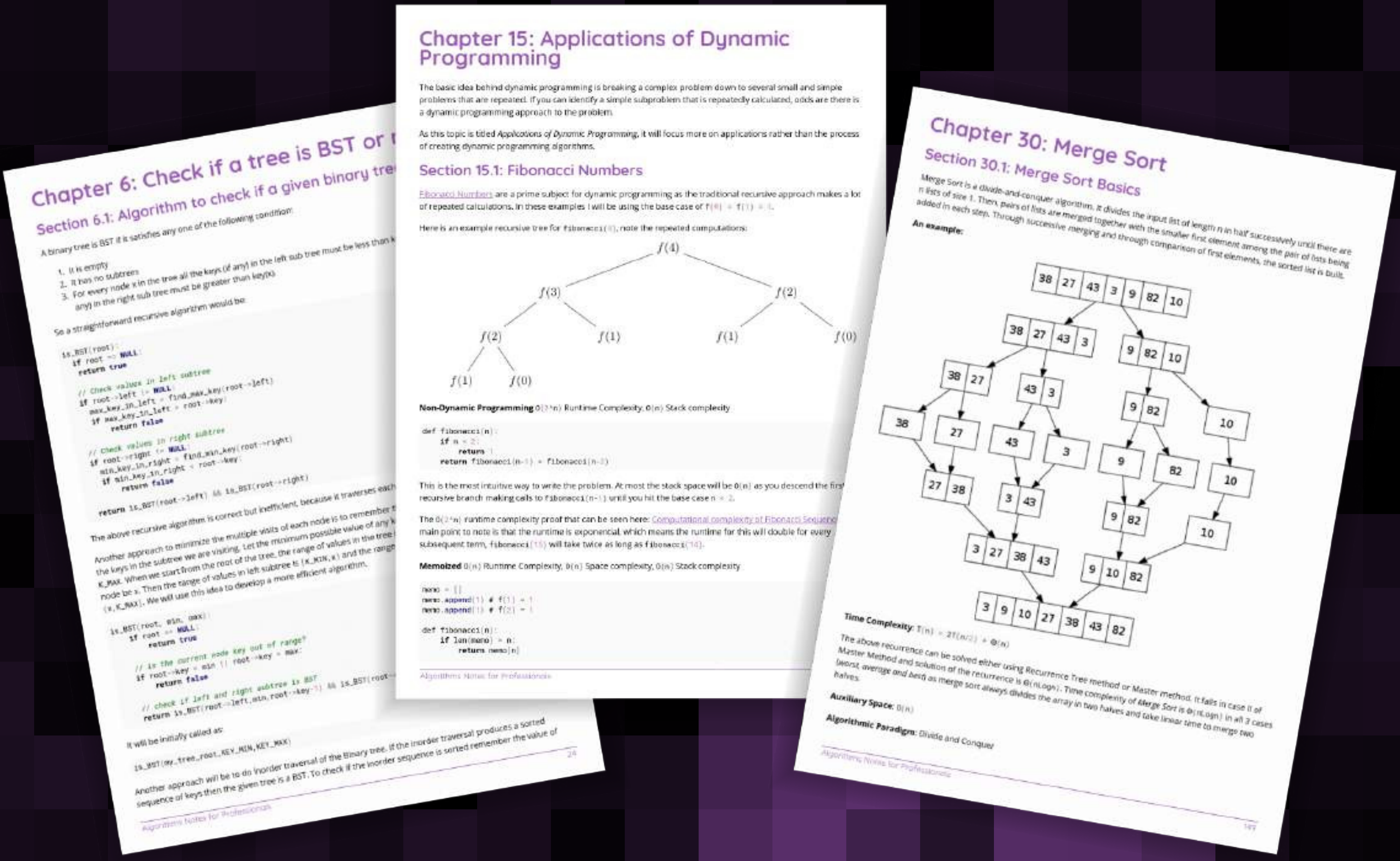
Algorithms

Notes for Professionals



Chapter 6: Check if a tree is BST or not

Section 6.1: Algorithm to check if a given binary tree is BST

Chapter 15: Applications of Dynamic Programming

Section 15.1: Fibonacci Numbers

Chapter 30: Merge Sort

Section 30.1: Merge Sort Basics

200+ pages

of professional hints and tricks

GoalKicker.com
Free Programming Books

Disclaimer
This is an unofficial free book created for educational purposes and is not affiliated with official Algorithms group(s) or company(s).
All trademarks and registered trademarks are the property of their respective owners

Contents

About 1

Chapter 1: Getting started with algorithms 2

- Section 1.1: A sample algorithmic problem 2
- Section 1.2: Getting Started with Simple Fizz Buzz Algorithm in Swift 2

Chapter 2: Algorithm Complexity 5

- Section 2.1: Big-Theta notation 5
- Section 2.2: Comparison of the asymptotic notations 6
- Section 2.3: Big-Omega Notation 6

Chapter 3: Big-O Notation 8

- Section 3.1: A Simple Loop 9
- Section 3.2: A Nested Loop 9
- Section 3.3: O(log n) types of Algorithms 10
- Section 3.4: An O(log n) example 12

Chapter 4: Trees 14

- Section 4.1: Typical anary tree representation 14
- Section 4.2: Introduction 14
- Section 4.3: To check if two Binary trees are same or not 15

Chapter 5: Binary Search Trees 18

- Section 5.1: Binary Search Tree - Insertion (Python) 18
- Section 5.2: Binary Search Tree - Deletion(C++) 20
- Section 5.3: Lowest common ancestor in a BST 21
- Section 5.4: Binary Search Tree - Python 22

Chapter 6: Check if a tree is BST or not 24

- Section 6.1: Algorithm to check if a given binary tree is BST 24
- Section 6.2: If a given input tree follows Binary search tree property or not 25

Chapter 7: Binary Tree traversals 26

- Section 7.1: Level Order traversal - Implementation 26
- Section 7.2: Pre-order, Inorder and Post Order traversal of a Binary Tree 27

Chapter 8: Lowest common ancestor of a Binary Tree 29

- Section 8.1: Finding lowest common ancestor 29

Chapter 9: Graph 30

- Section 9.1: Storing Graphs (Adjacency Matrix) 30
- Section 9.2: Introduction To Graph Theory 33
- Section 9.3: Storing Graphs (Adjacency List) 37
- Section 9.4: Topological Sort 39
- Section 9.5: Detecting a cycle in a directed graph using Depth First Traversal 40
- Section 9.6: Thorup's algorithm 41

Chapter 10: Graph Traversals 43

- Section 10.1: Depth First Search traversal function 43

Chapter 11: Dijkstra’s Algorithm 44

- Section 11.1: Dijkstra's Shortest Path Algorithm 44

Chapter 12: A* Pathfinding 49

- Section 12.1: Introduction to A* 49
- Section 12.2: A* Pathfinding through a maze with no obstacles 49
- Section 12.3: Solving 8-puzzle problem using A* algorithm 56

Chapter 13: A* Pathfinding Algorithm 59
Section 13.1: Simple Example of A* Pathfinding: A maze with no obstacles 59
Chapter 14: Dynamic Programming 66
Section 14.1: Edit Distance 66
Section 14.2: Weighted Job Scheduling Algorithm 66
Section 14.3: Longest Common Subsequence 70
Section 14.4: Fibonacci Number 71
Section 14.5: Longest Common Substring 72
Chapter 15: Applications of Dynamic Programming 73
Section 15.1: Fibonacci Numbers 73
Chapter 16: Kruskal's Algorithm 76
Section 16.1: Optimal, disjoint-set based implementation 76
Section 16.2: Simple, more detailed implementation 77
Section 16.3: Simple, disjoint-set based implementation 77
Section 16.4: Simple, high level implementation 77
Chapter 17: Greedy Algorithms 79
Section 17.1: Huffman Coding 79
Section 17.2: Activity Selection Problem 82
Section 17.3: Change-making problem 84
Chapter 18: Applications of Greedy technique 86
Section 18.1: Offline Caching 86
Section 18.2: Ticket automat 94
Section 18.3: Interval Scheduling 97
Section 18.4: Minimizing Lateness 101
Chapter 19: Prim's Algorithm 105
Section 19.1: Introduction To Prim's Algorithm 105
Chapter 20: Bellman–Ford Algorithm 113
Section 20.1: Single Source Shortest Path Algorithm (Given there is a negative cycle in a graph) 113
Section 20.2: Detecting Negative Cycle in a Graph 116
Section 20.3: Why do we need to relax all the edges at most (V-1) times 118
Chapter 21: Line Algorithm 121
Section 21.1: Bresenham Line Drawing Algorithm 121
Chapter 22: Floyd-Warshall Algorithm 124
Section 22.1: All Pair Shortest Path Algorithm 124
Chapter 23: Catalan Number Algorithm 127
Section 23.1: Catalan Number Algorithm Basic Information 127
Chapter 24: Multithreaded Algorithms 129
Section 24.1: Square matrix multiplication multithread 129
Section 24.2: Multiplication matrix vector multithread 129
Section 24.3: merge-sort multithread 129
Chapter 25: Knuth Morris Pratt (KMP) Algorithm 131
Section 25.1: KMP-Example 131
Chapter 26: Edit Distance Dynamic Algorithm 133
Section 26.1: Minimum Edits required to convert string 1 to string 2 133
Chapter 27: Online algorithms 136
Section 27.1: Paging (Online Caching) 137
Chapter 28: Sorting 143
Section 28.1: Stability in Sorting 143

Chapter 29: Bubble Sort 144
Section 29.1: Bubble Sort 144
Section 29.2: Implementation in C & C++ 144
Section 29.3: Implementation in C# 145
Section 29.4: Python Implementation 146
Section 29.5: Implementation in Java 147
Section 29.6: Implementation in Javascript 147
Chapter 30: Merge Sort 149
Section 30.1: Merge Sort Basics 149
Section 30.2: Merge Sort Implementation in Go 150
Section 30.3: Merge Sort Implementation in C & C# 150
Section 30.4: Merge Sort Implementation in Java 152
Section 30.5: Merge Sort Implementation in Python 153
Section 30.6: Bottoms-up Java Implementation 154
Chapter 31: Insertion Sort 156
Section 31.1: Haskell Implementation 156
Chapter 32: Bucket Sort 157
Section 32.1: C# Implementation 157
Chapter 33: Quicksort 158
Section 33.1: Quicksort Basics 158
Section 33.2: Quicksort in Python 160
Section 33.3: Lomuto partition java implementation 160
Chapter 34: Counting Sort 162
Section 34.1: Counting Sort Basic Information 162
Section 34.2: Psuedocode Implementation 162
Chapter 35: Heap Sort 164
Section 35.1: C# Implementation 164
Section 35.2: Heap Sort Basic Information 164
Chapter 36: Cycle Sort 166
Section 36.1: Pseudocode Implementation 166
Chapter 37: Odd-Even Sort 167
Section 37.1: Odd-Even Sort Basic Information 167
Chapter 38: Selection Sort 170
Section 38.1: Elixir Implementation 170
Section 38.2: Selection Sort Basic Information 170
Section 38.3: Implementation of Selection sort in C# 172
Chapter 39: Searching 174
Section 39.1: Binary Search 174
Section 39.2: Rabin Karp 175
Section 39.3: Analysis of Linear search (Worst, Average and Best Cases) 176
Section 39.4: Binary Search: On Sorted Numbers 178
Section 39.5: Linear search 178
Chapter 40: Substring Search 180
Section 40.1: Introduction To Knuth-Morris-Pratt (KMP) Algorithm 180
Section 40.2: Introduction to Rabin-Karp Algorithm 183
Section 40.3: Python Implementation of KMP algorithm 186
Section 40.4: KMP Algorithm in C 187
Chapter 41: Breadth-First Search 190

Section 41.1: Finding the Shortest Path from Source to other Nodes ... 190
Section 41.2: Finding Shortest Path from Source in a 2D graph ... 196
Section 41.3: Connected Components Of Undirected Graph Using BFS ... 197
Chapter 42: Depth First Search ... 202
Section 42.1: Introduction To Depth-First Search ... 202
Chapter 43: Hash Functions ... 207
Section 43.1: Hash codes for common types in C# ... 207
Section 43.2: Introduction to hash functions ... 208
Chapter 44: Travelling Salesman ... 210
Section 44.1: Brute Force Algorithm ... 210
Section 44.2: Dynamic Programming Algorithm ... 210
Chapter 45: Knapsack Problem ... 212
Section 45.1: Knapsack Problem Basics ... 212
Section 45.2: Solution Implemented in C# ... 212
Chapter 46: Equation Solving ... 214
Section 46.1: Linear Equation ... 214
Section 46.2: Non-Linear Equation ... 216
Chapter 47: Longest Common Subsequence ... 220
Section 47.1: Longest Common Subsequence Explanation ... 220
Chapter 48: Longest Increasing Subsequence ... 225
Section 48.1: Longest Increasing Subsequence Basic Information ... 225
Chapter 49: Check two strings are anagrams ... 228
Section 49.1: Sample input and output ... 228
Section 49.2: Generic Code for Anagrams ... 229
Chapter 50: Pascal's Triangle ... 231
Section 50.1: Pascal triangle in C ... 231
Chapter 51: Algo:- Print a m*n matrix in square wise ... 232
Section 51.1: Sample Example ... 232
Section 51.2: Write the generic code ... 232
Chapter 52: Matrix Exponentiation ... 233
Section 52.1: Matrix Exponentiation to Solve Example Problems ... 233
Chapter 53: polynomial-time bounded algorithm for Minimum Vertex Cover ... 237
Section 53.1: Algorithm Pseudo Code ... 237
Chapter 54: Dynamic Time Warping ... 238
Section 54.1: Introduction To Dynamic Time Warping ... 238
Chapter 55: Fast Fourier Transform ... 242
Section 55.1: Radix 2 FFT ... 242
Section 55.2: Radix 2 Inverse FFT ... 247
Appendix A: Pseudocode ... 249
Section A.1: Variable affectations ... 249
Section A.2: Functions ... 249
Credits ... 250
You may also like ... 252

Click here to download full PDF material