Algorithms

Introduction. Growth of functions. Asymptotic analysis of upper and expected complexity bounds. Algorithm Analysis, Complexity of an algorithm. Big O notation: Complexity classes: constant, logarithmic, linear, quadratic, and exponential. Empirical measurements of performance. Time and space trade-offs in algorithms. Review of Elementary Data Structures:Stack, Queue, Priority Queues, Heaps, Hash Tables, Sorting: MergeSort, QuickSort, Sets, Sorting lower bound, RadixSort, Selection. Fundamental techniques: The greedy method, Divide-and-conquer. Dynamic Programming: Longest Common Subsequence problem, Maximum Subarray Problem. Graphs. Representations of graphs (e.g., adjacency list, adjacency matrix). Graph algorithms: Depth-first search. Breadth-first search. Directed Graphs, Shortest paths (Dijkstra's and Floyd's algorithms), Minimum Spanning Trees (Prim's and Kruskal's algorithms). Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms). Network flow. NP-Completeness. Approximation algorithms.