Preface

Data are the blood of any application that we develop. A data structure is like the viscera that pumps and carries that blood. In other words, data structure is the heart of any application we design. Selection of a data structure can drastically enhance the performance of the system.

The main motivation behind this book is to show the application of several data structures for different diverse problems. Matrix multiplication can be used to calculate the total revenues of a chain of departmental stores. In other data structure books we find the logic of matrix multiplication but their applications are not available. Moreover, data structure is taught as a stand-alone subject devoid of application of other fields of science. In this book the relationship between data structure and other fields of science like numerical methods, applied statistics, physics, etc., are shown with a variety of problems. While I studied data structure, there was no such book available which covered all the basic algorithms of different data structures and their applications. So I decided to write one for future students of data structures.

This book is mainly written for the students of BE, BSc (Comp Sc.), BSc (IT), BCA, and MCA but it can be used by students of any data structure course because it covers a variety of data structures and teaches its readers how to create their own data structures to serve a specific need. Students who are preparing for GATE examination on Computer Science will also find this book useful. Undergraduate students who know about C arrays and structures can skip chapters 1 and 2. Other than this exception, the chapters are arranged in an order in which they should be read.

"Before we start" discusses what a data structure is and why it is needed. This section gives the reader an overview of a data structure and its importance in programming different applications.

Chapter 1 discusses arrays, starting from declaring an array to how arrays can be used in console based animation programs.

Chapter 2 discusses structures that is the basic building block of all the data structures.

Chapter 3 discusses all kinds of linked lists. Linked list is the most basic pointer-based data structure and is the building block of other different data structures.

Chapter 4 discusses string handling. This chapter covers different string processing functions and their applications in solving some interesting problems like "UPC product code verification", "Credit card number verification", etc.

Chapter 5 discusses recursion, a powerful programming technique. This chapter shows how recursion can be useful in diverse programmatic situations starting from a very common example of generating Fibonacci numbers to solving nonlinear equations using recursive root finding methods and recursive creation of fractals like Koch Snowflake.

Chapter 6 discusses stack data structure. Besides describing the usual push, pop algorithms, in this chapter a new data structure, MTF list, has been introduced and modeled using stacks. A tray of stack is also modeled under the name of saguaro stacks. Stacks are used greatly in parsers. An XML parser is written to introduce the capacity which this simple data structure has to offer.

Chapter 7 discusses Queue data structure. In this chapter it has been shown how real-time queues can be simulated using queues. MTF is also modeled using queues.

Chapter 8 discusses Tree data structure. In this chapter different types of tree data structure and their methods have been discussed. Trees are very important data structures and find applications in lot of problems. A diverse set of problems have been identified and presented in order to showcase this immense capacity of tree data structures.

Chapter 9 discusses Graph data structure. Graph is probably the most complicated data structure and has application in almost all fields of science and technology. In this chapter the basic graph theory algorithms are implemented in order to get the reader interested in this subject. Graph theory is so massive that complete coverage of the algorithms available till date are out of scope of this book and can easily be accommodated as content of another title.

Chapter 10 discusses sorting algorithms. The chapter starts with a broader classification of algorithms and then implementation of the algorithms with their time and space complexities shown on tabular and graphical ways. There are good comparison graphs that show which algorithm is better. Apart from these, the chapter has a list of problems where sorting is the key to solve.

Chapter 11 discusses hashing techniques. Broad classification of hashing algorithms have been classified and implemented in full-length programs. Besides this, it has been shown how hashing can be used in computer security softwares.

Chapter 12 discusses ADT. This chapter shows how to create a new ADT, what are the different types of methods an ADT can have and how to distinguish them, etc. This chapter must be read before the chapters 13, 14 and 15.

Chapter 13 discusses Date data structure. This chapter shows how to create different functions that deal with dates. Knowledge of these will enable the reader to implement date-involved calculations in different applications.

Chapter 14 discusses Map data structure. A map is basically a hash table that is nothing but a collection of key-value pairs. This chapter shows how a simple map can be used in design of a phone book, a dictionary and a Random Ciphering Machine (RCM).

Chapter 15 discusses the Currency data structure. A Currency can also be represented as a double. But that doesn't convey the intention of the programmer. Thus the code becomes less readable and in turn, less maintainable. So in order to create a more programmer-friendly code, a separate data type for currency is needed. In this chapter it has been shown how different currencies can be modeled using C structures and several methods are also defined that operate on this data structure.

Chapter 16 discusses File Handling in C. This chapter mainly shows how to save and retrieve data from permanent storage files. Before reading this chapter, reading the chapter on string is mandatory.

Appendix A: Project Ideas. This section gives some exciting and innovative ideas to the reader to implement.

Appendix B: Bibliography, for further reading.

I am open to constructive criticism, comments and suggestion for the improvement of this book. Please let me know if you find anything worth reporting.

O'Fallon, MO — SUDIPTA MUKHERJEE