DESIGN AND ANALYSIS OF ALGORITHMS

PROF. MADHAVAN MUKUND
Department of Computer Science and Engineering
Chennai Mathematical Institute

INTENDED AUDIENCE: Students in BE/BTech Computer Science, 2nd/3rd year.

PRE-REQUISITES: Exposure to introductory courses on programming and data structures.

INDUSTRY SUPPORT: This course should be of value to any company working in the area of software services and products.

COURSE OUTLINE:

This course will cover basic concepts in the design and analysis of algorithms.Asymptotic complexity, O() notation Sorting and search.Algorithms on graphs: exploration, connectivity, shortest paths, directed acyclic graphs, spanning trees.Design techniques: divide and conquer, greedy, dynamic programming.Data structures: heaps, union of disjoint sets, search trees Intractability.

ABOUT INSTRUCTOR :

Prof. Madhavan Mukund studied at IIT Bombay (BTech) and Aarhus University (PhD). He has been a faculty member at Chennai Mathematical Institute since 1992, where he is presently Professor and Director. His main research area is formal verification. He has active research collaborations within and outside India and serves on international conference programme committees and editorial boards of journals.

COURSE PLAN:

Week 1 Module 1: Introduction | Module 2: Examples and motivation | Module 3: Examples and motivation | Module 4: Asymptotic complexity: informal concepts | Module 5: Asymptotic complexity: formal notation | Module 6: Asymptotic complexity: examples | Assignments MCQ/Fill in blanks (unique answer)

Week 2 Module 1: Searching in list: binary search | Module 2: Sorting: insertion sort | Module 3: Sorting: selection sort | Module 4: Sorting: merge sort | Module 5: Sorting: quicksort | Module 6: Sorting: stability and other issues | Assignments MCQ/Fill in blanks, programming assignment

Week 3 Module 1: Graphs: Motivation | Module 2: Graph exploration: BFS | Module 3: Graph exploration: DFS | Module 4: DFS numbering and applications | Module 5: Directed acyclic graphs | Module 6: Directed acyclic graphs | Assignments MCQ/Fill in blanks, programming assignment

Week 4 Module 1: Shortest paths: unweighted and weighted | Module 2: Single source shortest paths: Dijkstra | Module 3: Single source shortest paths: Dijkstra | Module 4: Minimum cost spanning trees: Prim's algorithm | Module 5: Minimum cost spanning trees: Kruskal's Algorithm | Module 6: Union-Find data structure | Assignments MCQ/Fill in blanks, programming assignment

Week 5 Module 1: Divide and conquer: counting inversions | Module 2: Divide and conquer: nearest pair of points | Module 3: Priority queues, heaps | Module 4: Priority queues, heaps | Module 5: Dijstra/Prims revisited using heaps | Module 6: Search Trees: Introduction | Assignments MCQ/Fill in blanks, programming assignment

Week 6 Module 1: Search Trees: Traversals, insertions, deletions | Module 2: Search Trees: Balancing | Module 3: Greedy : Interval scheduling | Module 4: Greedy : Proof strategies | Module 5: Greedy : Huffman coding | Module 6: Dynamic Programming: weighted interval scheduling | Assignments MCQ/Fill in blanks, programming assignment

Week 7 Module 1: Dynamic Programming: memoization | Module 2: Dynamic Programming: edit distance | Module 3: Dynamic Programming: longest ascending subsequence | Module 4: Dynamic Programming: matrix multiplication | Module 5: Dynamic Programming: shortest paths: Bellman Ford | Module 6: Dynamic Programming: shortest paths: Floyd Warshall | Assignments MCQ/Fill in blanks, programming assignment

Week 8 Module 1: Intractability: NP completeness | Module 2: Intractability: reductions | Module 3: Intractability: examples | Module 4: Intractability: more examples | Module 5: Misc topics | Module 6: Misc topics | Assignments MCQ/Fill in blanks