

Coding Interview Questions

**By
Narasimha Karumanchi**

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

Copyright ©2017 by *CareerMonk.com*

All rights reserved.

Designed by *Narasimha Karumanchi*

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 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.
- *A.Vamshi Krishna*, IIT Kanpur, Mentor Graphics Inc.
- *Ramanaiah*, Lecturer, Nagarjuna Institute of Technology and Sciences, MLG

-*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.

This book assumes you have some basic knowledge about computer science. 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.

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.

In most 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 the *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 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 get full understanding of all the topics that are covered. In subsequent readings you can skip directly to any chapter to refer to a specific topic. Even though, enough readings have been done for the purpose of correcting 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 that you frequently 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

Table of Contents

1. Programming Basics	13
1.1 Variables	13
1.2 Data types	13
1.3 Data Structure	13
1.4 Abstract Data Types (ADTs)	14
1.5 Memory and Variables	14
1.6 Pointers	15
1.7 Techniques of Parameter Passing	17
1.8 Binding	20
1.9 Scope	20
1.10 Storage Classes	21
1.11 Storage Organization	25
1.12 Programming Techniques	27
1.13 Basic Concepts of OOPS	28
2. Scripting Languages	71
2.1 Interpreter versus Compiler	71
2.2 What Are Scripting Languages?	72
2.3 Shell Scripting	72
2.4 PERL [Practical Extraction and Report Language]	78
2.5 Python	95
3. Design Interview Questions	100
3.1 Glossary	100
3.2 Tips	101
3.3 Sample Design Questions For Practice	142
4. Operating System Concepts	143
4.1 Glossary	143
4.2 Questions on Operating System Concepts	144
5. Computer Networking Basics	148
5.1 Introduction	148
5.2 LAN vs. WAN	148
5.2 Segmentation and Multiplexing	149
5.3 End Devices	149
5.4 Intermediary Devices	149
5.5 Hub, Switch, and Router Defined	149
5.6 Medium	150
5.7 Peer-to-peer and Client/server networks	150
5.8 How does Internet works?	151
5.9 Difference between OSI and TCP/IP models	153
5.10 Client/Server Computing and the Internet	154
5.11 ARP and RARP	154
5.12 Subnetting	155
5.13 How Routing Works?	155
5.14 Unicast, Broadcast and Multicast	156
5.15 How traceroute (or tracert) and ping works?	156
5.16 What is QoS?	157
6. Database Concepts	158
6.1 Glossary	158
6.2 Questions on Database Concepts	159
7. Brain Teasers	163
7.1 Questions on Brain Teasers	163
8. Algorithms Introduction	165
8.1 What is an Algorithm?	165
8.2 Why the Analysis of Algorithms?	165
8.3 Goal of the Analysis of Algorithms	165
8.4 What is Running Time Analysis?	165
8.5 How to Compare Algorithms	165
8.6 What is Rate of Growth?	166
8.7 Commonly Used Rates of Growth	166

8.8 Types of Analysis	166
8.9 Asymptotic Notation.....	167
8.10 Big-O Notation.....	167
8.15 Omega- Ω Notation	169
8.16 Theta- Θ Notation	169
8.17 Why is it called Asymptotic Analysis?	170
8.18 Guidelines for Asymptotic Analysis	170
8.19 Simplifying properties of asymptotic notations	171
8.20 Commonly used Logarithms and Summations.....	171
8.21 Master Theorem for Divide and Conquer Recurrences.....	172
8.22 Problems on Divide and Conquer Master Theorem.....	172
8.23 Master Theorem for Subtract and Conquer Recurrences.....	173
8.24 Variant of Subtraction and Conquer Master Theorem	173
8.25 Method of Guessing and Confirm	173
8.26 Amortized Analysis.....	174
8.27 Problems with Solutions on Algorithms Analysis	175
9. Recursion and Backtracking.....	184
9.1 Introduction	184
9.2 What is Recursion?.....	184
9.3 Why Recursion?.....	184
9.4 Format of a Recursive Function	184
9.5 Recursion and Memory (Visualization)	185
9.6 Recursion versus Iteration	185
9.7 Notes on Recursion.....	185
9.8 Example Algorithms of Recursion.....	186
9.9 Problems with Solutions on Recursion.....	186
9.10 What is Backtracking?	187
9.11 Example Algorithms of Backtracking	187
9.12 Problems with Solutions on Backtracking	187
10. Linked Lists	190
10.1 What is a Linked List?.....	190
10.2 Linked Lists ADT.....	190
10.3 Why Linked Lists?	190
10.4 Arrays Overview	190
10.5 Comparison of Linked Lists with Arrays & Dynamic Arrays	191
10.6 Singly Linked Lists.....	191
10.7 Doubly Linked Lists	196
10.8 Circular Linked Lists	200
10.9 Memory-efficient Doubly Linked List.....	205
10.10 Unrolled Linked Lists	206
10.11 Skip Lists	210
10.12 Problems with Solutions on Linked Lists	214
11. Stacks.....	231
11.1 What is a Stack?	231
11.2 How Stacks are used?.....	231
11.3 Stack ADT.....	231
11.4 Applications.....	232
11.5 Implementation.....	232
11.6 Comparison of Implementations.....	236
11.7 Problems with Solutions on Stacks	236
12. Queues	252
12.1 What is a Queue?	252
12.2 How are Queues Used?.....	252
12.3 Queue ADT.....	252
12.4 Exceptions	253
12.5 Applications.....	253
12.6 Implementation.....	253
12.7 Problems with Solutions on Queues	257
13. Trees.....	263
13.1 What is a Tree?	263
13.2 Glossary	263
13.3 Binary Trees	264
13.4 Types of Binary Trees.....	264
13.5 Properties of Binary Trees	265

13.6 Binary Tree Traversals	266
13.7 Generic Trees (N-ary Trees)	283
13.8 Threaded Binary Tree [Stack/Queue less] Traversals	288
13.9 Expression Trees	293
13.10 XOR Trees	295
13.11 Binary Search Trees (BSTs)	296
13.12 Balanced Binary Search Trees	308
13.13 AVL (Adelson-Velskii and Landis) Trees	309
13.14 Other Variations in Trees	322
14. Priority Queue and Heaps	326
14.2 What is a Priority Queue?	326
14.2 Priority Queue ADT	326
14.3 Priority Queue Applications	326
14.4 Priority Queue Implementations	326
14.5 Heaps and Binary Heaps	327
14.6 Binary Heaps	328
14.7 Heapsort	333
14.8 Priority Queues [Heaps]: Problems & Solutions	334
15. Graph Algorithms	343
15.1 Introduction	343
15.2 Glossary	343
15.3 Applications of Graphs	345
15.4 Graph Representation	346
15.5 Graph Traversals	348
15.6 Topological Sort	354
15.7 Shortest Path Algorithms	355
15.8 Minimal Spanning Tree	360
15.9 Problems with Solutions on Graph Algorithms	363
16. Sorting	368
16.1 What is Sorting?	368
16.2 Why is Sorting Necessary?	368
16.3 Classification of Sorting Algorithms	368
16.4 Other Classifications	368
16.5 Bubble Sort	369
16.6 Selection Sort	369
16.7 Insertion Sort	370
16.8 Shell Sort	372
16.9 Merge Sort	373
16.10 Heap Sort	374
16.11 Quick Sort	374
16.12 Tree Sort	376
16.13 Comparison of Sorting Algorithms	376
16.14 Linear Sorting Algorithms	377
16.15 Counting Sort	377
16.16 Bucket Sort (or Bin Sort)	377
16.17 Radix Sort	378
16.18 Topological Sort	378
16.19 External Sorting	378
16.20 Sorting: Problems & Solutions	379
17. Searching	387
17.1 What is Searching?	387
17.2 Why do we need Searching?	387
17.3 Types of Searching	387
17.4 Unordered Linear Search	387
17.5 Sorted/Ordered Linear Search	387
17.6 Binary Search	388
17.7 Interpolation Search	388
17.8 Comparing Basic Searching Algorithms	389
17.9 Symbol Tables and Hashing	390
17.10 String Searching Algorithms	390
17.11 Searching: Problems & Solutions	390
18. Selection Algorithms [Medians]	409
18.1 What are Selection Algorithms?	409
18.2 Selection by Sorting	409

18.3 Partition-based Selection Algorithm-----	409
18.4 Linear Selection algorithm - Median of Medians algorithm -----	409
18.5 Finding the K Smallest Elements in Sorted Order -----	409
18.6 Problems with Solutions on Selection Algorithms -----	409
19. Symbol Tables.....	417
19.1 Introduction-----	417
19.2 What are Symbol Tables?-----	417
19.3 Symbol Table Implementations -----	417
19.4 Comparison Table of Symbols for Implementations -----	418
20. Hashing	419
20.1 What is Hashing? -----	419
20.2 Why Hashing? -----	419
20.3 HashTable ADT-----	419
20.4 Understanding Hashing-----	419
20.5 Components of Hashing-----	420
20.6 Hash Table -----	420
20.7 Hash Function-----	420
20.8 Load Factor -----	421
20.9 Collisions -----	421
20.10 Collision Resolution Techniques -----	421
20.11 Separate Chaining-----	422
20.12 Open Addressing -----	422
20.13 Comparison of Collision Resolution Techniques-----	423
20.14 How Hashing Gets $O(1)$ Complexity? -----	424
20.15 Hashing Techniques -----	424
20.16 Problems for which Hash Tables are not suitable-----	424
20.17 Bloom Filters -----	424
20.18 Hashing: Problems & Solutions -----	426
21. String Algorithms.....	434
21.1 Introduction-----	434
21.2 String Matching Algorithms -----	434
21.3 Brute Force Method-----	434
21.4 Rabin-Karp String Matching Algorithm-----	435
21.5 String Matching with Finite Automata-----	435
21.6 KMP Algorithm -----	436
21.7 Boyer-Moore Algorithm-----	439
21.8 Data Structures for Storing Strings -----	439
21.9 Hash Tables for Strings -----	439
21.10 Binary Search Trees for Strings -----	439
21.11 Tries -----	440
21.12 Ternary Search Trees -----	441
21.13 Comparing BSTs, Tries and TSTs -----	445
21.14 Suffix Trees -----	445
21.15 String Algorithms: Problems & Solutions-----	447
22. Algorithms Design Techniques	455
22.1 Introduction-----	455
22.2 Classification -----	455
22.3 Classification by Implementation Method-----	455
22.4 Classification by Design Method-----	455
22.5 Other Classifications -----	456
23. Greedy Algorithms	457
23.1 Introduction-----	457
23.2 Greedy Strategy -----	457
23.3 Elements of Greedy Algorithms -----	457
23.4 Does Greedy Always Work?-----	457
23.5 Advantages and Disadvantages of Greedy Method -----	457
23.6 Greedy Applications -----	457
23.7 Understanding Greedy Technique -----	458
23.8 Greedy Algorithms: Problems & Solutions-----	460
24. Divide and Conquer Algorithms	465
24.1 Introduction-----	465
24.2 What is the Divide and Conquer Strategy? -----	465
24.3 Does Divide and Conquer Always Work?-----	465
24.4 Divide and Conquer Visualization -----	465

24.5 Understanding Divide and Conquer-----	466
24.6 Advantages of Divide and Conquer-----	466
24.7 Disadvantages of Divide and Conquer-----	466
24.8 Master Theorem -----	466
24.9 Divide and Conquer Applications-----	467
24.10 Divide and Conquer: Problems & Solutions -----	467
25. Dynamic Programming	476
25.1 Introduction-----	476
25.2 What is Dynamic Programming Strategy? -----	476
25.3 Properties of Dynamic Programming Strategy-----	476
25.4 Can Dynamic Programming Solve All Problems? -----	476
25.5 Dynamic Programming Approaches-----	476
25.6 Examples of Dynamic Programming Algorithms -----	477
25.7 Understanding Dynamic Programming-----	477
25.8 Longest Common Subsequence-----	479
25.9 Dynamic Programming: Problems & Solutions -----	481
26. Miscellaneous Concepts	506
26.1 Introduction-----	506
26.2 Hacks on Bitwise Programming-----	506
26.3 Other Programming Questions-----	509
27. Non-Technical Help	515
27.1 Tips-----	515
27.2 Non-Technical Questions -----	516

Coding Interview Questions

Other titles by *Narasimha Karumanchi*

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