and $q(\text{Rolex}) = 5/1000 = 0.005$.

Because we are assuming that it is equally likely for an incoming message to be spam as it is not to be spam, we can estimate the probability that an incoming message containing the word “Rolex” is spam by

$$r(\text{Rolex}) = \frac{p(\text{Rolex})}{p(\text{Rolex}) + q(\text{Rolex})} = \frac{0.125}{0.125 + 0.005} = \frac{0.125}{0.130} \approx 0.962.$$

Because $r(\text{Rolex})$ is greater than the threshold 0.9, we reject such messages as spam. 

Detecting spam based on the presence of a single word can lead to excessive false positives and false negatives. Consequently, spam filters look at the presence of multiple words. For example, suppose that the message contains the words w_1 and w_2 . Let E_1 and E_2 denote the events that the message contains the words w_1 and w_2 , respectively. To make our computations simpler, we assume that E_1 and E_2 are independent events and that $E_1 | S$ and $E_2 | S$ are independent events and that we have no prior knowledge regarding whether or not the message is spam. (The assumptions that E_1 and E_2 are independent and that $E_1 | S$ and $E_2 | S$ are independent may introduce some error into our computations; we assume that this error is small.) Using Bayes’ theorem and our assumptions, we can show (see Exercise 23) that $p(S | E_1 \cap E_2)$, the probability that the message is spam given that it contains both w_1 and w_2 , is

$$p(S | E_1 \cap E_2) = \frac{p(E_1 | S)p(E_2 | S)}{p(E_1 | S)p(E_2 | S) + p(E_1 | \bar{S})p(E_2 | \bar{S})}.$$

We estimate the probability $p(S | E_1 \cap E_2)$ by

$$r(w_1, w_2) = \frac{p(w_1)p(w_2)}{p(w_1)p(w_2) + q(w_1)q(w_2)}.$$

That is, $r(w_1, w_2)$ estimates the probability that the message is spam, given that it contains the words w_1 and w_2 . When $r(w_1, w_2)$ is greater than a preset threshold, such as 0.9, we determine that the message is likely spam.

EXAMPLE 4

Suppose that we train a Bayesian spam filter on a set of 2000 spam messages and 1000 messages that are not spam. The word “stock” appears in 400 spam messages and 60 messages that are not spam, and the word “undervalued” appears in 200 spam messages and 25 messages that are not spam. Estimate the probability that an incoming message containing both the words “stock” and “undervalued” is spam, assuming that we have no prior knowledge about whether it is spam. Will we reject such messages as spam when we set the threshold at 0.9?

Solution: Using the counts of each of these two words in messages known to be spam or known not to be spam, we obtain the following estimates: $p(\text{stock}) = 400/2000 = 0.2$, $q(\text{stock}) = 60/1000 = 0.06$, $p(\text{undervalued}) = 200/2000 = 0.1$, and $q(\text{undervalued}) = 25/1000 = 0.025$. Using these probabilities, we can estimate the probability that the message is spam by

$$\begin{aligned} r(\text{stock, undervalued}) &= \frac{p(\text{stock})p(\text{undervalued})}{p(\text{stock})p(\text{undervalued}) + q(\text{stock})q(\text{undervalued})} \\ &= \frac{(0.2)(0.1)}{(0.2)(0.1) + (0.06)(0.025)} \approx 0.930. \end{aligned}$$

Because we have set the threshold for rejecting messages at 0.9, such messages will be rejected by the filter. 

The more words we use to estimate the probability that an incoming mail message is spam, the better is our chance that we correctly determine whether it is spam. In general, if E_i is the

event that the message contains word w_i , assuming that the number of incoming spam messages is approximately the same as the number of incoming messages that are not spam, and that the events $E_i | S$ are independent, then by Bayes' theorem the probability that a message containing all the words w_1, w_2, \dots, w_k is spam is

$$p(S | \bigcap_{i=1}^k E_i) = \frac{\prod_{i=1}^k p(E_i | S)}{\prod_{i=1}^k p(E_i | S) + \prod_{i=1}^k p(E_i | \bar{S})}.$$

We can estimate this probability by

$$r(w_1, w_2, \dots, w_k) = \frac{\prod_{i=1}^k p(w_i)}{\prod_{i=1}^k p(w_i) + \prod_{i=1}^k q(w_i)}.$$

For the most effective spam filter, we choose words for which the probability that each of these words appears in spam is either very high or very low. When we compute this value for a particular message, we reject the message as spam if $r(w_1, w_2, \dots, w_k)$ exceeds a preset threshold, such as 0.9.

Another way to improve the performance of a Bayesian spam filter is to look at the probabilities that particular pairs of words appear in spam and in messages that are not spam. We then treat appearances of these pairs of words as appearance of a single block, rather than as the appearance of two separate words. For example, the pair of words “enhance performance” most likely indicates spam, while “operatic performance” indicates a message that is not spam. Similarly, we can assess the likelihood that a message is spam by examining the structure of a message to determine where words appear in it. Also, spam filters look at appearances of certain types of strings of characters rather than just words. For example, a message with the valid e-mail address of one of your friends is less likely to be spam (if not sent by a worm) than one containing an e-mail address that came from a country known to originate a lot of spam. There is an ongoing war between people who create spam and those trying to filter their messages out. This leads to the introduction of many new techniques to defeat spam filters, including inserting into spam messages long strings of words that appear in messages that are not spam, as well as including words inside pictures. The techniques we have discussed here are only the first steps in fighting this war on spam.

Bayesian poisoning, the insertion of extra words to defeat spam filters, can use random or purposefully selected words.

Exercises

1. Suppose that E and F are events in a sample space and $p(E) = 1/3$, $p(F) = 1/2$, and $p(E | F) = 2/5$. Find $p(F | E)$.
2. Suppose that E and F are events in a sample space and $p(E) = 2/3$, $p(F) = 3/4$, and $p(F | E) = 5/8$. Find $p(E | F)$.
3. Suppose that Frida selects a ball by first picking one of two boxes at random and then selecting a ball from this box at random. The first box contains two white balls and three blue balls, and the second box contains four white balls and one blue ball. What is the probability that Frida picked a ball from the first box if she has selected a blue ball?
4. Suppose that Ann selects a ball by first picking one of two boxes at random and then selecting a ball from this box. The first box contains three orange balls and four black balls, and the second box contains five orange balls and six black balls. What is the probability that Ann picked a ball from the second box if she has selected an orange ball?
5. Suppose that 8% of all bicycle racers use steroids, that a bicyclist who uses steroids tests positive for steroids 96% of the time, and that a bicyclist who does not use steroids tests positive for steroids 9% of the time. What is the probability that a randomly selected bicyclist who tests positive for steroids actually uses steroids?
6. When a test for steroids is given to soccer players, 98% of the players taking steroids test positive and 12% of the players not taking steroids test positive. Suppose that 5% of soccer players take steroids. What is the probability that a soccer player who tests positive takes steroids?
7. Suppose that a test for opium use has a 2% false positive rate and a 5% false negative rate. That is, 2% of people who do not use opium test positive for opium, and

- 5% of opium users test negative for opium. Furthermore, suppose that 1% of people actually use opium.
- Find the probability that someone who tests negative for opium use does not use opium.
 - Find the probability that someone who tests positive for opium use actually uses opium.
8. Suppose that one person in 10,000 people has a rare genetic disease. There is an excellent test for the disease; 99.9% of people with the disease test positive and only 0.02% who do not have the disease test positive.
- What is the probability that someone who tests positive has the genetic disease?
 - What is the probability that someone who tests negative does not have the disease?
9. Suppose that 8% of the patients tested in a clinic are infected with HIV. Furthermore, suppose that when a blood test for HIV is given, 98% of the patients infected with HIV test positive and that 3% of the patients not infected with HIV test positive. What is the probability that
- a patient testing positive for HIV with this test is infected with it?
 - a patient testing positive for HIV with this test is not infected with it?
 - a patient testing negative for HIV with this test is infected with it?
 - a patient testing negative for HIV with this test is not infected with it?
10. Suppose that 4% of the patients tested in a clinic are infected with avian influenza. Furthermore, suppose that when a blood test for avian influenza is given, 97% of the patients infected with avian influenza test positive and that 2% of the patients not infected with avian influenza test positive. What is the probability that
- a patient testing positive for avian influenza with this test is infected with it?
 - a patient testing positive for avian influenza with this test is not infected with it?
 - a patient testing negative for avian influenza with this test is infected with it?
 - a patient testing negative for avian influenza with this test is not infected with it?
11. An electronics company is planning to introduce a new camera phone. The company commissions a marketing report for each new product that predicts either the success or the failure of the product. Of new products introduced by the company, 60% have been successes. Furthermore, 70% of their successful products were predicted to be successes, while 40% of failed products were predicted to be successes. Find the probability that this new camera phone will be successful if its success has been predicted.
- *12. A space probe near Neptune communicates with Earth using bit strings. Suppose that in its transmissions it sends a 1 one-third of the time and a 0 two-thirds of the time. When a 0 is sent, the probability that it is received correctly is 0.9, and the probability that it is received incorrectly (as a 1) is 0.1. When a 1 is sent, the probability that it is received correctly is 0.8, and the probability that it is received incorrectly (as a 0) is 0.2.
- a) Find the probability that a 0 is received.
- b) Use Bayes' theorem to find the probability that a 0 was transmitted, given that a 0 was received.
13. Suppose that E , F_1 , F_2 , and F_3 are events from a sample space S and that F_1 , F_2 , and F_3 are pairwise disjoint and their union is S . Find $p(F_1 | E)$ if $p(E | F_1) = 1/8$, $p(E | F_2) = 1/4$, $p(E | F_3) = 1/6$, $p(F_1) = 1/4$, $p(F_2) = 1/4$, and $p(F_3) = 1/2$.
14. Suppose that E , F_1 , F_2 , and F_3 are events from a sample space S and that F_1 , F_2 , and F_3 are pairwise disjoint and their union is S . Find $p(F_2 | E)$ if $p(E | F_1) = 2/7$, $p(E | F_2) = 3/8$, $p(E | F_3) = 1/2$, $p(F_1) = 1/6$, $p(F_2) = 1/2$, and $p(F_3) = 1/3$.
15. In this exercise we will use Bayes' theorem to solve the Monty Hall puzzle (Example 10 in Section 7.1). Recall that in this puzzle you are asked to select one of three doors to open. There is a large prize behind one of the three doors and the other two doors are losers. After you select a door, Monty Hall opens one of the two doors you did not select that he knows is a losing door, selecting at random if both are losing doors. Monty asks you whether you would like to switch doors. Suppose that the three doors in the puzzle are labeled 1, 2, and 3. Let W be the random variable whose value is the number of the winning door; assume that $p(W = k) = 1/3$ for $k = 1, 2, 3$. Let M denote the random variable whose value is the number of the door that Monty opens. Suppose you choose door i .
- What is the probability that you will win the prize if the game ends without Monty asking you whether you want to change doors?
 - Find $p(M = j | W = k)$ for $j = 1, 2, 3$ and $k = 1, 2, 3$.
 - Use Bayes' theorem to find $p(W = j | M = k)$ where i and j and k are distinct values.
 - Explain why the answer to part (c) tells you whether you should change doors when Monty gives you the chance to do so.
16. Ramesh can get to work in three different ways: by bicycle, by car, or by bus. Because of commuter traffic, there is a 50% chance that he will be late when he drives his car. When he takes the bus, which uses a special lane reserved for buses, there is a 20% chance that he will be late. The probability that he is late when he rides his bicycle is only 5%. Ramesh arrives late one day. His boss wants to estimate the probability that he drove his car to work that day.
- Suppose the boss assumes that there is a $1/3$ chance that Ramesh takes each of the three ways he can get to work. What estimate for the probability that Ramesh drove his car does the boss obtain from Bayes' theorem under this assumption?
 - Suppose the boss knows that Ramesh drives 30% of the time, takes the bus only 10% of the time, and takes his bicycle 60% of the time. What estimate for the probability that Ramesh drove his car does the boss obtain from Bayes' theorem using this information?

- *17. Prove Theorem 2, the extended form of Bayes' theorem. That is, suppose that E is an event from a sample space S and that F_1, F_2, \dots, F_n are mutually exclusive events such that $\bigcup_{i=1}^n F_i = S$. Assume that $p(E) \neq 0$ and $p(F_i) \neq 0$ for $i = 1, 2, \dots, n$. Show that

$$p(F_j | E) = \frac{p(E | F_j)p(F_j)}{\sum_{i=1}^n p(E | F_i)p(F_i)}.$$

[Hint: Use the fact that $E = \bigcup_{i=1}^n (E \cap F_i)$.]

18. Suppose that a Bayesian spam filter is trained on a set of 500 spam messages and 200 messages that are not spam. The word “exciting” appears in 40 spam messages and in 25 messages that are not spam. Would an incoming message be rejected as spam if it contains the word “exciting” and the threshold for rejecting spam is 0.9?
19. Suppose that a Bayesian spam filter is trained on a set of 1000 spam messages and 400 messages that are not spam. The word “opportunity” appears in 175 spam messages and 20 messages that are not spam. Would an incoming message be rejected as spam if it contains the word “opportunity” and the threshold for rejecting a message is 0.9?
20. Would we reject a message as spam in Example 4
- using just the fact that the word “undervalued” occurs in the message?
 - using just the fact that the word “stock” occurs in the message?
21. Suppose that a Bayesian spam filter is trained on a set of 10,000 spam messages and 5000 messages that are not spam. The word “enhancement” appears in 1500 spam

messages and 20 messages that are not spam, while the word “herbal” appears in 800 spam messages and 200 messages that are not spam. Estimate the probability that a received message containing both the words “enhancement” and “herbal” is spam. Will the message be rejected as spam if the threshold for rejecting spam is 0.9?

22. Suppose that we have prior information concerning whether a random incoming message is spam. In particular, suppose that over a time period, we find that s spam messages arrive and h messages arrive that are not spam.
- Use this information to estimate $p(S)$, the probability that an incoming message is spam, and $p(\bar{S})$, the probability an incoming message is not spam.
 - Use Bayes' theorem and part (a) to estimate the probability that an incoming message containing the word w is spam, where $p(w)$ is the probability that w occurs in a spam message and $q(w)$ is the probability that w occurs in a message that is not spam.
23. Suppose that E_1 and E_2 are the events that an incoming mail message contains the words w_1 and w_2 , respectively. Assuming that E_1 and E_2 are independent events and that $E_1 | S$ and $E_2 | S$ are independent events, where S is the event that an incoming message is spam, and that we have no prior knowledge regarding whether or not the message is spam, show that

$$\begin{aligned} p(S | E_1 \cap E_2) \\ = \frac{p(E_1 | S)p(E_2 | S)}{p(E_1 | S)p(E_2 | S) + p(E_1 | \bar{S})p(E_2 | \bar{S})}. \end{aligned}$$

7.4 Expected Value and Variance

Introduction

The **expected value** of a random variable is the sum over all elements in a sample space of the product of the probability of the element and the value of the random variable at this element. Consequently, the expected value is a weighted average of the values of a random variable. The expected value of a random variable provides a central point for the distribution of values of this random variable. We can solve many problems using the notion of the expected value of a random variable, such as determining who has an advantage in gambling games and computing the average-case complexity of algorithms. Another useful measure of a random variable is its **variance**, which tells us how spread out the values of this random variable are. We can use the variance of a random variable to help us estimate the probability that a random variable takes values far removed from its expected value.

Expected Values



Many questions can be formulated in terms of the value we expect a random variable to take, or more precisely, the average value of a random variable when an experiment is performed a large number of times. Questions of this kind include: How many heads are expected to appear

when a coin is flipped 100 times? What is the expected number of comparisons used to find an element in a list using a linear search? To study such questions we introduce the concept of the expected value of a random variable.

DEFINITION 1

The *expected value*, also called the *expectation* or *mean*, of the random variable X on the sample space S is equal to

$$E(X) = \sum_{s \in S} p(s)X(s).$$

The *deviation* of X at $s \in S$ is $X(s) - E(X)$, the difference between the value of X and the mean of X .

Note that when the sample space S has n elements $S = \{x_1, x_2, \dots, x_n\}$, $E(X) = \sum_{i=1}^n p(x_i)X(x_i)$.

Remark: When there are infinitely many elements of the sample space, the expectation is defined only when the infinite series in the definition is absolutely convergent. In particular, the expectation of a random variable on an infinite sample space is finite if it exists.

EXAMPLE 1 **Expected Value of a Die** Let X be the number that comes up when a fair die is rolled. What is the expected value of X ?

Solution: The random variable X takes the values 1, 2, 3, 4, 5, or 6, each with probability 1/6. It follows that

$$E(X) = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = \frac{21}{6} = \frac{7}{2}. \quad \blacktriangleleft$$

EXAMPLE 2 A fair coin is flipped three times. Let S be the sample space of the eight possible outcomes, and let X be the random variable that assigns to an outcome the number of heads in this outcome. What is the expected value of X ?



Solution: In Example 10 of Section 7.2 we listed the values of X for the eight possible outcomes when a coin is flipped three times. Because the coin is fair and the flips are independent, the probability of each outcome is 1/8. Consequently,

$$\begin{aligned} E(X) &= \frac{1}{8}[X(HHH) + X(HHT) + X(HTH) + X(THH) + X(TTH) \\ &\quad + X(THT) + X(HTT) + X(TTT)] \\ &= \frac{1}{8}(3 + 2 + 2 + 2 + 1 + 1 + 1 + 0) = \frac{12}{8} \\ &= \frac{3}{2}. \end{aligned}$$

Consequently, the expected number of heads that come up when a fair coin is flipped three times is 3/2. \blacktriangleleft

When an experiment has relatively few outcomes, we can compute the expected value of a random variable directly from its definition, as was done in Example 2. However, when an experiment has a large number of outcomes, it may be inconvenient to compute the expected value of a random variable directly from its definition. Instead, we can find the expected value

of a random variable by grouping together all outcomes assigned the same value by the random variable, as Theorem 1 shows.

THEOREM 1

If X is a random variable and $p(X = r)$ is the probability that $X = r$, so that $p(X = r) = \sum_{s \in S, X(s)=r} p(s)$, then

$$E(X) = \sum_{r \in X(S)} p(X = r)r.$$

Proof: Suppose that X is a random variable with range $X(S)$, and let $p(X = r)$ be the probability that the random variable X takes the value r . Consequently, $p(X = r)$ is the sum of the probabilities of the outcomes s such that $X(s) = r$. It follows that

$$E(X) = \sum_{r \in X(S)} p(X = r)r. \quad \triangleleft$$

Example 3 and the proof of Theorem 2 will illustrate the use of this formula. In Example 3 we will find the expected value of the sum of the numbers that appear on two fair dice when they are rolled. In Theorem 2 we will find the expected value of the number of successes when n Bernoulli trials are performed.

EXAMPLE 3 What is the expected value of the sum of the numbers that appear when a pair of fair dice is rolled?

Solution: Let X be the random variable equal to the sum of the numbers that appear when a pair of dice is rolled. In Example 12 of Section 7.2 we listed the value of X for the 36 outcomes of this experiment. The range of X is $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. By Example 12 of Section 7.2 we see that

$$\begin{aligned} p(X = 2) &= p(X = 12) = 1/36, \\ p(X = 3) &= p(X = 11) = 2/36 = 1/18, \\ p(X = 4) &= p(X = 10) = 3/36 = 1/12, \\ p(X = 5) &= p(X = 9) = 4/36 = 1/9, \\ p(X = 6) &= p(X = 8) = 5/36, \\ p(X = 7) &= 6/36 = 1/6. \end{aligned}$$

Substituting these values in the formula, we have

$$\begin{aligned} E(X) &= 2 \cdot \frac{1}{36} + 3 \cdot \frac{1}{18} + 4 \cdot \frac{1}{12} + 5 \cdot \frac{1}{9} + 6 \cdot \frac{5}{36} + 7 \cdot \frac{1}{6} \\ &\quad + 8 \cdot \frac{5}{36} + 9 \cdot \frac{1}{9} + 10 \cdot \frac{1}{12} + 11 \cdot \frac{1}{18} + 12 \cdot \frac{1}{36} \\ &= 7. \end{aligned} \quad \triangleleft$$

THEOREM 2

The expected number of successes when n mutually independent Bernoulli trials are performed, where p is the probability of success on each trial, is np .

Proof: Let X be the random variable equal to the number of successes in n trials. By Theorem 2 of Section 7.2 we see that $p(X = k) = C(n, k)p^k q^{n-k}$. Hence, we have

$$\begin{aligned}
 E(X) &= \sum_{k=1}^n kp(X = k) && \text{by Theorem 1} \\
 &= \sum_{k=1}^n kC(n, k)p^k q^{n-k} && \text{by Theorem 2 in Section 7.2} \\
 &= \sum_{k=1}^n nC(n-1, k-1)p^k q^{n-k} && \text{by Exercise 21 in Section 6.4} \\
 &= np \sum_{k=1}^n C(n-1, k-1)p^{k-1} q^{n-k} && \text{factoring } np \text{ from each term} \\
 &= np \sum_{j=0}^{n-1} C(n-1, j)p^j q^{n-1-j} && \text{shifting index of summation with } j = k - 1 \\
 &= np(p+q)^{n-1} && \text{by the binomial theorem} \\
 &= np. && \text{because } p+q=1
 \end{aligned}$$

This completes the proof because it shows that the expected number of successes in n mutually independent Bernoulli trials is np . \triangleleft

We will also show that the hypothesis that the Bernoulli trials are mutually independent in Theorem 2 is not necessary.

Linearity of Expectations

Theorem 3 tells us that expected values are linear. For example, the expected value of the sum of random variables is the sum of their expected values. We will find this property exceedingly useful.

THEOREM 3

If $X_i, i = 1, 2, \dots, n$ with n a positive integer, are random variables on S , and if a and b are real numbers, then

- (i) $E(X_1 + X_2 + \dots + X_n) = E(X_1) + E(X_2) + \dots + E(X_n)$
- (ii) $E(aX + b) = aE(X) + b$.

Proof: Part (i) follows for $n = 2$ directly from the definition of expected value, because

$$\begin{aligned}
 E(X_1 + X_2) &= \sum_{s \in S} p(s)(X_1(s) + X_2(s)) \\
 &= \sum_{s \in S} p(s)X_1(s) + \sum_{s \in S} p(s)X_2(s) \\
 &= E(X_1) + E(X_2).
 \end{aligned}$$

The case for n random variables follows easily by mathematical induction using the case of two random variables. (We leave it to the reader to complete the proof.)

To prove part (ii), note that

$$\begin{aligned} E(aX + b) &= \sum_{s \in S} p(s)(aX(s) + b) \\ &= a \sum_{s \in S} p(s)X(s) + b \sum_{s \in S} p(s) \\ &= aE(X) + b \text{ because } \sum_{s \in S} p(s) = 1. \end{aligned}$$



Examples 4 and 5 illustrate how to use Theorem 3.

EXAMPLE 4 Use Theorem 3 to find the expected value of the sum of the numbers that appear when a pair of fair dice is rolled. (This was done in Example 3 without the benefit of this theorem.)

Solution: Let X_1 and X_2 be the random variables with $X_1((i, j)) = i$ and $X_2((i, j)) = j$, so that X_1 is the number appearing on the first die and X_2 is the number appearing on the second die. It is easy to see that $E(X_1) = E(X_2) = 7/2$ because both equal $(1 + 2 + 3 + 4 + 5 + 6)/6 = 21/6 = 7/2$. The sum of the two numbers that appear when the two dice are rolled is the sum $X_1 + X_2$. By Theorem 3, the expected value of the sum is $E(X_1 + X_2) = E(X_1) + E(X_2) = 7/2 + 7/2 = 7$.



EXAMPLE 5 In the proof of Theorem 2 we found the expected value of the number of successes when n independent Bernoulli trials are performed, where p is the probability of success on each trial by direct computation. Show how Theorem 3 can be used to derive this result where the Bernoulli trials are not necessarily independent.

Solution: Let X_i be the random variable with $X_i((t_1, t_2, \dots, t_n)) = 1$ if t_i is a success and $X_i((t_1, t_2, \dots, t_n)) = 0$ if t_i is a failure. The expected value of X_i is $E(X_i) = 1 \cdot p + 0 \cdot (1 - p) = p$ for $i = 1, 2, \dots, n$. Let $X = X_1 + X_2 + \dots + X_n$, so that X counts the number of successes when these n Bernoulli trials are performed. Theorem 3, applied to the sum of n random variables, shows that $E(X) = E(X_1) + E(X_2) + \dots + E(X_n) = np$.



We can take advantage of the linearity of expectations to find the solutions of many seemingly difficult problems. The key step is to express a random variable whose expectation we wish to find as the sum of random variables whose expectations are easy to find. Examples 6 and 7 illustrate this technique.

EXAMPLE 6 **Expected Value in the Hatchet Problem** A new employee checks the hats of n people at a restaurant, forgetting to put claim check numbers on the hats. When customers return for their hats, the checker gives them back hats chosen at random from the remaining hats. What is the expected number of hats that are returned correctly?

Solution: Let X be the random variable that equals the number of people who receive the correct hat from the checker. Let X_i be the random variable with $X_i = 1$ if the i th person receives the correct hat and $X_i = 0$ otherwise. It follows that

$$X = X_1 + X_2 + \dots + X_n.$$

Because it is equally likely that the checker returns any of the hats to this person, it follows that the probability that the i th person receives the correct hat is $1/n$. Consequently, by Theorem 1, for all i we have

$$E(X_i) = 1 \cdot p(X_i = 1) + 0 \cdot p(X_i = 0) = 1 \cdot 1/n + 0 = 1/n.$$

By the linearity of expectations (Theorem 3), it follows that

$$E(X) = E(X_1) + E(X_2) + \cdots + E(X_n) = n \cdot 1/n = 1.$$

Consequently, the average number of people who receive the correct hat is exactly 1. Note that this answer is independent of the number of people who have checked their hats! (We will find an explicit formula for the probability that no one receives the correct hat in Example 4 of Section 8.6.) 

EXAMPLE 7

Expected Number of Inversions in a Permutation The ordered pair (i, j) is called an **inversion** in a permutation of the first n positive integers if $i < j$ but j precedes i in the permutation. For instance, there are six inversions in the permutation 3, 5, 1, 4, 2; these inversions are

$$(1, 3), (1, 5), (2, 3), (2, 4), (2, 5), (4, 5).$$

Let $I_{i,j}$ be the random variable on the set of all permutations of the first n positive integers with $I_{i,j} = 1$ if (i, j) is an inversion of the permutation and $I_{i,j} = 0$ otherwise. It follows that if X is the random variable equal to the number of inversions in the permutation, then

$$X = \sum_{1 \leq i < j \leq n} I_{i,j}.$$



Note that it is equally likely for i to precede j in a randomly chosen permutation as it is for j to precede i . (To see this, note that there are an equal number of permutations with each of these properties.) Consequently, for all pairs i and j we have

$$E(I_{i,j}) = 1 \cdot p(I_{i,j} = 1) + 0 \cdot p(I_{i,j} = 0) = 1 \cdot 1/2 + 0 = 1/2.$$

Because there are $\binom{n}{2}$ pairs i and j with $1 \leq i < j \leq n$ and by the linearity of expectations (Theorem 3), we have

$$E(X) = \sum_{1 \leq i < j \leq n} E(I_{i,j}) = \binom{n}{2} \cdot \frac{1}{2} = \frac{n(n-1)}{4}.$$

It follows that there are an average of $n(n-1)/4$ inversions in a permutation of the first n positive integers. 

Average-Case Computational Complexity



Computing the average-case computational complexity of an algorithm can be interpreted as computing the expected value of a random variable. Let the sample space of an experiment be the set of possible inputs a_j , $j = 1, 2, \dots, n$, and let X be the random variable that assigns to a_j the number of operations used by the algorithm when given a_j as input. Based on our knowledge of the input, we assign a probability $p(a_j)$ to each possible input value a_j . Then, the average-case complexity of the algorithm is

$$E(X) = \sum_{j=1}^n p(a_j)X(a_j).$$

This is the expected value of X .

Finding the average-case computational complexity of an algorithm is usually much more difficult than finding its worst-case computational complexity, and often involves the use of sophisticated methods. However, there are some algorithms for which the analysis required to find the average-case computational complexity is not difficult. For instance, in Example 8 we will illustrate how to find the average-case computational complexity of the linear search algorithm under different assumptions concerning the probability that the element for which we search is an element of the list.

EXAMPLE 8 **Average-Case Complexity of the Linear Search Algorithm** We are given a real number x and a list of n distinct real numbers. The linear search algorithm, described in Section 3.1, locates x by successively comparing it to each element in the list, terminating when x is located or when all the elements have been examined and it has been determined that x is not in the list. What is the average-case computational complexity of the linear search algorithm if the probability that x is in the list is p and it is equally likely that x is any of the n elements in the list? (There are $n + 1$ possible types of input: one type for each of the n numbers in the list and a last type for numbers not in the list, which we treat as a single input.)

Solution: In Example 4 of Section 3.3 we showed that $2i + 1$ comparisons are used if x equals the i th element of the list and, in Example 2 of Section 3.3, we showed that $2n + 2$ comparisons are used if x is not in the list. The probability that x equals a_i , the i th element in the list, is p/n , and the probability that x is not in the list is $q = 1 - p$. It follows that the average-case computational complexity of the linear search algorithm is

$$\begin{aligned} E &= \frac{3p}{n} + \frac{5p}{n} + \cdots + \frac{(2n+1)p}{n} + (2n+2)q \\ &= \frac{p}{n}(3 + 5 + \cdots + (2n+1)) + (2n+2)q \\ &= \frac{p}{n}((n+1)^2 - 1) + (2n+2)q \\ &= p(n+2) + (2n+2)q. \end{aligned}$$

(The third equality follows from Example 2 of Section 5.1.) For instance, when x is guaranteed to be in the list, we have $p = 1$ (so the probability that $x = a_i$ is $1/n$ for each i) and $q = 0$. Then $E = n + 2$, as we showed in Example 4 in Section 3.3.

When p , the probability that x is in the list, is $1/2$, it follows that $q = 1 - p = 1/2$, so $E = (n+2)/2 + n + 1 = (3n+4)/2$. Similarly, if the probability that x is in the list is $3/4$, we have $p = 3/4$ and $q = 1/4$, so $E = 3(n+2)/4 + (n+1)/2 = (5n+8)/4$.

Finally, when x is guaranteed not to be in the list, we have $p = 0$ and $q = 1$. It follows that $E = 2n + 2$, which is not surprising because we have to search the entire list. 

Example 9 illustrates how the linearity of expectations can help us find the average-case complexity of a sorting algorithm, the insertion sort.

EXAMPLE 9 **Average-Case Complexity of the Insertion Sort** What is the average number of comparisons used by the insertion sort to sort n distinct elements?

Solution: We first suppose that X is the random variable equal to the number of comparisons used by the insertion sort (described in Section 3.1) to sort a list a_1, a_2, \dots, a_n of n distinct elements. Then $E(X)$ is the average number of comparisons used. (Recall that at step i for $i = 2, \dots, n$, the insertion sort inserts the i th element in the original list into the correct position in the sorted list of the first $i - 1$ elements of the original list.)

We let X_i be the random variable equal to the number of comparisons used to insert a_i into the proper position after the first $i - 1$ elements a_1, a_2, \dots, a_{i-1} have been sorted. Because

$$X = X_2 + X_3 + \cdots + X_n,$$

we can use the linearity of expectations to conclude that

$$E(X) = E(X_2 + X_3 + \cdots + X_n) = E(X_2) + E(X_3) + \cdots + E(X_n).$$

To find $E(X_i)$ for $i = 2, 3, \dots, n$, let $p_j(k)$ denote the probability that the largest of the first j elements in the list occurs at the k th position, that is, that $\max(a_1, a_2, \dots, a_j) = a_k$, where $1 \leq k \leq j$. Because the elements of the list are randomly distributed, it is equally likely for the largest element among the first j elements to occur at any position. Consequently, $p_j(k) = 1/j$. If $X_i(k)$ equals the number of comparisons used by the insertion sort if a_i is inserted into the k th position in the list once a_1, a_2, \dots, a_{i-1} have been sorted, it follows that $X_i(k) = k$. Because it is possible that a_i is inserted in any of the first i positions, we find that

$$E(X_i) = \sum_{k=1}^i p_i(k) \cdot X_i(k) = \sum_{k=1}^i \frac{1}{i} \cdot k = \frac{1}{i} \cdot \sum_{k=1}^i k = \frac{1}{i} \cdot \frac{i(i+1)}{2} = \frac{i+1}{2}.$$

It follows that

$$\begin{aligned} E(X) &= \sum_{i=2}^n E(X_i) = \sum_{i=2}^n \frac{i+1}{2} = \frac{1}{2} \sum_{j=3}^{n+1} j \\ &= \frac{1}{2} \frac{(n+1)(n+2)}{2} - \frac{1}{2}(1+2) = \frac{n^2 + 3n - 4}{4}. \end{aligned}$$

To obtain the third of these equalities we shifted the index of summation, setting $j = i + 1$. To obtain the fourth equality, we used the formula $\sum_{k=1}^m k = m(m+1)/2$ (from Table 2 in Section 2.4) with $m = n + 1$, subtracting off the missing terms with $j = 1$ and $j = 2$. We conclude that the average number of comparisons used by the insertion sort to sort n elements equals $(n^2 + 3n - 4)/4$, which is $\Theta(n^2)$. 

The Geometric Distribution

We now turn our attention to a random variable with infinitely many possible outcomes.

EXAMPLE 10 Suppose that the probability that a coin comes up tails is p . This coin is flipped repeatedly until it comes up tails. What is the expected number of flips until this coin comes up tails?



Solution: We first note that the sample space consists of all sequences that begin with any number of heads, denoted by H , followed by a tail, denoted by T . Therefore, the sample space is the set $\{T, HT, HHT, HHHT, HHHHT, \dots\}$. Note that this is an infinite sample space. We can determine the probability of an element of the sample space by noting that the coin flips are independent and that the probability of a head is $1 - p$. Therefore, $p(T) = p$, $p(HT) = (1 - p)p$, $p(HHT) = (1 - p)^2 p$, and in general the probability that the coin is flipped n times before a tail comes up, that is, that $n - 1$ heads come up followed by a tail, is $(1 - p)^{n-1} p$. (Exercise 14 asks for a verification that the sum of the probabilities of the points in the sample space is 1.)

Now let X be the random variable equal to the number of flips in an element in the sample space. That is, $X(T) = 1$, $X(HT) = 2$, $X(HHT) = 3$, and so on. Note that $p(X = j) = (1 - p)^{j-1} p$. The expected number of flips until the coin comes up tails equals $E(X)$.

Using Theorem 1, we find that

$$E(X) = \sum_{j=1}^{\infty} j \cdot p(X = j) = \sum_{j=1}^{\infty} j(1 - p)^{j-1} p = p \sum_{j=1}^{\infty} j(1 - p)^{j-1} = p \cdot \frac{1}{p^2} = \frac{1}{p}.$$

[The third equality in this chain follows from Table 2 in Section 2.4, which tells us that $\sum_{j=1}^{\infty} j(1 - p)^{j-1} = 1/(1 - (1 - p))^2 = 1/p^2$.] It follows that the expected number of times the coin is flipped until tails comes up is $1/p$. Note that when the coin is fair we have $p = 1/2$, so the expected number of flips until it comes up tails is $1/(1/2) = 2$. 

The random variable X that equals the number of flips expected before a coin comes up tails is an example of a random variable with a **geometric distribution**.

DEFINITION 2

A random variable X has a *geometric distribution with parameter p* if $p(X = k) = (1 - p)^{k-1} p$ for $k = 1, 2, 3, \dots$, where p is a real number with $0 \leq p \leq 1$.

Geometric distributions arise in many applications because they are used to study the time required before a particular event happens, such as the time required before we find an object with a certain property, the number of attempts before an experiment succeeds, the number of times a product can be used before it fails, and so on.

When we computed the expected value of the number of flips required before a coin comes up tails, we proved Theorem 4.

THEOREM 4

If the random variable X has the geometric distribution with parameter p , then $E(X) = 1/p$.

Independent Random Variables

We have already discussed independent events. We will now define what it means for two random variables to be independent.

DEFINITION 3

The random variables X and Y on a sample space S are *independent* if

$$p(X = r_1 \text{ and } Y = r_2) = p(X = r_1) \cdot p(Y = r_2),$$

or in words, if the probability that $X = r_1$ and $Y = r_2$ equals the product of the probabilities that $X = r_1$ and $Y = r_2$, for all real numbers r_1 and r_2 .

EXAMPLE 11

Are the random variables X_1 and X_2 from Example 4 independent?



Solution: Let $S = \{1, 2, 3, 4, 5, 6\}$, and let $i \in S$ and $j \in S$. Because there are 36 possible outcomes when the pair of dice is rolled and each is equally likely, we have

$$p(X_1 = i \text{ and } X_2 = j) = 1/36.$$

Furthermore, $p(X_1 = i) = 1/6$ and $p(X_2 = j) = 1/6$, because the probability that i appears on the first die and the probability that j appears on the second die are both $1/6$. It follows that

$$p(X_1 = i \text{ and } X_2 = j) = \frac{1}{36} \quad \text{and} \quad p(X_1 = i)p(X_2 = j) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36},$$

so X_1 and X_2 are independent. 

EXAMPLE 12 Show that the random variables X_1 and $X = X_1 + X_2$, where X_1 and X_2 are as defined in Example 4, are not independent.

Solution: Note that $p(X_1 = 1 \text{ and } X = 12) = 0$, because $X_1 = 1$ means the number appearing on the first die is 1, which implies that the sum of the numbers appearing on the two dice cannot equal 12. On the other hand, $p(X_1 = 1) = 1/6$ and $p(X = 12) = 1/36$. Hence $p(X_1 = 1 \text{ and } X = 12) \neq p(X_1 = 1) \cdot p(X = 12)$. This counterexample shows that X_1 and X are not independent. 

The expected value of the product of two independent random variables is the product of their expected values, as Theorem 5 shows.

THEOREM 5

If X and Y are independent random variables on a sample space S , then $E(XY) = E(X)E(Y)$.

Proof: To prove this formula, we use the key observation that the event $XY = r$ is the disjoint union of the events $X = r_1$ and $Y = r_2$ over all $r_1 \in X(S)$ and $r_2 \in Y(S)$ with $r = r_1r_2$. We have

$$\begin{aligned} E(XY) &= \sum_{r \in XY(S)} r \cdot p(XY = r) && \text{by Theorem 1} \\ &= \sum_{r_1 \in X(S), r_2 \in Y(S)} r_1 r_2 \cdot p(X = r_1 \text{ and } Y = r_2) && \text{expressing } XY = r \text{ as a disjoint union} \\ &= \sum_{r_1 \in X(S)} \sum_{r_2 \in Y(S)} r_1 r_2 \cdot p(X = r_1 \text{ and } Y = r_2) && \text{using a double sum to order the terms} \\ &= \sum_{r_1 \in X(S)} \sum_{r_2 \in Y(S)} r_1 r_2 \cdot p(X = r_1) \cdot p(Y = r_2) && \text{by the independence of } X \text{ and } Y \\ &= \sum_{r_1 \in X(S)} (r_1 \cdot p(X = r_1) \cdot \sum_{r_2 \in Y(S)} r_2 \cdot p(Y = r_2)) && \text{by factoring out } r_1 \cdot p(X = r_1) \\ &= \sum_{r_1 \in X(S)} r_1 \cdot p(X = r_1) \cdot E(Y) && \text{by the definition of } E(Y) \\ &= E(Y) \left(\sum_{r_1 \in X(S)} r_1 \cdot p(X = r_1) \right) && \text{by factoring out } E(Y) \\ &= E(Y)E(X) && \text{by the definition of } E(X) \end{aligned}$$

We complete the proof by noting that $E(Y)E(X) = E(X)E(Y)$, which is a consequence of the commutative law for multiplication. 

Note that when X and Y are random variables that are not independent, we cannot conclude that $E(XY) = E(X)E(Y)$, as Example 13 shows.

EXAMPLE 13 Let X and Y be random variables that count the number of heads and the number of tails when a coin is flipped twice. Because $p(X = 2) = 1/4$, $p(X = 1) = 1/2$, and $p(X = 0) = 1/4$, by Theorem 1 we have

$$E(X) = 2 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} = 1.$$

A similar computation shows that $E(Y) = 1$. We note that $XY = 0$ when either two heads and no tails or two tails and no heads come up and that $XY = 1$ when one head and one tail come up. Hence,

$$E(XY) = 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}.$$

It follows that

$$E(XY) \neq E(X)E(Y).$$

This does not contradict Theorem 5 because X and Y are not independent, as the reader should verify (see Exercise 16). 

Variance



The expected value of a random variable tells us its average value, but nothing about how widely its values are distributed. For example, if X and Y are the random variables on the set $S = \{1, 2, 3, 4, 5, 6\}$, with $X(s) = 0$ for all $s \in S$ and $Y(s) = -1$ if $s \in \{1, 2, 3\}$ and $Y(s) = 1$ if $s \in \{4, 5, 6\}$, then the expected values of X and Y are both zero. However, the random variable X never varies from 0, while the random variable Y always differs from 0 by 1. The variance of a random variable helps us characterize how widely a random variable is distributed. In particular, it provides a measure of how widely X is distributed about its expected value.

DEFINITION 4

Let X be a random variable on a sample space S . The *variance* of X , denoted by $V(X)$, is

$$V(X) = \sum_{s \in S} (X(s) - E(X))^2 p(s).$$

That is, $V(X)$ is the weighted average of the square of the deviation of X . The *standard deviation* of X , denoted $\sigma(X)$, is defined to be $\sqrt{V(X)}$.

Theorem 6 provides a useful simple expression for the variance of a random variable.

THEOREM 6

If X is a random variable on a sample space S , then $V(X) = E(X^2) - E(X)^2$.

Proof: Note that

$$\begin{aligned} V(X) &= \sum_{s \in S} (X(s) - E(X))^2 p(s) \\ &= \sum_{s \in S} X(s)^2 p(s) - 2E(X) \sum_{s \in S} X(s)p(s) + E(X)^2 \sum_{s \in S} p(s) \\ &= E(X^2) - 2E(X)E(X) + E(X)^2 \\ &= E(X^2) - E(X)^2. \end{aligned}$$

We have used the fact that $\sum_{s \in S} p(s) = 1$ in the next-to-last step. 

We can use Theorems 3 and 6 to derive an alternative formula for $V(X)$ that provides some insight into the meaning of the variance of a random variable.

COROLLARY 1

If X is a random variable on a sample space S and $E(X) = \mu$, then $V(X) = E((X - \mu)^2)$.

μ is the Greek letter mu.

Proof: If X is a random variable with $E(X) = \mu$, then

$$\begin{aligned}
 E((X - \mu)^2) &= E(X^2 - 2\mu X + \mu^2) && \text{expanding } (X - \mu)^2 \\
 &= E(X^2) - E(2\mu X) + E(\mu^2) && \text{by part (i) of Theorem 3} \\
 &= E(X^2) - 2\mu E(X) + E(\mu^2) && \text{by part (ii) of Theorem 3, noting that } \mu \text{ is a constant} \\
 &= E(X^2) - 2\mu E(X) + \mu^2 && \text{as } E(\mu^2) = \mu^2, \text{ because } \mu^2 \text{ is a constant} \\
 &= E(X^2) - 2\mu^2 + \mu^2 && \text{because } E(X) = \mu \\
 &= E(X^2) - \mu^2 && \text{simplifying} \\
 &= V(X) && \text{by Theorem 6 and noting that } E(X) = \mu.
 \end{aligned}$$

This completes the proof. \triangleleft

Corollary 1 tells us that the variance of a random variable X is the expected value of the square of the difference between X and its own expected value. This is commonly expressed as saying that the variance of X is the mean of the square of its deviation. We also say that the standard deviation of X is the square root of the mean of the square of its deviation (often read as the “root mean square” of the deviation).

We now compute the variance of some random variables.

EXAMPLE 14

What is the variance of the random variable X with $X(t) = 1$ if a Bernoulli trial is a success and $X(t) = 0$ if it is a failure, where p is the probability of success and q is the probability of failure?



Solution: Because X takes only the values 0 and 1, it follows that $X^2(t) = X(t)$. Hence,

$$V(X) = E(X^2) - E(X)^2 = p - p^2 = p(1 - p) = pq.$$

**EXAMPLE 15**

Variance of the Value of a Die What is the variance of the random variable X , where X is the number that comes up when a fair die is rolled?

Solution: We have $V(X) = E(X^2) - E(X)^2$. By Example 1 we know that $E(X) = 7/2$. To find $E(X^2)$ note that X^2 takes the values i^2 , $i = 1, 2, \dots, 6$, each with probability $1/6$. It follows that

$$E(X^2) = \frac{1}{6}(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2) = \frac{91}{6}.$$

We conclude that

$$V(X) = \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{35}{12}. \quad \blacktriangleleft$$

EXAMPLE 16 What is the variance of the random variable $X((i, j)) = 2i$, where i is the number appearing on the first die and j is the number appearing on the second die, when two fair dice are rolled?

Solution: We will use Theorem 6 to find the variance of X . To do so, we need to find the expected values of X and X^2 . Note that because $p(X = k)$ is $1/6$ for $k = 2, 4, 6, 8, 10, 12$ and is 0 otherwise,

$$E(X) = (2 + 4 + 6 + 8 + 10 + 12)/6 = 7,$$

and

$$E(X^2) = (2^2 + 4^2 + 6^2 + 8^2 + 10^2 + 12^2)/6 = 182/3.$$

It follows from Theorem 6 that

$$V(X) = E(X^2) - E(X)^2 = 182/3 - 49 = 35/3. \quad \blacktriangleleft$$

Another useful property is that the variance of the sum of two or more independent random variables is the sum of their variances. The formula that expresses this property is known as **Bienaym 's formula**, after Iren  -Jules Bienaym , the French mathematician who discovered it in 1853. Bienaym 's formula is useful for computing the variance of the result of n independent Bernoulli trials, for instance.

THEOREM 7

BIENAYM 'S FORMULA If X and Y are two independent random variables on a sample space S , then $V(X + Y) = V(X) + V(Y)$. Furthermore, if X_i , $i = 1, 2, \dots, n$, with n a positive integer, are pairwise independent random variables on S , then $V(X_1 + X_2 + \dots + X_n) = V(X_1) + V(X_2) + \dots + V(X_n)$.



IREN  -JULES BIENAYM  (1796–1878) Bienaym , born in Paris, moved with his family to Bruges in 1803 when his father became a government administrator. Bienaym  attended the Lyc   imp  rial in Bruges, and when his family returned to Paris in 1811, the Lyc   Louis-le-Grand. As a teenager, he helped defend Paris during the 1814 Napoleonic Wars; in 1815, he became a student at the ´Ecole Polytechnique. In 1816 he joined the Ministry of Finances to help support his family. In 1819, he left the civil service, taking a job lecturing mathematics at the Acad  mie militaire de Saint-Cyr. Unhappy with conditions there, he soon returned to the Ministry of Finances. He attained the position of inspector general, remaining until forced to retire in 1848 for political reasons. He was able to return as inspector general in 1850, but he retired a second time in 1852. In 1851 he briefly was professor at the Sorbonne and also served as an expert statistician for Napoleon III.

Bienaym  was one of the founders of the Soci  t   Math  matique de France, and in 1875 was its president.

Bienaym  was noted for his ingenuity, but his papers frustrated readers by omitting important proofs. He published sparsely, often in obscure journals. However, he made important contributions to probability and statistics, and to their applications to the social sciences and to finance. Among his important contributions are the Bienaym -Chebyshev inequality, which provides a simple proof of the law of large numbers, a generalization of Laplace's least square method, and Bienaym 's formula for the variance of a sum of random variables. He studied the extinction of aristocratic families, declining despite general population growth. Bienaym  was a skilled linguist; he translated the works of Chebyshev, a close friend, from Russian to French. It has been suggested that his relative obscurity results from his modesty, his lack of interest in asserting the priority of his discoveries, and the fact that his work was often ahead of its time. He and his brother married two sisters who were daughters of a family friend. Bienaym  and his wife had two sons and three daughters.

Proof: From Theorem 6, we have

$$V(X + Y) = E((X + Y)^2) - E(X + Y)^2.$$

It follows that

$$\begin{aligned} V(X + Y) &= E(X^2 + 2XY + Y^2) - (E(X) + E(Y))^2 \\ &= E(X^2) + 2E(XY) + E(Y^2) - E(X)^2 - 2E(X)E(Y) - E(Y)^2. \end{aligned}$$

Because X and Y are independent, by Theorem 5 we have $E(XY) = E(X)E(Y)$. It follows that

$$\begin{aligned} V(X + Y) &= (E(X^2) - E(X)^2) + (E(Y^2) - E(Y)^2) \\ &= V(X) + V(Y). \end{aligned}$$

We leave the proof of the case for n pairwise independent random variables to the reader (Exercise 34). Such a proof can be constructed by generalizing the proof we have given for the case for two random variables. Note that it is not possible to use mathematical induction in a straightforward way to prove the general case (see Exercise 33). \triangleleft

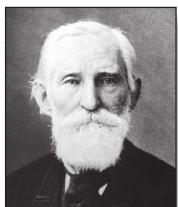
EXAMPLE 17 Find the variance and standard deviation of the random variable X whose value when two fair dice are rolled is $X((i, j)) = i + j$, where i is the number appearing on the first die and j is the number appearing on the second die.

Solution: Let X_1 and X_2 be the random variables defined by $X_1((i, j)) = i$ and $X_2((i, j)) = j$ for a roll of the dice. Then $X = X_1 + X_2$, and X_1 and X_2 are independent, as Example 11 showed. From Theorem 7 it follows that $V(X) = V(X_1) + V(X_2)$. A simple computation as in Example 16, together with Exercise 29 in the Supplementary Exercises, tells us that $V(X_1) = V(X_2) = 35/12$. Hence, $V(X) = 35/12 + 35/12 = 35/6$ and $\sigma(X) = \sqrt{35/6}$. \triangleleft

We will now find the variance of the random variable that counts the number of successes when n independent Bernoulli trials are carried out.

EXAMPLE 18 What is the variance of the number of successes when n independent Bernoulli trials are performed, where, on each trial, p is the probability of success and q is the probability of failure?

Solution: Let X_i be the random variable with $X_i((t_1, t_2, \dots, t_n)) = 1$ if trial t_i is a success and $X_i((t_1, t_2, \dots, t_n)) = 0$ if trial t_i is a failure. Let $X = X_1 + X_2 + \dots + X_n$. Then X counts the number of successes in the n trials. From Theorem 7 it follows that $V(X) = V(X_1) + V(X_2) + \dots + V(X_n)$. Using Example 14 we have $V(X_i) = pq$ for $i = 1, 2, \dots, n$. It follows that $V(X) = npq$. \triangleleft



PAFNUTY LVOVICH CHEBYSHEV (1821–1894) Chebyshev was born into the gentry in Okatovo, Russia. His father was a retired army officer who had fought against Napoleon. In 1832 the family, with its nine children, moved to Moscow, where Pafnuty completed his high school education at home. He entered the Department of Physics and Mathematics at Moscow University. As a student, he developed a new method for approximating the roots of equations. He graduated from Moscow University in 1841 with a degree in mathematics, and he continued his studies, passing his master's exam in 1843 and completing his master's thesis in 1846.

Chebyshev was appointed in 1847 to a position as an assistant at the University of St. Petersburg. He wrote and defended a thesis in 1847. He became a professor at St. Petersburg in 1860, a position he held until 1882. His book on the theory of congruences written in 1849 was influential in the development of number theory. His work on the distribution of prime numbers was seminal. He proved Bertrand's conjecture that for every integer $n > 3$, there is a prime between n and $2n - 2$. Chebyshev helped develop ideas that were later used to prove the prime number theorem. Chebyshev's work on the approximation of functions using polynomials is used extensively when computers are used to find values of functions. Chebyshev was also interested in mechanics. He studied the conversion of rotary motion into rectilinear motion by mechanical coupling. The Chebyshev parallel motion is three linked bars approximating rectilinear motion.

Chebyshev's Inequality

How likely is it that a random variable takes a value far from its expected value? Theorem 8, called Chebyshev's inequality, helps answer this question by providing an upper bound on the probability that the value of a random variable differs from the expected value of the random variable by more than a specified amount.

THEOREM 8

CHEBYSHEV'S INEQUALITY Let X be a random variable on a sample space S with probability function p . If r is a positive real number, then

$$p(|X(s) - E(X)| \geq r) \leq V(X)/r^2.$$

Proof: Let A be the event

$$A = \{s \in S \mid |X(s) - E(X)| \geq r\}.$$

What we want to prove is that $p(A) \leq V(X)/r^2$. Note that

$$\begin{aligned} V(X) &= \sum_{s \in S} (X(s) - E(X))^2 p(s) \\ &= \sum_{s \in A} (X(s) - E(X))^2 p(s) + \sum_{s \notin A} (X(s) - E(X))^2 p(s). \end{aligned}$$

The second sum in this expression is nonnegative, because each of its summands is nonnegative. Also, because for each element s in A , $(X(s) - E(X))^2 \geq r^2$, the first sum in this expression is at least $\sum_{s \in A} r^2 p(s)$. Hence, $V(X) \geq \sum_{s \in A} r^2 p(s) = r^2 p(A)$. It follows that $V(X)/r^2 \geq p(A)$, so $p(A) \leq V(X)/r^2$, completing the proof. \triangleleft

EXAMPLE 19

Deviation from the Mean when Counting Tails Suppose that X is the random variable that counts the number of tails when a fair coin is tossed n times. Note that X is the number of successes when n independent Bernoulli trials, each with probability of success $1/2$, are performed. It follows that $E(X) = n/2$ (by Theorem 2) and $V(X) = n/4$ (by Example 18). Applying Chebyshev's inequality with $r = \sqrt{n}$ shows that

$$p(|X(s) - n/2| \geq \sqrt{n}) \leq (n/4)/(\sqrt{n})^2 = 1/4.$$

Consequently, the probability is no more than $1/4$ that the number of tails that come up when a fair coin is tossed n times deviates from the mean by more than \sqrt{n} . \triangleleft

Chebyshev's inequality, although applicable to any random variable, often fails to provide a practical estimate for the probability that the value of a random variable exceeds its mean by a large amount. This is illustrated by Example 20.

EXAMPLE 20

Let X be the random variable whose value is the number appearing when a fair die is rolled. We have $E(X) = 7/2$ (see Example 1) and $V(X) = 35/12$ (see Example 15). Because the only possible values of X are 1, 2, 3, 4, 5, and 6, X cannot take a value more than $5/2$ from its mean, $E(X) = 7/2$. Hence, $p(|X - 7/2| \geq r) = 0$ if $r > 5/2$. By Chebyshev's inequality we know that $p(|X - 7/2| \geq r) \leq (35/12)/r^2$.

For example, when $r = 3$, Chebyshev's inequality tells us that $p(|X - 7/2| \geq 3) \leq (35/12)/9 = 35/108 \approx 0.324$, which is a poor estimate, because $p(|X - 7/2| \geq 3) = 0$. \triangleleft

Exercises

1. What is the expected number of heads that come up when a fair coin is flipped five times?
 2. What is the expected number of heads that come up when a fair coin is flipped 10 times?
 3. What is the expected number of times a 6 appears when a fair die is rolled 10 times?
 4. A coin is biased so that the probability a head comes up when it is flipped is 0.6. What is the expected number of heads that come up when it is flipped 10 times?
 5. What is the expected sum of the numbers that appear on two dice, each biased so that a 3 comes up twice as often as each other number?
 6. What is the expected value when a \$1 lottery ticket is bought in which the purchaser wins exactly \$10 million if the ticket contains the six winning numbers chosen from the set $\{1, 2, 3, \dots, 50\}$ and the purchaser wins nothing otherwise?
 7. The final exam of a discrete mathematics course consists of 50 true/false questions, each worth two points, and 25 multiple-choice questions, each worth four points. The probability that Linda answers a true/false question correctly is 0.9, and the probability that she answers a multiple-choice question correctly is 0.8. What is her expected score on the final?
 8. What is the expected sum of the numbers that appear when three fair dice are rolled?
 9. Suppose that the probability that x is in a list of n distinct integers is $2/3$ and that it is equally likely that x equals any element in the list. Find the average number of comparisons used by the linear search algorithm to find x or to determine that it is not in the list.
 10. Suppose that we flip a fair coin until either it comes up tails twice or we have flipped it six times. What is the expected number of times we flip the coin?
 11. Suppose that we roll a fair die until a 6 comes up or we have rolled it 10 times. What is the expected number of times we roll the die?
 12. Suppose that we roll a fair die until a 6 comes up.
 - What is the probability that we roll the die n times?
 - What is the expected number of times we roll the die?
 13. Suppose that we roll a pair of fair dice until the sum of the numbers on the dice is seven. What is the expected number of times we roll the dice?
 14. Show that the sum of the probabilities of a random variable with geometric distribution with parameter p , where $0 < p \leq 1$, equals 1.
 15. Show that if the random variable X has the geometric distribution with parameter p , and j is a positive integer, then $p(X \geq j) = (1 - p)^{j-1}$.
 16. Let X and Y be the random variables that count the number of heads and the number of tails that come up when two fair coins are flipped. Show that X and Y are not independent.
 17. Estimate the expected number of integers with 1000 digits that need to be selected at random to find a prime, if the probability a number with 1000 digits is prime is approximately $1/2302$.
 18. Suppose that X and Y are random variables and that X and Y are nonnegative for all points in a sample space S . Let Z be the random variable defined by $Z(s) = \max(X(s), Y(s))$ for all elements $s \in S$. Show that $E(Z) \leq E(X) + E(Y)$.
 19. Let X be the number appearing on the first die when two fair dice are rolled and let Y be the sum of the numbers appearing on the two dice. Show that $E(X)E(Y) \neq E(XY)$.
 - *20. Show that if X_1, X_2, \dots, X_n are mutually independent random variables, then $E(\prod_{i=1}^n X_i) = \prod_{i=1}^n E(X_i)$.
- The **conditional expectation** of the random variable X given the event A from the sample space S is $E(X|A) = \sum_{r \in X(S)} r \cdot P(X = r|A)$.
21. What is expected value of the sum of the numbers appearing on two fair dice when they are rolled given that the sum of these numbers is at least nine. That is, what is $E(X|A)$ where X is the sum of the numbers appearing on the two dice and A is the event that $X \geq 9$?
- The **law of total expectation** states that if the sample space S is the disjoint union of the events S_1, S_2, \dots, S_n and X is a random variable, then $E(X) = \sum_{j=1}^n E(X|S_j)P(S_j)$.
22. Prove the law of total expectations.
 23. Use the law of total expectation to find the average weight of a breeding elephant seal, given that 12% of the breeding elephant seals are male and the rest are female, and the expected weights of a breeding elephant seal is 4,200 pounds for a male and 1,100 pounds for a female.
 24. Let A be an event. Then I_A , the **indicator random variable** of A , equals 1 if A occurs and equals 0 otherwise. Show that the expectation of the indicator random variable of A equals the probability of A , that is, $E(I_A) = p(A)$.
 25. A **run** is a maximal sequence of successes in a sequence of Bernoulli trials. For example, in the sequence $S, S, S, F, S, S, F, F, S$, where S represents success and F represents failure, there are three runs consisting of three successes, two successes, and one success, respectively. Let R denote the random variable on the set of sequences of n independent Bernoulli trials that counts the number of runs in this sequence. Find $E(R)$. [Hint: Show that $R = \sum_{j=1}^n I_j$, where $I_j = 1$ if a run begins at the j th Bernoulli trial and $I_j = 0$ otherwise. Find $E(I_1)$ and then find $E(I_j)$, where $1 < j \leq n$.]
 26. Let $X(s)$ be a random variable, where $X(s)$ is a nonnegative integer for all $s \in S$, and let A_k be the event that $X(s) \geq k$. Show that $E(X) = \sum_{k=1}^{\infty} p(A_k)$.
 27. What is the variance of the number of heads that come up when a fair coin is flipped 10 times?

- 28.** What is the variance of the number of times a 6 appears when a fair die is rolled 10 times?
- 29.** Let X_n be the random variable that equals the number of tails minus the number of heads when n fair coins are flipped.
- What is the expected value of X_n ?
 - What is the variance of X_n ?
- 30.** Show that if X and Y are independent random variables, then $V(XY) = E(X)^2V(Y) + E(Y)^2V(X) + V(X)V(Y)$.
- 31.** Let $A(X) = E(|X - E(X)|)$, the expected value of the absolute value of the deviation of X , where X is a random variable. Prove or disprove that $A(X + Y) = A(X) + A(Y)$ for all random variables X and Y .
- 32.** Provide an example that shows that the variance of the sum of two random variables is not necessarily equal to the sum of their variances when the random variables are not independent.
- 33.** Suppose that X_1 and X_2 are independent Bernoulli trials each with probability $1/2$, and let $X_3 = (X_1 + X_2) \bmod 2$.
- Show that X_1 , X_2 , and X_3 are pairwise independent, but X_3 and $X_1 + X_2$ are not independent.
 - Show that $V(X_1 + X_2 + X_3) = V(X_1) + V(X_2) + V(X_3)$.
 - Explain why a proof by mathematical induction of Theorem 7 does not work by considering the random variables X_1 , X_2 , and X_3 .
- *34.** Prove the general case of Theorem 7. That is, show that if X_1, X_2, \dots, X_n are pairwise independent random variables on a sample space S , where n is a positive integer, then $V(X_1 + X_2 + \dots + X_n) = V(X_1) + V(X_2) + \dots + V(X_n)$. [Hint: Generalize the proof given in Theorem 7 for two random variables. Note that a proof using mathematical induction does not work; see Exercise 33.]
- 35.** Use Chebyshev's inequality to find an upper bound on the probability that the number of tails that come up when a fair coin is tossed n times deviates from the mean by more than $5\sqrt{n}$.
- 36.** Use Chebyshev's inequality to find an upper bound on the probability that the number of tails that come up when a biased coin with probability of heads equal to 0.6 is tossed n times deviates from the mean by more than \sqrt{n} .
- 37.** Let X be a random variable on a sample space S such that $X(s) \geq 0$ for all $s \in S$. Show that $p(X(s) \geq a) \leq E(X)/a$ for every positive real number a . This inequality is called **Markov's inequality**.
- 38.** Suppose that the number of cans of soda pop filled in a day at a bottling plant is a random variable with an expected value of 10,000 and a variance of 1000.
- Use Markov's inequality (Exercise 37) to obtain an upper bound on the probability that the plant will fill more than 11,000 cans on a particular day.
 - Use Chebyshev's inequality to obtain a lower bound on the probability that the plant will fill between 9000 and 11,000 cans on a particular day.
- 39.** Suppose that the number of tin cans recycled in a day at a recycling center is a random variable with an expected value of 50,000 and a variance of 10,000.
- Use Markov's inequality (Exercise 37) to find an upper bound on the probability that the center will recycle more than 55,000 cans on a particular day.
 - Use Chebyshev's inequality to provide a lower bound on the probability that the center will recycle 40,000 to 60,000 cans on a certain day.
- *40.** Suppose the probability that x is the i th element in a list of n distinct integers is $i/[n(n + 1)]$. Find the average number of comparisons used by the linear search algorithm to find x or to determine that it is not in the list.
- *41.** In this exercise we derive an estimate of the average-case complexity of the variant of the bubble sort algorithm that terminates once a pass has been made with no interchanges. Let X be the random variable on the set of permutations of a set of n distinct integers $\{a_1, a_2, \dots, a_n\}$ with $a_1 < a_2 < \dots < a_n$ such that $X(P)$ equals the number of comparisons used by the bubble sort to put these integers into increasing order.
- Show that, under the assumption that the input is equally likely to be any of the $n!$ permutations of these integers, the average number of comparisons used by the bubble sort equals $E(X)$.
 - Use Example 5 in Section 3.3 to show that $E(X) \leq n(n - 1)/2$.
 - Show that the sort makes at least one comparison for every inversion of two integers in the input.
 - Let $I(P)$ be the random variable that equals the number of inversions in the permutation P . Show that $E(X) \geq E(I)$.
 - Let $I_{j,k}$ be the random variable with $I_{j,k}(P) = 1$ if a_k precedes a_j in P and $I_{j,k} = 0$ otherwise. Show that $I(P) = \sum_k \sum_{j < k} I_{j,k}(P)$.
 - Show that $E(I) = \sum_k \sum_{j < k} E(I_{j,k})$.
 - Show that $E(I_{j,k}) = 1/2$. [Hint: Show that $E(I_{j,k}) =$ probability that a_k precedes a_j in a permutation P . Then show it is equally likely for a_k to precede a_j as it is for a_j to precede a_k in a permutation.]
 - Use parts (f) and (g) to show that $E(I) = n(n - 1)/4$.
 - Conclude from parts (b), (d), and (h) that the average number of comparisons used to sort n integers is $\Theta(n^2)$.
- *42.** In this exercise we find the average-case complexity of the quick sort algorithm, described in the preamble to Exercise 50 in Section 5.4, assuming a uniform distribution on the set of permutations.
- Let X be the number of comparisons used by the quick sort algorithm to sort a list of n distinct integers. Show that the average number of comparisons used by the quick sort algorithm is $E(X)$ (where the sample space is the set of all $n!$ permutations of n integers).

- b) Let $I_{j,k}$ denote the random variable that equals 1 if the j th smallest element and the k th smallest element of the initial list are ever compared as the quick sort algorithm sorts the list and equals 0 otherwise. Show that $X = \sum_{k=2}^n \sum_{j=1}^{k-1} I_{j,k}$.
- c) Show that $E(X) = \sum_{k=2}^n \sum_{j=1}^{k-1} p(\text{the } j\text{th smallest element and the } k\text{th smallest element are compared})$.
- d) Show that $p(\text{the } j\text{th smallest element and the } k\text{th smallest element are compared})$, where $k > j$, equals $2/(k-j+1)$.
- e) Use parts (c) and (d) to show that $E(X) = 2(n+1)(\sum_{i=2}^n 1/i) - 2(n-1)$.
- f) Conclude from part (e) and the fact that $\sum_{j=1}^n 1/j \approx \ln n + \gamma$, where $\gamma = 0.57721\dots$ is Euler's constant, that the average number of comparisons used by the quick sort algorithm is $\Theta(n \log n)$.
- *43. What is the variance of the number of **fixed elements**, that is, elements left in the same position, of a randomly selected permutation of n elements? [Hint: Let X denote the number of fixed points of a random permutation. Write $X = X_1 + X_2 + \dots + X_n$, where $X_i = 1$ if the permutation fixes the i th element and $X_i = 0$ otherwise.]

The **covariance** of two random variables X and Y on a sample space S , denoted by $\text{Cov}(X, Y)$, is defined to be the expected value of the random variable $(X - E(X))(Y - E(Y))$. That is, $\text{Cov}(X, Y) = E((X - E(X))(Y - E(Y)))$.

44. Show that $\text{Cov}(X, Y) = E(XY) - E(X)E(Y)$, and use this result to conclude that $\text{Cov}(X, Y) = 0$ if X and Y are independent random variables.
45. Show that $V(X + Y) = V(X) + V(Y) + 2 \text{Cov}(X, Y)$.
46. Find $\text{Cov}(X, Y)$ if X and Y are the random variables with $X((i, j)) = 2i$ and $Y((i, j)) = i + j$, where i and j are the numbers that appear on the first and second of two dice when they are rolled.
47. When m balls are distributed into n bins uniformly at random, what is the probability that the first bin remains empty?
48. What is the expected number of balls that fall into the first bin when m balls are distributed into n bins uniformly at random?
49. What is the expected number of bins that remain empty when m balls are distributed into n bins uniformly at random?

Key Terms and Results

TERMS

sample space: the set of possible outcomes of an experiment
event: a subset of the sample space of an experiment

probability of an event (Laplace's definition): the number of successful outcomes of this event divided by the number of possible outcomes

probability distribution: a function p from the set of all outcomes of a sample space S for which $0 \leq p(x_i) \leq 1$ for $i = 1, 2, \dots, n$ and $\sum_{i=1}^n p(x_i) = 1$, where x_1, \dots, x_n are the possible outcomes

probability of an event E : the sum of the probabilities of the outcomes in E

$p(E|F)$ (**conditional probability of E given F**): the ratio $p(E \cap F)/p(F)$

independent events: events E and F such that $p(E \cap F) = p(E)p(F)$

pairwise independent events: events E_1, E_2, \dots, E_n such that $p(E_i \cap E_j) = p(E_i)p(E_j)$ for all pairs of integers i and j with $1 \leq j < k \leq n$

mutually independent events: events E_1, E_2, \dots, E_n such that $p(E_{i_1} \cap E_{i_2} \cap \dots \cap E_{i_m}) = p(E_{i_1})p(E_{i_2}) \dots p(E_{i_m})$ whenever $i_j, j = 1, 2, \dots, m$, are integers with $1 \leq i_1 < i_2 < \dots < i_m \leq n$ and $m \geq 2$

random variable: a function that assigns a real number to each possible outcome of an experiment

distribution of a random variable X : the set of pairs $(r, p(X = r))$ for $r \in X(S)$

uniform distribution: the assignment of equal probabilities to the elements of a finite set

expected value of a random variable: the weighted average of a random variable, with values of the random variable weighted by the probability of outcomes, that is, $E(X) = \sum_{s \in S} p(s)X(s)$

geometric distribution: the distribution of a random variable X such that $p(X = k) = (1-p)^{k-1}p$ for $k = 1, 2, \dots$ for some real number p with $0 \leq p \leq 1$.

independent random variables: random variables X and Y such that $p(X = r_1 \text{ and } Y = r_2) = p(X = r_1)p(Y = r_2)$ for all real numbers r_1 and r_2

variance of a random variable X : the weighted average of the square of the difference between the value of X and its expected value $E(X)$, with weights given by the probability of outcomes, that is, $V(X) = \sum_{s \in S} (X(s) - E(X))^2 p(s)$

standard deviation of a random variable X : the square root of the variance of X , that is, $\sigma(X) = \sqrt{V(X)}$

Bernoulli trial: an experiment with two possible outcomes

probabilistic (or Monte Carlo) algorithm: an algorithm in which random choices are made at one or more steps

probabilistic method: a technique for proving the existence of objects in a set with certain properties that proceeds by assigning probabilities to objects and showing that the probability that an object has these properties is positive

RESULTS

The probability of exactly k successes when n independent Bernoulli trials are carried out equals $C(n, k)p^kq^{n-k}$, where p is the probability of success and $q = 1 - p$ is the probability of failure.

Bayes' theorem: If E and F are events from a sample space S such that $p(E) \neq 0$ and $p(F) \neq 0$, then

$$p(F | E) = \frac{p(E | F)p(F)}{p(E | F)p(F) + p(E | \bar{F})p(\bar{F})}.$$

$$E(X) = \sum_{r \in X(S)} p(X = r)r.$$

linearity of expectations: $E(X_1 + X_2 + \dots + X_n) = E(X_1) + E(X_2) + \dots + E(X_n)$ if X_1, X_2, \dots, X_n are random variables

If X and Y are independent random variables, then $E(XY) = E(X)E(Y)$.

Bienaymé's formula: If X_1, X_2, \dots, X_n are independent random variables, then $V(X_1 + X_2 + \dots + X_n) = V(X_1) + V(X_2) + \dots + V(X_n)$.

Chebyshev's inequality: $p(|X(s) - E(X)| \geq r) \leq V(X)/r^2$, where X is a random variable with probability function p and r is a positive real number.

Review Questions

1. a) Define the probability of an event when all outcomes are equally likely.
b) What is the probability that you select the six winning numbers in a lottery if the six different winning numbers are selected from the first 50 positive integers?
2. a) What conditions should be met by the probabilities assigned to the outcomes from a finite sample space?
b) What probabilities should be assigned to the outcome of heads and the outcome of tails if heads comes up three times as often as tails?
3. a) Define the conditional probability of an event E given an event F .
b) Suppose E is the event that when a die is rolled it comes up an even number, and F is the event that when a die is rolled it comes up 1, 2, or 3. What is the probability of F given E ?
4. a) When are two events E and F independent?
b) Suppose E is the event that an even number appears when a fair die is rolled, and F is the event that a 5 or 6 comes up. Are E and F independent?
5. a) What is a random variable?
b) What are the possible values assigned by the random variable X that assigns to a roll of two dice the larger number that appears on the two dice?
6. a) Define the expected value of a random variable X .
b) What is the expected value of the random variable X that assigns to a roll of two dice the larger number that appears on the two dice?
7. a) Explain how the average-case computational complexity of an algorithm, with finitely many possible input values, can be interpreted as an expected value.
b) What is the average-case computational complexity of the linear search algorithm, if the probability that the element for which we search is in the list is $1/3$, and it is equally likely that this element is any of the n elements in the list?
8. a) What is meant by a Bernoulli trial?
b) What is the probability of k successes in n independent Bernoulli trials?
c) What is the expected value of the number of successes in n independent Bernoulli trials?
9. a) What does the linearity of expectations of random variables mean?
b) How can the linearity of expectations help us find the expected number of people who receive the correct hat when a hatcheck person returns hats at random?
10. a) How can probability be used to solve a decision problem, if a small probability of error is acceptable?
b) How can we quickly determine whether a positive integer is prime, if we are willing to accept a small probability of making an error?
11. State Bayes' theorem and use it to find $p(F | E)$ if $p(E | F) = 1/3$, $p(E | \bar{F}) = 1/4$, and $p(F) = 2/3$, where E and F are events from a sample space S .
12. a) What does it mean to say that a random variable has a geometric distribution with parameter p ?
b) What is the mean of a geometric distribution with parameter p ?
13. a) What is the variance of a random variable?
b) What is the variance of a Bernoulli trial with probability p of success?
14. a) What is the variance of the sum of n independent random variables?
b) What is the variance of the number of successes when n independent Bernoulli trials, each with probability p of success, are carried out?
15. What does Chebyshev's inequality tell us about the probability that a random variable deviates from its mean by more than a specified amount?

Supplementary Exercises

1. What is the probability that six consecutive integers will be chosen as the winning numbers in a lottery where each number chosen is an integer between 1 and 40 (inclusive)?
 2. A player in the Mega Millions lottery picks five different integers between 1 and 56, inclusive, and a sixth integer between 1 and 46, which may duplicate one of the earlier five integers. The player wins the jackpot if the first five numbers picked match the first five numbers drawn and the sixth number matches the sixth number drawn.
 - a) What is the probability that a player wins the jackpot?
 - b) What is the probability that a player wins \$250,000, which is the prize for matching the first five numbers, but not the sixth number, drawn?
 - c) What is the probability that a player wins \$150 by matching exactly three of the first five numbers and the sixth number or by matching four of the first five numbers but not the sixth number?
 - d) What is the probability that a player wins a prize, if a prize is given when the player matches at least three of the first five numbers or the last number?
 3. A player in the Powerball lottery picks five different integers between 1 and 59, inclusive, and a sixth integer between 1 and 39, which may duplicate one of the earlier five integers. The player wins the jackpot if the first five numbers picked match the first five number drawn and the sixth number matches the sixth number drawn.
 - a) What is the probability that a player wins the jackpot?
 - b) What is the probability that a player wins \$200,000, which is the prize for matching the first five numbers, but not the sixth number, drawn?
 - c) What is the probability that a player wins \$100 by matching exactly three of the first five and the sixth numbers or four of the first five numbers but not the sixth number?
 - d) What is the probability that a player wins a prize, if a prize is given when the player matches at least three of the first five numbers or the last number?
 4. What is the probability that a hand of 13 cards contains no pairs?
 5. What is the probability that a 13-card bridge hand contains
 - a) all 13 hearts?
 - b) 13 cards of the same suit?
 - c) seven spades and six clubs?
 - d) seven cards of one suit and six cards of a second suit?
 - e) four diamonds, six hearts, two spades, and one club?
 - f) four cards of one suit, six cards of a second suit, two cards of a third suit, and one card of the fourth suit?
 6. What is the probability that a seven-card poker hand contains
 - a) four cards of one kind and three cards of a second kind?
 - b) three cards of one kind and pairs of each of two different kinds?
 7. a) pairs of each of three different kinds and a single card of a fourth kind?
 b) pairs of each of two different kinds and three cards of a third, fourth, and fifth kind?
 c) cards of seven different kinds?
 d) a seven-card flush?
 e) a seven-card straight?
 f) a seven-card straight flush?
- An **octahedral die** has eight faces that are numbered 1 through 8.
7. a) What is the expected value of the number that comes up when a fair octahedral die is rolled?
 - b) What is the variance of the number that comes up when a fair octahedral die is rolled?
- A **dodecahedral die** has 12 faces that are numbered 1 through 12.
8. a) What is the expected value of the number that comes up when a fair dodecahedral die is rolled?
 - b) What is the variance of the number that comes up when a fair dodecahedral die is rolled?
 9. Suppose that a pair of fair octahedral dice is rolled.
 - a) What is the expected value of the sum of the numbers that come up?
 - b) What is the variance of the sum of the numbers that come up?
 10. Suppose that a pair of fair dodecahedral dice is rolled.
 - a) What is the expected value of the sum of the numbers that come up?
 - b) What is the variance of the sum of the numbers that come up?
 11. Suppose that a fair standard (cubic) die and a fair octahedral die are rolled together.
 - a) What is the expected value of the sum of the numbers that come up?
 - b) What is the variance of the sum of the numbers that come up?
 12. Suppose that a fair octahedral die and a fair dodecahedral die are rolled together.
 - a) What is the expected value of the sum of the numbers that come up?
 - b) What is the variance of the sum of the numbers that come up?
 13. Suppose n people, $n \geq 3$, play “odd person out” to decide who will buy the next round of refreshments. The n people each flip a fair coin simultaneously. If all the coins but one come up the same, the person whose coin comes up different buys the refreshments. Otherwise, the people flip the coins again and continue until just one coin comes up different from all the others.
 - a) What is the probability that the odd person out is decided in just one coin flip?

- b)** What is the probability that the odd person out is decided with the k th flip?
- c)** What is the expected number of flips needed to decide odd person out with n people?
- 14.** Suppose that p and q are primes and $n = pq$. What is the probability that a randomly chosen positive integer less than n is not divisible by either p or q ?
- *15.** Suppose that m and n are positive integers. What is the probability that a randomly chosen positive integer less than mn is not divisible by either m or n ?
- 16.** Suppose that E_1, E_2, \dots, E_n are n events with $p(E_i) > 0$ for $i = 1, 2, \dots, n$. Show that
- $$\begin{aligned} p(E_1 \cap E_2 \cap \dots \cap E_n) \\ = p(E_1)p(E_2 | E_1)p(E_3 | E_1 \cap E_2) \\ \dots p(E_n | E_1 \cap E_2 \cap \dots \cap E_{n-1}). \end{aligned}$$
- 17.** There are three cards in a box. Both sides of one card are black, both sides of one card are red, and the third card has one black side and one red side. We pick a card at random and observe only one side.
- a)** If the side is black, what is the probability that the other side is also black?
- b)** What is the probability that the opposite side is the same color as the one we observed?
- 18.** What is the probability that when a fair coin is flipped n times an equal number of heads and tails appear?
- 19.** What is the probability that a randomly selected bit string of length 10 is a palindrome?
- 20.** What is the probability that a randomly selected bit string of length 11 is a palindrome?
- 21.** Consider the following game. A person flips a coin repeatedly until a head comes up. This person receives a payment of 2^n dollars if the first head comes up at the n th flip.
- a)** Let X be a random variable equal to the amount of money the person wins. Show that the expected value of X does not exist (that is, it is infinite). Show that a rational gambler, that is, someone willing to pay to play the game as long as the price to play is not more than the expected payoff, should be willing to wager any amount of money to play this game. (This is known as the **St. Petersburg paradox**. Why do you suppose it is called a paradox?)
- b)** Suppose that the person receives 2^n dollars if the first head comes up on the n th flip where $n < 8$ and $2^8 = 256$ dollars if the first head comes up on or after the eighth flip. What is the expected value of the amount of money the person wins? How much money should a person be willing to pay to play this game?
- 22.** Suppose that n balls are tossed into b bins so that each ball is equally likely to fall into any of the bins and that the tosses are independent.
- a)** Find the probability that a particular ball lands in a specified bin.
- b)** What is the expected number of balls that land in a particular bin?
- c)** What is the expected number of balls tossed until a particular bin contains a ball?
- *d)** What is the expected number of balls tossed until all bins contain a ball? [Hint: Let X_i denote the number of tosses required to have a ball land in an i th bin once $i - 1$ bins contain a ball. Find $E(X_i)$ and use the linearity of expectations.]
- 23.** Suppose that A and B are events with probabilities $p(A) = 3/4$ and $p(B) = 1/3$.
- a)** What is the largest $p(A \cap B)$ can be? What is the smallest it can be? Give examples to show that both extremes for $p(A \cap B)$ are possible.
- b)** What is the largest $p(A \cup B)$ can be? What is the smallest it can be? Give examples to show that both extremes for $p(A \cup B)$ are possible.
- 24.** Suppose that A and B are events with probabilities $p(A) = 2/3$ and $p(B) = 1/2$.
- a)** What is the largest $p(A \cap B)$ can be? What is the smallest it can be? Give examples to show that both extremes for $p(A \cap B)$ are possible.
- b)** What is the largest $p(A \cup B)$ can be? What is the smallest it can be? Give examples to show that both extremes for $p(A \cup B)$ are possible.
- 25.** Recall from Definition 5 in Section 7.2 that the events E_1, E_2, \dots, E_n are **mutually independent** if $p(E_{i_1} \cap E_{i_2} \cap \dots \cap E_{i_m}) = p(E_{i_1})p(E_{i_2}) \dots p(E_{i_m})$ whenever $i_j, j = 1, 2, \dots, m$, are integers with $1 \leq i_1 < i_2 < \dots < i_m \leq n$ and $m \geq 2$.
- a)** Write out the conditions required for three events E_1, E_2 , and E_3 to be mutually independent.
- b)** Let E_1, E_2 , and E_3 be the events that the first flip comes up heads, that the second flip comes up tails, and that the third flip comes up tails, respectively, when a fair coin is flipped three times. Are E_1, E_2 , and E_3 mutually independent?
- c)** Let E_1, E_2 , and E_3 be the events that the first flip comes up heads, that the third flip comes up heads, and that an even number of heads come up, respectively, when a fair coin is flipped three times. Are E_1, E_2 , and E_3 pairwise independent? Are they mutually independent?
- d)** Let E_1, E_2 , and E_3 be the events that the first flip comes up heads, that the third flip comes up heads, and that exactly one of the first flip and third flip come up heads, respectively, when a fair coin is flipped three times. Are E_1, E_2 , and E_3 pairwise independent? Are they mutually independent?
- e)** How many conditions must be checked to show that n events are mutually independent?
- 26.** Suppose that A and B are events from a sample space S such that $p(A) \neq 0$ and $p(B) \neq 0$. Show that if $p(B | A) < p(B)$, then $p(A | B) < p(A)$.

In Exercise 27 we consider the **two children problem**, introduced in 1959 by Martin Gardner in his Mathematical Games column in *Scientific American*. A version of the puzzle asks: “We meet Mr. Smith as he is walking down the street with a young child whom he introduces as his son. He also tells us that he has two children. What is the probability that his other child is a son?” We will show that this puzzle is ambiguous, leading to a paradox, by showing that there are two reasonable answers to this problem and we will describe how to make the puzzle unambiguous.

- *27. a) Solve this puzzle in two different ways. First, answer the problem by considering the probability of the gender of the second child. Then, determine the probability differently, by considering the four different possibilities for a family of two children.

- b) Show that the answer to the puzzle becomes unambiguous if we also know that Mr. Smith chose his walking companion at random from his two children.
 c) Another variation of this puzzle asks “When we meet Mr. Smith, he tells us that he has two children and at least one is a son. What is the probability that his other child is a son?” Solve this variation of the puzzle, explaining why it is unambiguous.

28. In 2010, the puzzle designer Gary Foshee posed this problem: “Mr. Smith has two children, one of whom is a son born on a Tuesday. What is the probability that Mr. Smith has two sons?” Show that there are two different answers to this puzzle, depending on whether Mr. Smith specifically mentioned his son because he was born on a Tuesday or whether he randomly chose a child and reported its gender and birth day of the week. [Hint: For the first possibility, enumerate all the equally likely possibilities for the gender and birth day of the week of the other child. To do, this consider first the cases where the older child is a boy born on a Tuesday and then the case where the older child is not a boy born on a Tuesday.]

29. Let X be a random variable on a sample space S . Show that $V(aX + b) = a^2V(X)$ whenever a and b are real numbers.

30. Use Chebyshev’s inequality to show that the probability that more than 10 people get the correct hat back when a hatcheck person returns hats at random does not exceed $1/100$ no matter how many people check their hats. [Hint: Use Example 6 and Exercise 43 in Section 7.4.]

31. Suppose that at least one of the events E_j , $j = 1, 2, \dots, m$, is guaranteed to occur and no more than two can occur. Show that if $p(E_j) = q$ for $j = 1, 2, \dots, m$ and $p(E_j \cap E_k) = r$ for $1 \leq j < k \leq m$, then $q \geq 1/m$ and $r \leq 2/m$.

32. Show that if m is a positive integer, then the probability that the m th success occurs on the $(m + n)$ th trial when independent Bernoulli trials, each with probability p of success, are run, is $\binom{n+m-1}{n}q^n p^m$.

33. There are n different types of collectible cards you can get as prizes when you buy a particular product. Suppose that every time you buy this product it is equally likely that you get any type of these cards. Let X be the random

variable equal to the number of products that need to be purchased to obtain at least one of each type of card and let X_j be the random variable equal to the number of additional products that must be purchased after j different cards have been collected until a new card is obtained for $j = 0, 1, \dots, n - 1$.

- a) Show that $X = \sum_{j=0}^{n-1} X_j$.
 b) Show that after j distinct types of cards have been obtained, the card obtained with the next purchase will be a card of a new type with probability $(n - j)/n$.
 c) Show that X_j has a geometric distribution with parameter $(n - j)/n$.
 d) Use parts (a) and (c) to show that $E(X) = n \sum_{j=1}^n 1/j$.
 e) Use the approximation $\sum_{j=1}^n 1/j \approx \ln n + \gamma$, where $\gamma = 0.57721\dots$ is Euler’s constant, to find the expected number of products that you need to buy to get one card of each type if there are 50 different types of cards.

34. The **maximum satisfiability problem** asks for an assignment of truth values to the variables in a compound proposition in conjunctive normal form (which expresses a compound proposition as the conjunction of clauses where each clause is the disjunction of two or more variables or their negations) that makes as many of these clauses true as possible. For example, three but not four of the clauses in

$$(p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$

can be made true by an assignment of truth values to p , q , and r . We will show that probabilistic methods can provide a lower bound for the number of clauses that can be made true by an assignment of truth values to the variables.

- a) Suppose that there are n variables in a compound proposition in conjunctive normal form. If we pick a truth value for each variable randomly by flipping a coin and assigning true to the variable if the coin comes up heads and false if it comes up tails, what is the probability of each possible assignment of truth values to the n variables?
 b) Assuming that each clause is the disjunction of exactly two distinct variables or their negations, what is the probability that a given clause is true, given the random assignment of truth values from part (a)?
 c) Suppose that there are D clauses in the compound proposition. What is the expected number of these clauses that are true, given the random assignment of truth values of the variables?
 d) Use part (c) to show that for every compound proposition in conjunctive normal form there is an assignment of truth values to the variables that makes at least $3/4$ of the clauses true.
 35. What is the probability that each player has a hand containing an ace when the 52 cards of a standard deck are dealt to four players?

- *36.** The following method can be used to generate a random permutation of a sequence of n terms. First, interchange the n th term and the $r(n)$ th term where $r(n)$ is a randomly selected integer with $1 \leq r(n) \leq n$. Next, interchange the $(n - 1)$ st term of the resulting sequence with its $r(n - 1)$ st term where $r(n - 1)$ is a randomly selected integer with $1 \leq r(n - 1) \leq n - 1$. Continue this process until $j = n$, where at the j th step you interchange the $(n - j + 1)$ st term

of the resulting sequence with its $r(n - j + 1)$ st term, where $r(n - j + 1)$ is a randomly selected integer with $1 \leq r(n - j + 1) \leq n - j + 1$. Show that when this method is followed, each of the $n!$ different permutations of the terms of the sequence is equally likely to be generated. [Hint: Use mathematical induction, assuming that the probability that each of the permutations of $n - 1$ terms produced by this procedure for a sequence of $n - 1$ terms is $1/(n - 1)!$.]

Computer Projects

Write programs with these input and output.

1. Given a real number p with $0 \leq p \leq 1$, generate random numbers taken from a Bernoulli distribution with probability p .
2. Given a positive integer n , generate a random permutation of the set $\{1, 2, 3, \dots, n\}$. (See Exercise 36 in the Supplementary Exercises.)
3. Given positive integers m and n , generate m random permutations of the first n positive integers. Find the number of inversions in each permutation and determine the average number of these inversions.
4. Given a positive integer n , simulate n repeated flips of a biased coin with probability p of heads and determine the number of heads that come up. Display the cumulative results.
5. Given positive integers n and m , generate m random permutations of the first n positive integers. Sort each permutation using the insertion sort, counting the number of comparisons used. Determine the average number of comparisons used over all m permutations.
6. Given positive integers n and m , generate m random permutations of the first n positive integers. Sort each permutation using the version of the bubble sort that terminates
- when a pass has been made with no interchanges, counting the number of comparisons used. Determine the average number of comparisons used over all m permutations.
7. Given a positive integer m , simulate the collection of cards that come with the purchase of products to find the number of products that must be purchased to obtain a full set of m different collector cards. (See Supplementary Exercise 33.)
8. Given positive integers m and n , simulate the placement of n keys, where a record with key k is placed at location $h(k) = k \bmod m$ and determine whether there is at least one collision.
9. Given a positive integer n , find the probability of selecting the six integers from the set $\{1, 2, \dots, n\}$ that were mechanically selected in a lottery.
10. Simulate repeated trials of the Monty Hall Three-Door problem (Example 10 in Section 7.1) to calculate the probability of winning with each strategy.
11. Given a list of words and the empirical probabilities they occur in spam e-mails and in e-mails that are not spam, determine the probability that a new e-mail message is spam.

Computations and Explorations

Use a computational program or programs you have written to do these exercises.

1. Find the probabilities of each type of hand in five-card poker and rank the types of hands by their probability.
2. Find some conditions such that the expected value of buying a \$1 lottery ticket in the New Jersey Pick-6 lottery has an expected value of more than \$1. To win you have to select the six numbers drawn, where order does not matter, from the positive integers 1 to 49, inclusive. The winnings are split evenly among holders of winning tickets. Be sure to consider the total size of the pot going into the drawing and the number of people buying tickets.
3. Estimate the probability that two integers selected at random are relatively prime by testing a large number of randomly selected pairs of integers. Look up the theorem that gives this probability and compare your results with the correct probability.
4. Determine the number of people needed to ensure that the probability at least two of them have the same day of the year as their birthday is at least 70%, at least 80%, at least 90%, at least 95%, at least 98%, and at least 99%.

5. Generate a list of 100 randomly selected permutations of the set of the first 100 positive integers. (See Exercise 36 in the Supplementary Exercises.)
6. Given a collection of e-mail messages, each determined to be spam or not to be spam, develop a Bayesian filter based on the appearance of particular words in these messages.
7. Simulate the odd-person-out procedure (described in Exercise 13 of the Supplementary Exercises) for n people with $3 \leq n \leq 10$. Run a large number of trials for

each value of n and use the results to estimate the expected number of flips needed to find the odd person out. Does your result agree with that found in Exercise 29 in Section 7.2? Vary the problem by supposing that exactly one person has a biased coin with probability of heads $p \neq 0.5$.

8. Given a positive integer n , simulate a hatcheck person randomly giving hats back to people. Determine the number of people who get the correct hat back.

Writing Projects

Respond to these with essays using outside sources.

1. Describe the origins of probability theory and the first uses of this theory, including those by Cardano, Pascal, and Laplace.
2. Describe the different bets you can make when you play roulette. Find the probability of each of these bets in the American version where the wheel contains the numbers 0 and 00. Which is the best bet and which is the worst for you?
3. Discuss the probability of winning when you play the game of blackjack versus a casino. Is there a winning strategy for the person playing against the house?
4. Investigate the game of craps and discuss the probability that the shooter wins and how close to a fair game it is.

5. Discuss issues involved in developing successful spam filters and the current situation in the war between spammers and people trying to filter spam out.
6. Discuss the history and solution of what is known as the Newton–Pepys problem, which asks which is most likely: rolling at least one six when six dice are rolled, rolling at least two sixes when 12 dice are rolled, or rolling at least three sixes when 18 dice are rolled.
7. Explain how Erdős and Rényi first used the probabilistic method and describe some other applications of this method.
8. Discuss the different types of probabilistic algorithms and describe some examples of each type.

8

Advanced Counting Techniques

- 8.1** Applications of Recurrence Relations
- 8.2** Solving Linear Recurrence Relations
- 8.3** Divide-and-Conquer Algorithms and Recurrence Relations
- 8.4** Generating Functions
- 8.5** Inclusion–Exclusion
- 8.6** Applications of Inclusion–Exclusion

Many counting problems cannot be solved easily using the methods discussed in Chapter 6. One such problem is: How many bit strings of length n do not contain two consecutive zeros? To solve this problem, let a_n be the number of such strings of length n . An argument can be given that shows that the sequence $\{a_n\}$ satisfies the recurrence relation $a_{n+1} = a_n + a_{n-1}$ and the initial conditions $a_1 = 2$ and $a_2 = 3$. This recurrence relation and the initial conditions determine the sequence $\{a_n\}$. Moreover, an explicit formula can be found for a_n from the equation relating the terms of the sequence. As we will see, a similar technique can be used to solve many different types of counting problems.

We will discuss two ways that recurrence relations play important roles in the study of algorithms. First, we will introduce an important algorithmic paradigm known as dynamic programming. Algorithms that follow this paradigm break down a problem into overlapping subproblems. The solution to the problem is then found from the solutions to the subproblems through the use of a recurrence relation. Second, we will study another important algorithmic paradigm, divide-and-conquer. Algorithms that follow this paradigm can be used to solve a problem by recursively breaking it into a fixed number of nonoverlapping subproblems until these problems can be solved directly. The complexity of such algorithms can be analyzed using a special type of recurrence relation. In this chapter we will discuss a variety of divide-and-conquer algorithms and analyze their complexity using recurrence relations.

We will also see that many counting problems can be solved using formal power series, called generating functions, where the coefficients of powers of x represent terms of the sequence we are interested in. Besides solving counting problems, we will also be able to use generating functions to solve recurrence relations and to prove combinatorial identities.

Many other kinds of counting problems cannot be solved using the techniques discussed in Chapter 6, such as: How many ways are there to assign seven jobs to three employees so that each employee is assigned at least one job? How many primes are there less than 1000? Both of these problems can be solved by counting the number of elements in the union of sets. We will develop a technique, called the principle of inclusion–exclusion, that counts the number of elements in a union of sets, and we will show how this principle can be used to solve counting problems.

The techniques studied in this chapter, together with the basic techniques of Chapter 6, can be used to solve many counting problems.

8.1 Applications of Recurrence Relations

Introduction

Recall from Chapter 2 that a recursive definition of a sequence specifies one or more initial terms and a rule for determining subsequent terms from those that precede them. Also, recall that a rule of the latter sort (whether or not it is part of a recursive definition) is called a **recurrence relation** and that a sequence is called a *solution* of a recurrence relation if its terms satisfy the recurrence relation.

In this section we will show that such relations can be used to study and to solve counting problems. For example, suppose that the number of bacteria in a colony doubles every hour. If a colony begins with five bacteria, how many will be present in n hours? To solve this problem,

let a_n be the number of bacteria at the end of n hours. Because the number of bacteria doubles every hour, the relationship $a_n = 2a_{n-1}$ holds whenever n is a positive integer. This recurrence relation, together with the initial condition $a_0 = 5$, uniquely determines a_n for all nonnegative integers n . We can find a formula for a_n using the iterative approach followed in Chapter 2, namely that $a_n = 5 \cdot 2^n$ for all nonnegative integers n .

Some of the counting problems that cannot be solved using the techniques discussed in Chapter 6 can be solved by finding recurrence relations involving the terms of a sequence, as was done in the problem involving bacteria. In this section we will study a variety of counting problems that can be modeled using recurrence relations. In Chapter 2 we developed methods for solving certain recurrence relation. In Section 8.2 we will study methods for finding explicit formulae for the terms of sequences that satisfy certain types of recurrence relations.

We conclude this section by introducing the algorithmic paradigm of dynamic programming. After explaining how this paradigm works, we will illustrate its use with an example.

Modeling With Recurrence Relations



We can use recurrence relations to model a wide variety of problems, such as finding compound interest (see Example 11 in Section 2.4), counting rabbits on an island, determining the number of moves in the Tower of Hanoi puzzle, and counting bit strings with certain properties.



Example 1 shows how the population of rabbits on an island can be modeled using a recurrence relation.

EXAMPLE 1



Rabbits and the Fibonacci Numbers Consider this problem, which was originally posed by Leonardo Pisano, also known as Fibonacci, in the thirteenth century in his book *Liber abaci*. A young pair of rabbits (one of each sex) is placed on an island. A pair of rabbits does not breed until they are 2 months old. After they are 2 months old, each pair of rabbits produces another pair each month, as shown in Figure 1. Find a recurrence relation for the number of pairs of rabbits on the island after n months, assuming that no rabbits ever die.

Reproducing pairs (at least two months old)	Young pairs (less than two months old)	Month	Reproducing pairs	Young pairs	Total pairs
		1	0	1	1
		2	0	1	1
		3	1	1	2
		4	1	2	3
		5	2	3	5
		6	3	5	8

FIGURE 1 Rabbits on an Island.

The Fibonacci numbers appear in many other places in nature, including the number of petals on flowers and the number of spirals on seedheads.

Solution: Denote by f_n the number of pairs of rabbits after n months. We will show that f_n , $n = 1, 2, 3, \dots$, are the terms of the Fibonacci sequence.

The rabbit population can be modeled using a recurrence relation. At the end of the first month, the number of pairs of rabbits on the island is $f_1 = 1$. Because this pair does not breed during the second month, $f_2 = 1$ also. To find the number of pairs after n months, add the number on the island the previous month, f_{n-1} , and the number of newborn pairs, which equals f_{n-2} , because each newborn pair comes from a pair at least 2 months old.

Consequently, the sequence $\{f_n\}$ satisfies the recurrence relation

$$f_n = f_{n-1} + f_{n-2}$$

for $n \geq 3$ together with the initial conditions $f_1 = 1$ and $f_2 = 1$. Because this recurrence relation and the initial conditions uniquely determine this sequence, the number of pairs of rabbits on the island after n months is given by the n th Fibonacci number. 



Example 2 involves a famous puzzle.

EXAMPLE 2



The Tower of Hanoi A popular puzzle of the late nineteenth century invented by the French mathematician Édouard Lucas, called the Tower of Hanoi, consists of three pegs mounted on a board together with disks of different sizes. Initially these disks are placed on the first peg in order of size, with the largest on the bottom (as shown in Figure 2). The rules of the puzzle allow disks to be moved one at a time from one peg to another as long as a disk is never placed on top of a smaller disk. The goal of the puzzle is to have all the disks on the second peg in order of size, with the largest on the bottom.

Let H_n denote the number of moves needed to solve the Tower of Hanoi problem with n disks. Set up a recurrence relation for the sequence $\{H_n\}$.

Solution: Begin with n disks on peg 1. We can transfer the top $n - 1$ disks, following the rules of the puzzle, to peg 3 using H_{n-1} moves (see Figure 3 for an illustration of the pegs and disks at this point). We keep the largest disk fixed during these moves. Then, we use one move to transfer the largest disk to the second peg. We can transfer the $n - 1$ disks on peg 3 to peg 2 using H_{n-1} additional moves, placing them on top of the largest disk, which always stays fixed on the bottom of peg 2. Moreover, it is easy to see that the puzzle cannot be solved using fewer steps. This shows that

$$H_n = 2H_{n-1} + 1.$$

The initial condition is $H_1 = 1$, because one disk can be transferred from peg 1 to peg 2, according to the rules of the puzzle, in one move.

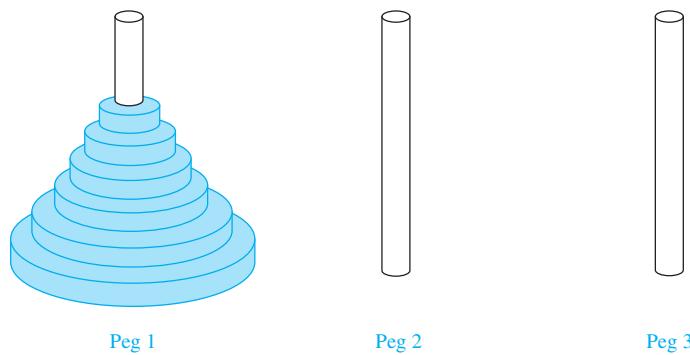


FIGURE 2 The Initial Position in the Tower of Hanoi.

Schemes for efficiently backing up computer files on multiple tapes or other media are based on the moves used to solve the Tower of Hanoi puzzle.

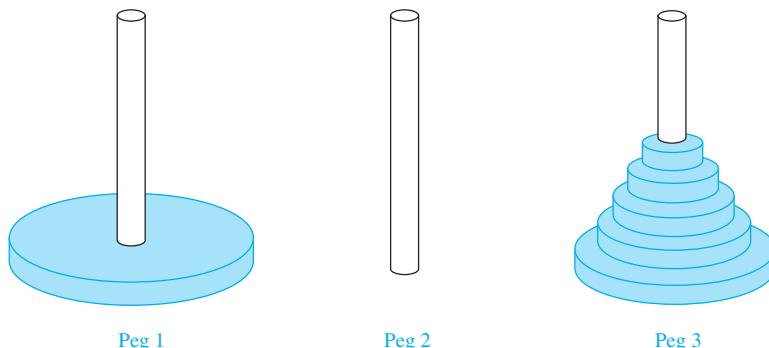


FIGURE 3 An Intermediate Position in the Tower of Hanoi.

We can use an iterative approach to solve this recurrence relation. Note that

$$\begin{aligned}
 H_n &= 2H_{n-1} + 1 \\
 &= 2(2H_{n-2} + 1) + 1 = 2^2H_{n-2} + 2 + 1 \\
 &= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3H_{n-3} + 2^2 + 2 + 1 \\
 &\vdots \\
 &= 2^{n-1}H_1 + 2^{n-2} + 2^{n-3} + \cdots + 2 + 1 \\
 &= 2^{n-1} + 2^{n-2} + \cdots + 2 + 1 \\
 &= 2^n - 1.
 \end{aligned}$$

We have used the recurrence relation repeatedly to express H_n in terms of previous terms of the sequence. In the next to last equality, the initial condition $H_1 = 1$ has been used. The last equality is based on the formula for the sum of the terms of a geometric series, which can be found in Theorem 1 in Section 2.4.

The iterative approach has produced the solution to the recurrence relation $H_n = 2H_{n-1} + 1$ with the initial condition $H_1 = 1$. This formula can be proved using mathematical induction. This is left for the reader as Exercise 1.

A myth created to accompany the puzzle tells of a tower in Hanoi where monks are transferring 64 gold disks from one peg to another, according to the rules of the puzzle. The myth says that the world will end when they finish the puzzle. How long after the monks started will the world end if the monks take one second to move a disk?

From the explicit formula, the monks require

$$2^{64} - 1 = 18,446,744,073,709,551,615$$

moves to transfer the disks. Making one move per second, it will take them more than 500 billion years to complete the transfer, so the world should survive a while longer than it already has. 



Remark: Many people have studied variations of the original Tower of Hanoi puzzle discussed in Example 2. Some variations use more pegs, some allow disks to be of the same size, and some restrict the types of allowable disk moves. One of the oldest and most interesting variations is the **Reve's puzzle**,* proposed in 1907 by Henry Dudeney in his book *The Canterbury Puzzles*. The Reve's puzzle involves pilgrims challenged by the Reve to move a stack of cheeses of varying sizes from the first of four stools to another stool without ever placing a cheese on one of smaller diameter. The Reve's puzzle, expressed in terms of pegs and disks, follows the same rules as the

*Reve, more commonly spelled *reeve*, is an archaic word for *governor*.

Tower of Hanoi puzzle, except that four pegs are used. You may find it surprising that no one has been able to establish the minimum number of moves required to solve this puzzle for n disks. However, there is a conjecture, now more than 50 years old, that the minimum number of moves required equals the number of moves used by an algorithm invented by Frame and Stewart in 1939. (See Exercises 38–45 and [St94] for more information.)

Example 3 illustrates how recurrence relations can be used to count bit strings of a specified length that have a certain property.

EXAMPLE 3 Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive 0s. How many such bit strings are there of length five?

Solution: Let a_n denote the number of bit strings of length n that do not have two consecutive 0s. To obtain a recurrence relation for $\{a_n\}$, note that by the sum rule, the number of bit strings of length n that do not have two consecutive 0s equals the number of such bit strings ending with a 0 plus the number of such bit strings ending with a 1. We will assume that $n \geq 3$, so that the bit string has at least three bits.

The bit strings of length n ending with 1 that do not have two consecutive 0s are precisely the bit strings of length $n - 1$ with no two consecutive 0s with a 1 added at the end. Consequently, there are a_{n-1} such bit strings.

Bit strings of length n ending with a 0 that do not have two consecutive 0s must have 1 as their $(n - 1)$ st bit; otherwise they would end with a pair of 0s. It follows that the bit strings of length n ending with a 0 that have no two consecutive 0s are precisely the bit strings of length $n - 2$ with no two consecutive 0s with 10 added at the end. Consequently, there are a_{n-2} such bit strings.

We conclude, as illustrated in Figure 4, that

$$a_n = a_{n-1} + a_{n-2}$$

for $n \geq 3$.

The initial conditions are $a_1 = 2$, because both bit strings of length one, 0 and 1 do not have consecutive 0s, and $a_2 = 3$, because the valid bit strings of length two are 01, 10, and 11. To obtain a_5 , we use the recurrence relation three times to find that

$$\begin{aligned} a_3 &= a_2 + a_1 = 3 + 2 = 5, \\ a_4 &= a_3 + a_2 = 5 + 3 = 8, \\ a_5 &= a_4 + a_3 = 8 + 5 = 13. \end{aligned}$$

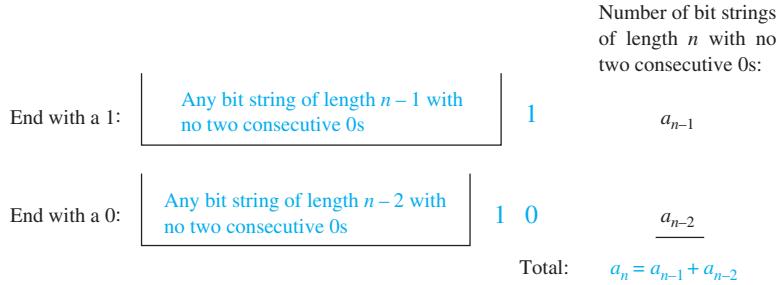


FIGURE 4 Counting Bit Strings of Length n with No Two Consecutive 0s.

Remark: Note that $\{a_n\}$ satisfies the same recurrence relation as the Fibonacci sequence. Because $a_1 = f_3$ and $a_2 = f_4$ it follows that $a_n = f_{n+2}$.

Example 4 shows how a recurrence relation can be used to model the number of codewords that are allowable using certain validity checks.

EXAMPLE 4 Codeword Enumeration A computer system considers a string of decimal digits a valid codeword if it contains an even number of 0 digits. For instance, 1230407869 is valid, whereas 120987045608 is not valid. Let a_n be the number of valid n -digit codewords. Find a recurrence relation for a_n .

Solution: Note that $a_1 = 9$ because there are 10 one-digit strings, and only one, namely, the string 0, is not valid. A recurrence relation can be derived for this sequence by considering how a valid n -digit string can be obtained from strings of $n - 1$ digits. There are two ways to form a valid string with n digits from a string with one fewer digit.

First, a valid string of n digits can be obtained by appending a valid string of $n - 1$ digits with a digit other than 0. This appending can be done in nine ways. Hence, a valid string with n digits can be formed in this manner in $9a_{n-1}$ ways.

Second, a valid string of n digits can be obtained by appending a 0 to a string of length $n - 1$ that is not valid. (This produces a string with an even number of 0 digits because the invalid string of length $n - 1$ has an odd number of 0 digits.) The number of ways that this can be done equals the number of invalid $(n - 1)$ -digit strings. Because there are 10^{n-1} strings of length $n - 1$, and a_{n-1} are valid, there are $10^{n-1} - a_{n-1}$ valid n -digit strings obtained by appending an invalid string of length $n - 1$ with a 0.

Because all valid strings of length n are produced in one of these two ways, it follows that there are

$$\begin{aligned} a_n &= 9a_{n-1} + (10^{n-1} - a_{n-1}) \\ &= 8a_{n-1} + 10^{n-1} \end{aligned}$$

valid strings of length n . 

Example 5 establishes a recurrence relation that appears in many different contexts.

EXAMPLE 5 Find a recurrence relation for C_n , the number of ways to parenthesize the product of $n + 1$ numbers, $x_0 \cdot x_1 \cdot x_2 \cdots \cdot x_n$, to specify the order of multiplication. For example, $C_3 = 5$ because there are five ways to parenthesize $x_0 \cdot x_1 \cdot x_2 \cdot x_3$ to determine the order of multiplication:

$$\begin{array}{lll} ((x_0 \cdot x_1) \cdot x_2) \cdot x_3 & (x_0 \cdot (x_1 \cdot x_2)) \cdot x_3 & (x_0 \cdot x_1) \cdot (x_2 \cdot x_3) \\ x_0 \cdot ((x_1 \cdot x_2) \cdot x_3) & x_0 \cdot (x_1 \cdot (x_2 \cdot x_3)). \end{array}$$

Solution: To develop a recurrence relation for C_n , we note that however we insert parentheses in the product $x_0 \cdot x_1 \cdot x_2 \cdots \cdot x_n$, one “.” operator remains outside all parentheses, namely, the operator for the final multiplication to be performed. [For example, in $(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$, it is the final “.”, while in $(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$ it is the second “.”.] This final operator appears between two of the $n + 1$ numbers, say, x_k and x_{k+1} . There are $C_k C_{n-k-1}$ ways to insert parentheses to determine the order of the $n + 1$ numbers to be multiplied when the final operator appears between x_k and x_{k+1} , because there are C_k ways to insert parentheses in the product $x_0 \cdot x_1 \cdots \cdot x_k$ to determine the order in which these $k + 1$ numbers are to be multiplied and C_{n-k-1} ways to insert parentheses in the product $x_{k+1} \cdot x_{k+2} \cdots \cdot x_n$ to determine

the order in which these $n - k$ numbers are to be multiplied. Because this final operator can appear between any two of the $n + 1$ numbers, it follows that

$$\begin{aligned} C_n &= C_0 C_{n-1} + C_1 C_{n-2} + \cdots + C_{n-2} C_1 + C_{n-1} C_0 \\ &= \sum_{k=0}^{n-1} C_k C_{n-k-1}. \end{aligned}$$

Note that the initial conditions are $C_0 = 1$ and $C_1 = 1$. 

The recurrence relation in Example 5 can be solved using the method of generating functions, which will be discussed in Section 8.4. It can be shown that $C_n = C(2n, n)/(n + 1)$ (see Exercise 41 in Section 8.4) and that $C_n \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$ (see [GrKnPa94]). The sequence $\{C_n\}$ is the sequence of **Catalan numbers**, named after Eugène Charles Catalan. This sequence appears as the solution of many different counting problems besides the one considered here (see the chapter on Catalan numbers in [MiRo91] or [Ro84a] for details).



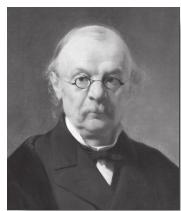
Algorithms and Recurrence Relations

Recurrence relations play an important role in many aspects of the study of algorithms and their complexity. In Section 8.3, we will show how recurrence relations can be used to analyze the complexity of divide-and-conquer algorithms, such as the merge sort algorithm introduced in Section 5.4. As we will see in Section 8.3, divide-and-conquer algorithms recursively divide a problem into a fixed number of non-overlapping subproblems until they become simple enough to solve directly. We conclude this section by introducing another algorithmic paradigm known as **dynamic programming**, which can be used to solve many optimization problems efficiently.

An algorithm follows the dynamic programming paradigm when it recursively breaks down a problem into simpler overlapping subproblems, and computes the solution using the solutions of the subproblems. Generally, recurrence relations are used to find the overall solution from the solutions of the subproblems. Dynamic programming has been used to solve important problems in such diverse areas as economics, computer vision, speech recognition, artificial intelligence, computer graphics, and bioinformatics. In this section we will illustrate the use of dynamic programming by constructing an algorithm for solving a scheduling problem. Before doing so, we will relate the amusing origin of the name *dynamic programming*, which was



EUGÈNE CHARLES CATALAN (1814–1894) Eugène Catalan was born in Bruges, then part of France. His father became a successful architect in Paris while Eugène was a boy. Catalan attended a Parisian school for design hoping to follow in his father's footsteps. At 15, he won the job of teaching geometry to his design school classmates. After graduating, Catalan attended a school for the fine arts, but because of his mathematical aptitude his instructors recommended that he enter the École Polytechnique. He became a student there, but after his first year, he was expelled because of his politics. However, he was readmitted, and in 1835, he graduated and won a position at the Collège de Châlons sur Marne.



In 1838, Catalan returned to Paris where he founded a preparatory school with two other mathematicians, Sturm and Liouville. After teaching there for a short time, he was appointed to a position at the École Polytechnique. He received his doctorate from the École Polytechnique in 1841, but his political activity in favor of the French Republic hurt his career prospects. In 1846 Catalan held a position at the Collège de Charlemagne; he was appointed to the Lycée Saint Louis in 1849. However, when Catalan would not take a required oath of allegiance to the new Emperor Louis-Napoleon Bonaparte, he lost his job. For 13 years he held no permanent position. Finally, in 1865 he was appointed to a chair of mathematics at the University of Liège, Belgium, a position he held until his 1884 retirement.

Catalan made many contributions to number theory and to the related subject of continued fractions. He defined what are now known as the Catalan numbers when he solved the problem of dissecting a polygon into triangles using non-intersecting diagonals. Catalan is also well known for formulating what was known as the *Catalan conjecture*. This asserted that 8 and 9 are the only consecutive powers of integers, a conjecture not solved until 2003. Catalan wrote many textbooks, including several that became quite popular and appeared in as many as 12 editions. Perhaps this textbook will have a 12th edition someday!

introduced by the mathematician Richard Bellman in the 1950s. Bellman was working at the RAND Corporation on projects for the U.S. military, and at that time, the U.S. Secretary of Defense was hostile to mathematical research. Bellman decided that to ensure funding, he needed a name not containing the word mathematics for his method for solving scheduling and planning problems. He decided to use the adjective *dynamic* because, as he said “it’s impossible to use the word dynamic in a pejorative sense” and he thought that dynamic programming was “something not even a Congressman could object to.”

AN EXAMPLE OF DYNAMIC PROGRAMMING The problem we use to illustrate dynamic programming is related to the problem studied in Example 7 in Section 3.1. In that problem our goal was to schedule as many talks as possible in a single lecture hall. These talks have preset start and end times; once a talk starts, it continues until it ends; no two talks can proceed at the same time; and a talk can begin at the same time another one ends. We developed a greedy algorithm that always produces an optimal schedule, as we proved in Example 12 in Section 5.1. Now suppose that our goal is not to schedule the most talks possible, but rather to have the largest possible combined attendance of the scheduled talks.

We formalize this problem by supposing that we have n talks, where talk j begins at time t_j , ends at time e_j , and will be attended by w_j students. We want a schedule that maximizes the total number of student attendees. That is, we wish to schedule a subset of talks to maximize the sum of w_j over all scheduled talks. (Note that when a student attends more than one talk, this student is counted according to the number of talks attended.) We denote by $T(j)$ the maximum number of total attendees for an optimal schedule from the first j talks, so $T(n)$ is the maximal number of total attendees for an optimal schedule for all n talks.

We first sort the talks in order of increasing end time. After doing this, we renumber the talks so that $e_1 \leq e_2 \leq \dots \leq e_n$. We say that two talks are **compatible** if they can be part of the same schedule, that is, if the times they are scheduled do not overlap (other than the possibility one ends and the other starts at the same time). We define $p(j)$ to be largest integer i , $i < j$, for which $e_i \leq s_j$, if such an integer exists, and $p(j) = 0$ otherwise. That is, talk $p(j)$ is the talk ending latest among talks compatible with talk j that end before talk j ends, if such a talk exists, and $p(j) = 0$ if there are no such talks.



RICHARD BELLMAN (1920–1984) Richard Bellman, born in Brooklyn, where his father was a grocer, spent many hours in the museums and libraries of New York as a child. After graduating high school, he studied mathematics at Brooklyn College and graduated in 1941. He began postgraduate work at Johns Hopkins University, but because of the war, left to teach electronics at the University of Wisconsin. He was able to continue his mathematics studies at Wisconsin, and in 1943 he received his masters degree there. Later, Bellman entered Princeton University, teaching in a special U.S. Army program. In late 1944, he was drafted into the army. He was assigned to the Manhattan Project at Los Alamos where he worked in theoretical physics. After the war, he returned to Princeton and received his Ph.D. in 1946.

After briefly teaching at Princeton, he moved to Stanford University, where he attained tenure. At Stanford he pursued his fascination with number theory. However, Bellman decided to focus on mathematical questions arising from real-world problems. In 1952, he joined the RAND Corporation, working on multistage decision processes, operations research problems, and applications to the social sciences and medicine. He worked on many military projects while at RAND. In 1965 he left RAND to become professor of mathematics, electrical and biomedical engineering and medicine at the University of Southern California.

In the 1950s Bellman pioneered the use of dynamic programming, a technique invented earlier, in a wide range of settings. He is also known for his work on stochastic control processes, in which he introduced what is now called the Bellman equation. He coined the term *curse of dimensionality* to describe problems caused by the exponential increase in volume associated with adding extra dimensions to a space. He wrote an amazing number of books and research papers with many coauthors, including many on industrial production and economic systems. His work led to the application of computing techniques in a wide variety of areas ranging from the design of guidance systems for space vehicles, to network optimization, and even to pest control.

Tragically, in 1973 Bellman was diagnosed with a brain tumor. Although it was removed successfully, complications left him severely disabled. Fortunately, he managed to continue his research and writing during his remaining ten years of life. Bellman received many prizes and awards, including the first Norbert Wiener Prize in Applied Mathematics and the IEEE Gold Medal of Honor. He was elected to the National Academy of Sciences. He was held in high regard for his achievements, courage, and admirable qualities. Bellman was the father of two children.

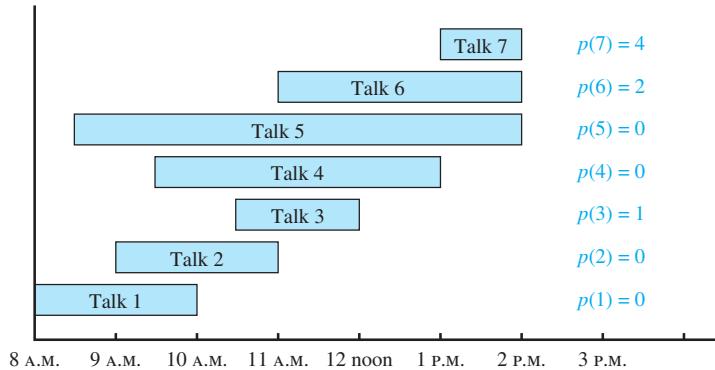


FIGURE 5 A Schedule of Lectures with the Values of $p(n)$ Shown.

EXAMPLE 6 Consider seven talks with these start times and end times, as illustrated in Figure 5.

Talk 1: start 8 A.M., end 10 A.M.

Talk 2: start 9 A.M., end 11 A.M.

Talk 3: start 10:30 A.M., end 12 noon

Talk 4: start 9:30 A.M., end 1 P.M.

Talk 5: start 8:30 A.M., end 2 P.M.

Talk 6: start 11 A.M., end 2 P.M.

Talk 7: start 1 P.M., end 2 P.M.

Find $p(j)$ for $j = 1, 2, \dots, 7$.

Solution: We have $p(1) = 0$ and $p(2) = 0$, because no talks end before either of the first two talks begin. We have $p(3) = 1$ because talk 3 and talk 1 are compatible, but talk 3 and talk 2 are not compatible; $p(4) = 0$ because talk 4 is not compatible with any of talks 1, 2, and 3; $p(5) = 0$ because talk 5 is not compatible with any of talks 1, 2, 3, and 4; and $p(6) = 2$ because talk 6 and talk 2 are compatible, but talk 6 is not compatible with any of talks 3, 4, and 5. Finally, $p(7) = 4$, because talk 7 and talk 4 are compatible, but talk 7 is not compatible with either of talks 5 or 6. \blacktriangleleft

To develop a dynamic programming algorithm for this problem, we first develop a key recurrence relation. To do this, first note that if $j \leq n$, there are two possibilities for an optimal schedule of the first j talks (recall that we are assuming that the n talks are ordered by increasing end time): (i) talk j belongs to the optimal schedule or (ii) it does not.

Case (i): We know that talks $p(j) + 1, \dots, j - 1$ do not belong to this schedule, for none of these other talks are compatible with talk j . Furthermore, the other talks in this optimal schedule must comprise an optimal schedule for talks $1, 2, \dots, p(j)$. For if there were a better schedule for talks $1, 2, \dots, p(j)$, by adding talk j , we will have a schedule better than the overall optimal schedule. Consequently, in case (i), we have $T(j) = w_j + T(p(j))$.

Case (ii): When talk j does not belong to an optimal schedule, it follows that an optimal schedule from talks $1, 2, \dots, j$ is the same as an optimal schedule from talks $1, 2, \dots, j - 1$. Consequently, in case (ii), we have $T(j) = T(j - 1)$. Combining cases (i) and (ii) leads us to the recurrence relation

$$T(j) = \max(w_j + T(p(j)), T(j - 1)).$$

Now that we have developed this recurrence relation, we can construct an efficient algorithm, Algorithm 1, for computing the maximum total number of attendees. We ensure that the algorithm is efficient by storing the value of each $T(j)$ after we compute it. This allows us to compute $T(j)$

only once. If we did not do this, the algorithm would have exponential worst-case complexity. The process of storing the values as each is computed is known as **memoization** and is an important technique for making recursive algorithms efficient.

ALGORITHM 1 Dynamic Programming Algorithm for Scheduling Talks.

```

procedure Maximum Attendees ( $s_1, s_2, \dots, s_n$ : start times of talks;
 $e_1, e_2, \dots, e_n$ : end times of talks;  $w_1, w_2, \dots, w_n$ : number of attendees to talks)
    sort talks by end time and relabel so that  $e_1 \leq e_2 \leq \dots \leq e_n$ 
    for  $j := 1$  to  $n$ 
        if no job  $i$  with  $i < j$  is compatible with job  $j$ 
             $p(j) = 0$ 
        else  $p(j) := \max\{i \mid i < j \text{ and job } i \text{ is compatible with job } j\}$ 
         $T(0) := 0$ 
        for  $j := 1$  to  $n$ 
             $T(j) := \max(w_j + T(p(j)), T(j - 1))$ 
    return  $T(n)$  { $T(n)$  is the maximum number of attendees}

```

In Algorithm 1 we determine the maximum number of attendees that can be achieved by a schedule of talks, but we do not find a schedule that achieves this maximum. To find talks we need to schedule, we use the fact that talk j belongs to an optimal solution for the first j talks if and only if $w_j + T(p(j)) \geq T(j - 1)$. We leave it as Exercise 53 to construct an algorithm based on this observation that determines which talks should be scheduled to achieve the maximum total number of attendees.

Algorithm 1 is a good example of dynamic programming as the maximum total attendance is found using the optimal solutions of the overlapping subproblems, each of which determines the maximum total attendance of the first j talks for some j with $1 \leq j \leq n - 1$. See Exercises 56 and 57 and Supplementary Exercises 14 and 17 for other examples of dynamic programming.

Exercises

1. Use mathematical induction to verify the formula derived in Example 2 for the number of moves required to complete the Tower of Hanoi puzzle.
2. a) Find a recurrence relation for the number of permutations of a set with n elements.
b) Use this recurrence relation to find the number of permutations of a set with n elements using iteration.
3. A vending machine dispensing books of stamps accepts only one-dollar coins, \$1 bills, and \$5 bills.
a) Find a recurrence relation for the number of ways to deposit n dollars in the vending machine, where the order in which the coins and bills are deposited matters.
b) What are the initial conditions?
c) How many ways are there to deposit \$10 for a book of stamps?
4. A country uses as currency coins with values of 1 peso, 2 pesos, 5 pesos, and 10 pesos and bills with values of 5 pesos, 10 pesos, 20 pesos, 50 pesos, and 100 pesos. Find a recurrence relation for the number of ways to pay a bill of n pesos if the order in which the coins and bills are paid matters.
5. How many ways are there to pay a bill of 17 pesos using the currency described in Exercise 4, where the order in which coins and bills are paid matters?
- *6. a) Find a recurrence relation for the number of strictly increasing sequences of positive integers that have 1 as their first term and n as their last term, where n is a positive integer. That is, sequences a_1, a_2, \dots, a_k , where $a_1 = 1$, $a_k = n$, and $a_j < a_{j+1}$ for $j = 1, 2, \dots, k - 1$.
b) What are the initial conditions?
c) How many sequences of the type described in (a) are there when n is an integer with $n \geq 2$?
7. a) Find a recurrence relation for the number of bit strings of length n that contain a pair of consecutive 0s.

- b)** What are the initial conditions?
c) How many bit strings of length seven contain two consecutive 0s?
- 8.** **a)** Find a recurrence relation for the number of bit strings of length n that contain three consecutive 0s.
b) What are the initial conditions?
c) How many bit strings of length seven contain three consecutive 0s?
- 9.** **a)** Find a recurrence relation for the number of bit strings of length n that do not contain three consecutive 0s.
b) What are the initial conditions?
c) How many bit strings of length seven do not contain three consecutive 0s?
- *10.** **a)** Find a recurrence relation for the number of bit strings of length n that contain the string 01.
b) What are the initial conditions?
c) How many bit strings of length seven contain the string 01?
- 11.** **a)** Find a recurrence relation for the number of ways to climb n stairs if the person climbing the stairs can take one stair or two stairs at a time.
b) What are the initial conditions?
c) In how many ways can this person climb a flight of eight stairs?
- 12.** **a)** Find a recurrence relation for the number of ways to climb n stairs if the person climbing the stairs can take one, two, or three stairs at a time.
b) What are the initial conditions?
c) In how many ways can this person climb a flight of eight stairs?
- A string that contains only 0s, 1s, and 2s is called a **ternary string**.
- 13.** **a)** Find a recurrence relation for the number of ternary strings of length n that do not contain two consecutive 0s.
b) What are the initial conditions?
c) How many ternary strings of length six do not contain two consecutive 0s?
- 14.** **a)** Find a recurrence relation for the number of ternary strings of length n that contain two consecutive 0s.
b) What are the initial conditions?
c) How many ternary strings of length six contain two consecutive 0s?
- *15.** **a)** Find a recurrence relation for the number of ternary strings of length n that do not contain two consecutive 0s or two consecutive 1s.
b) What are the initial conditions?
c) How many ternary strings of length six do not contain two consecutive 0s or two consecutive 1s?
- *16.** **a)** Find a recurrence relation for the number of ternary strings of length n that contain either two consecutive 0s or two consecutive 1s.
b) What are the initial conditions?
c) How many ternary strings of length six contain two consecutive 0s or two consecutive 1s?
- *17. a)** Find a recurrence relation for the number of ternary strings of length n that do not contain consecutive symbols that are the same.
b) What are the initial conditions?
c) How many ternary strings of length six do not contain consecutive symbols that are the same?
- **18. a)** Find a recurrence relation for the number of ternary strings of length n that contain two consecutive symbols that are the same.
b) What are the initial conditions?
c) How many ternary strings of length six contain consecutive symbols that are the same?
- 19.** Messages are transmitted over a communications channel using two signals. The transmittal of one signal requires 1 microsecond, and the transmittal of the other signal requires 2 microseconds.
a) Find a recurrence relation for the number of different messages consisting of sequences of these two signals, where each signal in the message is immediately followed by the next signal, that can be sent in n microseconds.
b) What are the initial conditions?
c) How many different messages can be sent in 10 microseconds using these two signals?
- 20.** A bus driver pays all tolls, using only nickels and dimes, by throwing one coin at a time into the mechanical toll collector.
a) Find a recurrence relation for the number of different ways the bus driver can pay a toll of n cents (where the order in which the coins are used matters).
b) In how many different ways can the driver pay a toll of 45 cents?
- 21. a)** Find the recurrence relation satisfied by R_n , where R_n is the number of regions that a plane is divided into by n lines, if no two of the lines are parallel and no three of the lines go through the same point.
b) Find R_n using iteration.
- *22. a)** Find the recurrence relation satisfied by R_n , where R_n is the number of regions into which the surface of a sphere is divided by n great circles (which are the intersections of the sphere and planes passing through the center of the sphere), if no three of the great circles go through the same point.
b) Find R_n using iteration.
- *23. a)** Find the recurrence relation satisfied by S_n , where S_n is the number of regions into which three-dimensional space is divided by n planes if every three of the planes meet in one point, but no four of the planes go through the same point.
b) Find S_n using iteration.
- 24.** Find a recurrence relation for the number of bit sequences of length n with an even number of 0s.
- 25.** How many bit sequences of length seven contain an even number of 0s?

- 26.** a) Find a recurrence relation for the number of ways to completely cover a $2 \times n$ checkerboard with 1×2 dominoes. [Hint: Consider separately the coverings where the position in the top right corner of the checkerboard is covered by a domino positioned horizontally and where it is covered by a domino positioned vertically.]

- b) What are the initial conditions for the recurrence relation in part (a)?
 c) How many ways are there to completely cover a 2×17 checkerboard with 1×2 dominoes?

- 27.** a) Find a recurrence relation for the number of ways to lay out a walkway with slate tiles if the tiles are red, green, or gray, so that no two red tiles are adjacent and tiles of the same color are considered indistinguishable.

- b) What are the initial conditions for the recurrence relation in part (a)?
 c) How many ways are there to lay out a path of seven tiles as described in part (a)?

- 28.** Show that the Fibonacci numbers satisfy the recurrence relation $f_n = 5f_{n-4} + 3f_{n-5}$ for $n = 5, 6, 7, \dots$, together with the initial conditions $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2$, and $f_4 = 3$. Use this recurrence relation to show that f_{5n} is divisible by 5, for $n = 1, 2, 3, \dots$.

- *29.** Let $S(m, n)$ denote the number of onto functions from a set with m elements to a set with n elements. Show that $S(m, n)$ satisfies the recurrence relation

$$S(m, n) = n^m - \sum_{k=1}^{n-1} C(n, k)S(m, k)$$

whenever $m \geq n$ and $n > 1$, with the initial condition $S(m, 1) = 1$.

- 30.** a) Write out all the ways the product $x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4$ can be parenthesized to determine the order of multiplication.

- b) Use the recurrence relation developed in Example 5 to calculate C_4 , the number of ways to parenthesize the product of five numbers so as to determine the order of multiplication. Verify that you listed the correct number of ways in part (a).
 c) Check your result in part (b) by finding C_4 , using the closed formula for C_n mentioned in the solution of Example 5.

- 31.** a) Use the recurrence relation developed in Example 5 to determine C_5 , the number of ways to parenthesize the product of six numbers so as to determine the order of multiplication.

- b) Check your result with the closed formula for C_5 mentioned in the solution of Example 5.

- *32.** In the Tower of Hanoi puzzle, suppose our goal is to transfer all n disks from peg 1 to peg 3, but we cannot move a disk directly between pegs 1 and 3. Each move of a disk must be a move involving peg 2. As usual, we cannot place a disk on top of a smaller disk.

- a) Find a recurrence relation for the number of moves required to solve the puzzle for n disks with this added restriction.
 b) Solve this recurrence relation to find a formula for the number of moves required to solve the puzzle for n disks.
 c) How many different arrangements are there of the n disks on three pegs so that no disk is on top of a smaller disk?
 d) Show that every allowable arrangement of the n disks occurs in the solution of this variation of the puzzle.



Exercises 33–37 deal with a variation of the **Josephus problem** described by Graham, Knuth, and Patashnik in [GrKnPa94]. This problem is based on an account by the historian Flavius Josephus, who was part of a band of 41 Jewish rebels trapped in a cave by the Romans during the Jewish-Roman war of the first century. The rebels preferred suicide to capture; they decided to form a circle and to repeatedly count off around the circle, killing every third rebel left alive. However, Josephus and another rebel did not want to be killed this way; they determined the positions where they should stand to be the last two rebels remaining alive. The variation we consider begins with n people, numbered 1 to n , standing around a circle. In each stage, every second person still left alive is eliminated until only one survives. We denote the number of the survivor by $J(n)$.

33. Determine the value of $J(n)$ for each integer n with $1 \leq n \leq 16$.
 34. Use the values you found in Exercise 33 to conjecture a formula for $J(n)$. [Hint: Write $n = 2^m + k$, where m is a nonnegative integer and k is a nonnegative integer less than 2^m .]
 35. Show that $J(n)$ satisfies the recurrence relation $J(2n) = 2J(n) - 1$ and $J(2n+1) = 2J(n) + 1$, for $n \geq 1$, and $J(1) = 1$.
 36. Use mathematical induction to prove the formula you conjectured in Exercise 34, making use of the recurrence relation from Exercise 35.
 37. Determine $J(100)$, $J(1000)$, and $J(10,000)$ from your formula for $J(n)$.

Exercises 38–45 involve the Reve's puzzle, the variation of the Tower of Hanoi puzzle with four pegs and n disks. Before presenting these exercises, we describe the Frame–Stewart algorithm for moving the disks from peg 1 to peg 4 so that no disk is ever on top of a smaller one. This algorithm, given the number of disks n as input, depends on a choice of an integer k with $1 \leq k \leq n$. When there is only one disk, move it from peg 1 to peg 4 and stop. For $n > 1$, the algorithm proceeds recursively, using these three steps. Recursively move the stack of the $n - k$ smallest disks from peg 1 to peg 2, using all four pegs. Next move the stack of the k largest disks from peg 1 to peg 4, using the three-peg algorithm from the Tower of Hanoi puzzle without using the peg holding the $n - k$ smallest disks. Finally, recursively move the smallest $n - k$ disks to peg 4, using all four pegs. Frame and Stewart showed that to produce the fewest moves using their algorithm, k should be chosen to be the smallest integer

such that n does not exceed $t_k = k(k + 1)/2$, the k th triangular number, that is, $t_{k-1} < n \leq t_k$. The unsettled conjecture, known as **Frame's conjecture**, is that this algorithm uses the fewest number of moves required to solve the puzzle, no matter how the disks are moved.

38. Show that the Reve's puzzle with three disks can be solved using five, and no fewer, moves.
 39. Show that the Reve's puzzle with four disks can be solved using nine, and no fewer, moves.
 40. Describe the moves made by the Frame–Stewart algorithm, with k chosen so that the fewest moves are required, for
 - a) 5 disks.
 - b) 6 disks.
 - c) 7 disks.
 - d) 8 disks.
 - *41. Show that if $R(n)$ is the number of moves used by the Frame–Stewart algorithm to solve the Reve's puzzle with n disks, where k is chosen to be the smallest integer with $n \leq k(k + 1)/2$, then $R(n)$ satisfies the recurrence relation $R(n) = 2R(n - k) + 2^k - 1$, with $R(0) = 0$ and $R(1) = 1$.
 - *42. Show that if k is as chosen in Exercise 41, then $R(n) - R(n - 1) = 2^{k-1}$.
 - *43. Show that if k is as chosen in Exercise 41, then $R(n) = \sum_{i=1}^k i2^{i-1} - (t_k - n)2^{k-1}$.
 - *44. Use Exercise 43 to give an upper bound on the number of moves required to solve the Reve's puzzle for all integers n with $1 \leq n \leq 25$.
 - *45. Show that $R(n)$ is $O(\sqrt{n}2^{\sqrt{2n}})$.
- Let $\{a_n\}$ be a sequence of real numbers. The **backward differences** of this sequence are defined recursively as shown next. The **first difference** ∇a_n is
- $$\nabla a_n = a_n - a_{n-1}.$$
- The **($k + 1$)st difference** $\nabla^{k+1} a_n$ is obtained from $\nabla^k a_n$ by
- $$\nabla^{k+1} a_n = \nabla^k a_n - \nabla^k a_{n-1}.$$
46. Find ∇a_n for the sequence $\{a_n\}$, where
 - a) $a_n = 4$.
 - b) $a_n = 2n$.
 - c) $a_n = n^2$.
 - d) $a_n = 2^n$.
 47. Find $\nabla^2 a_n$ for the sequences in Exercise 46.
 48. Show that $a_{n-1} = a_n - \nabla a_n$.
 49. Show that $a_{n-2} = a_n - 2\nabla a_n + \nabla^2 a_n$.
 - *50. Prove that a_{n-k} can be expressed in terms of a_n , ∇a_n , $\nabla^2 a_n$, ..., $\nabla^k a_n$.
 51. Express the recurrence relation $a_n = a_{n-1} + a_{n-2}$ in terms of a_n , ∇a_n , and $\nabla^2 a_n$.
 52. Show that any recurrence relation for the sequence $\{a_n\}$ can be written in terms of a_n , ∇a_n , $\nabla^2 a_n$, The resulting equation involving the sequences and its differences is called a **difference equation**.
- *53. Construct the algorithm described in the text after Algorithm 1 for determining which talks should be scheduled to maximize the total number of attendees and not just the maximum total number of attendees determined by Algorithm 1.
54. Use Algorithm 1 to determine the maximum number of total attendees in the talks in Example 6 if w_i , the number of attendees of talk i , $i = 1, 2, \dots, 7$, is
 - a) 20, 10, 50, 30, 15, 25, 40.
 - b) 100, 5, 10, 20, 25, 40, 30.
 - c) 2, 3, 8, 5, 4, 7, 10.
 - d) 10, 8, 7, 25, 20, 30, 5.
55. For each part of Exercise 54, use your algorithm from Exercise 53 to find the optimal schedule for talks so that the total number of attendees is maximized.
56. In this exercise we will develop a dynamic programming algorithm for finding the maximum sum of consecutive terms of a sequence of real numbers. That is, given a sequence of real numbers a_1, a_2, \dots, a_n , the algorithm computes the maximum sum $\sum_{i=j}^k a_i$ where $1 \leq j \leq k \leq n$.
 - a) Show that if all terms of the sequence are nonnegative, this problem is solved by taking the sum of all terms. Then, give an example where the maximum sum of consecutive terms is not the sum of all terms.
 - b) Let $M(k)$ be the maximum of the sums of consecutive terms of the sequence ending at a_k . That is, $M(k) = \max_{1 \leq j \leq k} \sum_{i=j}^k a_i$. Explain why the recurrence relation $M(k) = \max(M(k-1) + a_k, a_k)$ holds for $k = 2, \dots, n$.
 - c) Use part (b) to develop a dynamic programming algorithm for solving this problem.
 - d) Show each step your algorithm from part (c) uses to find the maximum sum of consecutive terms of the sequence 2, -3, 4, 1, -2, 3.
 - e) Show that the worst-case complexity in terms of the number of additions and comparisons of your algorithm from part (c) is linear.
- *57. Dynamic programming can be used to develop an algorithm for solving the matrix-chain multiplication problem introduced in Section 3.3. This is the problem of determining how the product $A_1 A_2 \cdots A_n$ can be computed using the fewest integer multiplications, where A_1, A_2, \dots, A_n are $m_1 \times m_2, m_2 \times m_3, \dots, m_n \times m_{n+1}$ matrices, respectively, and each matrix has integer entries. Recall that by the associative law, the product does not depend on the order in which the matrices are multiplied.
 - a) Show that the brute-force method of determining the minimum number of integer multiplications needed to solve a matrix-chain multiplication problem has exponential worst-case complexity. [Hint: Do this by first showing that the order of multiplication of matrices is specified by parenthesizing the product. Then, use Example 5 and the result of part (c) of Exercise 41 in Section 8.4.]

b) Denote by \mathbf{A}_{ij} the product $\mathbf{A}_i \mathbf{A}_{i+1} \dots \mathbf{A}_j$, and $M(i, j)$ the minimum number of integer multiplications required to find \mathbf{A}_{ij} . Show that if the least number of integer multiplications are used to compute \mathbf{A}_{ij} , where $i < j$, by splitting the product into the product of \mathbf{A}_i through \mathbf{A}_k and the product of \mathbf{A}_{k+1} through \mathbf{A}_j , then the first k terms must be parenthesized so that \mathbf{A}_{ik} is computed in the optimal way using $M(i, k)$ integer multiplications and $\mathbf{A}_{k+1,j}$ must be parenthesized so that $\mathbf{A}_{k+1,j}$ is computed in the optimal way using $M(k+1, j)$ integer multiplications.

- c) Explain why part (b) leads to the recurrence relation $M(i, j) = \min_{i \leq k < j} (M(i, k) + M(k+1, j) + m_i m_{k+1} m_{j+1})$ if $1 \leq i \leq j \leq n$.
- d) Use the recurrence relation in part (c) to construct an efficient algorithm for determining the order the n matrices should be multiplied to use the minimum number of integer multiplications. Store the partial results $M(i, j)$ as you find them so that your algorithm will not have exponential complexity.
- e) Show that your algorithm from part (d) has $O(n^3)$ worst-case complexity in terms of multiplications of integers.

8.2 Solving Linear Recurrence Relations

Introduction



A wide variety of recurrence relations occur in models. Some of these recurrence relations can be solved using iteration or some other ad hoc technique. However, one important class of recurrence relations can be explicitly solved in a systematic way. These are recurrence relations that express the terms of a sequence as linear combinations of previous terms.

DEFINITION 1

A *linear homogeneous recurrence relation of degree k with constant coefficients* is a recurrence relation of the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k},$$

where c_1, c_2, \dots, c_k are real numbers, and $c_k \neq 0$.

The recurrence relation in the definition is **linear** because the right-hand side is a sum of previous terms of the sequence each multiplied by a function of n . The recurrence relation is **homogeneous** because no terms occur that are not multiples of the a_j s. The coefficients of the terms of the sequence are all **constants**, rather than functions that depend on n . The **degree** is k because a_n is expressed in terms of the previous k terms of the sequence.

A consequence of the second principle of mathematical induction is that a sequence satisfying the recurrence relation in the definition is uniquely determined by this recurrence relation and the k initial conditions

$$a_0 = C_0, a_1 = C_1, \dots, a_{k-1} = C_{k-1}.$$

EXAMPLE 1 The recurrence relation $P_n = (1.11)P_{n-1}$ is a linear homogeneous recurrence relation of degree one. The recurrence relation $f_n = f_{n-1} + f_{n-2}$ is a linear homogeneous recurrence relation of degree two. The recurrence relation $a_n = a_{n-5}$ is a linear homogeneous recurrence relation of degree five.

Example 2 presents some examples of recurrence relations that are not linear homogeneous recurrence relations with constant coefficients.

EXAMPLE 2 The recurrence relation $a_n = a_{n-1} + a_{n-2}^2$ is not linear. The recurrence relation $H_n = 2H_{n-1} + 1$ is not homogeneous. The recurrence relation $B_n = nB_{n-1}$ does not have constant coefficients.