Preface

The purpose of this book is to provide a modern and comprehensive introduction to the subject of discrete structures. *Discrete structures*, also called *discrete mathematics*, is an exciting and active subject, particularly due to its extreme relevance to both mathematics and to computer science and algorithms. The subject forms a common foundation for rigorous mathematical logical reasoning and proofs, as well as a formal introduction to abstract objects that are essential tools in an assortment of applications and effective computer implementations. Computing skills are now an integral part of most all scientific fields, and students are very enthusiastic about being able to harness the full computing power of these tools. Courses in discrete structures are offered at most all universities and in an increasingly large portion of community colleges as well, and are required by both math and computer science programs.

How the book evolved: Over the past 12 years the author has been regularly teaching the designated one-semester course in discrete structures (to both mathematics and computer science majors) at two universities: the University of Guam and the California State University–Dominguez Hills, at both of which he has been appointed as a professor. In different semesters, he has been working to supplement the core material with exposure to an assortment of the new exciting applications-oriented topics that fall under the heading of "discrete structures," but are not always part of the standard curriculum. This standard curriculum is nonetheless important, and it should be included in any decent book or course on the subject. At the same time, it is also important to develop materials that reflect many of the advances and recent trends and applications of this area. Examples of some of the applications that are extensively treated in this book include: simulation, genetic algorithms, network flows, probabilistic primality tests, public key cryptography, and many applications to coding theory. There have been numerous new developments in this vast subject, and the tasks that can be accomplished by students on their PCs are now very different than what was feasible even just 15 years ago. It is these exciting application areas that really make the topics of discrete structures so interesting and useful in applications, and is the reason that it is a required course for computer science students (and increasingly for mathematics students as well). Some of the topics covered in this book were introduced in various semesters in the author's discrete structures classes, and others were further expounded upon in subsequent special topics classes. Examples include separate upper-level courses in each of the areas of graph theory and algorithms, simulation, genetic algorithms, and in cryptography. Not being able to find books that included many of these new topics or had a good selection of exercises, the author found it necessary to prepare his own notes and exercise sets for many of these classes and topics. His courses in cryptography have resulted in a separate book by the author on this subject that was published (also by CRC Press) shortly before this book. Starting about five years ago, with the aim of expanding his materials into a complete discrete structures textbook, the author began to prepare book materials and exercises for the core topics in discrete structures (i.e., those included in the ACM's recommendations).[1] Thus, the book has been evolving through classroom testing on a number of different courses first with the applications, and then with the core material.

How the book is organized: The resulting chapters are written in an easy-going, yet rigorous, and extremely conscientious style. Each section is replete with clear definitions and theorems; the proofs are carefully explained, and written in a cordial style that students find appealing. There are numerous completely worked-out examples that illustrate key concepts, and figures and tables are employed to help students grasp the more subtle and difficult concepts. The text proper is punctuated with "Exercises for the Reader" that give readers frequent opportunities to assess their understanding of the material. These are meant to be done during a careful reading, and complete solutions of these Exercises for the Reader are provided in an appendix at the end of the book. In addition to these, each section ends with an extensive and well-thought-out set of section exercises that range in difficulty from routine to nontrivial, and sometimes include developments of new topics that complement or extend the material in the section proper. Some groups of the exercises (usually toward the end of a set) introduce and develop new topics and methods. Another appendix supplies answers, and in many cases, solution outlines to most of the odd-numbered exercises. A separate instructor's manual containing solutions to the even-numbered exercises is freely available from CRC Press to all qualified instructors. Many sections have appendices that cover either a new and related topic, or material that is more advanced. These appendices, as well as some of the exercise groups on new material, form an excellent basis for student projects. In addition to the ordinary exercise sets, most sections contain separate sets of Computer Exercises that are intended to be done with the aid of a computer. Except for a few rare exceptions that are clearly indicated (in Section 6.2, in Chapter 10, and in some of the section appendices), the text proper and ordinary exercises can be done without using computers. This feature makes the book suitable for courses that depend on a variety of computer usage: from none at all to courses where students write their own programs. The computer exercises contain extensive and useful material on how students can learn to write their own programs on any computing platform for most all of the major algorithms that are considered in this book. The separation of the various sections was done with learning and pedagogy in mind. One extreme is to have exercises only at the end of the chapters, and this tends to make it difficult for instructors to assign exercises on a daily basis; the other extreme is

[1] ACM is the acronym for The Association for Computing Machinery, the premier professional organization for computer science of the United States. It periodically drafts recommendations for discrete structures courses, since the area is so fundamental for computer science.

to split the chapters into many small sections, each one covering a single topic. There is much interaction of concepts in discrete structures, so an intermediate approach of splitting chapters into sections of reasonable sizes has been taken. The sections were conceived to provide an effective separation of learning objectives and quiz units. The large number, scope, organization, and variety of carefully crafted exercises form one of the major strengths of the book.

A few comments about the numbering scheme are in order. All theoretical results—theorems, lemmas, propositions, and corollaries—are numbered sequentially in each chapter. Definitions, examples, labeled equations, figures, tables, algorithms, and exercises for the reader each have their own separate counters. The general index at the end of the book is preceded by separate indices for theoretical results and algorithms.

How technology is incorporated: Although the book highly encourages the use of computing platforms, it can still be used without any computers. The text proper is written for non-computer users, as are all of the ordinary exercises. As a general rule, technical comments and implementation suggestions for computer users are restricted the computer exercise sets located at the end of most sections. A few clearly isolated examples do discuss computing (for example, the simulation material of Section 6.2, and some of the appendices), but these passages may simply be read by non-computer users. The main exception is Chapter 10. By its nature, this chapter makes extensive use of computing, so the exercises and computer exercises have been combined into single sets. Nonetheless, Chapter 10 has been written in a way that will be understandable to all readers, whether or not they are using a computer. Indeed, for non-computer users, it should serve to enlighten them about the synergistic power when computing is effectively combined with discrete structure skills.

The explanations of the algorithms and the computer exercises are platformindependent. Algorithms are explained first in ordinary English, and when appropriate using a natural and easy-to-understand pseudocode that can be readily translated into any computer language (an appendix summarizes the pseudo code). A Web page for the book is being maintained that includes, in particular, sample programs and programming notes for all of the computer material for the MATLAB[2] computing platform (which the author has used for computer implementations of all of the relevant topics in this book). These programs are replete with explanatory comments, and since MATLAB is a very user-friendly language, they should be useful to users of other computing platforms in their efforts to write programs. Rather than print the URL of the Web page, which may change as servers get updated or as (the author's physical or electronic) addresses may change, the easiest way to access this page is through the author's homepage, which can be obtained by a simple Web search of the

[2] MATLAB® is a computing platform that is among the top three used by mathematics departments in the world. It is gaining popularity in mathematics and computer science departments because of its wide usage across the spectrum in science and engineering departments.

author's last name. The Web page should also be navigable via the publisher's Web site.

How to get the most out of this book: Although everything in this book is carefully defined from scratch and there are no formal prerequisites, it is tacitly assumed that the reader has passed a course in precalculus. In particular, a course in calculus is not required. Many schools make calculus a prerequisite for the discrete structures course, not so much for the subject matter dependence, but more for the reason that calculus courses give students a good, possibly first, experience with mathematical rigor. There are a few rare and isolated comments and footnotes for the benefit of readers who are familiar with calculus, but these are clearly marked and may be skipped by non-calculus readers. There is more than enough material in this book for two semesters of discrete structures. The chapters are organized in a way that would allow instructors to build many different courses. A chapter/section dependence chart is provided (after this preface) for instructors wishing to plan courses. That being said, readers should not feel constrained from flipping ahead if they wish to read any particular topics of interest. Great care has been taken to make it possible to do this, and to look up and read about any earlier topics when this is necessary.

Acknowledgments: Most of the writing of this book was motivated by the author's teaching of discrete structures classes and related special topics courses over the past 12 years. He would like to express his gratitude to the math departments, computer science departments, and to his colleagues at the University of Guam and at California State University–Dominguez Hills for allowing him to frequently teach such classes, often with new topics and with various levels of technology. He is grateful to many students over the years who have suffered through preliminary drafts of various portions, and have provided much useful feedback that has helped him to make numerous improvements. Two colleagues deserve very special mention: It was delightful to have George Jennings involved in this project. He has done an exemplary job in preparing an instructor solutions manual for the book (instructors should contact CRC Press to obtain a copy), and in the process has read through the entire book and provided numerous useful suggestions for improvement. The author was also highly appreciative to have had his colleague Frank Miles on board. Frank has the precision of a Swiss watch, and he has carefully read through the entire manuscript and provided a plethora of scholarly suggestions and comments. Neighboring colleague Will Murray at California State University–Long Beach has read through significant portions and provided some very useful suggestions. The author thanks Yumi Nishimura for fine work in creating some of the technical drawings, and help with the cover design.[3] The anonymous reviewers have provided some very useful feedback and suggestions that have led to many

[3] The cover design is based on a photograph of a green sea turtle taken by the author during a recent dive trip to the beautiful island of Palau. The JPG technology of such digital photographs relies on Huffman codes, which are discussed in Chapter 8.

improvements, including some major restructuring of the book. As with all of his previous books, the author's mother Christa Stanoyevitch has continued to proofread through all of the early drafts (and thus has had the most arduous of all the proofreader jobs), and has helped him to correct many of the initial errors and to make stylistic improvements before the drafts make it to the students. The book's editor, Sunil Nair (who also holds a Ph.D. in mathematics), has been extraordinarily farsighted and helpful in his ideas that have led to significant improvements in the organization and coverage, as well as very nicely and promptly taking care of the numerous issues or concerns that came up during the latter part of the writing and the production process. The author was very pleased to be able to work again on his second CRC book with production editor Tara Nieuwesteeg; she has always been exceptionally hard-working, wise, and accommodating throughout the project. Thanks go also to the copy editor Cindy Gallardo, who has carefully read over the drafts and offered numerous helpful suggestions and corrections. The author takes full responsibility for any errors that remain, and would be grateful to any readers who could direct his attention to any such oversights.

About the Author

Alexander Stanoyevitch completed his doctorate in mathematical analysis at the University of Michigan–Ann Arbor, has held academic positions at the University of Hawaii and the University of Guam, and is presently a professor at California State University–Dominguez Hills. He has published several articles in leading mathematical journals and has been an invited speaker at numerous lectures and conferences in the United States, Europe, and Asia. His research interests include areas of both pure and applied mathematics, and he has taught many upper-level classes to mathematics students as well as computer science students.

Dependency Chart

The following chart should be helpful to readers or instructors aiming to plan courses with this book. Major dependencies are indicated with solid arrows, minor ones with dashed arrows. The dependencies on Section 2.2 (*) pertain only to the material on equivalence relations. The dependence on Section 4.2 by Section 4.3 (**) is only for the material on modular matrices. This material is not used in Section 8.1 (which has a minor dependence on Section 4.3).