DISCRETE AND COMBINATORIAL MATHEMATICS

ABRAHAM P. HILLMAN GERALD L. ALEXANDERSON RICHARD M. GRASSL

DISCRETE AND COMBINATORIAL MATHEMATICS

ABRAHAM P. HILLMAN
University of New Mexico

GERALD L. ALEXANDERSON
Santa Clara University

RICHARD M. GRASSL
University of New Mexico

DELLEN PUBLISHING COMPANY
San Francisco, California

COLLIER MACMILLAN PUBLISHERS
London

Division of
Macmillan, Inc.

On the cover: "Bruges III," by Al Held, 1981, acrylic on canvas, 5 feet by 5 feet. The work of New York artist Al Held can be seen in many major American museums, as well as at the André Emmerich Gallery in New York City. The cover art is reproduced through the courtesy of the André Emmerich Gallery.

Printed in the United States of America

Permissions: Dellen Publishing Company
400 Pacific Avenue
San Francisco, California 94133

Orders: Dellen Publishing Company
c/o Macmillan Publishing Company
Front and Brown Streets
Riverside, New Jersey 08075

Collier Macmillan Canada, Inc.

LIBRARY OF CONGRESS CATALOGING-IN-PUBLICATION DATA

Hillman, Abraham P.
Discrete and combinatorial mathematics.

Bibliography: p.
Includes index.
1. Mathematics—1961– . 2. Electronic data processing—Mathematics. 3. Combinatorial analysis.
I. Alexanderson, Gerald L. II. Grassl, Richard M.
III Title.
QA39.2.H555 1986 510 86-9037
ISBN 0-02-354580-1

Printing 2 3 4 5 6 7 8 *Year* 6 7 8 9 0

ISBN 0-02-354580-1

PREFACE

This text provides mathematical foundations for students preparing to use or design computers and for students interested in combinatorial theory and its applications. Some choices of sections for possible courses are listed in charts below. These selections of topics have been successfully class tested, using preliminary editions.

The success of the students in mastering and retaining desired material is attributed to a number of features built into the text. Most important are the well organized graded problem sets consisting of more than 2100 problems. The early problems on each topic help the students to become familiar with the new terms and symbols and to build confidence for later work. Some problems are broken into "bite size" parts; such parts are labeled (i), (ii), and so on. Parts that can be assigned separately are labeled (a), (b), etc. Occasionally a hint, such as a reference to earlier material, is given. Some problem sets contain a series of related problems leading the student through concrete examples to a more general result. Answers or solutions are given to almost all odd-numbered problems.

Several recurrent themes in this text help to unify material presented as unrelated in other sources. For example, bit strings are introduced in the first section and are used in the development of binomial coefficients, subsets of a finite set, algebraic structures, boolean functions, induction, combinatorics and probability, binary codes, languages, and finite state machines. This text also provides a gradual approach to important topics by the foreshadowing of concepts to be discussed more fully, in later sections.

The diversity of topics in a course on discrete mathematics requires an above average amount of new vocabulary and symbolism. The format of the text helps to cope with this difficulty by making it easy to find any definition or notation. Every definition, notation, algorithm, and theorem has a name. Each chapter has a summary of terms introduced in the chapter followed by a set of review problems and a set of supplementary and challenging problems.

The examples and worked out solutions to many odd-numbered problems provide illustrations of all the basic techniques. It is not assumed that the readers are already familiar with binomial coefficients, factorials, sum and product notation, arithmetic and geometric progressions, and so on. No previous knowledge of calculus is required.

Biographical notes are given on some of the people who have made special contributions to the fields discussed in this text. An annotated bibliography is furnished for those seeking more advanced material or additional applications of discrete and combinatorial mathematics.

This text contains enough material for a full year course, and there are many ways of selecting sections for a coherent quarter or semester course. The following table indicates some possible choices of 24 sections for a course.

COURSE	SUGGESTED SECTIONS
Discrete and Combinatorial Mathematics	1.1–1.4, 1.8, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2, 6.1–6.3, 7.1, 7.2, 8.1, 8.3, 9.1–9.3, 10.1
Combinatorial Theory	1.1–1.4, 1.8, 4.1, 4.2, 5.1–5.8, 8.1, 8.3, 9.1–9.6, 10.1
Discrete Structures	1.1–1.8, 2.1–2.5, 3.1–3.3, 4.1, 4.2, 6.1–6.3, 7.1–7.3

The material can also be selected so as to achieve a particular main objective, as illustrated in the following chart.

OBJECTIVE	SUGGESTED SECTIONS
Solving Recursions	1.1–1.4, 1.8, 4.1, 4.2, 5.1, 5.2, 8.1, 8.3, 9.1–9.6
Complexity of Algorithms	1.1–1.4, 1.8, 4.1, 4.2, 5.1, 5.2, 8.1, 8.3, 9.1–9.3, 9.6, 10.1–10.3
Finite State Machines	1.1–1.4, 1.8, 2.1, 2.2, 2.5, 4.1, 6.1, 6.2, 11.1, 12.1, 12.2
Planar Graphs	1.1–1.8, 4.1, 4.2, 5.1, 5.2, 6.1–6.6
Graphs and Their Applications	1.1–1.8, 2.1, 2.2, 4.1, 4.2, 5.1, 5.2, 7.1–7.7, 11.1, 11.2

Flexibility in the selection of topics to be covered is enhanced by the explicit recalling of concepts from previous sections, together with the citations of earlier usage. Other aids are the material on the inside front and back covers, the index at the end of the book, and the summaries of terms for each chapter.

An instructors' manual is available.

ACKNOWLEDGMENTS

The authors are grateful for the valuable suggestions of the following mathematicians, each of whom reviewed one or more of the chapters of a preliminary edition: Don Alton, University of Iowa; Robert Beezer, University of Puget Sound; James Calhoun, Sangamon State University; Philip Feinsilver, Southern Illinois University; David Gale, University of California, Berkeley; Glenn Hopkins, University of Mississippi; Gordon Hughes, California State University, Chico; Takayuki D. Kimura, Washington University; Kenneth D. Lane, Colby College; Zane Motteler, California Polytechnic State University, San Luis Obispo; Peter O'Neil, University of Alabama, Birmingham; Robert Patenaude, College of the Canyons; K. Brooks Reid, Louisiana State University; Karen J. Schroeder, Bentley College; Karen Sharp, C. S. Mott Community College; and Ronald D. Whittekin, Metropolitan State College.

We also express our deepest appreciation to Josephine Hillman who helped in too many ways to mention. Sandra Yeary and Gloria Padilla are thanked for their assistance with the manuscript.

We also recognize with special gratitude the invaluable assistance from Dellen Publishing Company and its editorial and production staffs, including Don Dellen, Luana Morimoto, Phyllis Niklas, and Elliot Simon.

Abraham P. Hillman
Gerald L. Alexanderson
Richard M. Grassl

INTRODUCTION FOR STUDENTS

Students are encouraged to take advantage of several features of this text. The inside front cover has the Greek alphabet to aid in pronouncing these letters and a list of symbols that have fixed meaning when appearing in boldface type. Its facing page has some fundamental symbolism together with the symbolism for structures developed in this book. The inside back cover and its facing page have the other symbolism of the text.

There are answers or solutions near the end of the book for all odd-numbered problems except some "show that" or "prove that" problems. When the parts of a problem are labeled (i), (ii), and so on, this indicates that the problem has been broken into "bite size" parts which should be tackled in the given order. If an even-numbered problem seems difficult, it may be helpful to look at the previous odd-numbered problem and its answer or solution.

Each chapter ends with a summary of the terms introduced and is followed by a set of review problems. Readers desiring a greater test of their knowledge, ability, and perseverance may tackle the supplementary and challenging problems. Also, the annotated bibliography might provide other doors to knowledge.

The chapter summaries are merged in the index at the end of the text.

CONTENTS

3

4

5

6

1

SETS AND RELATIONS

This chapter introduces and develops some important tools for studying the mathematics needed in computer science and computer engineering. Operations on sets are fundamental in all of mathematics. They are presented here as a concrete example of a boolean algebra, the type of algebra involved in the working of electronic circuits and symbolic logic. The study of relations on sets helps in the construction of efficient relational databases, which provide easy access to the accumulated knowledge and techniques of our complicated civilization. The chapter concludes with the study of mappings from a set X to a set Y. This concept is used in many of the topics that follow and is developed in a way that also helps in the combinatorial aspects, which can be applied to appraising the cost in space, time, or money of various procedures.

1.1 BINARY STRINGS

Binary strings are useful in situations that feature dichotomy, that is, separation into two categories. In electrical circuits the dichotomy is that of a switch being either in the "on" or in the "off" position. In logic it involves an assertion's being "true" or "false." In set theory, elements are either "in" or "out" of a set.

In computer science, the symbols used most frequently to represent the two categories of any such dichotomy are the digits 0 and 1. In the applications just mentioned, we may be dealing with several switches, or assertions, or elements of a set; this motivates us to introduce the following definition.

DEFINITION 1 **SET B_n OF BINARY STRINGS**

The digits 0 and 1 are called ***bits*** or ***binary digits***. A string $b_1b_2 \dots b_n$ with each b_i either 0 or 1 is an ***n-bit string*** or an ***n-digit binary string***. We use $\boldsymbol{B_n}$ to denote the set of all n-bit strings.

For example, there are two 1-bit strings: 0 and 1; and there are four 2-digit binary strings: 00, 01, 10, and 11.

In certain algebraic expressions, braces { } have the same meaning as parentheses ( ); however there are contexts in which this is not true. In a ***listing*** for a finite set, the elements should be enclosed within braces and no element should be listed more than once. Thus $\boldsymbol{B_1} = \{0, 1\}$ is the set of 1-digit binary strings, and in this text $\{0, 1, 0\}$ is *not* an allowable use of set notation. Changing the order in which the elements appear creates a new listing but not a new set. For example,

$$\{1, 2, 3\} = \{1, 3, 2\} = \{2, 1, 3\} = \{2, 3, 1\} = \{3, 1, 2\} = \{3, 2, 1\}$$

gives six different listings for the same set. In general, $S = \{s_1, s_2, \dots, s_n\}$ means that $s_1, s_2, \dots, s_n$ are the n distinct elements of the set S. We have noted that $\boldsymbol{B_1} = \{0, 1\}$ and that

$$\boldsymbol{B_2} = \{00, 01, 10, 11\}. \tag{1}$$

The following notation is used in our systematic procedure for obtaining the strings of $\boldsymbol{B_{n+1}}$ from those of $\boldsymbol{B_n}$.

NOTATION 1 **PREFIXING A DIGIT**

Let $\beta = b_1b_2 \dots b_n$ be in the set $\boldsymbol{B_n}$. Then 0β and 1β denote the $(n + 1)$-digit strings $0b_1b_2 \dots b_n$ and $1b_1b_2 \dots b_n$, respectively.

For example, if the four strings of $\boldsymbol{B_2}$ are $\beta_1 = 00$, $\beta_2 = 01$, $\beta_3 = 10$, and $\beta_4 = 11$, then $\{0\beta_1, 0\beta_2, 0\beta_3, 0\beta_4, 1\beta_1, 1\beta_2, 1\beta_3, 1\beta_4\}$ is the set of all 3-digit binary strings

$$\boldsymbol{B_3} = \{000, 001, 010, 011, 100, 101, 110, 111\}. \tag{2}$$

When listing the strings of $\boldsymbol{B_3}$, how could one guard against omitting a string? It helps to know that there are eight strings in $\boldsymbol{B_3}$, as can be seen from Display (2). But suppose one repeated a string and left out another one; this could make it seem that there were the right number of strings. Having an agreed-upon order in which to list the strings lessens the probability of such errors and helps in checking the work of certain applications, such as the case table proofs to be introduced in Section 1.3. A standard order also makes it easier to find a given item in a listing stored in a computer memory.

For these reasons, we introduce a standard way of listing the elements of certain sets, including the sets $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots$. Since 0 precedes 1 in the usual numerical order, we adopt $\{0, 1\}$ as the standard listing for $\boldsymbol{B}_1$. For the same reason, we begin our standard listing of $\boldsymbol{B}_2$ with the strings that have 0 as the leftmost digit, 00 and 01, and then write the strings that start with 1, namely 10 and 11. Thus our standard listing for $\boldsymbol{B}_2$ is $\{00, 01, 10, 11\}$. We can then use this listing for $\boldsymbol{B}_2$ to define the standard listing for $\boldsymbol{B}_3$, and so on.

The following algorithm states the procedure formally. It is an example of a "recursive procedure," or an "inductive definition," because it gives the desired result for the first set $\boldsymbol{B}_1$ and then tells how to use the desired listing for a set $\boldsymbol{B}_k$ in the sequence $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots$ to obtain the chosen listing for the next set $\boldsymbol{B}_{k+1}$. Such procedures are discussed further in Section 4.1.

ALGORITHM 1 **STANDARD LISTING FOR $\boldsymbol{B}_n$**

The standard listing for $\boldsymbol{B}_1$ is $\{0, 1\}$. Given the standard listing $\{\beta_1, \beta_2, \ldots, \beta_r\}$ for $\boldsymbol{B}_k$, then the standard listing for $\boldsymbol{B}_{k+1}$ is

$$\{0\beta_1, 0\beta_2, \ldots, 0\beta_r, 1\beta_1, 1\beta_2, \ldots, 1\beta_r\}.$$

We note that Algorithm 1 produces the standard listing $\boldsymbol{B}_2 = \{00, 01, 10, 11\}$ of Display (1) from the standard listing $\boldsymbol{B}_1 = \{0, 1\}$ and also produces Display (2) from Display (1). Hence Display (2) is the standard listing for $\boldsymbol{B}_3$. As it turns out, the strings of $\boldsymbol{B}_n$ are the base-2 numerals for the numbers $0, 1, 2, \ldots, 2^n - 1$, and the order of appearance in the standard listing is the appropriate order for this application.

NOTATION 2 **NUMBER OF ELEMENTS IN A SET S**

If S is a finite set, $\# S$ stands for the number of elements in S. (One reads "$\# S$" as "the *size* of S" or as "the ***cardinal number*** of S.")

Thus, if $S = \{s_1, s_2, \ldots, s_n\}$, then $\# S = n$. If $\# S = 1$, the set S is called a ***singleton***. If $\# S = 2$, S is called a ***pair*** or a ***doubleton***. If $\# S = 3$, S is a ***triple***.

Since $\boldsymbol{B}_1 = \{0, 1\}$ and $\boldsymbol{B}_2 = \{00, 01, 10, 11\}$, we have $\#\boldsymbol{B}_1 = 2$ and $\#\boldsymbol{B}_2 = 4$. We next give $\#\boldsymbol{B}_n$ for n in $\{1, 2, 3, \ldots\}$.

THEOREM 1 **FORMULA FOR $\#\boldsymbol{B}_n$**

For all positive integers n, $\#\boldsymbol{B}_n = 2^n$.

INFORMAL PROOF Every string $\beta = b_1 b_2 \ldots b_{n+1}$ in $\boldsymbol{B}_{n+1}$ has as its leftmost digit b_1 either 0 or 1; that is, β is either of the form 0α or of the form 1α, with $\alpha = b_2 \ldots b_{n+1}$ in $\boldsymbol{B}_n$. The number of β's in $\boldsymbol{B}_{n+1}$ with $b_1 = 0$ is $\#\boldsymbol{B}_n$, as is the number of β's in $\boldsymbol{B}_{n+1}$ with $b_1 = 1$. Hence

$$\#\boldsymbol{B}_{n+1} = \#\boldsymbol{B}_n + \#\boldsymbol{B}_n = 2(\#\boldsymbol{B}_n).$$

This and $\#\boldsymbol{B}_1 = 2$ imply that $\#\boldsymbol{B}_2 = 2 \cdot 2 = 2^2$, $\#\boldsymbol{B}_3 = 2 \cdot 2^2 = 2^3$, and so on. Thus $\#\boldsymbol{B}_n = 2^n$.

We call this proof "informal" because, for many mathematicians, a formal proof of this result requires the steps of "mathematical induction" (which will be discussed in Section 4.2). □

It is important not only to produce the objects of certain sets, such as the strings of a desired $\boldsymbol{B}_n$, preferably in an easily recognized order, but also to know how many objects are in these sets. One reason for knowing the size of a set is that it helps us estimate the time and space needed for its listing, an important cost consideration when using computers. For example, it follows from Theorem 1 that $\#\boldsymbol{B}_{20} = 2^{20} = 1{,}048{,}576$ and hence that a listing for $\boldsymbol{B}_{20}$ using four strings per line and sixty-six lines per page would require 3972 pages.

DEFINITION 2 **LENGTH AND WEIGHT**

A binary string $\beta = b_1 b_2 \ldots b_n$ with n bits has ***length*** n. The number of 1's in the string β is the ***weight*** of β and is denoted by wgt β.

For example, 11001 and 00110 have length 5 with wgt 11001 = 3 and wgt 00110 = 2. The weight of β is the sum of its binary digits; that is,

$$\text{wgt}(b_1 b_2 \ldots b_n) = b_1 + b_2 + \cdots + b_n. \tag{3}$$

Also, if β is in $\boldsymbol{B}_n$, wgt β must be in the set $\{0, 1, \ldots, n\}$.

NOTATION 3 **SUBSET $\boldsymbol{B}_{n,k}$ OF $\boldsymbol{B}_n$**

The set of all binary strings of length n and weight k is denoted as $\boldsymbol{B}_{n,k}$.

For example, the sets of 3-bit strings of weight 0, 1, 2, and 3, respectively, are

$$B_{3,0} = \{000\}, \qquad B_{3,1} = \{001, 010, 100\},$$
$$B_{3,2} = \{011, 101, 110\}, \qquad B_{3,3} = \{111\}. \tag{4}$$

In each of these listings, the strings appear in the same relative order as in the standard listing for B_3.

LEMMA 1 WEIGHT OF 0β AND 1β

$$\text{wgt}(0\beta) = \text{wgt } \beta \qquad \text{and} \qquad \text{wgt}(1\beta) = 1 + \text{wgt } \beta.$$

PROOF Let $\beta = b_1 b_2 \ldots b_n$. Since the weight of a binary string is the sum of its digits,

$$\text{wgt}(0\beta) = 0 + b_1 + b_2 + \cdots + b_n = \text{wgt } \beta,$$
$$\text{wgt}(1\beta) = 1 + b_1 + b_2 + \cdots + b_n = 1 + \text{wgt } \beta. \qquad \square$$

Example 1 Here we use Lemma 1 and listings for $B_{3,3}$ and $B_{3,2}$ to obtain a listing for $B_{4,3}$. Let $\beta = b_1 b_2 b_3 b_4 \in B_{4,3}$. (The notation "$\alpha \in S$" means "$\alpha$ is an element of the set S.") Since b_1 is either 0 or 1, β is of the form 0α or the form 1α, where $\alpha = b_2 b_3 b_4$. If $b_1 = 0$, then wgt $\alpha = 3$ (by Lemma 1) and so $\alpha \in B_{3,3}$. If $b_1 = 1$, then wgt $\alpha = 2$ and $\alpha \in B_{3,2}$. Thus we prefix a 0 to 111, the only string in $B_{3,3}$, to obtain the only β in $B_{4,3}$ with 0 as the leftmost digit, and we prefix a 1 to each of the three strings of $B_{3,2} = \{011, 101, 110\}$ to get the three strings of $B_{4,3}$ with 1 as the leftmost digit. Together these give the listing

$$B_{4,3} = \{0111, 1011, 1101, 1110\} \tag{5}$$

for the set of all 4-digit binary strings of weight 3. $\square$

It is easy to see that

$$B_{1,0} = \{0\}, \qquad B_{2,0} = \{00\}, \qquad B_{3,0} = \{000\}, \qquad \ldots \tag{6}$$

and

$$B_{1,1} = \{1\}, \qquad B_{2,2} = \{11\}, \qquad B_{3,3} = \{111\}, \qquad \ldots. \tag{7}$$

That is, the only string in $B_{n,0}$ is the n-digit string $00 \ldots 0$, with each digit a 0, and the only string in $B_{n,n}$ is the n-digit string $11 \ldots 1$, with each digit a 1. Since $B_{n,0}$ is a singleton and $B_{n,n}$ is a singleton for every n in $Z^+ = \{1, 2, \ldots\}$, each of these sets has a unique listing. Hence we adopt the listings of Displays (6) and (7) as the standard listings.

We next give a recursive procedure for obtaining the standard listings for the other $B_{n,k}$, that is, those with $0 < k < n$.

ALGORITHM 2 STANDARD LISTING FOR $B_{n,k}$

Let n and k be integers with $0 < k < n$. Let the standard listings for $B_{n-1,k}$ and $B_{n-1,k-1}$ be

$$B_{n-1,k} = \{\alpha_1, \alpha_2, \ldots, \alpha_s\}$$

and

$$B_{n-1,k-1} = \{\gamma_1, \gamma_2, \ldots, \gamma_t\}.$$

Then the standard listing for $B_{n,k}$ is

$$\{0\alpha_1, 0\alpha_2, \ldots, 0\alpha_s, 1\gamma_1, 1\gamma_2, \ldots, 1\gamma_t\}.$$

Using this algorithm and the standard listings $B_{1,1} = \{1\}$ and $B_{1,0} = \{0\}$, we obtain the standard listing $B_{2,1} = \{01, 10\}$. The algorithm together with the standard listings $B_{2,2} = \{11\}$ and $B_{2,1} = \{01, 10\}$ give us the standard listing $B_{3,2} = \{011, 101, 110\}$.

The numbers that give the sizes of the sets $B_{n,k}$ appear frequently in mathematics. The following definition introduces the usual notation for these integers.

DEFINITION 3 BINOMIAL COEFFICIENT $\binom{n}{k}$

For n in $\{1, 2, 3, \ldots\}$ and k in $\{0, 1, \ldots, n\}$, let $\binom{n}{k}$ denote the number of n-digit binary strings of weight k; that is, $\binom{n}{k} = \# B_{n,k}$. Also let $\binom{0}{0} = 1$. [One reads $\binom{n}{k}$ as "n choose k" or as "***binomial coefficient*** n choose k."]

For example, the number of binary strings of length 8 and weight 6 is denoted as $\binom{8}{6}$. One obtains a string $\beta = b_1 b_2 \ldots b_n$ in $B_{n,k}$ by choosing k of the n integers of $\{1, 2, \ldots, n\}$ as the subscripts i for which $b_i = 1$ (and letting $b_j = 0$ for the other $n - k$ subscripts j). This indicates why $\binom{n}{k}$ is read as "n choose k." In Theorem 2(b) of Section 1.2, it is shown that $\binom{n}{k}$ is the number of ways of choosing k elements from any set of size n. In Section 5.2, we will see why the $\binom{n}{k}$ are called "binomial coefficients."

We note from Display (4) that $\binom{3}{0} = 1 = \binom{3}{3}$ and $\binom{3}{1} = 3 = \binom{3}{2}$. Example 1 shows that $\binom{4}{3} = \binom{3}{3} + \binom{3}{2} = 1 + 3 = 4$. These illustrate the following formulas.

THEOREM 2 BORDERS, RECURSION, AND SYMMETRY FORMULAS

(a) $\binom{n}{0} = 1 = \binom{n}{n}$ for $n = 0, 1, 2, \ldots$. (*Borders Formula*)
(b) $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ for $0 < k < n$. (*Recursion Formula*)
(c) $\binom{n}{k} = \binom{n}{n-k}$ for $0 \le k \le n$. (*Symmetry Formula*)