Design and Analysis of Algorithms

Effective: January 2020

1. Basic Tools and Techniques: order notation, induction, recurrence relations, summing sequences, lower bound for comparison-based sorting algorithms. (CLRS: 1–3, 9.1)

2. Divide and Conquer Algorithms: quick sort, insertion sort, heap sort, linear time selection. (CLRS: 4, 6–8)

3. Basic and Advanced Data Structures. (CLRS: 6.5, 11, 12, 18, 19, 21)

4. Greedy Algorithms: minimum spanning trees, graph traversal, shortest path algorithms, rational Knapsack problems. (CLRS: 16, 23)

5. Dynamic Programming and Greedy Algorithms: 0-1 Knapsack problems, shortest paths, optimal binary search trees, matric chain products. (CLRS: 15, 16)

6. Graph Algorithms: breadth-first search, depth-first search, topological sort, strongly connected components, all pair shortest paths, maximum flow and branch & bound. (CLRS: 22, 24–26)

7. NP-Completeness and Approximation Algorithms: P and NP classes, NP-complete problems, Hamiltonian circuit and other NP-complete problems, dealing with NP-complete and NP-hard problems, approximation algorithms. (CLRS: 34, 35)

[CLRS] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein, *Introduction to Algorithms*, Third Edition, MIT Press, 2009.