
Contents

| | |
|--|------|
| Preface | xi |
| Acknowledgments | xvii |
| Chapter 13. Hash Tables | 1 |
| 13.1. Introduction | 1 |
| 13.1.1. Dictionaries | 1 |
| 13.2. Hash table structure | 2 |
| 13.3. Hash functions | 4 |
| 13.3.1. Building hash functions | 4 |
| 13.4. Linear open addressing | 5 |
| 13.4.1. Operations on linear open addressed hash tables. | 8 |
| 13.4.2. Performance analysis | 10 |
| 13.4.3. Other collision resolution techniques with open addressing | 11 |
| 13.5. Chaining | 13 |
| 13.5.1. Operations on chained hash tables | 15 |
| 13.5.2. Performance analysis | 17 |
| 13.6. Applications. | 18 |
| 13.6.1. Representation of a keyword table in a compiler. | 18 |
| 13.6.2. Hash tables in the evaluation of a join operation on relational databases. | 19 |
| 13.6.3. Hash tables in a direct file organization | 22 |
| 13.7. Illustrative problems | 23 |

| | |
|---|-----|
| Chapter 14. File Organizations | 33 |
| 14.1. Introduction | 33 |
| 14.2. Files | 34 |
| 14.3. Keys | 36 |
| 14.4. Basic file operations | 38 |
| 14.5. Heap or pile organization | 38 |
| 14.5.1. Insert, delete and update operations | 39 |
| 14.6. Sequential file organization | 39 |
| 14.6.1. Insert, delete and update operations | 39 |
| 14.6.2. Making use of overflow blocks | 40 |
| 14.7. Indexed sequential file organization | 41 |
| 14.7.1. Structure of the ISAM files | 41 |
| 14.7.2. Insert, delete and update operations for a naïve ISAM file. | 42 |
| 14.7.3. Types of indexing | 43 |
| 14.8. Direct file organization | 48 |
| 14.9. Illustrative problems | 50 |
| | |
| Chapter 15. k-d Trees and Treaps | 61 |
| 15.1. Introduction | 61 |
| 15.2. k -d trees: structure and operations. | 61 |
| 15.2.1. Construction of a k -d tree | 65 |
| 15.2.2. Insert operation on k -d trees | 69 |
| 15.2.3. Find minimum operation on k -d trees | 70 |
| 15.2.4. Delete operation on k -d trees | 72 |
| 15.2.5. Complexity analysis and applications of k -d trees | 74 |
| 15.3. Treaps: structure and operations. | 76 |
| 15.3.1. Treap structure. | 76 |
| 15.3.2. Operations on treaps | 77 |
| 15.3.3. Complexity analysis and applications of treaps | 82 |
| 15.4. Illustrative problems. | 83 |
| | |
| Chapter 16. Searching | 93 |
| 16.1. Introduction | 93 |
| 16.2. Linear search | 94 |
| 16.2.1. Ordered linear search | 94 |
| 16.2.2. Unordered linear search | 94 |
| 16.3. Transpose sequential search | 96 |
| 16.4. Interpolation search | 98 |
| 16.5. Binary search | 100 |
| 16.5.1. Decision tree for binary search | 101 |

| | |
|--|-----|
| 16.6. Fibonacci search | 104 |
| 16.6.1. Decision tree for Fibonacci search | 105 |
| 16.7. Skip list search | 108 |
| 16.7.1. Implementing skip lists. | 112 |
| 16.7.2. Insert operation in a skip list | 113 |
| 16.7.3. Delete operation in a skip list | 114 |
| 16.8. Other search techniques. | 116 |
| 16.8.1. Tree search | 116 |
| 16.8.2. Graph search | 116 |
| 16.8.3. Indexed sequential search | 116 |
| 16.9. Illustrative problems. | 118 |

Chapter 17. Internal Sorting 131

| | |
|--|-----|
| 17.1. Introduction | 131 |
| 17.2. Bubble sort | 132 |
| 17.2.1. Stability and performance analysis. | 134 |
| 17.3. Insertion sort | 135 |
| 17.3.1. Stability and performance analysis. | 136 |
| 17.4. Selection sort | 138 |
| 17.4.1. Stability and performance analysis. | 140 |
| 17.5. Merge sort. | 140 |
| 17.5.1. Two-way merging. | 141 |
| 17.5.2. k -way merging. | 143 |
| 17.5.3. Non-recursive merge sort procedure | 144 |
| 17.5.4. Recursive merge sort procedure | 145 |
| 17.6. Shell sort. | 147 |
| 17.6.1. Analysis of shell sort | 153 |
| 17.7. Quick sort | 153 |
| 17.7.1. Partitioning. | 153 |
| 17.7.2. Quick sort procedure | 156 |
| 17.7.3. Stability and performance analysis. | 158 |
| 17.8. Heap sort | 159 |
| 17.8.1. Heap. | 159 |
| 17.8.2. Construction of heap | 160 |
| 17.8.3. Heap sort procedure. | 163 |
| 17.8.4. Stability and performance analysis. | 167 |
| 17.9. Radix sort | 167 |
| 17.9.1. Radix sort method. | 167 |
| 17.9.2. Most significant digit first sort | 171 |
| 17.9.3. Performance analysis. | 171 |
| 17.10. Counting sort | 171 |

| | |
|---|------------|
| 17.10.1. Performance analysis | 175 |
| 17.11. Bucket sort. | 175 |
| 17.11.1. Performance analysis | 178 |
| 17.12. Illustrative problems | 179 |
| | |
| Chapter 18. External Sorting | 197 |
| 18.1. Introduction | 197 |
| 18.1.1. The principle behind external sorting | 197 |
| 18.2. External storage devices | 198 |
| 18.2.1. Magnetic tapes. | 199 |
| 18.2.2. Magnetic disks. | 200 |
| 18.3. Sorting with tapes: balanced merge | 202 |
| 18.3.1. Buffer handling | 204 |
| 18.3.2. Balanced P-way merging on tapes | 205 |
| 18.4. Sorting with disks: balanced merge | 206 |
| 18.4.1. Balanced k-way merging on disks | 207 |
| 18.4.2. Selection tree. | 208 |
| 18.5. Polyphase merge sort | 212 |
| 18.6. Cascade merge sort | 214 |
| 18.7. Illustrative problems. | 216 |
| | |
| Chapter 19. Divide and Conquer | 229 |
| 19.1. Introduction | 229 |
| 19.2. Principle and abstraction | 229 |
| 19.3. Finding maximum and minimum | 231 |
| 19.3.1. Time complexity analysis | 232 |
| 19.4. Merge sort. | 233 |
| 19.4.1. Time complexity analysis | 233 |
| 19.5. Matrix multiplication | 234 |
| 19.5.1. Divide and Conquer-based approach to “high school” method of matrix multiplication | 234 |
| 19.5.2. Strassen’s matrix multiplication algorithm | 236 |
| 19.6. Illustrative problems. | 239 |
| | |
| Chapter 20. Greedy Method | 245 |
| 20.1. Introduction | 245 |
| 20.2. Abstraction | 245 |
| 20.3. Knapsack problem | 246 |
| 20.3.1. Greedy solution to the knapsack problem. | 247 |

| | |
|---|------------|
| 20.4. Minimum cost spanning tree algorithms | 249 |
| 20.4.1. Prim’s algorithm as a greedy method | 250 |
| 20.4.2. Kruskal’s algorithm as a greedy method | 250 |
| 20.5. Dijkstra’s algorithm | 251 |
| 20.6. Illustrative problems | 251 |
| Chapter 21. Dynamic Programming | 261 |
| 21.1. Introduction | 261 |
| 21.2. 0/1 knapsack problem | 263 |
| 21.2.1. Dynamic programming-based solution | 264 |
| 21.3. Traveling salesperson problem. | 266 |
| 21.3.1. Dynamic programming-based solution | 267 |
| 21.3.2. Time complexity analysis and applications of traveling salesperson problem | 269 |
| 21.4. All-pairs shortest path problem | 269 |
| 21.4.1. Dynamic programming-based solution | 270 |
| 21.4.2. Time complexity analysis | 272 |
| 21.5. Optimal binary search trees. | 272 |
| 21.5.1. Dynamic programming-based solution | 274 |
| 21.5.2. Construction of the optimal binary search tree | 276 |
| 21.5.3. Time complexity analysis | 279 |
| 21.6. Illustrative problems. | 280 |
| Chapter 22. P and NP Class of Problems | 287 |
| 22.1. Introduction | 287 |
| 22.2. Deterministic and nondeterministic algorithms | 289 |
| 22.3. Satisfiability problem | 292 |
| 22.3.1. Conjunctive normal form and Disjunctive normal form | 294 |
| 22.3.2. Definition of the satisfiability problem | 294 |
| 22.3.3. Construction of CNF and DNF from a logical formula | 295 |
| 22.3.4. Transformation of a CNF into a 3-CNF | 296 |
| 22.3.5. Deterministic algorithm for the satisfiability problem | 297 |
| 22.3.6. Nondeterministic algorithm for the satisfiability problem | 297 |
| 22.4. NP-complete and NP-hard problems | 297 |
| 22.4.1. Definitions | 298 |
| 22.5. Examples of NP-hard and NP-complete problems | 300 |
| 22.6. Cook’s theorem. | 302 |
| 22.7. The unsolved problem $P \stackrel{?}{=} NP$ | 303 |
| 22.8. Illustrative problems. | 304 |

References 311

Index 313

Summaries of other volumes 317