DATA STRUCTURES **USING C**

REEMA THAREJA
Assistant Professor
Department of Computer Science
Shyama Prasad Mukherjee College for Women
University of Delhi

OXFORD
UNIVERSITY PRESS

Contents

Preface *v*

1. INTRODUCTION TO C **1**

1.1 Introduction *1*
1.2 Keywords and Identifiers *1*
1.3 Basic Data Types in C *3*
1.4 Constants and Variables *4*
1.5 Writing the First C Program *5*
1.6 Header Files *7*
1.7 Input/Output Statement in C *8*
1.8 Operators in C *12*
1.9 Type Conversion and Typecasting *24*
1.10 Decision Control Statements *26*
1.11 Iterative Statements *37*
1.12 Break and Continue Statement *58*
Annexure ASCII Table and Description 67

2. FUNCTIONS **68**

2.1 Introduction *68*
2.2 Function Declaration *70*
2.3 Function Definition *71*
2.4 Function Call *71*
2.5 Passing Parameters to the Function *74*
2.6 Scope of Variables *78*
2.7 Storage Classes *80*
2.8 Recursive Functions *83*
2.9 Types of Recursion *88*
2.10 Recursion Vs Iteration *90*
2.11 Tower of Hanoi *91*

3. POINTERS **97**

3.1 Introduction *97*
3.2 Declaring Pointer Variables *98*
3.3 Pointer Expressions and Pointer Arithmetic *100*
3.4 Null Pointers *106*
3.5 Generic Pointers *107*
3.6 Passing Arguments to Function Using Pointers *108*
3.7 Pointer to Function *110*
3.8 Pointer to Pointers *111*

4. INTRODUCTION TO DATA STRUCTURES 115

4.1 Introduction *115*
4.2 Types of Data Structures *115*
4.3 Abstract Data Type *119*
4.4 Algorithm *121*
4.5 Key Features of an Algorithm *121*
4.6 Time and Space Complexity *124*
4.7 Big-Oh Notation *127*

5. ARRAYS 133

5.1 Introduction *133*
5.2 Declaration of Arrays *134*
5.3 Accessing Array Elements *135*
5.4 Storing Values in Arrays *136*
5.5 Calculating the Length of an Array *138*
5.6 Operations on Arrays *138*
5.7 One-dimensional Arrays for Inter-function Communication *160*
5.8 Pointers and Arrays *166*
5.9 Array of Pointers *170*
5.10 Two-dimensional Arrays *171*
5.11 Operations on 2D Arrays *174*
5.12 2D Arrays for Inter-function Communication *179*
5.13 Pointers and Two-dimensional Arrays *183*
5.14 Array of Function Pointers *185*
5.15 Multi-dimensional Arrays *186*
5.16 Sparse Matrices *188*

6. STRINGS 194

6.1 Introduction *194*
6.2 String Operations *197*
6.3 Array of Strings *214*
6.4 Pointers and Strings *226*

7. STRUCTURES 240

7.1 Introduction *240*
7.2 Nested Structures *246*
7.3 Arrays of Structures *248*
7.4 Structures and Functions *251*
7.5 Self-referential Structures *263*

Annexure Memory Allocation in C Programs 267

8. LINKED LISTS 274

8.1 Introduction *274*
8.2 Linked List Versus Arrays *276*
8.3 Memory Allocation and De-allocation for a Linked List *277*
8.4 Singly Linked List *278*
8.5 Polynomial Representation *297*
8.6 Circular Linked List *302*

8.7 Doubly Linked List *315*
8.8 Circular Doubly Linked List *332*
8.9 Header Linked List *345*

9. STACKS AND QUEUES **351**

9.1 Stacks *351*
9.2 Array Representation of Stacks *352*
9.3 Operations on a Stack *353*
9.4 Linked Representation of Stack *357*
9.5 Operations on a Linked Stack *357*
9.6 Multiple Stacks *361*
9.7 Infix, Postfix, and Prefix Notation *364*
9.8 Evaluation of an Infix Expression *366*
9.9 Convert Infix Expression to Prefix Expression *372*
9.10 Applications of Stack *376*
9.11 Queues *378*
9.12 Array Representation of Queues *379*
9.13 Circular Queue *382*
9.14 Linked Representation of a Queue *387*
9.15 Operations on a Queue *387*
9.16 Deques *392*
9.17 Priority Queues *397*
9.18 Multiple Queues *400*
Annexure The Stack Abstract Data Type 404

10. TREES **406**

10.1 Binary Trees *406*
10.2 Expression Trees *412*
10.3 Tournament Trees *413*
10.4 Traversing of a Binary Tree *414*
10.5 Huffman's Tree *417*

11. EFFICIENT BINARY TREES **428**

11.1 Binary Search Trees *428*
11.2 Operations on Binary Search Trees *430*
11.3 Threaded Binary Tree *447*
11.4 AVL Trees *451*
11.5 M-way Search Trees *462*
11.6 B-trees *464*
11.7 B+ trees *470*
11.8 Trie *471*
11.9 Red-black Tree *473*
11.10 Splay Tree *485*
Annexure Program for Simple Right In-threaded Binary Tree 494

12. HEAPS **501**

12.1 Binary Heaps *501*
12.2 Binomial Trees and Binomial Heaps *506*

12.3 Fibonacci Heaps *519*
12.4 Comparison of Binary, Binomial, and Fibonacci Heaps *528*
12.5 Applications of Heaps *529*

13. GRAPHS **533**

13.1 Introduction *533*
13.2 Transitive Closure of a Directed Graph *536*
13.3 Bi-connected Graphs *537*
13.4 Representation of Graphs *538*
13.5 Graph Traversal Algorithms *544*
13.6 Topological Sorting *551*
13.7 Shortest Path Algorithms *557*
13.8 Applications of Graphs *574*
Annexure Program for Finding Minimum Spanning Tree 579

14. SORTING **581**

14.1 Introduction *581*
14.2 Bubble Sort *581*
14.3 Insertion Sort *586*
14.4 Selection Sort *588*
14.5 Merge Sort *591*
14.6 Quick Sort *595*
14.7 Radix Sort *599*
14.8 Heap Sort *603*
14.9 Shell Sort *606*
14.10 Comparison of Sorting Algorithms *609*

15. HASHING AND COLLISION **613**

15.1 Introduction *613*
15.2 Hash Table *614*
15.3 Hash Function *615*
15.4 Different Hash Functions *615*
15.5 Collisions *618*
15.6 Collision Resolution by Chaining *629*
15.7 Pros and Cons of Hashing *633*
15.8 Applications of Hashing *633*

16. FILES AND THEIR ORGANIZATION **638**

16.1 Introduction *638*
16.2 Data Hierarchy *638*
16.3 File Attributes *639*
16.4 Text Files and Binary Files *640*
16.5 Basic File Operations *641*
16.6 File Organization *642*
16.7 Indexing *646*
16.8 External Sorting *651*

Index 655