Data Structures And Algorithms Made Easy

-To All My Readers

By
Narasimha Karumanchi

☼ Concepts ☼ Problems ☼ Interview Questions

Copyright© 2015 by *CareerMonk.com*

All rights reserved.

Designed by *Narasimha Karumanchi*

Copyright© 2015 CareerMonk Publications. All rights reserved.

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without written permission from the publisher or author.

Acknowledgements

Mother and *father*, it is impossible to thank you adequately for everything you have done, from loving me unconditionally to raising me in a stable household, where you persistent efforts traditional values and taught your children to celebrate and embrace life. I could not have asked for better parents or role-models. You showed me that anything is possible with faith, hard work and determination.

This book would not have been possible without the help of many people. I would like to express my gratitude to many people who saw me through this book, to all those who provided support, talked things over, read, wrote, offered comments, allowed me to quote their remarks and assisted in the editing, proofreading and design. In particular, I would like to thank the following individuals.

- *Mohan Mullapudi*, IIT Bombay, Architect, dataRPM Pvt. Ltd.
- *Navin Kumar Jaiswal*, Senior Consultant, Juniper Networks Inc.
- *A.Vamshi Krishna*, IIT Kanpur, Mentor Graphics Inc.
- *Kondrakunta Murali Krishna,* B-Tech., Technical Lead, HCL
- *Prof.Girish P.Saraph, Founder,Vegayan Systems,IIT Bombay*
- *Kishore Kumar Jinka,* IIT Bombay
- *Prof.Hsin − mu Tsai, National Taiwan University, Taiwan*
- *Prof.Chintapalli Sobhan Babu. IIT, Hyderabad*
- *Prof.Meda Sreenivasa Rao, JNTU, Hyderabad*

Last but not least, I would like to thank *Directors* of *Guntur Vikas College*, *Prof.Y.V.Gopala Krishna Murthy* & *Prof.Ayub Khan* [*ACE Engineering Academy*], *T.R.C.Bose* [*Ex. Director* of *APTransco*], *Ch.Venkateswara Rao VNR Vignanajyothi* [*Engineering College*, *Hyderabad*], *Ch.Venkata Narasaiah* [*IPS*], *Yarapathineni Lakshmaiah* [*Manchikallu*, *Gurazala*] & *all our well − wishers* for helping me and my family during our studies.

−Narasimha Karumanchi
M-Tech, *IIT Bombay*
Founder, *CareerMonk.com*

Preface

Dear Reader,

Please Hold on! I know many people do not read the preface. But I would strongly recommend that you go through the preface of this book at least. The reason for this is that this preface has *something different* to offer.

The main objective of the book is not to present you the theorems and proofs about *data structures* and *algorithms*. I have followed a pattern of improving the problem solutions with different complexities (for each problem, you will find multiple solutions with different, and reduced complexities). Basically, it's an enumeration of possible solutions. With this approach, even if you get a new question it will show you a way to think about all possible solutions. This book is very useful for interview preparation, competitive exams preparation, and campus interview preparations.

As a *job seeker* if you read the complete book with good understanding, I am sure you will challenge the interviewers and that is the objective of this book. If you read it as an *instructor*, you will deliver better lectures with an easy approach and as a result your students will appreciate the fact that they have opted for Computer Science / Information Technology as their degree.

This book is very useful for the *students* of *Engineering Degree* and *Masters* during their academic preparations. In all the chapters you will see that more importance has been given to problems and their analysis instead of theory. For each chapter, first you will read about the basic required theory and this will be followed by a section on problem sets. There are approximately 700 algorithmic problems and all of them are with solutions.

If you read as a *student* preparing for competitive exams for Computer Science/Information Technology, the content of this book covers *all* the *required* topics in full detail. While writing this book, my focus has been to help students who are preparing for these exams.

In all the chapters you will see more importance given to problems and analyzing them instead of concentrating more on theory. For each chapter, first you will see the basic required theory and then followed by problems.

For many problems, *multiple* solutions are provided with different levels of complexities. We start with *brute force* solution and slowly move towards the *best solution* possible for that problem. For each problem we will try to understand how much time the algorithm is taking and how much memory the algorithm is taking.

It is *recommended* that the reader does at least one complete reading of this book to get full understanding of all the topics. In the subsequent readings, you can go directly to any chapter and refer. Even though, enough readings were given for correcting the errors, there could be some minor typos in the book. If any such typos are found, they will be updated at *www.CareerMonk.com*. I request you to constantly monitor this site for any corrections, new problems and solutions. Also, please provide your valuable suggestions at: *Info@CareerMonk.com*.

Wish you all the best. I am sure that you will find this book useful.

—Narasimha Karumanchi
M-Tech, *IIT Bombay*
Founder, *CareerMonk.com*

Other Titles by Narasimha Karumanchi

- IT Interview Questions
- Data Structures and Algorithms for GATE
- Data Structures and Algorithms Made Easy in Java
- Coding Interview Questions
- Peeling Design Patterns
- Elements of Computer Networking
- Data Structure and Algorithmic Thinking with Python

Table of Contents

1. Introduction-- 13

1.1 Variables -- 13

1.2 Data types-- 13

1.3 Data Structures --- 14

1.4 Abstract Data Types (ADTs) --- 14

1.5 What is an Algorithm? -- 14

1.6 Why Analysis of Algorithms?--- 15

1.7 Goal of Analysis of Algorithms --- 15

1.8 What is Running Time Analysis? --- 15

1.9 How to Compare Algorithms? -- 15

1.10 What is Rate of Growth? -- 15

1.11 Commonly used Rate of Growths --- 16

1.12 Types of Analysis --- 17

1.13 Asymptotic Notation --- 17

1.14 Big-O Notation -- 17

1.15 Omega-Ω Notation-- 19

1.16 Theta-Θ Notation --- 19

1.17 Why is it called Asymptotic Analysis?-- 20

1.18 Guidelines for Asymptotic Analysis--- 20

1.19 Properties of Notations -- 22

1.20 Commonly used Logarithms and Summations -- 22

1.21 Master Theorem for Divide and Conquer -- 22

1.22 Problems on Divide and Conquer Master Theorem -- 23

1.23 Master Theorem for Subtract and Conquer Recurrences ----------------------------------- 23

1.24 Variant of subtraction and conquer master theorem -- 24

1.25 Method of Guessing and Confirm --- 24

1.26 Amortized Analysis --- 25

1.27 Problems on Algorithms Analysis -- 26

2. Recursion and Backtracking --- 35

2.1 Introduction-- 35

2.2 What is Recursion? -- 35

2.3 Why Recursion? -- 35

2.4 Format of a Recursive Function --- 35

2.5 Recursion and Memory (Visualization) --- 36

2.6 Recursion versus Iteration --- 37

2.7 Notes on Recursion -- 37

2.8 Example Algorithms of Recursion --- 37

2.9 Problems on Recursion-- 37

2.10 What is Backtracking?--- 38

2.11 Example Algorithms of Backtracking -- 38
2.12 Problems on Backtracking --- 38
3. Linked Lists --- 41
3.1 What is a Linked List? -- 41
3.2 Linked Lists ADT -- 41
3.3 Why Linked Lists? --- 41
3.4 Arrays Overview --- 42
3.5 Comparison of Linked Lists with Arrays & Dynamic Arrays -------------------------------------- 43
3.6 Singly Linked Lists -- 43
3.7 Doubly Linked Lists --- 48
3.8 Circular Linked Lists -- 53
3.9 A Memory-Efficient Doubly Linked List -- 58
3.10 Unrolled Linked Lists -- 59
3.11 Skip Lists --- 64
3.12 Problems on Linked Lists -- 66
4. Stacks --- 87
4.1 What is a Stack? --- 87
4.2 How Stacks are used? -- 87
4.3 Stack ADT --- 88
4.4 Applications --- 88
4.5 Implementation -- 88
4.6 Comparison of Implementations --- 93
4.7 Problems on Stacks -- 93
5. Queues -- 109
5.1 What is a Queue? -- 109
5.2 How are Queues Used? -- 109
5.3 Queue ADT -- 109
5.4 Exceptions -- 110
5.5 Applications --- 110
5.6 Implementation -- 110
5.7 Problems on Queues --- 115
6. Trees -- 121
6.1 What is a Tree? -- 121
6.2 Glossary -- 121
6.3 Binary Trees -- 122
6.4 Types of Binary Trees --- 123
6.5 Properties of Binary Trees --- 123
6.6 Binary Tree Traversals -- 125
6.7 Generic Trees (N-ary Trees) -- 144
6.8 Threaded Binary Tree Traversals [Stack or Queue less Traversals] -------------------------------- 150
6.9 Expression Trees -- 155
6.10 XOR Trees --- 157

6.11 Binary Search Trees (BSTs) ---------- 158
6.12 Balanced Binary Search Trees ---------- 172
6.13 AVL (Adelson-Velskii and Landis) Trees ---------- 173
6.14 Other Variations in Trees ---------- 188
7. Priority Queues and Heaps ---------- 192
7.1 What is a Priority Queue? ---------- 192
7.2 Priority Queue ADT ---------- 192
7.3 Priority Queue Applications ---------- 193
7.4 Priority Queue Implementations ---------- 193
7.5 Heaps and Binary Heap ---------- 194
7.6 Binary Heaps ---------- 195
7.7 Heapsort ---------- 200
7.8 Problems on Priority Queues [Heaps] ---------- 201
8. Disjoint Sets ADT ---------- 211
8.1 Introduction ---------- 211
8.2 Equivalence Relations and Equivalence Classes ---------- 211
8.3 Disjoint Sets ADT ---------- 212
8.4 Applications ---------- 212
8.5 Tradeoffs in Implementing Disjoint Sets ADT ---------- 212
8.8 Fast UNION implementation (Slow FIND) ---------- 213
8.9 Fast UNION implementations (Quick FIND) ---------- 216
8.10 Summary ---------- 218
8.11 Problems on Disjoint Sets ---------- 218
9. Graph Algorithms ---------- 220
9.1 Introduction ---------- 220
9.2 Glossary ---------- 220
9.3 Applications of Graphs ---------- 223
9.4 Graph Representation ---------- 223
9.5 Graph Traversals ---------- 226
9.6 Topological Sort ---------- 232
9.7 Shortest Path Algorithms ---------- 234
9.8 Minimal Spanning Tree ---------- 238
9.9 Problems on Graph Algorithms ---------- 242
10. Sorting ---------- 260
10.1 What is Sorting? ---------- 260
10.2 Why is Sorting necessary? ---------- 260
10.3 Classification of Sorting Algorithms ---------- 260
10.4 Other Classifications ---------- 261
10.5 Bubble sort ---------- 261
10.6 Selection Sort ---------- 262
10.7 Insertion sort ---------- 263
10.8 Shell sort ---------- 264

10.9 Merge sort ---265
10.10 Heapsort---267
10.11 Quicksort---267
10.12 Tree Sort---269
10.13 Comparison of Sorting Algorithms ---269
10.14 Linear Sorting Algorithms ---270
10.15 Counting Sort---270
10.16 Bucket sort [or Bin Sort] ---270
10.17 Radix sort ---271
10.18 Topological Sort ---271
10.19 External Sorting---271
10.20 Problems on Sorting ---272
11. Searching --- 281
11.1 What is Searching?---281
11.2 Why do we need Searching? ---281
11.3 Types of Searching ---281
11.4 Unordered Linear Search---281
11.5 Sorted/Ordered Linear Search ---282
11.6 Binary Search ---282
11.7 Comparing Basic Searching Algorithms ---283
11.8 Symbol Tables and Hashing ---283
11.9 String Searching Algorithms ---283
11.10 Problems on Searching---283
12. Selection Algorithms [Medians] --- 304
12.1 What are Selection Algorithms? ---304
12.2 Selection by Sorting ---304
12.3 Partition-based Selection Algorithm ---304
12.4 Linear Selection algorithm - Median of Medians algorithm ---304
12.5 Finding the K Smallest Elements in Sorted Order ---305
12.6 Problems on Selection Algorithms ---305
13. Symbol Tables --- 313
13.1 Introduction ---313
13.2 What are Symbol Tables? ---313
13.3 Symbol Table Implementations ---313
13.4 Comparison of Symbol Table Implementations ---314
14. Hashing--- 315
14.1 What is Hashing?---315
14.2 Why Hashing?---315
14.3 HashTable ADT ---315
14.4 Understanding Hashing ---315
14.5 Components of Hashing ---317
14.6 Hash Table ---317

14.7 Hash Function ---317
14.8 Load Factor---317
14.9 Collisions ---318
14.10 Collision Resolution Techniques---318
14.11 Separate Chaining ---318
14.12 Open Addressing---318
14.13 Comparison of Collision Resolution Techniques ---320
14.14 How Hashing Gets O(1) Complexity? ---320
14.15 Hashing Techniques---321
14.16 Problems for which Hash Tables are not suitable ---321
14.17 Bloom Filters ---321
14.18 Problems on Hashing---323

15. String Algorithms -- 333

15.1 Introduction ---333
15.2 String Matching Algorithms ---333
15.3 Brute Force Method ---334
15.4 Robin-Karp String Matching Algorithm ---334
15.5 String Matching with Finite Automata ---335
15.6 KMP Algorithm ---336
15.7 Boyce-Moore Algorithm ---339
15.8 Data Structures for Storing Strings ---339
15.9 Hash Tables for Strings---340
15.10 Binary Search Trees for Strings ---340
15.11 Tries---340
15.12 Ternary Search Trees---342
15.13 Comparing BSTs, Tries and TSTs ---346
15.14 Suffix Trees ---346
15.15 Problems on Strings ---349

16. Algorithms Design Techniques-- 357

16.1 Introduction ---357
16.2 Classification---357
16.3 Classification by Implementation Method ---357
16.4 Classification by Design Method ---358
16.5 Other Classifications---359

17. Greedy Algorithms -- 360

17.1 Introduction ---360
17.2 Greedy strategy ---360
17.3 Elements of Greedy Algorithms ---360
17.4 Does Greedy Always Work? ---360
17.5 Advantages and Disadvantages of Greedy Method---361
17.6 Greedy Applications---361
17.7 Understanding Greedy Technique ---361

17.8 Problems on Greedy Algorithms--363

18. Divide and Conquer Algorithms -- 369

18.1 Introduction --369

18.2 What is Divide and Conquer Strategy?--369

18.3 Does Divide and Conquer Always Work? ---369

18.4 Divide and Conquer Visualization ---369

18.5 Understanding Divide and Conquer --370

18.6 Advantages of Divide and Conquer --370

18.7 Disadvantages of Divide and Conquer --371

18.8 Master Theorem ---371

18.9 Divide and Conquer Applications --371

18.10 Problems on Divide and Conquer --371

19. Dynamic Programming-- 381

19.1 Introduction --381

19.2 What is Dynamic Programming Strategy?---381

19.3 Properties of Dynamic Programming Strategy --381

19.4 Can Dynamic Programming Solve All Problems?--------------------------------------381

19.5 Dynamic Programming Approaches ---381

19.6 Examples of Dynamic Programming Algorithms --382

19.7 Understanding Dynamic Programming --382

19.8 Longest Common Subsequence ---385

19.9 Problems on Dynamic Programming --386

20. Complexity Classes -- 415

20.1 Introduction --415

20.2 Polynomial/Exponential time---415

20.3 What is Decision Problem?---415

20.4 Decision Procedure---416

20.5 What is a Complexity Class? --416

20.6 Types of Complexity Classes ---416

20.7 Reductions---418

20.8 Problems on Complexity Classes---420

21. Miscellaneous Concepts -- 423

21.1 Introduction --423

21.2 Hacks on Bitwise Programming ---423

21.3 Other Programming Questions --427

References --- 433