

DATA STRUCTURES AND ALGORITHMS MADE EASY

Data Structures and Algorithmic Puzzles

5TH
EDITION



Narasimha Karumanchi, M.Tech, IIT Bombay
Founder, CareerMonk.com

Data Structures And Algorithms Made Easy

-To All My Readers

**By
Narasimha Karumanchi**

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

Copyright© 2017 by CareerMonk.com

All rights reserved.

Designed by *Narasimha Karumanchi*

Copyright© 2017 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.
- *Cathy Reed*, BA, MA, Copy Editor

–*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. 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.

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

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 Structures and Algorithmic Thinking with Python

Table of Contents

1. Introduction

- 1.1 Variables
- 1.2 Data Types
- 1.3 Data Structures
- 1.4 Abstract Data Types (ADTs)
- 1.5 What is an Algorithm?
- 1.6 Why the Analysis of Algorithms?
- 1.7 Goal of the Analysis of Algorithms
- 1.8 What is Running Time Analysis?
- 1.9 How to Compare Algorithms
- 1.10 What is Rate of Growth?
- 1.11 Commonly Used Rates of Growth
- 1.12 Types of Analysis
- 1.13 Asymptotic Notation
- 1.14 Big-O Notation [Upper Bounding Function]
- 1.15 Omega- Ω Notation [Lower Bounding Function]
- 1.16 Theta- Θ Notation [Order Function]
- 1.17 Important Notes
- 1.18 Why is it called Asymptotic Analysis?
- 1.19 Guidelines for Asymptotic Analysis
- 1.20 Simplifying properties of asymptotic notations
- 1.21 Commonly used Logarithms and Summations
- 1.22 Master Theorem for Divide and Conquer Recurrences
- 1.23 Divide and Conquer Master Theorem: Problems & Solutions
- 1.24 Master Theorem for Subtract and Conquer Recurrences
- 1.25 Variant of Subtraction and Conquer Master Theorem
- 1.26 Method of Guessing and Confirming

1.27 Amortized Analysis

1.28 Algorithms Analysis: Problems & Solutions

2. Recursion and Backtracking

2.1 Introduction

2.2 What is Recursion?

2.3 Why Recursion?

2.4 Format of a Recursive Function

2.5 Recursion and Memory (Visualization)

2.6 Recursion versus Iteration

2.7 Notes on Recursion

2.8 Example Algorithms of Recursion

2.9 Recursion: Problems & Solutions

2.10 What is Backtracking?

2.11 Example Algorithms of Backtracking

2.12 Backtracking: Problems & Solutions

3. Linked Lists

3.1 What is a Linked List?

3.2 Linked Lists ADT

3.3 Why Linked Lists?

3.4 Arrays Overview

3.5 Comparison of Linked Lists with Arrays & Dynamic Arrays

3.6 Singly Linked Lists

3.7 Doubly Linked Lists

3.8 Circular Linked Lists

3.9 A Memory-efficient Doubly Linked List

3.10 Unrolled Linked Lists

3.11 Skip Lists

3.12 Linked Lists: Problems & Solutions

4. Stacks

4.1 What is a Stack?

4.2 How Stacks are used

4.3 Stack ADT

- 4.4 Applications
- 4.5 Implementation
- 4.6 Comparison of Implementations
- 4.7 Stacks: Problems & Solutions

5. Queues

- 5.1 What is a Queue?
- 5.2 How are Queues Used?
- 5.3 Queue ADT
- 5.4 Exceptions
- 5.5 Applications
- 5.6 Implementation
- 5.7 Queues: Problems & Solutions

6. Trees

- 6.1 What is a Tree?
- 6.2 Glossary
- 6.3 Binary Trees
- 6.4 Types of Binary Trees
- 6.5 Properties of Binary Trees
- 6.6 Binary Tree Traversals
- 6.7 Generic Trees (N -ary Trees)
- 6.8 Threaded Binary Tree Traversals (Stack or Queue-less Traversals)
- 6.9 Expression Trees
- 6.10 XOR Trees
- 6.11 Binary Search Trees (BSTs)
- 6.12 Balanced Binary Search Trees
- 6.13 AVL (Adelson-Velskii and Landis) Trees
- 6.14 Other Variations on Trees

7. Priority Queues and Heaps

- 7.1 What is a Priority Queue?
- 7.2 Priority Queue ADT
- 7.3 Priority Queue Applications
- 7.4 Priority Queue Implementations