Data Structures
And
Algorithms
Made Easy In JAVA

Data Structure and Algorithmic Puzzles

By

Narasimha Karumanchi

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 thank them for their efforts in improving the end result. Before we do so, however, I should mention that I have done my best to correct the mistakes that the reviewers have pointed out and to accurately describe the protocols and mechanisms. I alone am responsible for any remaining errors.

First and foremost, 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.
- *Kishore Kumar Jinka*, *IIT Bombay*
- *A.Vamshi Krishna*, IIT Kanpur, Mentor Graphics Inc.
- *Hirak Chatterjee*, Yahoo Inc.
- *Kondrakunta Murali Krishna,* B-Tech., Technical Lead, HCL
- *Chaganti Siva Rama Krishna Prasad*, Founder, StockMonks Pvt. Ltd.
- *Naveen Valsakumar*, Co-Founder, NotionPress Pvt. Ltd.
- *Ramanaiah*, Lecturer, Nagarjuna Institute of Technology and Sciences, MLG

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*] and *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 give 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 of *CareerMonk.com*

Other Titles by Narasimha Karumanchi

- IT Interview Questions
- Elements of Computer Networking
- Data Structures and Algorithmic Thinking With Python
- Data Structures and Algorithms Made Easy (C/C++)
- Coding Interview Questions
- Data Structures and Algorithms for GATE
- Peeling Design Patterns

Table of Contents

1. Introduction ---15

1.1 Variables--- 15

1.2 Data types -- 15

1.3 Data Structure--- 16

1.4 Abstract Data Types (ADTs) --- 16

1.5 What is an Algorithm?--- 16

1.6 Why Analysis of Algorithms? -- 17

1.7 Goal of Analysis of Algorithms -- 17

1.8 What is Running Time Analysis? --- 17

1.9 How to Compare Algorithms? -- 17

1.10 What is Rate of Growth?-- 17

1.11 Commonly used Rate of Growths -- 18

1.12 Types of Analysis -- 19

1.13 Asymptotic Notation--- 19

1.14 Big-O Notation --- 19

1.15 Omega-Ω Notation -- 21

1.16 Theta-Θ Notation -- 21

1.17 Important Notes-- 22

1.18 Why is it called Asymptotic Analysis? -- 22

1.19 Guidelines for Asymptotic Analysis --- 23

1.20 Properties of Notations-- 24

1.21 Commonly used Logarithms and Summations --- 24

1.22 Master Theorem for Divide and Conquer -- 24

1.23 Problems on Divide and Conquer Master Theorem--- 25

1.24 Master Theorem for Subtract and Conquer Recurrences ------------------------------------ 26

1.25 Variant of Subtraction and Conquer master theorem -- 26

1.26 Method of Guessing and Confirm --- 26

1.27 Amortized Analysis --- 27

1.28 Problems on Algorithms Analysis --- 28

2. Recursion and Backtracking---38

2.1 Introduction -- 38

2.2 What is Recursion? --- 38

2.3 Why Recursion? --- 38

2.4 Format of a Recursive Function -- 38

2.5 Recursion and Memory (Visualization)-- 39

2.6 Recursion versus Iteration-- 39

2.7 Notes on Recursion -- 40

2.8 Example Algorithms of Recursion -- 40
2.9 Problems on Recursion -- 40
2.10 What is Backtracking? -- 41
2.11 Example Algorithms of Backtracking -- 41
2.12 Problems on Backtracking -- 41
3. Linked Lists --43
3.1 What is a Linked List? -- 43
3.2 Linked Lists ADT -- 43
3.3 Why Linked Lists? -- 43
3.4 Arrays Overview -- 43
3.5 Comparison of Linked Lists with Arrays & Dynamic Arrays -- 45
3.6 Singly Linked Lists -- 45
3.7 Doubly Linked Lists -- 52
3.8 Circular Linked Lists -- 59
3.9 A Memory-Efficient Doubly Linked List -- 65
3.10 Unrolled Linked Lists -- 66
3.11 Skip Lists -- 74
3.11 Problems on Linked Lists -- 77
4. Stacks --99
4.1 What is a Stack? -- 99
4.2 How Stacks are used? -- 99
4.3 Stack ADT -- 99
4.4 Exceptions --100
4.5 Applications --100
4.6 Implementation --100
4.7 Comparison of Implementations --105
4.8 Problems on Stacks --106
5. Queues -- 125
5.1 What is a Queue? --125
5.2 How are Queues Used? --125
5.3 Queue ADT --125
5.4 Exceptions --126
5.5 Applications --126
5.6 Implementation --126
5.7 Problems on Queues --132
6. Trees -- 136
6.1 What is a Tree? --136
6.2 Glossary --136

6.3 Binary Trees--137

6.4 Binary Tree Traversals ---140

6.5 Generic Trees (N-ary Trees)---162

6.6 Threaded [Stack or Queue less] Binary Tree Traversals ---168

6.7 Expression Trees ---173

6.8 XOR Trees ---175

6.9 Binary Search Trees (BSTs)--176

6.10 Balanced Binary Search Trees--188

6.11 AVL (Adelson-Velskii and Landis) Trees--189

6.12 Other Variations in Trees---201

7. Priority Queues and Heaps--- 205

7.1 What is a Priority Queue?---205

7.2 Priority Queue ADT ---205

7.3 Priority Queue Applications --205

7.4 Priority Queue Implementations ---206

7.5 Heaps and Binary Heap --206

7.6 Binary Heaps ---207

7.7 Problems on Priority Queues [Heaps] ---213

8. Disjoint Sets ADT --- 223

8.1 Introduction---223

8.2 Equivalence Relations and Equivalence Classes --223

8.3 Disjoint Sets ADT --224

8.4 Applications---224

8.5 Tradeoffs in Implementing Disjoint Sets ADT---224

8.6 Fast UNION implementation (Slow FIND) --225

8.7 Fast UNION implementations (Quick FIND)--227

8.8 Path Compression---229

8.9 Summary ---230

8.10 Problems on Disjoint Sets --230

9. Graph Algorithms--- 231

9.1 Introduction---231

9.2 Glossary ---231

9.3 Applications of Graphs ---234

9.4 Graph Representation ---234

9.5 Graph Traversals--236

9.6 Topological Sort ---244

9.7 Shortest Path Algorithms --245

9.8 Minimal Spanning Tree --250

9.9 Problems on Graph Algorithms ---254

10. Sorting -- 271

10.1 What is Sorting? --271

10.2 Why is Sorting Necessary? --271

10.3 Classification --271

10.4 Other Classifications --272

10.5 Bubble sort --272

10.6 Selection Sort---273

10.7 Insertion sort ---273

10.8 Shell sort --275

10.9 Merge sort---276

10.10 Heapsort ---277

10.11 Quicksort --278

10.12 Tree Sort---280

10.13 Comparison of Sorting Algorithms ---280

10.14 Linear Sorting Algorithms ---280

10.15 Counting Sort ---281

10.16 Bucket sort [or Bin Sort] --281

10.17 Radix sort ---281

10.18 Topological Sort --282

10.19 External Sorting --282

10.20 Problems on Sorting ---283

11. Searching -- 291

11.1 What is Searching? --291

11.2 Why do we need Searching? --291

11.3 Types of Searching---291

11.4 Symbol Tables and Hashing --293

11.5 String Searching Algorithms --293

11.6 Problems on Searching --293

12. Selection Algorithms [Medians]-- 314

12.1 What are Selection Algorithms? ---314

12.2 Selection by Sorting ---314

12.3 Partition-based Selection Algorithm --314

12.4 Linear Selection algorithm - Median of Medians algorithm ---314

12.5 Finding the K Smallest Elements in Sorted Order--314

12.6 Problems on Selection Algorithms --315

13. Symbol Tables --- 323

13.1 Introduction --323

13.2 What are Symbol Tables? ---323
13.3 Symbol Table Implementations---323
13.4 Comparison of Symbol Table Implementations ---324
14. Hashing--- 325
14.1 What is Hashing?---325
14.2 Why Hashing?---325
14.3 HashTable ADT ---325
14.4 Understanding Hashing ---325
14.5 Components of Hashing ---326
14.6 Hash Table---326
14.7 Hash Function ---327
14.8 Load Factor ---327
14.9 Collisions---327
14.10 Collision Resolution Techniques---327
14.11 Separate Chaining ---328
14.12 Open Addressing ---328
14.13 Comparison of Collision Resolution Techniques ---329
14.14 How Hashing Gets O(1) Complexity?---330
14.15 Hashing Techniques ---330
14.16 Problems for which Hash Tables are not Suitable ---330
14.17 Bloom Filters---330
14.18 Problems on Hashing---332
15. String Algorithms --- 340
5.1 Introduction---340
15.2 String Matching Algorithms---340
15.3 Brute Force Method ---340
15.4 Robin-Karp String Matching Algorithm ---341
15.5 String Matching with Finite Automata ---342
15.6 KMP Algorithm ---343
15.7 Boyce-Moore Algorithm ---345
15.8 Data Structures for Storing Strings ---346
15.9 Hash Tables for Strings---346
15.10 Binary Search Trees for Strings---346
15.11 Tries ---346
15.12 Ternary Search Trees---348
15.13 Comparing BSTs, Tries and TSTs ---352
15.14 Suffix Trees ---352
15.15 Problems on Strings ---355
16. Algorithms Design Techniques --- 365

16.1 Introduction --365
16.2 Classification --365
16.3 Classification by Implementation Method ---365
16.4 Classification by Design Method ---366
16.5 Other Classifications --366
17. Greedy Algorithms --- 368
17.1 Introduction --368
17.2 Greedy strategy --368
17.3 Elements of Greedy Algorithms--368
17.4 Does Greedy Always Work? ---368
17.5 Advantages and Disadvantages of Greedy Method --368
17.6 Greedy Applications --369
17.7 Understanding Greedy Technique ---369
17.8 Problems on Greedy Algorithms--372
18. Divide and Conquer Algorithms -- 378
18.1 Introduction --378
18.2 What is Divide and Conquer Strategy?--378
18.3 Does Divide and Conquer Always Work? --378
18.4 Divide and Conquer Visualization ---378
18.5 Understanding Divide and Conquer --379
18.6 Master Theorem--380
18.7 Divide and Conquer Applications --380
18.8 Problems on Divide and Conquer --380
19. Dynamic Programming -- 391
19.1 Introduction --391
19.2 What is Dynamic Programming Strategy? ---391
19.3 Properties of Dynamic Programming Strategy --391
19.4 Can Dynamic Programming Solve All Problems? --391
19.5 Dynamic Programming Approaches ---391
19.6 Examples of Dynamic Programming Algorithms--392
19.7 Understanding Dynamic Programming --392
19.8 Problems on Dynamic Programming --396
20. Complexity Classes --- 421
20.1 Introduction --421
20.2 Polynomial/Exponential time--421
20.3 What is Decision Problem?--421
20.4 Decision Procedure --422
20.5 What is a Complexity Class?--422

20.6 Types of Complexity Classes ---422

20.7 Reductions --424

20.8 Problems on Complexity Classes --426

21. Miscellaneous Concepts -- 428

21.1 Introduction ---428

21.2 Hacks on Bitwise Programming --428

21.3 Other Programming Questions ---432