

# **Data Structures And Algorithms Made Easy**

**-To All My Readers**

**By  
Narasimha Karumanchi**

 **Concepts**    **Problems**    **Interview Questions**

Copyright© 2016 by *CareerMonk.com*

All rights reserved.

Designed by *Narasimha Karumanchi*

Copyright© 2016 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 your persistent efforts and traditional values 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 all of the people 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
- *Cathy Reed*, BA, MA, Copy Editor
- *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 the *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 Vignanjyothi [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 typically do not read the Preface of a book. But I strongly recommend that you read this particular Preface.

It is not the main objective of this book to present you with the theorems and proofs on *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 the possible solutions. You will find this book useful for interview preparation, competitive exams preparation, and campus interview preparations.

As a *job seeker*, if you read the complete book, I am sure you will be able to challenge the interviewers. If you read it as an *instructor*, it will help you to deliver lectures with an approach that is easy to follow, 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 also useful for *Engineering degree students* and *Masters degree students* during their academic preparations. In all the chapters you will see that there is more emphasis on problems and their analysis rather than on theory. In each chapter, you will first read about the basic required theory, which is then followed by a section on problem sets. In total, there are approximately 700 algorithmic problems, all with solutions.

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

In all the chapters you will see more emphasis on problems and analysis rather than on theory. In each chapter, you will first see the basic required theory followed by various problems.








For many problems, *multiple* solutions are provided with different levels of complexity. We start with the *brute force* solution and slowly move toward the *best solution* possible for that problem. For each problem, we endeavor to understand how much time the algorithm takes and how much memory the algorithm uses.

It is recommended that the reader does at least one *complete* reading of this book to gain a full understanding of all the topics that are covered. Then, in subsequent readings you can skip directly to any chapter to refer to a specific topic. Even though many readings have been done for the purpose of correcting errors, there could still be some minor typos in the book. If any are found, they will be updated at [www.CareerMonk.com](http://www.CareerMonk.com). You can monitor this site for any corrections and also for new problems and solutions. Also, please provide your valuable suggestions at: [Info@CareerMonk.com](mailto:Info@CareerMonk.com).

I wish you all the best and I am confident that you will find this book useful.

—Narasimha Karumanchi  
M-Tech, IIT Bombay  
Founder, [CareerMonk.com](http://CareerMonk.com)

## Other Books 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 the Analysis of Algorithms? -----	15
1.7 Goal of the 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 Rates of Growth -----	16
1.12 Types of Analysis -----	17
1.13 Asymptotic Notation -----	17
1.14 Big-O Notation -----	17
1.15 Omega- $\Omega$ Notation -----	19
1.16 Theta- $\Theta$ 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 Divide and Conquer Master Theorem: Problems & Solutions -----	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 Confirming -----	24
1.26 Amortized Analysis -----	25
1.27 Algorithms Analysis: Problems & Solutions -----	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 Recursion: Problems & Solutions -----	37
2.10 What is Backtracking? -----	38

2.11 Example Algorithms of Backtracking	38
2.12 Backtracking: Problems & Solutions	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 Linked Lists: Problems & Solutions	66
4. Stacks	86
4.1 What is a Stack?	86
4.2 How Stacks are used	86
4.3 Stack ADT	87
4.4 Applications	87
4.5 Implementation	87
4.6 Comparison of Implementations	92
4.7 Stacks: Problems & Solutions	92
5. Queues	108
5.1 What is a Queue?	108
5.2 How are Queues Used?	108
5.3 Queue ADT	108
5.4 Exceptions	109
5.5 Applications	109
5.6 Implementation	109
5.7 Queues: Problems & Solutions	114
6. Trees	119
6.1 What is a Tree?	119
6.2 Glossary	119
6.3 Binary Trees	120
6.4 Types of Binary Trees	121
6.5 Properties of Binary Trees	121
6.6 Binary Tree Traversals	123
6.7 Generic Trees (N-ary Trees)	142
6.8 Threaded Binary Tree Traversals (Stack or Queue-less Traversals)	148
6.9 Expression Trees	153
6.10 XOR Trees	155



6.11 Binary Search Trees (BSTs)	156
6.12 Balanced Binary Search Trees	170
6.13 AVL (Adelson-Velskii and Landis) Trees	170
6.14 Other Variations on Trees	186
7. Priority Queues and Heaps	190
7.1 What is a Priority Queue?	190
7.2 Priority Queue ADT	190
7.3 Priority Queue Applications	191
7.4 Priority Queue Implementations	191
7.5 Heaps and Binary Heaps	192
7.6 Binary Heaps	193
7.7 Heapsort	198
7.8 Priority Queues [Heaps]: Problems & Solutions	199
8. Disjoint Sets ADT	210
8.1 Introduction	210
8.2 Equivalence Relations and Equivalence Classes	210
8.3 Disjoint Sets ADT	211
8.4 Applications	211
8.5 Tradeoffs in Implementing Disjoint Sets ADT	211
8.8 Fast UNION Implementation (Slow FIND)	212
8.9 Fast UNION Implementations (Quick FIND)	215
8.10 Summary	217
8.11 Disjoint Sets: Problems & Solutions	217
9. Graph Algorithms	219
9.1 Introduction	219
9.2 Glossary	219
9.3 Applications of Graphs	222
9.4 Graph Representation	222
9.5 Graph Traversals	225
9.6 Topological Sort	231
9.7 Shortest Path Algorithms	233
9.8 Minimal Spanning Tree	238
9.9 Graph Algorithms: Problems & Solutions	241
10. Sorting	259
10.1 What is Sorting?	259
10.2 Why is Sorting Necessary?	259
10.3 Classification of Sorting Algorithms	259
10.4 Other Classifications	260
10.5 Bubble Sort	260
10.6 Selection Sort	261
10.7 Insertion Sort	262
10.8 Shell Sort	263

10.9 Merge Sort	264
10.10 Heap Sort	266
10.11 Quick Sort	266
10.12 Tree Sort	268
10.13 Comparison of Sorting Algorithms	268
10.14 Linear Sorting Algorithms	269
10.15 Counting Sort	269
10.16 Bucket Sort (or Bin Sort)	269
10.17 Radix Sort	270
10.18 Topological Sort	270
10.19 External Sorting	271
10.20 Sorting: Problems & Solutions	271
11. Searching	280
11.1 What is Searching?	280
11.2 Why do we need Searching?	280
11.3 Types of Searching	280
11.4 Unordered Linear Search	280
11.5 Sorted/Ordered Linear Search	281
11.6 Binary Search	281
11.7 Comparing Basic Searching Algorithms	282
11.8 Symbol Tables and Hashing	282
11.9 String Searching Algorithms	282
11.10 Searching: Problems & Solutions	282
12. Selection Algorithms [Medians]	302
12.1 What are Selection Algorithms?	302
12.2 Selection by Sorting	302
12.3 Partition-based Selection Algorithm	302
12.4 Linear Selection Algorithm - Median of Medians Algorithm	302
12.5 Finding the K Smallest Elements in Sorted Order	302
12.6 Selection Algorithms: Problems & Solutions	303
13. Symbol Tables	310
13.1 Introduction	310
13.2 What are Symbol Tables?	310
13.3 Symbol Table Implementations	310
13.4 Comparison Table of Symbols for Implementations	311
14. Hashing	312
14.1 What is Hashing?	312
14.2 Why Hashing?	312
14.3 HashTable ADT	312
14.4 Understanding Hashing	312
14.5 Components of Hashing	313
14.6 Hash Table	313

14.7 Hash Function	314
14.8 Load Factor	315
14.9 Collisions	315
14.10 Collision Resolution Techniques	315
14.11 Separate Chaining	315
14.12 Open Addressing	315
14.13 Comparison of Collision Resolution Techniques	317
14.14 How Hashing Gets $O(1)$ Complexity?	317
14.15 Hashing Techniques	318
14.16 Problems for which Hash Tables are not suitable	318
14.17 Bloom Filters	318
14.18 Hashing: Problems & Solutions	319
15. String Algorithms	329
15.1 Introduction	329
15.2 String Matching Algorithms	329
15.3 Brute Force Method	329
15.4 Robin-Karp String Matching Algorithm	330
15.5 String Matching with Finite Automata	331
15.6 KMP Algorithm	332
15.7 Boyce-Moore Algorithm	335
15.8 Data Structures for Storing Strings	335
15.9 Hash Tables for Strings	335
15.10 Binary Search Trees for Strings	335
15.11 Tries	336
15.12 Ternary Search Trees	337
15.13 Comparing BSTs, Tries and TSTs	341
15.14 Suffix Trees	341
15.15 Strings: Problems & Solutions	344
16. Algorithms Design Techniques	352
16.1 Introduction	352
16.2 Classification	352
16.3 Classification by Implementation Method	352
16.4 Classification by Design Method	353
16.5 Other Classifications	354
17. Greedy Algorithms	355
17.1 Introduction	355
17.2 Greedy Strategy	355
17.3 Elements of Greedy Algorithms	355
17.4 Does Greedy Always Work?	355
17.5 Advantages and Disadvantages of Greedy Method	355
17.6 Greedy Applications	356
17.7 Understanding Greedy Technique	356

17.8 Greedy Algorithms: Problems & Solutions-----	358
18. Divide and Conquer Algorithms -----	364
18.1 Introduction -----	364
18.2 What is the Divide and Conquer Strategy?-----	364
18.3 Does Divide and Conquer Always Work? -----	364
18.4 Divide and Conquer Visualization -----	364
18.5 Understanding Divide and Conquer -----	365
18.6 Advantages of Divide and Conquer -----	365
18.7 Disadvantages of Divide and Conquer -----	366
18.8 Master Theorem -----	366
18.9 Divide and Conquer Applications -----	366
18.10 Divide and Conquer: Problems & Solutions -----	366
19. Dynamic Programming-----	376
19.1 Introduction -----	376
19.2 What is Dynamic Programming Strategy?-----	376
19.3 Properties of Dynamic Programming Strategy -----	376
19.4 Can Dynamic Programming Solve All Problems?-----	376
19.5 Dynamic Programming Approaches -----	376
19.6 Examples of Dynamic Programming Algorithms -----	377
19.7 Understanding Dynamic Programming -----	377
19.8 Longest Common Subsequence -----	380
19.9 Dynamic Programming: Problems & Solutions -----	381
20. Complexity Classes -----	409
20.1 Introduction -----	409
20.2 Polynomial/Exponential Time -----	409
20.3 What is a Decision Problem? -----	409
20.4 Decision Procedure-----	410
20.5 What is a Complexity Class? -----	410
20.6 Types of Complexity Classes-----	410
20.7 Reductions-----	412
20.8 Complexity Classes: Problems & Solutions-----	414
21. Miscellaneous Concepts -----	416
21.1 Introduction -----	416
21.2 Hacks on Bitwise Programming-----	416
21.3 Other Programming Questions -----	420
References-----	426