

Data Structures & Algorithm Analysis In

C++

**3rd
edition**

Data Structures and Algorithm Analysis in C++



3200
SEMINAR LIBRARY
Department of Computer Science
UNIVERSITY OF KARACHI
27.7.16

Summary	59
Exercises	59
References	61

Chapter 2 Algorithm Analysis

63

2.1	Mathematical Background	63
2.2	Model	66
2.3	What to Analyze	66
2.4	Running Time Calculations	69
2.4.1	A Simple Example	69
2.4.2	General Rules	70
2.4.3	Solutions for the Maximum Subsequence Sum Problem	72
2.4.4	Logarithms in the Running Time	78
2.4.5	Checking Your Analysis	82
2.4.6	A Grain of Salt	83
	Summary	83
	Exercises	84
	References	89

Chapter 3 Lists, Stacks, and Queues

91

3.1	Abstract Data Types (ADTs)	91
3.2	The List ADT	92
3.2.1	Simple Array Implementation of Lists	92
3.2.2	Simple Linked Lists	93
3.3	vector and list in the STL	94
3.3.1	Iterators	95
3.3.2	Example: Using erase on a List	97
3.3.3	const_iterators	97
3.4	Implementation of vector	99
3.5	Implementation of list	103
3.6	The Stack ADT	114
3.6.1	Stack Model	114
3.6.2	Implementation of Stacks	115
3.6.3	Applications	116
3.7	The Queue ADT	124
3.7.1	Queue Model	124
3.7.2	Array Implementation of Queues	124
3.7.3	Applications of Queues	126
	Summary	127
	Exercises	128

Chapter 4 Trees**133**

- 4.1 Preliminaries 133
 - 4.1.1 Implementation of Trees 134
 - 4.1.2 Tree Traversals with an Application 135
- 4.2 Binary Trees 139
 - 4.2.1 Implementation 140
 - 4.2.2 An Example: Expression Trees 141
- 4.3 The Search Tree ADT—Binary Search Trees 144
 - 4.3.1 contains 145
 - 4.3.2 findMin and findMax 145
 - 4.3.3 insert 149
 - 4.3.4 remove 150
 - 4.3.5 Destructor and Copy Assignment Operator 152
 - 4.3.6 Average-Case Analysis 153
- 4.4 AVL Trees 156
 - 4.4.1 Single Rotation 159
 - 4.4.2 Double Rotation 162
- 4.5 Splay Trees 169
 - 4.5.1 A Simple Idea (That Does Not Work) 170
 - 4.5.2 Splaying 172
- 4.6 Tree Traversals (Revisited) 178
- 4.7 B-Trees 179
- 4.8 Sets and Maps in the Standard Library 185
 - 4.8.1 Sets 185
 - 4.8.2 Maps 186
 - 4.8.3 Implementation of set and map 187
 - 4.8.4 An Example That Uses Several Maps 188
- Summary 194
- Exercises 194
- References 201

Chapter 5 Hashing**205**

- 5.1 General Idea 205
- 5.2 Hash Function 206
- 5.3 Separate Chaining 208
- 5.4 Hash Tables Without Linked Lists 212
 - 5.4.1 Linear Probing 213
 - 5.4.2 Quadratic Probing 215
 - 5.4.3 Double Hashing 219
- 5.5 Rehashing 220

8.4	Smart Union Algorithms	341
8.5	Path Compression	344
8.6	Worst Case for Union-by-Rank and Path Compression	345
8.6.1	Analysis of the Union/Find Algorithm	346
8.7	An Application	351
	Summary	354
	Exercises	355
	References	356

Chapter 9 Graph Algorithms

359

9.1	Definitions	359
9.1.1	Representation of Graphs	360
9.2	Topological Sort	362
9.3	Shortest-Path Algorithms	365
9.3.1	Unweighted Shortest Paths	367
9.3.2	Dijkstra's Algorithm	371
9.3.3	Graphs with Negative Edge Costs	380
9.3.4	Acyclic Graphs	380
9.3.5	All-Pairs Shortest Path	384
9.3.6	Shortest Path Example	385
9.4	Network Flow Problems	387
9.4.1	A Simple Maximum-Flow Algorithm	387
9.5	Minimum Spanning Tree	392
9.5.1	Prim's Algorithm	393
9.5.2	Kruskal's Algorithm	396
9.6	Applications of Depth-First Search	398
9.6.1	Undirected Graphs	399
9.6.2	Biconnectivity	401
9.6.3	Euler Circuits	405
9.6.4	Directed Graphs	408
9.6.5	Finding Strong Components	410
9.7	Introduction to NP-Completeness	412
9.7.1	Easy vs. Hard	412
9.7.2	The Class NP	413
9.7.3	NP-Complete Problems	414
	Summary	416
	Exercises	416
	References	424

Chapter 10 Algorithm Design Techniques**429**

- 10.1 Greedy Algorithms 429
 - 10.1.1 A Simple Scheduling Problem 430
 - 10.1.2 Huffman Codes 433
 - 10.1.3 Approximate Bin Packing 439
- 10.2 Divide and Conquer 447
 - 10.2.1 Running Time of Divide and Conquer Algorithms 448
 - 10.2.2 Closest-Points Problem 450
 - 10.2.3 The Selection Problem 455
 - 10.2.4 Theoretical Improvements for Arithmetic Problems 458
- 10.3 Dynamic Programming 462
 - 10.3.1 Using a Table Instead of Recursion 462
 - 10.3.2 Ordering Matrix Multiplications 464
 - 10.3.3 Optimal Binary Search Tree 467
 - 10.3.4 All-Pairs Shortest Path 471
- 10.4 Randomized Algorithms 474
 - 10.4.1 Random Number Generators 475
 - 10.4.2 Skip Lists 479
 - 10.4.3 Primality Testing 481
- 10.5 Backtracking Algorithms 484
 - 10.5.1 The Turnpike Reconstruction Problem 485
 - 10.5.2 Games 489
- Summary 495
- Exercises 495
- References 505

Chapter 11 Amortized Analysis**511**

- 11.1 An Unrelated Puzzle 512
- 11.2 Binomial Queues 512
- 11.3 Skew Heaps 517
- 11.4 Fibonacci Heaps 519
 - 11.4.1 Cutting Nodes in Leftist Heaps 520
 - 11.4.2 Lazy Merging for Binomial Queues 522
 - 11.4.3 The Fibonacci Heap Operations 526
 - 11.4.4 Proof of the Time Bound 526
- 11.5 Splay Trees 529
 - Summary 533
 - Exercises 533
 - References 535