

SALIENT FEATURES OF BOOK

- ☞ All code written in C
- ☞ Enumeration of possible solutions for each problem
- ☞ Covers all topics for competitive exams
- ☞ Covers interview questions on data structures and algorithms
- ☞ Reference Manual for working people
- ☞ Campus Preparation
- ☞ Degree/Masters Course Preparation
- ☞ Big Job Hunters: Microsoft, Google, Amazon, Yahoo, Oracle, Facebook and many more

ABOUT THE AUTHOR



Narasimha Karumanchi is the Senior Software Developer at Amazon Corporation, India. Most recently he worked for IBM Labs, Hyderabad and prior to that he served for Mentor Graphics and Microsoft, Hyderabad. He received his B-TECH. in Computer Science from JNT University and his M-Tech. in Computer Science from IIT Bombay.

He has experience in teaching data structures and algorithms at various training centers and colleges. He was born and brought up in Kambhampadu, Macherla (Palnadu), Guntur, Andhra Pradesh.

CareerMonk Publications

ISBN 978-0-615-45981-3



9 780615 459813 >

DATA STRUCTURES AND ALGORITHMS MADE EASY

DATA STRUCTURES AND ALGORITHMS MADE EASY

Success Key for:

- Campus Preparation
- Degree/Masters Course Preparation
- Instructor's
- GATE Preparation
- Big Job hunters: Microsoft, Google, Amazon, Yahoo, Facebook, Adobe and many more
- Reference Manual for Working People

Multiple smart
solutions with
different
complexities

Narasimha Karumanchi

M-Tech, IIT Bombay

Founder of CareerMonk.com

CareerMonk Publications

To My Parents
-Laxmi and Modaiah

To My Family Members

To My Friends

To IIT Bombay

To All Hard Workers

Copyright ©2010 by CareerMonk.com

All rights reserved.

Designed by Narasimha Karumanchi

Printed in India

Acknowledgements

I would like to express my gratitude to the 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.

I would like to thank *Ram Mohan Mullapudi* for encouraging me when I was at IIT Bombay. He is the first person who taught me the importance of *algorithms* and its *design*. From that day I keep on updating myself.

I would like to thank *Vamshi Krishna* [*Mentor Graphics*] and *Kalyani Tummala* [*Xilinx*] for spending time in reviewing this book and providing me the valuable suggestions almost every day.

I would like to thank *Sobhan* [*Professor IIT, Hyderabad*] for spending his valuable time in reviewing the book and suggestions. His review gave me the confidence in the quality of the book.

I would like to thank *Kiran* and *Laxmi* [Founder's of *TheGATEMATE.com*] for approaching me for teaching Data Structures and Algorithms at their training centers. They are the primary reason for initiation of this book.

My *friends* and *colleagues* have contributed greatly to the quality of this book. I thank all of you for your help and suggestions.

Special thanks should go to my wife, *Sailaja* for her encouragement and help during writing of this book.

Last but not least, I would like to thank Director's of *Guntur Vikas College*, *Gopala Krishna Murthy* [Director of *ACE Engineering Academy*], *TRC Bose* [Former Director of *APTransco*] and *Venkateswara Rao* [*VNR Vignanjyothi Engineering College, Hyderabad*] for helping me and my family during our studies.

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

Preface

Dear Reader,

Please Hold on! I know many people do not read preface. But I would like to strongly recommend reading preface of this book at least. This preface has *something different* from regular prefaces.

As a *job seeker* if you read complete book with good understanding, I am sure you will challenge the interviewer's and that is the objective of this book.

If you read as an *instructor*, you will give better lectures with easy go approach and as a result your students will feel proud for selecting Computer Science / Information Technology as their degree.

This book is very much useful for the *students* of *Engineering* and *Masters* during their academic preparations. All the chapters of this book contain theory and their related problems as many as possible. There a total of approximately 700 algorithmic puzzles and all of them are with solutions.

If you read as a *student* preparing for competition exams for Computer Science/Information Technology], the content of this book covers *all* the *required* topics in full details. While writing the book, an intense care has been taken to help students who are preparing for these kinds of 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 of the problems, *multiple* solutions are provided with different 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, at least one complete reading of this book is required to get full understanding of all the topics. In the subsequent readings, readers can directly go to any chapter and refer. Even though, enough readings were given for correcting the errors, due to human tendency there could be some minor typos in the book. If any such typos found, they will be updated at www.CareerMonk.com. I request readers 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. Have a nice reading.

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

Table of Contents

Chapter 1	Introduction	29
	Variables	29
	Data types	29
	System defined data types (Primitive data types)	30
	User defined data types.....	30
	Data Structure	30
	Abstract Data Types (ADT's).....	31
	Memory and Variables	31
	Size of a Variable.....	32
	Address of a Variable	32
	Pointers.....	33
	Declaration of Pointers.....	33
	Pointers Usage.....	33
	Pointer Manipulation	34
	Arrays and Pointers	35
	Dynamic Memory Allocation.....	36
	Function Pointers.....	36
	Parameter Passing Techniques.....	37
	Actual and Formal Parameters.....	37
	Semantics of Parameter Passing.....	38
	Language Support for Parameter Passing Techniques	38
	Pass by Value.....	38
	Pass by Result.....	39
	Pass by Value-Result.....	40
	Pass by Reference (aliasing)	41
	Pass by Name.....	42
	Binding	43
	Binding Times	43
	Static Binding (Early binding).....	43
	Dynamic Binding (Late binding).....	43
	Scope.....	44

Static Scope.....	44
Dynamic Scope.....	45
Storage Classes.....	46
Auto Storage Class.....	46
Extern storage class.....	47
Register Storage Class	52
Static Storage Class.....	52
Storage Organization	53
Static Segment.....	54
Stack Segment	54
Heap Segment	56
Shallow Copy versus Deep Copy.....	57
Chapter 2 Analysis of Algorithms	58
Introduction	58
What is an Algorithm?	58
Why Analysis of Algorithms?	58
Goal of Analysis of Algorithms?.....	59
What is Running Time Analysis?.....	59
How to Compare Algorithms?	59
What is Rate of Growth?	60
Commonly used Rate of Growths.....	60
Types of Analysis	61
Asymptotic Notation?.....	62
Big-O Notation.....	62
Big-O Visualization.....	63
Big-O Examples.....	63
No Uniqueness?.....	64
Omega- Ω Notation.....	64
Ω Examples.....	65
Theta- θ Notation.....	65
Θ Examples.....	66
Important Notes.....	66
Why is it called Asymptotic Analysis?	67

Guidelines for Asymptotic Analysis?	67
Properties of Notations	69
Commonly used Logarithms and Summations	70
Master Theorem for Divide and Conquer	70
Problems Divide and Conquer Master Theorem	71
Master Theorem for Subtract and Conquer Recurrences	73
Variant of subtraction and conquer master theorem	73
Problems on Algorithms Analysis	73
Chapter 3 Recursion and Backtracking	88
Introduction	88
What is Recursion?	88
Why Recursion?	88
Format of a Recursive Function	88
Recursion and Memory (Visualization)	89
Recursion versus Iteration	90
Recursion	90
Iteration	91
Notes on Recursion	91
Example Algorithms of Recursion	91
Problems on Recursion	91
What is Backtracking?	92
Example Algorithms Of Backtracking	93
Problems On Backtracking	93
Chapter 4 Linked Lists	95
What is a Linked List?	95
Linked Lists ADT	95
Why Linked Lists?	95
Arrays Overview	96
Why Constant Time for Accessing Array Elements?	96
Advantages of Arrays	96
Disadvantages of Arrays	96
Dynamic Arrays	96
Advantages of Linked Lists	97

Issues with Linked Lists (Disadvantages).....	97
Comparison of Linked Lists with Arrays and Dynamic Arrays.....	97
Singly Linked Lists.....	98
Basic Operations on a List	98
Traversing the Linked List.....	98
Singly Linked List Insertion	99
Inserting a Node in Singly Linked List at the Beginning.....	99
Inserting a Node in Singly Linked List at the Ending.....	100
Inserting a Node in Singly Linked List in the Middle.....	100
Singly Linked List Deletion	102
Deleting the First Node in Singly Linked List.....	102
Deleting the last node in Singly Linked List	102
Deleting an Intermediate Node in Singly Linked List	103
Deleting Singly Linked List	104
Doubly Linked Lists.....	105
Doubly Linked List Insertion	105
Inserting a Node in Doubly Linked List at the Beginning.....	106
Inserting a Node in Doubly Linked List at the Ending.....	106
Inserting a Node in Doubly Linked List in the Middle.....	106
Doubly Linked List Deletion	108
Deleting the First Node in Doubly Linked List.....	108
Deleting the Last Node in Doubly Linked List.....	109
Deleting an Intermediate Node in Doubly Linked List	110
Circular Linked Lists.....	111
Counting Nodes in a Circular List.....	112
Printing the contents of a circular list	112
Inserting a Node at the End of a Circular Linked List	113
Inserting a Node at Front of a Circular Linked List	114
Deleting the Last Node in a Circular List	116
Deleting the First Node in a Circular List.....	117
Applications of Circular List.....	118
A Memory-Efficient Doubly Linked List	119
Problems on Linked Lists	120

Chapter 5	Stacks	143
What is a Stack?		143
How are Stacks Used?		143
Stack ADT		144
Main stack operations		144
Auxiliary stack operations		144
Exceptions		144
Applications		144
Implementation.....		145
Simple Array Implementation		145
Dynamic Array Implementation.....		147
Performance		150
Linked List Implementation.....		150
Performance		152
Comparison of Implementations.....		152
Comparing Incremental Strategy and Doubling Strategy.....		152
Comparing Array Implementation and Linked List Implementation.....		153
Problems on Stacks		153
Chapter 6	Queues	176
What is a Queue?		176
How are Queues Used?		176
Queue ADT		177
Main queue operations		177
Auxiliary queue operations		177
Exceptions		177
Applications		177
Direct applications		177
Indirect applications		177
Implementation.....		178
Why Circular Arrays?.....		178
Simple Circular Array Implementation.....		178
Performance & Limitations		181
Dynamic Circular Array Implementation		181

Performance	184
Linked List Implementation	184
Performance	186
Comparison of Implementations.....	186
Problems on Queues	186
Chapter 7 Trees.....	190
What is a Tree?	190
Glossary	190
Binary Trees	191
Types of Binary Trees	192
Properties of Binary Trees	193
Structure of Binary Trees.....	194
Operations on Binary Trees.....	194
Applications of Binary Trees	195
Binary Tree Traversals.....	195
Traversal Possibilities	195
Classifying the Traversals	196
PreOrder Traversal	196
InOrder Traversal	198
PostOrder Traversal.....	199
Level Order Traversal	201
Problems on Binary Trees	202
Generic Trees (N-ary Trees).....	226
Representation of Generic Trees.....	227
Problems on Generic Trees	228
Threaded Binary Tree Traversals [Stack or Queue less Traversals]	234
Issues with Regular Binary Trees	234
Motivation for Threaded Binary Trees	235
Classifying Threaded Binary Trees	235
Types of Threaded Binary Trees	236
Threaded Binary Tree structure.....	236
Difference between Binary Tree and Threaded Binary Tree Structures	236
Finding Inorder Successor in Inorder Threaded Binary Tree	238

Inorder Traversal in Inorder Threaded Binary Tree.....	238
Finding PreOrder Successor in InOrder Threaded Binary Tree	239
PreOrder Traversal of InOrder Threaded Binary Tree	239
Insertion of Nodes in InOrder Threaded Binary Trees.....	240
Problems on Threaded binary Trees.....	241
Expression Trees	243
Algorithm for Building Expression Tree from Postfix Expression.....	243
Example	244
XOR Trees	246
Binary Search Trees (BSTs)	247
Why Binary Search Trees?	247
Binary Search Tree Property	247
Binary Search Tree Declaration	248
Operations on Binary Search Trees.....	248
Important Notes on Binary Search Trees	248
Finding an Element in Binary Search Trees.....	249
Finding an Minimum Element in Binary Search Trees	250
Finding an Maximum Element in Binary Search Trees.....	251
Where is Inorder Predecessor and Successor?	252
Inserting an Element from Binary Search Tree.....	252
Deleting an Element from Binary Search Tree	253
Problems on Binary Search Trees	255
Balanced Binary Search Trees	265
Complete Balanced Binary Search Trees	266
AVL (Adelson-Velskii and Landis) trees	266
Properties of AVL Trees	266
Minimum/Maximum Number of Nodes in AVL Tree	267
AVL Tree Declaration.....	267
Finding Height of an AVL tree	268
Rotations.....	268
Observation	268
Types of Violations	269
Single Rotations	269

Double Rotations	271
Insertion into an AVL tree	273
Problems on AVL Trees.....	274
Other Variations in Trees.....	279
Red-Black Trees	279
Splay Trees	280
Augmented Trees.....	280
Interval Trees	281
Chapter 8 Priority Queue and Heaps	283
What is a Priority Queue?.....	283
Priority Queue ADT	283
Main Priority Queues Operations.....	283
Auxiliary Priority Queues Operations.....	284
Priority Queue Applications	284
Priority Queue Implementations.....	284
Unordered Array Implementation.....	284
Unordered List Implementation	284
Ordered Array Implementation.....	284
Ordered List Implementation.....	285
Binary Search Trees Implementation	285
Balanced Binary Search Trees Implementation	285
Binary Heap Implementation.....	285
Comparing Implementations.....	285
Heaps and Binary Heap	285
What is a Heap?	285
Types of Heaps?.....	286
Binary Heaps	287
Representing Heaps	287
Declaration of Heap.....	287
Creating Heap	287
Parent of a Node.....	288
Children of a Node.....	288
Getting the Maximum Element	288

Heapifying an Element	289
Deleting an Element	291
Inserting an Element	291
Destroying Heap	293
Heapifying the Array	293
Heapsort	294
Problems on Priority Queues [Heaps]	295
Chapter 9 Disjoint Sets ADT	307
Introduction	307
Equivalence Relations and Equivalence Classes	307
Disjoint Sets ADT	308
Applications	308
Tradeoffs in Implementing Disjoint Sets ADT	308
Fast FIND Implementation (Quick FIND)	309
Fast UNION Implementation (Quick UNION)	309
Fast UNION implementation (Slow FIND)	309
Fast UNION implementations (Quick FIND)	313
UNION by Size	313
UNION by Height (UNION by Rank)	314
Comparing UNION by Size and UNION by Height	315
Path Compression	316
Summary	317
Problems on Disjoint Sets	317
Chapter 10 Graph Algorithms	319
Introduction	319
Glossary	319
Applications of Graphs	322
Graph Representation	322
Adjacency Matrix	322
Adjacency List	324
Adjacency Set	326
Comparison of Graph Representations	326
Graph Traversals	327

Depth First Search [DFS].....	327
Breadth First Search [BFS].....	332
Comparing DFS and BFS	334
Topological Sort	335
Applications of Topological Sorting.....	336
Shortest Path Algorithms	337
Shortest Path in Unweighted Graph.....	337
Shortest path in Weighted Graph [Dijkstra's].....	339
Bellman-Ford Algorithm.....	343
Overview of Shortest Path Algorithms.....	344
Minimal Spanning Tree	344
Prim's Algorithm	344
Kruskal's Algorithm	345
Problems on Graph Algorithms	349
Chapter 11 Sorting	378
What is Sorting?.....	378
Why Sorting?	378
Classification	378
By Number of Comparisons	378
By Number of Swaps.....	378
By Memory Usage	378
By Recursion	379
By Stability	379
By Adaptability	379
Other Classifications.....	379
Internal Sort	379
External Sort.....	379
Bubble sort	379
Implementation.....	380
Performance	381
Selection Sort	381
Algorithm	381
Implementation.....	381

Performance	382
Insertion sort	382
Advantages	382
Algorithm	383
Implementation.....	383
Example	383
Analysis	384
Performance	384
Comparisons to Other Sorting Algorithms.....	384
Shell sort	385
Implementation.....	385
Analysis	386
Performance	386
Merge sort	386
Important Notes	386
Implementation.....	387
Analysis	388
Performance	388
Heapsort	388
Performance	389
Quicksort	389
Algorithm	389
Implementation.....	389
Analysis	390
Performance	392
Randomized Quick sort	392
Tree Sort	393
Performance	393
Comparison of Sorting Algorithms	393
Linear Sorting Algorithms.....	394
Counting Sort	394
Bucket sort [or Bin Sort].....	395
Radix sort.....	396

Topological Sort	396
External Sorting	397
Problems on Sorting	398
Chapter 12 Searching	414
What is Searching?	414
Why Searching?	414
Types of Searching.....	414
Unordered Linear Search.....	414
Sorted/Ordered Linear Search.....	415
Binary Search	415
Comparing Basic Searching Algorithms	417
Symbol Tables and Hashing	417
String Searching Algorithms	417
Problems on Searching.....	417
Chapter 13 Selection Algorithms [Medians].....	450
What are Selection Algorithms?	450
Selection by Sorting.....	450
Partition-based Selection Algorithm	450
Linear Selection algorithm - Median of Medians algorithm	450
Finding the k Smallest Elements in Sorted Order.....	451
Problems on Selection Algorithms	451
Chapter 14 Symbol Tables	464
Introduction	464
What are Symbol Tables?	464
Symbol Table Implementations	465
Unordered Array Implementation.....	465
Ordered [Sorted] Array Implementation.....	465
Unordered Linked List Implementation.....	465
Ordered Linked List Implementation.....	465
Binary Search Trees Implementation	465
Balanced Binary Search Trees Implementation	465
Ternary Search Implementation	466
Hashing Implementation.....	466

Comparison of Symbol Table Implementations.....	466
Chapter 15 Hashing.....	467
What is Hashing?	467
Why Hashing?.....	467
HashTable ADT.....	467
Understanding Hashing.....	467
If Arrays Are There Why Hashing?.....	468
Components in Hashing	469
Hash Table.....	469
Hash Function.....	470
How to Choose Hash Function?.....	470
Characteristics of Good Hash Functions.....	470
Load Factor.....	470
Collisions	470
Collision Resolution Techniques.....	471
Separate Chaining	471
Open Addressing.....	471
Linear Probing	472
Quadratic Probing.....	472
Double Hashing.....	473
Comparison of Collision Resolution Techniques	473
Comparisons: Linear Probing vs. Double Hashing.....	473
Comparisons: Open Addressing vs. Separate Chaining	474
Comparisons: Open Addressing methods.....	474
How Hashing Gets $O(1)$ Complexity?.....	474
Hashing Techniques	474
Static Hashing	475
Dynamic Hashing.....	475
Problems for which Hash Tables are not Suitable	475
Problems on Hashing.....	475
Chapter 16 String Algorithms.....	489
Introduction	489
String Matching Algorithms.....	489

Brute Force Method.....	490
Robin-Karp String Matching Algorithm	490
Selecting Hash Function.....	490
Step by Step explanation.....	492
String Matching with Finite Automata	492
Finite Automata	492
How Finite Automata Works?	492
Important Notes for Constructing the Finite Automata	493
Matching Algorithm	493
KMP Algorithm	493
Filling Prefix Table	494
Matching Algorithm	495
Boyce-Moore Algorithm	498
Data structures for Storing Strings.....	499
Hash Tables for Strings.....	499
Binary Search Trees for Strings.....	499
Issues with Binary Search Tree Representation	499
Tries	500
What is a Trie?	500
Why Tries?	500
Inserting a String in Trie	501
Searching a String in Trie	501
Issues with Tries Representation	502
Ternary Search Trees.....	502
Ternary Search Trees Declaration.....	502
Inserting strings in Ternary Search Tree	503
Searching in Ternary Search Tree.....	505
Displaying All Words of Ternary Search Tree	506
Finding Length of Largest Word in TST.....	507
Comparing BSTs, Tries and TSTs	507
Suffix Trees.....	507
Prefix and Suffix.....	507
Observation	508

What is a Suffix Tree?	508
The Construction of Suffix Trees	508
Applications of Suffix Trees.....	511
Problems on Strings	511
Chapter 17 Algorithms Design Techniques	520
Introduction	520
Classification	520
Classification by Implementation Method	520
Recursion or Iteration.....	520
Procedural or Declarative (Non-Procedural)	521
Serial or Parallel or Distributed.....	521
Deterministic or Non-Deterministic.....	521
Exact or Approximate	521
Classification by Design Method.....	521
Greedy Method	521
Divide and Conquer.....	522
Dynamic Programming	522
Linear Programming.....	522
Reduction [Transform and Conquer].....	522
Other Classifications	523
Classification by Research Area	523
Classification by Complexity	523
Randomized Algorithms.....	523
Branch and Bound Enumeration and Backtracking.....	523
Chapter 18 Greedy Algorithms.....	524
Introduction	524
Greedy strategy	524
Elements of Greedy Algorithms.....	524
Greedy choice property	524
Optimal substructure	524
Does Greedy Works Always?	525
Advantages and Disadvantages of Greedy Method	525
Greedy Applications	525

Understanding Greedy Technique	525
Huffman coding algorithm	525
Problems on greedy algorithms	529
Chapter 19 Divide and Conquer Algorithms	540
Introduction	540
What is Divide and Conquer Strategy?	540
Does Divide and Conquer Work Always?	540
Divide and Conquer Visualization	540
Understanding Divide and Conquer	541
Advantages of Divide and Conquer	542
Disadvantages of Divide and Conquer	542
Master Theorem	543
Divide and Conquer Applications	543
Problems on Divide and Conquer	543
Chapter 20 Dynamic Programming	560
Introduction	560
What is Dynamic Programming Strategy?	560
Can Dynamic Programming Solve Any Problem?	560
Dynamic Programming Approaches	560
Bottom-up Dynamic Programming	561
Top-down Dynamic Programming	561
Bottom-up versus Top-down Programming	561
Examples of Dynamic Programming Algorithms	561
Understanding Dynamic Programming	561
Fibonacci Series	562
Observations	564
Factorial of a Number	564
Problems on Dynamic Programming	566
Chapter 21 Complexity Classes	611
Introduction	611
Polynomial/exponential time	611
What is Decision Problem?	612
Decision Procedure	612