Texts in Computer Science

More information about this series at http://www.springer.com/series/3191

Sergei Kurgalin • Sergei Borzunov

The Discrete Math Workbook

A Companion Manual Using Python

Second Edition

Springer

Sergei Kurgalin
Voronezh State University
Voronezh, Russia

Sergei Borzunov
Voronezh State University
Voronezh, Russia

ISSN 1868-0941 ISSN 1868-095X (electronic)
Texts in Computer Science
ISBN 978-3-030-42220-2 ISBN 978-3-030-42221-9 (eBook)
https://doi.org/10.1007/978-3-030-42221-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface to the Second Edition

The second edition of the book comes out after a short period of time compared to the first one, which we believe is connected with the increased interest in discrete mathematics among specialists in the field of computer science. Now we use Python as a programming language. The choice of this language is determined by its universality and rapid growth in popularity in the world. In our opinion, Python is good enough for teaching methods of design and analysis of algorithms.

We have kept the structure of the material the same in the second edition: each chapter consists of a theoretical part containing basic definitions, theorems, and typical schemes of problem-solving, then we give problems for solving in a classroom led by a teacher or for self-study.

More than 50 new problems with solutions and answers have been added to the book, as well as control questions for each chapter to check the knowledge of basic definitions and theoretical facts. In a number of cases clarifying comments were made or observed for inaccuracies corrected in solutions and proofs.

Acknowledgments

The authors would like to express their grateful acknowledgment to all who expressed their wishes or suggested ways to improve the book after the publication of the first edition.

We are grateful to our colleagues Alexey Bukhovets, Valeria Kaverina, Alexander Loboda, Peter Meleshenko, and Mikhail Semenov for their useful discussions, advices, and critical comments. Also, at Springer, we would like to thank Senior Editor, Wayne Wheeler, and Associate Editor, Simon Rees.

Nikolai Paukov helped us a lot in debugging and checking the program code of the algorithms presented in the book.

All remaining errors and inaccuracies, of course, remain on the authors' conscience.

Voronezh, Russia — Sergei Kurgalin
January 2020 — Sergei Borzunov

Preface to the First Edition

Rapid development of information and communication technologies is a characteristic feature of the current time period. Training specialists who meet the requirements of the time in the field of such technologies leads to the necessity of paying special attention to their mathematical education. Herewith it is desirable to shape knowledge presented in the form of a practical part of mathematical courses in such a way that after performing these tasks students get the required competencies. One of the mathematical courses which is taught, as a rule, in early semesters is the course of discrete mathematics. A number of general courses and specialist disciplines included in the cycle of specialist training in the field of information technologies are based on the present course. Thus, mathematical logic and theory of algorithms form the theoretical foundation of informatics, Boolean algebra serves as a basis for methods of electronic circuit development, theory of graphs is used when constructing multiprocessor computing systems and computer networks, and in programming. To teach students the competent practical application of their theoretical knowledge is one of the methodological objectives the given study guide can help with.

The study guide was created over a number of years and is backed up by the teaching experience of the course at the Department of Computer Science of Voronezh State University. Its contents correspond to Federal State Educational Standards on the following training programmes: "Information Systems and Technologies", "Program Engineering", and "Radiophysics". The existing problem books on discrete mathematics and mathematical logic do not fully cover the necessary volume and information scope upon the abovementioned programmes to graduate fully skilled subject matter experts; therefore this study guide provides a large number of problems of a wide range of difficulty for these training programmes.

The present study guide is intended for practical training, laboratory practicals, and self-study. It contains basic theoretical concepts and methods of solving the most fundamental problems; it forms the perceptions of sets, relations on sets, and behaviur of different types of relations, as well as functions, general concepts of combinatorics, theory of graphs, Boolean algebra, basic principles of algorithms theory, etc. Some chapters of the study book, such as "Turing Machine" and "Asymptotic Analysis" go beyond the traditional course of discrete mathematics. Nevertheless, we believe it necessary to include them into the structure of the study guide, as they promote understanding of the methods of construction and analysis of algorithms. According to the famous Church–Turing thesis, the Turing machine is able to imitate all means of step-by-step computation, and is considered to be the model of any computing system existing today. The theory of algorithms' asymptotic analysis studies the methods of getting asymptotic estimations of algorithms' computational complexity, which has the paramount importance for the estimation of the need in resource requirement at the specific implementation of an algorithm.

It is only natural that we analyzed certain textbooks and problem books while preparing the study guide with a view to using the positive experience contained in them and to produce the most efficient method of presenting the material. Most of them are enumerated in the reference list; in particular, these are [4, 18, 28].

The study guide covers two semesters of the study course; some chapters might be used in one-semester courses.

Each chapter begins with a theoretical part, which albeit occupies little space, is of crucial importance as it reviews the basic concepts of the course, and sets the utilized terminology. Alongside of the basic concepts, theoretical provisions, and formulae in the text, there are also instructions on their practical use. Then the detailed solution of several most common problems is shown and the problems for solving in a classroom (a computer lab) are distributed. They can be used for self-study. You will find the answers and solutions to all the problems except for the simplest ones. Exercises differ in complexity level. The most difficult ones are marked with an asterisk ($*$), placed next to the exercise number. If the theoretical part is provided with an example, its end is designated by the symbol $\square$.

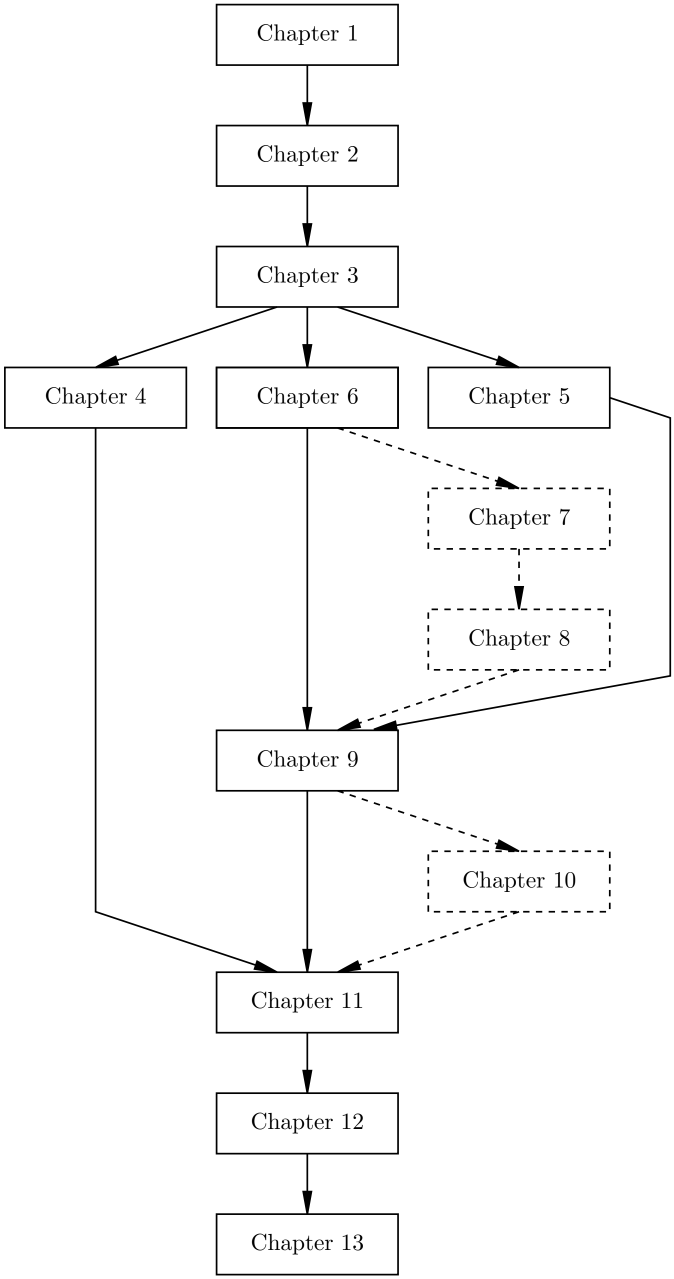
The differential characteristics of the study guide, in our opinion, are as follows: detailed problem solution containing typical methods and computational schemes; presence of a big amount of exercises and examples of different complexity levels; reflecting the subject field of information and communication technologies in problem definitions where we thought it is relevant. Moreover, taking into account the specificity of a bachelor's training programme in the field of such technologies, it is needed to implement a developed algorithm in a particular programming language.

In chapters "Concept of an Algorithm. Correctness of Algorithms" and "Basic Algorithms", fragments of code in the Python language are set in order to demonstrate the applicability of studied methods in programming practice. Headings of some problems in chapter "Basic Algorithms" are in bold; the abovementioned problems are recommended to do in a computer lab.

A separate chapter is dedicated to parallel algorithms. Multiprocessor computing systems are extensively used at present for solving tasks on mathematical and computer modeling, management of database and complex software packages, etc. Furthermore, the majority of modern computing systems back up parallel computations on a hardware level. From this perspective, questions connected with the construction and analysis of parallel algorithms are becoming increasingly relevant. It should be pointed out that in many cases developing an effective parallel algorithm for solution of some task requires attraction of new ideas and methods comparing to creating a sequential algorithm version. These are, for instance, practically important problems of searching a target element in data structures, evaluation of an algebraic expression, etc. The theoretical part of this chapter takes a relatively bigger place in comparison with other chapters, as it deals with more complex issues of applied aspects of parallel programming, rather difficult when first studied. This chapter requires good command of methods of problem solution described in the previous chapters and has, in the authors' opinion, increased complexity.

There is a sufficient amount of illustrations in the book, which help students visualize all objects under consideration and connectivity between them. In the final section you will find reference data which include the Greek alphabet, basic trigonometric formulae, brief summary on differential and integral calculi, and the most important finite summas, which will mitigate the need for resorting to the specialist reference literature. This edition is provided with the reference list, as well as the name and the subject indices. After the first mentioning of scientists' surnames, consult the references with their short biographies taken from [87].

Below you can see the scheme of the chapter information dependence in the form of an oriented graph reflecting the preferable order of covering the academic material. For instance, after having studied chapters 1, 2, and 3 you can move to one of the three chapters: 4, 5, or 6, whose contents are relatively independent. The dashed border marks the chapters which will be suitable for readers who want to reach a high level of the subject knowledge; the present sections contain more difficult academic material. This way, after studying chapter 6 you can either come to chapter 9, or study the material of chapters 7 and 8 more thoroughly for better digestion, and only then switch to chapter 9.



The chapter dependency chart

Contents

Notations

A **and** B	The conjunction of propositions A, B
A **or** B	The disjunction of propositions A, B
$A \Rightarrow B$	The conditional statement, or implication
not A	The negation of the proposition A
$\forall x\ (P(x))$	For all x, $P(x)$ is true
$\exists x\ (P(x))$	There exists x such that $P(x)$
F_n	The nth Fibonacci number
L_n	The nth Lucas number
H_n	The nth harmonic number
$\mathcal{C}_n$	The nth Catalan number
$\mathbb{N} = \{0, 1, 2, \ldots\}$	The set of natural numbers
$\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$	The set of integers
$\mathbb{R} = (-\infty, \infty)$	The set of real numbers
$\mathbb{C} = \mathbb{R} \times \mathbb{R}$	The set of complex numbers
$[a, b]$	The closed-interval from a to b: $\{x : a \le x \le b\}$
(a, b)	The open-interval from a to b: $\{x : a < x < b\}$
$A \subseteq B$	The set A is a subset of the B
$A \subset B$	The set A is a proper subset of the set B
U	The universal set
$\varnothing$	The empty set
$A \cup B$	The union of the sets A and B
$A \cap B$	The intersection of the sets A and B
$A \backslash B$	The complement of the set B with respect to the set A, or the difference of the sets A and B
$\overline{A}$	The complement of the set A
$A \Delta B$	The symmetric difference of sets A and B
$\|A\|$	The cardinality of the set A
$\mathcal{P}(A)$	The power set of the set A—that is the set of all subsets of A
$\aleph_0$	Aleph null, the cardinality of a countable set
$A \times B$	The Cartesian, or cross, or direct, product of the sets A and B
$(a_1, a_2, \ldots, a_n)$	The vector—that is the element of a set A^n
$x\ R\ y$	The element x is related to the element y
R^*	The closure of the relation R

$x \prec y$	x is the ancestor y, or the element x is less than the element y
$x \prec\!\cdot\, y$	The element y covers the element x
R^{-1}	The inverse relation of the relation R
$S \circ R$	The composite relation for R and S
$g \circ f$	The composition of the functions f and g
$\lfloor x \rfloor$	The floor function of x—that is the greatest integer less or equal to the real x
$\lceil x \rceil$	The ceiling function of x—that is the smallest integer greater or equal to the real x
$P(n, k)$	The number of (n, k)-permutations without repetition allowed
$\widetilde{P}(n, k)$	The number of (n, k)-permutations with repetition allowed
$C(n, k)$	The number of (n, k)-combinations without repetition allowed
$\widetilde{C}(n, k)$	The number of (n, k)-combinations with repetition allowed
$G(V, E)$	G is a graph with vertex set V and edge set E
$d(v)$	The degree of vertex v of a graph
$M(i,j)$	The elements of the adjacency matrix
$\widetilde{M}(i,j)$	The elements of the incidence matrix
$G_1 \sim G_2$	The graphs G_1 and G_2 are isomorphic
$\chi(G)$	The chromatic number of the graph G
$\chi'(G)$	The edge chromatic number of the graph G
$D(V, E)$	D is a digraph with vertex set V, E is a relation on V
$d^{+}(v)$	The out-degree of the vertex v in a digraph
$d^{-}(v)$	The in-degree of the vertex v in a digraph
$x_1 \wedge x_2$	The conjunction of x_1, x_2
$x_1 \vee x_2$	The disjunction of x_1, x_2
$x_1 \rightarrow x_2$	The implication of x_2 by x_1
$x_1 \leftrightarrow x_2$	The equivalence of x_1 and x_2
$x_1 \oplus x_2$	The Boolean sum of x_1 and x_2
$x_1 \mid x_2$	The Sheffer stroke
$x_1 \downarrow x_2$	The Peirce arrow
$O(g(n))$	The class of functions growing not faster than $g(n)$
$\Omega(g(n))$	The class of functions growing at least as fast as $g(n)$
$\Theta(g(n))$	The class of functions of the same order as $g(n)$
$B(n)$	The best-case time complexity of an algorithm to solve a problem of size n
$A(n)$	The average-case time complexity to solve a problem of size n
$W(n)$	The worst-case time complexity to solve a problem of size n
RAM	Random Access Machine
PRAM	Parallel Random Access Machine

p	The number of nodes in a computing system
S_p	The performance speedup using p processors
E_p	The efficiency of parallel computation using p processors
C_p	The cost of a parallel algorithm for p processors