Preface

Efficient problem-solving using computers, irrespective of the discipline or application, calls for the design of efficient algorithms. Inclusion of appropriate data structures is of critical importance to the design of efficient algorithms. In other words, *good algorithm design must go hand in hand with appropriate data structures for efficient program design to solve a problem*.

Data structures is a fundamental course in Computer Science which most undergraduate and graduate programmes in Computer Science, Computer Science and Engineering, and other allied engineering disciplines such as Computer Integrated Manufacturing, Product Design and Commerce and Communication Engineering , to list a few, offer during the first year or first semester of the programme. It is offered as a core or an elective course, enabling students to have the much needed foundation for efficient programming, leading to better problem-solving in their respective disciplines. Besides regular academic programmes, training programmes of the IT corporate sector and other institutes also offer a course on data structures either by way of certificate courses, diploma or post-diploma programmes.

Most of the well-known textbooks/monographs on this subject have discussed the concepts in relation to a programming language—beginning with Pascal and spanning a spectrum of them such as C, C++, C#, Java, and so on—essentially calling for a fair knowledge of the language, before one proceeds to understand the data structure. There does remain a justification in this, when one argues that the implementation of data structures in a specific programming language needs to be demonstrated or that the algorithms pertaining to the data structure need a convenient medium of presentation and when this is so, why not a programming language?

Again, while some authors have insisted on using their books for an advanced level course, there are some who insist on a working knowledge of the specific programming language as a pre-requisite to using the book. However, in the case of a core course, as it is in most academic programmes, it is not uncommon for a novice or a sophomore, to be bewildered by the 'miles of code' that demonstrate or explain a data structure, rendering the subject difficult. In fact, the effort that one needs to put in to comprehend the data structure and its applications, is distracted by the necessity to garner sufficient programming knowledge to follow the code. It is indeed ironical that while a novice is taught data structures to appreciate programming, in reality it turns out that one learns programming to appreciate data structures!

In my decades-old experience of offering the course to graduate programmes which admits students from heterogeneous undergraduate disciplines, with little or less strong knowledge of programming, I had several occasions to observe this malady.

In fact, it is not uncommon for some academic programmes, especially graduate programmes, which due to their shorter duration have a course in Programming and Data Structures running in parallel in the same semester, (much to the chagrin of the novice learner) that a novice is forced to learn data structures through its implementation (in a specific programming language), when in reality it ought to be learning augmented with implementation of the data structures, failure of which has been the reason behind the fallout.

A solution to this problem would be to (i) frame the course such that the theory deals with the concepts, techniques and applications of data structures, not taking recourse to any specific programming language, but instead settling for a pseudo-language which clearly expounds the data structure and supplementing the course material with illustrative problems and exercises to reinforce the students' grasp of the concepts, and (ii) augment the theory with laboratory sessions to enable the student implement the data structure in itself or as embedded in an application, in a language of his/her own choice or as insisted upon in the curriculum. This would enable the student who has acquired sufficient knowledge and insight into the data structures, to appreciate the beauty and the merits of employing the data structure by programming it himself or herself, rather than 'look' for the data structure in a pre-written code.

This means that textbooks catering to the fundamental understanding of the data structure concepts for use as course material in the classroom are as much needed as those books which cater to the implementation of data structures in a programming language for use in the laboratory sessions. While most books in the market conform to the latter, to bring out a book for use as classroom course material by instructors handling a course on data structures and comprehensive enough for the novice students to benefit, has been the main motivation in writing this book. In this direction, the book details concepts, techniques and applications pertaining to data structures, independent of any programming language, discusses several illustrative problems and poses review questions to reinforce the understanding of the theory, and presents a suggestive list of programming assignments to aid implementation of the data structures. In fact, the book may be independently used as a textbook since it is self-contained or serves as a companion for books discussing data structures implemented in a specific programming language such as C, C++, Java, etc.

The book lays an all-round emphasis on Theory, Applications, Illustrative Problems, Review Questions and Programming Assignments to enable the students comprehend, implement and appreciate data structures. The whole book is divided into five parts.

As an introduction, the need for data structures and some basic concepts pertaining to analysis of algorithms which is essential to appreciate algorithms associated with data structures, have been presented in chapters 1–2.

Part I details sequential linear data structures, viz., *arrays, stacks, queues, priority queues* and *dequeues*, and comprises chapters 3–5. **Part II** details linked linear data structures, viz., *linked lists, linked stacks* and *linked queues*, and comprises chapters 6–7. **Part III** elucidates the nonlinear data structures of *trees, binary trees* and *graphs* covering chapters 8–9. **Part IV** highlights the advanced data structures of *binary search trees, AVL trees, B trees, tries, red black trees, splay trees, hash tables* and *files,* which spans chapters 10–14. **Part V** spans chapters 15–17 and discusses searching algorithms of *linear search*, *transpose sequential search*, *interpolation search*, *binary search*, *Fibonacci search,* and *other search techniques,* and internal sorting techniques of *bubble sort, insertion sort, selection sort, merge sort, shell sort, quick sort, heap sort* and *radix sort*, and external sorting techniques of *sorting with tapes*, *sorting with disks*, *polyphase merge sort* and *cascade merge sort*.

The concepts and techniques behind each data structure and their applications have been explained. Every chapter includes a variety of Illustrative Problems pertaining to the data structure(s) detailed, a summary of the technical content of the chapter and a list of Review Questions, to reinforce the comprehension of the concepts. A set of Programming Assignments to be implemented in the laboratory sessions, have also been listed at the end of the appropriate chapters.

The book could be used both as an introductory or an advanced-level textbook for the undergraduate, graduate and research programmes which offer data structures as a core or an elective course. While the book is primarily meant to serve as a course material for use in the classroom, it could be used as a companion guide during the laboratory sessions to nurture better understanding of the theoretical concepts.

The book could also serve as a course material for various diploma, post-diploma programmes and certificate courses conducted by various IT and related institutes and corporate sectors.

An introductory level course for a duration of one semester, targeting an undergraduate programme or a first-year graduate programme or a diploma programme or a certificate course, could include chapters 1–2, PART I, PART II, chapter 8 of PART III, chapter 13 of PART IV, chapter 15 (Sec. 15.1–15.2, 15.5) and chapter 16 (Sec. 16.1–16.3, 16.5, 16.7) of PART V in its curriculum.

A middle-level course for a duration of one semester, targeting senior graduate-level programmes and research programmes such as MS/Ph D, could include chapters 1–2, PART I, PART II, PART III, chapters 10, 11 and 13 of PART IV, and selective sections of chapters 15–16 of PART V.

An advanced-level course could include parts IV and V besides selections from the rest, based on the prerequisite courses satisfied.

Chapters 8, 10, 11 (Sec. 11.10–11.3), 13, 14 and 17 could be useful for inclusion in a curriculum that serves as a prerequisite for a course on Database Management Systems.

The salient features of the book are as follows:

- All-round emphasis on theory, problems, applications and programming assignments
- Simple and lucid explanation of the theory
- Inclusion of several applications to illustrate the use of data structures
- Several worked-out examples as Illustrative Problems in each chapter
- List of Programming Assignments at the end of each chapter
- Review Questions to strengthen understanding
- Self-contained text for use as a textbook for either an introductory or advanced-level course

The book is accompanied by a web supplement that can be accessed at *www.mhhe.com/pai/dsa*. It includes the following online material:

- **Slide Presentation**
  The slides illustrative of the technical content in each chapter of the book could be effectively used by the instructor to supplement classroom teaching.
- **Solution Manual**
  Solutions to selected problems in each chapter are given here.
- **C Programs**
  C implementation of algorithms, demonstrative of selective data structures, discussed in the book have been given here.

I express my sincere thanks to the Management and Principal, PSG College of Technology, Coimbatore, for the encouragement and support provided by them. I also express my appreciation for the editorial and production teams of McGraw-Hill Education (India) Limited, New Delhi, for the excellent production values.

Thanks are also due to all the reviewers who went through the text and provided noteworthy suggestions and advice. Their names are listed below.

Sanjay Goswami	*Department of Computer Applications, Narula Institute of Technology, Kolkata*
Sanjoy Kumar Saha	*Department of Computer Science and Engineering, Jadavpur University, Kolkata*
Naveen Rajpal	*University School of Information Technology, Guru Gobind Singh Indraprastha University, New Delhi*
Amit Jain	*Computer Science/Information Technology, Radha Govind Engineering College, Meerut*
Nilima Fulmare	*Hindustan College of Science and Technology Agra*
T.V. Gopal	*Department of Computer Science and Engineering, Anna University, Chennai*
Suganthi Jeyaraj	*Department of Computer Science and Engineering, PSG College of Technology, Coimbatore*
Jibi Abraham	*Department of Computer Science and Engineering, M.S. Ramaiah Institute of Technology, Bangalore*

I would like to place on record my reverence for my mother whose blessings and prayers have been a source of inspiration and great strength. True to the Indian spiritual tradition, I offer my reverent salutations to my spiritual guru Srimat Swami Vireswaranandaji Maharaj, the tenth President of the Ramakrishna Math and Mission. Lastly, the infinite support, encouragement and help provided by my sisters Rekha and Udaya in all my endeavors, are affectionately remembered.

While I hope that the book would be beneficial to novices and sophomores alike, constructive feedback and suggestions for improvement may kindly be mailed to *vijipai@vsnl.com*

G A V Pai