Table of Contents

1. Introduction ---------- 15

1.2 Data Types ---------- 15

1.3 Data Structures ---------- 16

1.4 Abstract Data Types (ADTs) ---------- 16

1.5 What is an Algorithm? ---------- 16

1.6 Why the Analysis of Algorithms? ---------- 16

1.7 Goal of the Analysis of Algorithms ---------- 17

1.8 What is Running Time Analysis? ---------- 17

1.9 How to Compare Algorithms ---------- 17

1.10 What is Rate of Growth? ---------- 17

1.11 Commonly Used Rates of Growth ---------- 17

1.12 Types of Analysis ---------- 18

1.13 Asymptotic Notation ---------- 19

1.14 Big-O Notation ---------- 19

1.15 Omega-Ω Notation [Lower Bounding Function] ---------- 20

1.16 Theta-Θ Notation ---------- 20

1.17 Why is it called Asymptotic Analysis? ---------- 21

1.18 Guidelines for Asymptotic Analysis ---------- 21

1.20 Simplifying properties of asymptotic notations ---------- 22

1.21 Commonly used Logarithms and Summations ---------- 23

1.22 Master Theorem for Divide and Conquer Recurrences ---------- 23

1.23 Divide and Conquer Master Theorem: Problems & Solutions ---------- 23

1.24 Master Theorem for Subtract and Conquer Recurrences ---------- 24

1.25 Variant of Subtraction and Conquer Master Theorem ---------- 24

1.26 Method of Guessing and Confirming ---------- 25

1.27 Amortized Analysis ---------- 26

1.28 Algorithms Analysis: Problems & Solutions ---------- 26

2. Recursion and Backtracking ---------- 36

2.1 Introduction ---------- 36

2.2 What is Recursion? ---------- 36

2.3 Why Recursion? ---------- 36

2.4 Format of a Recursive Function ---------- 36

2.5 Recursion and Memory (Visualization) ---------- 37

2.6 Recursion versus Iteration ---------- 37

2.7 Notes on Recursion ---------- 37

2.8 Example Algorithms of Recursion ---------- 38

2.9 Recursion: Problems & Solutions ---------- 38

2.10 What is Backtracking? ---------- 38

2.11 Example Algorithms of Backtracking ---------- 39

2.12 Backtracking: Problems & Solutions ---------- 39

3. Linked Lists ---------- 41
3.1 What is a Linked List? ---------- 41
3.2 Linked Lists ADT ---------- 41
3.3 Why Linked Lists? ---------- 41
3.4 Arrays Overview ---------- 41
3.5 Comparison of Linked Lists with Arrays and Dynamic Arrays ---------- 42
3.6 Singly Linked Lists ---------- 43
3.7 Doubly Linked Lists ---------- 48
3.8 Circular Linked Lists ---------- 52
3.9 A Memory-efficient Doubly Linked List ---------- 56
3.10 Unrolled Linked Lists ---------- 57
3.11 Skip Lists ---------- 61
3.12 Linked Lists: Problems & Solutions ---------- 64
4. Stacks ---------- 85
4.1 What is a Stack? ---------- 85
4.2 How Stacks are used ---------- 85
4.3 Stack ADT ---------- 85
4.4 Applications ---------- 86
4.5 Implementation ---------- 86
4.6 Comparison of Implementations ---------- 92
4.7 Stacks: Problems & Solutions ---------- 92
5. Queues ---------- 112
5.1 What is a Queue? ---------- 112
5.2 How are Queues Used? ---------- 112
5.3 Queue ADT ---------- 112
5.4 Exceptions ---------- 112
5.5 Applications ---------- 112
5.6 Implementation ---------- 113
5.7 Queues: Problems & Solutions ---------- 120
6. Trees ---------- 125
6.1 What is a Tree? ---------- 125
6.2 Glossary ---------- 125
6.3 Binary Trees ---------- 126
6.4 Types of Binary Trees ---------- 126
6.5 Properties of Binary Trees ---------- 126
6.6 Binary Tree Traversals ---------- 127
6.7 Generic Trees (N-ary Trees) ---------- 147
6.8 Threaded Binary Tree Traversals (Stack or Queue-less Traversals) ---------- 152
6.9 Expression Trees ---------- 156
6.10 XOR Trees ---------- 158
6.11 Binary Search Trees (BSTs) ---------- 158
6.12 Balanced Binary Search Trees ---------- 170
6.13 AVL (Adelson-Velskii and Landis) Trees ---------- 171

6.14 Other Variations on Trees ---- 182
6.15 Supplementary Questions ---- 185

7. Priority Queues and Heaps ---- 186
7.1 What is a Priority Queue? ---- 186
7.2 Priority Queue ADT ---- 186
7.3 Priority Queue Applications ---- 186
7.4 Priority Queue Implementations ---- 186
7.5 Heaps and Binary Heaps ---- 187
7.6 Binary Heaps ---- 188
7.7 Heapsort ---- 192
7.8 Priority Queues [Heaps]: Problems & Solutions ---- 193

8. Disjoint Sets ADT ---- 202
8.1 Introduction ---- 202
8.2 Equivalence Relations and Equivalence Classes ---- 202
8.3 Disjoint Sets ADT ---- 202
8.4 Applications ---- 202
8.5 Tradeoffs in Implementing Disjoint Sets ADT ---- 203
8.8 Fast UNION Implementation (Slow FIND) ---- 203
8.9 Fast UNION Implementations (Quick FIND) ---- 205
8.10 Summary ---- 207
8.11 Disjoint Sets: Problems & Solutions ---- 207

9. Graph Algorithms ---- 209
9.1 Introduction ---- 209
9.2 Glossary ---- 209
9.3 Applications of Graphs ---- 211
9.4 Graph Representation ---- 211
9.5 Graph Traversals ---- 217
9.6 Topological Sort ---- 226
9.7 Shortest Path Algorithms ---- 227
9.8 Minimal Spanning Tree ---- 232
9.9 Graph Algorithms: Problems & Solutions ---- 235

10. Sorting ---- 251
10.1 What is Sorting? ---- 251
10.2 Why is Sorting Necessary? ---- 251
10.3 Classification of Sorting Algorithms ---- 251
10.4 Other Classifications ---- 251
10.5 Bubble Sort ---- 252
10.6 Selection Sort ---- 253
10.7 Insertion Sort ---- 254
10.8 Shell Sort ---- 256
10.9 Merge Sort ---- 257
10.10 Heap Sort ---- 259
10.11 Quick Sort ---- 259

10.12 Tree Sort 263
10.13 Comparison of Sorting Algorithms 263
10.14 Linear Sorting Algorithms 264
10.15 Counting Sort 264
10.16 Bucket Sort (or Bin Sort) 264
10.17 Radix Sort 265
10.18 Topological Sort 265
10.19 External Sorting 265
10.20 Sorting: Problems & Solutions 266
11. Searching 277
11.1 What is Searching? 277
11.2 Why do we need Searching? 277
11.3 Types of Searching 277
11.4 Unordered Linear Search 277
11.5 Sorted/Ordered Linear Search 277
11.6 Binary Search 278
11.7 Interpolation Search 278
11.8 Comparing Basic Searching Algorithms 279
11.9 Symbol Tables and Hashing 280
11.10 String Searching Algorithms 280
11.11 Searching: Problems & Solutions 280
12. Selection Algorithms [Medians] 299
12.1 What are Selection Algorithms? 299
12.2 Selection by Sorting 299
12.3 Partition-based Selection Algorithm 299
12.4 Linear Selection Algorithm - Median of Medians Algorithm 299
12.5 Finding the K Smallest Elements in Sorted Order 299
12.6 Selection Algorithms: Problems & Solutions 299
13. Symbol Tables 306
13.1 Introduction 306
13.2 What are Symbol Tables? 306
13.3 Symbol Table Implementations 306
13.4 Comparison Table of Symbols for Implementations 307
14. Hashing 308
14.1 What is Hashing? 308
14.2 Why Hashing? 308
14.3 Hash Table ADT 308
14.4 Understanding Hashing 308
14.5 Components of Hashing 309
14.6 Hash Table 309
14.7 Hash Function 310
14.8 Load Factor 311
14.9 Collisions 311

14.10 Collision Resolution Techniques ---------- 311
14.11 Separate Chaining ---------- 311
14.12 Open Addressing ---------- 312
14.13 Comparison of Collision Resolution Techniques ---------- 313
14.14 How Hashing Gets O(1) Complexity ---------- 313
14.15 Hashing Techniques ---------- 313
14.16 Problems for which Hash Tables are not suitable ---------- 314
14.17 Bloom Filters ---------- 314
14.18 Hashing: Problems & Solutions ---------- 315

15. String Algorithms ---------- 325

15.1 Introduction ---------- 325
15.2 String Matching Algorithms ---------- 325
15.3 Brute Force Method ---------- 325
15.4 Rabin-Karp String Matching Algorithm ---------- 326
15.5 String Matching with Finite Automata ---------- 326
15.6 KMP Algorithm ---------- 327
15.7 Boyer-Moore Algorithm ---------- 330
15.8 Data Structures for Storing Strings ---------- 330
15.9 Hash Tables for Strings ---------- 330
15.10 Binary Search Trees for Strings ---------- 330
15.11 Tries ---------- 330
15.12 Ternary Search Trees ---------- 332
15.13 Comparing BSTs, Tries and TSTs ---------- 335
15.14 Suffix Trees ---------- 335
15.15 String Algorithms: Problems & Solutions ---------- 337

16. Algorithms Design Techniques ---------- 345

16.1 Introduction ---------- 345
16.2 Classification ---------- 345
16.3 Classification by Implementation Method ---------- 345
16.4 Classification by Design Method ---------- 346
16.5 Other Classifications ---------- 346

17. Greedy Algorithms ---------- 348

17.1 Introduction ---------- 348
17.2 Greedy Strategy ---------- 348
17.3 Elements of Greedy Algorithms ---------- 348
17.4 Does Greedy Always Work? ---------- 348
17.5 Advantages and Disadvantages of Greedy Method ---------- 348
17.6 Greedy Applications ---------- 348
17.7 Understanding Greedy Technique ---------- 349
17.8 Greedy Algorithms: Problems & Solutions ---------- 351

18. Divide and Conquer Algorithms ---------- 356

18.1 Introduction ---------- 356
18.2 What is the Divide and Conquer Strategy? ---------- 356

18.3 Does Divide and Conquer Always Work? ---------- 356
18.4 Divide and Conquer Visualization ---------- 356
18.5 Understanding Divide and Conquer ---------- 357
18.6 Advantages of Divide and Conquer ---------- 357
18.7 Disadvantages of Divide and Conquer ---------- 357
18.8 Master Theorem ---------- 357
18.9 Divide and Conquer Applications ---------- 358
18.10 Divide and Conquer: Problems & Solutions ---------- 358

19. Dynamic Programming ---------- 369

19.1 Introduction ---------- 369
19.2 What is Dynamic Programming Strategy? ---------- 369
19.3 Properties of Dynamic Programming Strategy ---------- 369
19.4 Greedy vs Divide and Conquer vs DP ---------- 369
19.5 Can DP solve all problems? ---------- 370
19.6 Dynamic Programming Approaches ---------- 370
19.7 Understanding DP Approaches ---------- 370
19.8 Examples of DP Algorithms ---------- 373
19.9 Longest Common Subsequence ---------- 373
19.10 Dynamic Programming: Problems & Solutions ---------- 375

20. Complexity Classes ---------- 400

20.1 Introduction ---------- 400
20.2 Polynomial/Exponential Time ---------- 400
20.3 What is a Decision Problem? ---------- 400
20.4 Decision Procedure ---------- 400
20.5 What is a Complexity Class? ---------- 400
20.6 Types of Complexity Classes ---------- 401
20.7 Reductions ---------- 402
20.8 Complexity Classes: Problems & Solutions ---------- 404

21. Miscellaneous Concepts ---------- 406

21.1 Introduction ---------- 406
21.2 Hacks on Bit-wise Programming ---------- 406
21.3 Other Programming Questions ---------- 409

References ---------- 416