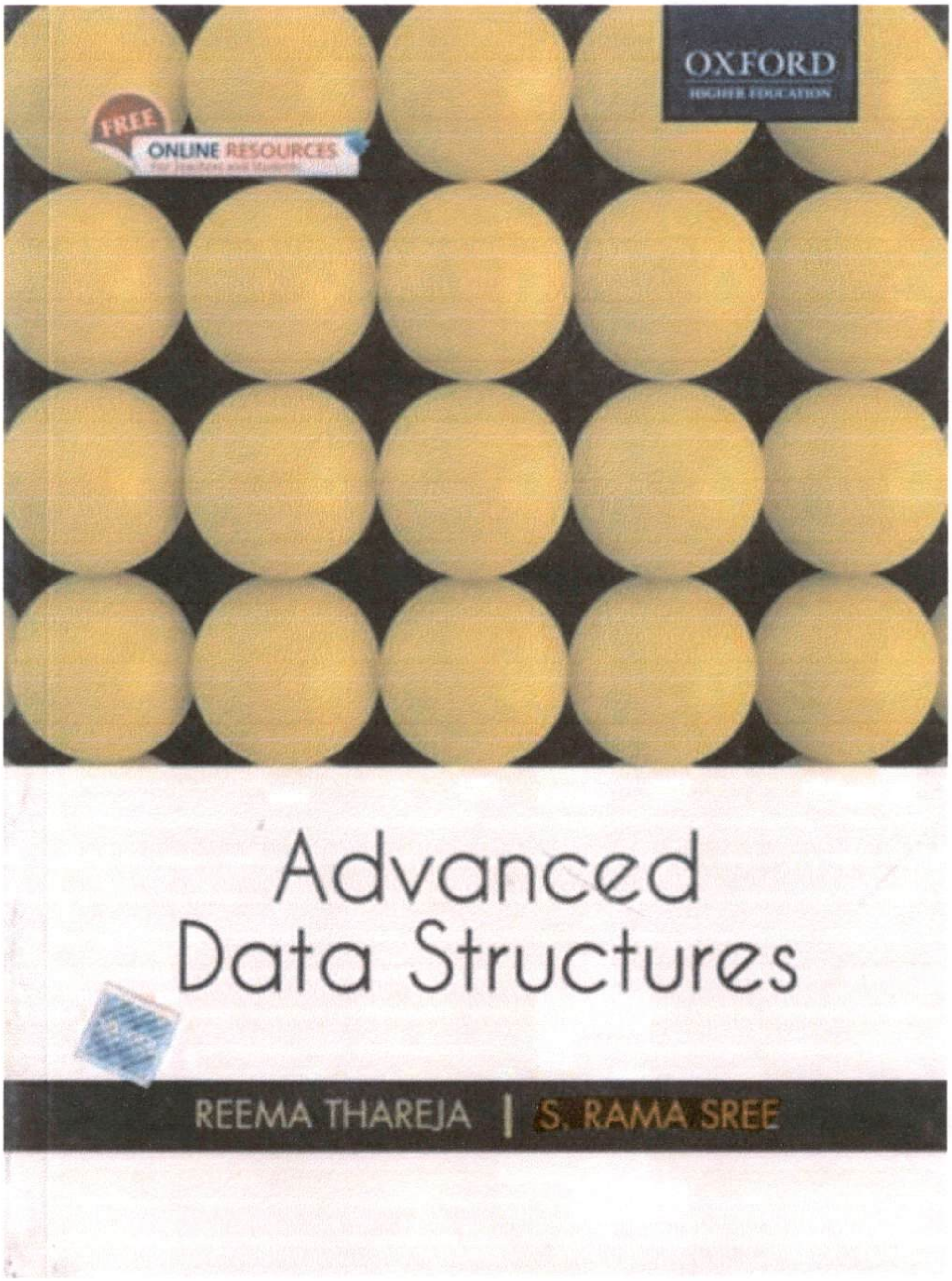
ADITYA ENGINEERING COLLEGE

An Autonomous Institution
Approved by AICTE • Permanently Affiliated to JNTUK • Accredited by NAAC with 'A' Grade
Recognised by UGC under sections 2(f) and 12(B) of UGC Act, 1956
Aditya Nagar, ADB Road, Surampalem - 533437, Near Kakinada, E.G.Dt., Ph:99498 76662

List of Books published during the year 2017

S. No.	Title of the Book	Page No.
1.	Advanced Data Structures	1-7



OXFORD
HIGHER EDUCATION

FREE
ONLINE RESOURCES
For Teachers and Students

Advanced Data Structures

REEMA THAREJA | S. RAMA SREE

PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437

Detailed Contents

Preface iv

Acknowlegements vii

1. External Sorting 1

1.1 Introduction *1*
1.2 k–way Merge Sort *2*
1.2.1 2-way Merge Sort *2*
1.2.2 3–way Merge Sort *3*
1.2.4 4–way Merge Sort *3*
1.2.4 Merging Runs *4*
1.2.5 Time Complexity *6*
1.3 Buffer Handling for Parallel Operation *7*
1.4 Run Generation *15*
1.5 Optimal Merging of Runs *17*

2. Hashing 27

2.1 Introduction to Static Hashing *27*
2.2 Hash Tables *28*
2.3 Hash Functions *29*
2.4 Different Hash Functions *30*
2.4.1 Division Method *30*
2.4.2 Multiplication Method *31*
2.4.3 Mid-Square Method *31*
2.4.4 Folding Method *32*
2.5 Secure Hash Functions *32*
2.6 Collision Resolution (or Overflow Handling) Techniques *33*
2.6.1 Open Addressing Techniques *33*
2.6.1.1 Linear Probing *33*
2.6.1.2 Quadratic Probing *39*
2.6.1.3 Double Hashing *42*
2.6.1.4 Rehashing *45*
2.6.2 Chaining Technique *49*
2.6.3 Comparison between Open Hashing and Closed Hashing Techniques used for Resolving Collisions *53*

PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437

2.7 Dynamic Hashing 53

2.7.1 Motivation for Dynamic Hashing 53

2.7.2 Dynamic Hashing (or Extended Hashing) Using Directories 54

2.7.3 Directory-less Dynamic Hashing 57

2.8 Pros and Cons of Hashing 58

2.9 Applications of Hashing 59

3. Trees 63

3.1 Introduction 63

3.1.1 Basic Terminology 63

3.2 Types of Trees 64

3.2.1 General Trees 64

3.2.2 Forests 64

3.2.3 Binary Trees 65

3.2.4 Binary Search Trees 69

3.2.5 Expression Trees 69

3.2.6 Tournament Trees 70

3.3 Creating a Binary Tree from a General Tree 70

3.4 Traversing a Binary Tree 72

3.4.1 Pre-order Traversal 72

3.4.2 In-order Traversal 73

3.4.3 Post-order Traversal 73

3.4.4 Level-order Traversal 74

3.4.5 Constructing a Binary Tree from Traversal Results 74

3.5 Applications of Trees 75

4. Priority Queues (Heaps) 79

4.1 Introduction—Priority Queues 79

4.2 Binary Heaps—Model and Simple Implementation 83

4.2. Binary Heap – Structure Property 84

4.2.2 Binary Heap – Order Property 84

4.3 Basic Heap Operations 85

4.3.1 Inserting a New Element in a Binary Heap 85

4.3.2 Deleting an Element from a Binary Heap 87

4.4 Other Heap Operations 88

4.5 Applications of Priority Queues 92

4.5.1 The Selection Problem 92

4.5.2 Event Simulation 93

4.6 Binomial Heaps (or Queues) 94

4.7 Binomial Heap (or Queue) Structure and Implementation 95

PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437

4.8 Binomial Queue Operations *96*
4.8.1 Creating a New Binomial Heap *96*
4.8.2 Finding the Node with Minimum Key *96*
4.8.3 Linking and Uniting Two Binomial Heaps *97*
4.8.4 Inserting a New Node *99*
4.8.5 Extracting the Node with Minimum Key *100*
4.8.6 Decreasing the Value of a Node *101*
4.8.7 Deleting a Node *101*
4.8.8 Lazy Binomial Queue *103*
4.9 Comparison between Binary and Binomial Heaps *103*

5. Efficient Binary Search Trees **107**
5.1 Binary Search Trees *107*
5.1.1 Operations on Binary Search Trees *109*
5.1.1.1 Searching for a Node in a Binary Search Tree *109*
5.1.1.2 Inserting a New Node in a Binary Search Tree *110*
5.1.1.3 Deleting a Node from a Binary Search Tree *111*
5.2 Optimal Binary Search Tree (OBST) *116*
5.3 Self-balancing Binary Search Tree *122*
5.4 AVL Trees *122*
5.4.1 Operations on AVL Trees *124*
5.5 Red-Black Trees *140*
5.5.1 Properties and Representation of Red-Black Trees *140*
5.5.2 Operations on Red-Black Trees *141*
5.5.2.1 Searching for a Node in a Red-Black Tree *141*
5.5.2.2 Inserting a Node in a Red-Black Tree *142*
5.4.2.3 Deleting a Node from a Red-Black Tree *148*
5.5.2.4 Joining and Splitting Red-Black Tree *152*
5.5.3 Applications of Red-Black Trees *155*

6. Multi-way Search Trees **159**
6.1 M-way Search Trees *159*
6.1.1 Definition and Properties *159*
6.1.2 Searching an M-way Search Tree *160*
6.2 B-Trees *161*
6.2.1 Definitions *161*
6.2.2 Properties of B-trees *161*
6.2.3 Number of Elements in a B-tree *161*
6.2.4 Searching for an Element in a B-tree *162*
6.2.5 Inserting a New Element in a B-trees *163*

PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437

6.2.6 Deleting an Element from a B-tree *164*

6.2.7 Applications of B-trees *172*

6.3 B+ Trees *172*

6.3.1 Searching a B+ Tree *173*

6.3.2 Inserting a New Element in a B+ Tree *173*

6.3.3 Deleting an Element from a B+ Tree *174*

6.4 2-3 Trees *175*

6.4.1 Searching for an Element in a 2-3 Tree *176*

6.4.2 Inserting a New Element in a 2-3 Tree *177*

6.4.3 Deleting an Element from a 2-3 Tree *178*

7. Digital Search Structures **183**

7.1 Introduction to Digital Search Tree *183*

7.1.1 Operations on Digital Search Trees *184*

7.1.1.1 Insertion *184*

7.1.1.2 Searching *186*

7.1.1.3 Deletion *188*

7.2 Binary Tries and Patricia *191*

7.2.1 Binary Tries *191*

7.2.2 Compressed Binary Trie *192*

7.2.3 Patricia *192*

7.2.3.1 Searching Patricia *193*

7.2.3.2 Inserting into Patricia *195*

7.2.3.3 Delete a node from Patricia *197*

7.3 Muti-way Tries *200*

7.3.1 Definition *200*

7.3.2 Searching a Trie *203*

7.3.3 Sampling Strategies *203*

7.3.4 Inserting into a Trie *205*

7.3.5 Deletion from a Trie *206*

7.3.6 Keys with Different Length *206*

7.3.7 Height of a Trie *207*

7.3.8 Space Required and Alternative Node Structures *207*

7.3.9 Prefix Search and Applications *209*

7.3.10 Compressed Tries *211*

7.3.11 Compressed Tries with Digit Numbers *211*

7.3.11.2 Inserting into a Compressed Trie with Digit Numbers *212*

7.3.11.3 Deletion of Element from Compressed tries with Digit Numbers *213*

7.3.12 Compressed Tries with Skip Fields *214*

7.3.12.1 Searching a Compressed Tries with Skip *215*

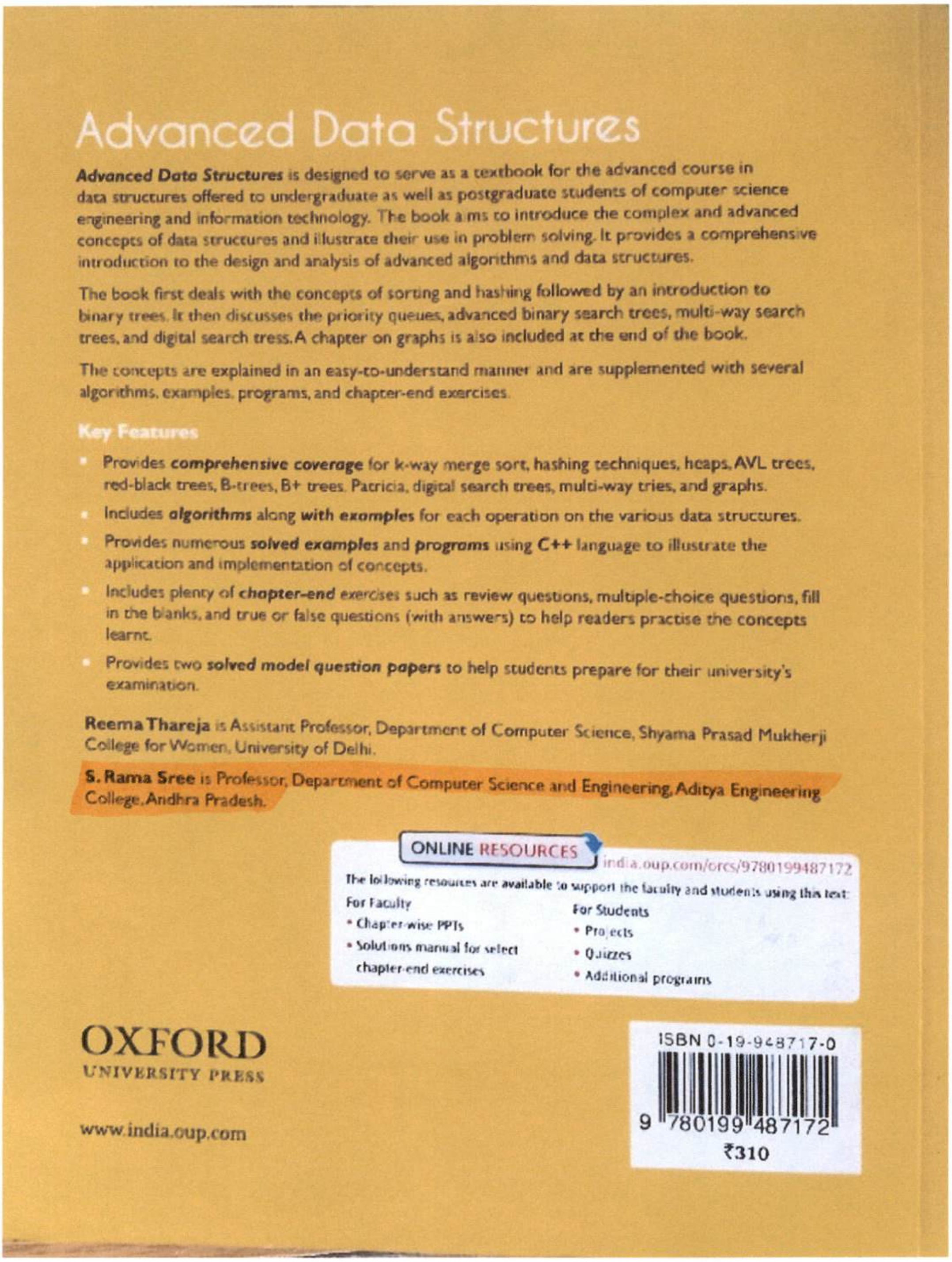
PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAM... M - 533 437

7.3.12.2 Inserting into a Compressed Trie with Skip Fields 215
7.3.12.3 Deleting an Element from a Compressed Trie with Skip Fields 217
7.3.13 Compressed Tries with Labeled Edges 217
7.3.13.1 Searching a Compressed Trie with Labeled Edges 218
7.3.13.2 Inserting into a Compressed Trie with Labeled Edges 218
7.3.13.3 Deleting an Element from a Compressed Trie with Labeled Edges 219
7.4 Tries and Internet Packet (IP) Forwarding 220
7.4.1 IP Routing 220
7.4.2 1-bit Tries 221
7.4.3 Fixed–stride Tries 221
7.4.4 Variable–stride Tries 222

8. Graphs 227
8.1 Introduction 227
8.2 Graph Terminology 228
8.3 Directed Graphs 229
8.3.1 Terminology of a Directed Graph 229
8.3.2 Transitive Closure of a Directed Graph 230
8.4 Bi-connected Components 232
8.5 Representation of Graphs 232
8.5.1 Adjacency Matrix Representation 232
8.5.2 Adjacency List Representation 235
8.6 Graph Traversal Algorithms 237
8.6.1 Breadth-First Search 238
8.6.2 Depth-First Search 241
8.7 Topological Sorting 244
8.8 Minimum Spanning Trees 250
8.8.1 Kruskal's Algorithm 252
8.8.2 Prim's Algorithm 256
8.9 Shortest Path Algorithms 260
8.9.1 Dijkstra's Algorithm 260
8.9.2 Bellman-Ford Algorithm 263
8.9.3 Warshall's Algorithm 267
8.9.4 Floyd-Warshall Algorithm 268
8.10 Applications of Graphs 271

Appendix: Answers to Objective-type Questions 275
Solved Model Question Paper-I 278
Solved Model Question Paper-II 280
About the Authors 282

PRINCIPAL
ADITYA ENGINEERING COLLEGE
SUR 4 - 5 37



Advanced Data Structures

Advanced Data Structures is designed to serve as a textbook for the advanced course in data structures offered to undergraduate as well as postgraduate students of computer science engineering and information technology. The book aims to introduce the complex and advanced concepts of data structures and illustrate their use in problem solving. It provides a comprehensive introduction to the design and analysis of advanced algorithms and data structures.

The book first deals with the concepts of sorting and hashing followed by an introduction to binary trees. It then discusses the priority queues, advanced binary search trees, multi-way search trees, and digital search tress. A chapter on graphs is also included at the end of the book.

The concepts are explained in an easy-to-understand manner and are supplemented with several algorithms, examples, programs, and chapter-end exercises.

Key Features

- Provides ***comprehensive coverage*** for k-way merge sort, hashing techniques, heaps, AVL trees, red-black trees, B-trees, B+ trees, Patricia, digital search trees, multi-way tries, and graphs.
- Includes ***algorithms*** along ***with examples*** for each operation on the various data structures.
- Provides numerous ***solved examples*** and ***programs*** using ***C++*** language to illustrate the application and implementation of concepts.
- Includes plenty of ***chapter-end*** *exercises* such as review questions, multiple-choice questions, fill in the blanks, and true or false questions (with answers) to help readers practise the concepts learnt.
- Provides two ***solved model question papers*** to help students prepare for their university's examination.

Reema Thareja is Assistant Professor, Department of Computer Science, Shyama Prasad Mukherji College for Women, University of Delhi.

S. Rama Sree is Professor, Department of Computer Science and Engineering, Aditya Engineering College, Andhra Pradesh.

ONLINE RESOURCES india.oup.com/orcs/9780199487172

The following resources are available to support the faculty and students using this text:

For Faculty
- Chapter-wise PPTs
- Solutions manual for select chapter-end exercises

For Students
- Projects
- Quizzes
- Additional programs

OXFORD
UNIVERSITY PRESS

www.india.oup.com

ISBN 0-19-948717-0
9 780199 487172
₹310

PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437