IV Semester, DSC 1D

Design and Analysis of Algorithms

Unit I

Fundamentals of the Analysis of Algorithm Efficiency: The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes.

Divide-and-Conquer: maximum-subarray problem, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences.

Dynamic Programming: Rod cutting, Matrix-chain multiplication, Elements of dynamic programming, longest common subsequence, Optimal binary search trees.

Greedy Algorithms: An activity-selection problem, Elements of the greedy strategy, Huffman codes, Matroids and greedy methods, task-scheduling problem as a matroid.

Unit II

Searching and Sorting Techniques: Review of elementary sorting techniques-selection sort, Bubble sort, insertion sort, more sorting techniques-quick sort, heap sort, merge sort, shell sort, external sorting.

Limitations of Algorithm: Lower-Bound Arguments, Decision Trees, *P*, *NP*, and *NP*-Complete Problems.

Polynomials and the FFT: Representing polynomials, The DFT and FFT, Efficient FFT implementations.

Number-Theoretic Algorithms: Elementary number-theoretic notions, Greatest common divisor(GCD), Modular arithmetic, Addition and Multiplication of two large numbers.

Unit III

String Matching: The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm.

NP-Completeness: Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs, NP-complete problems.

Approximation Algorithms: The vertex-cover problem, The traveling-salesman problem, The set-covering problem, Randomization and linear programming, The subset-sum problem.

Unit IV

Elementary Graph Algorithms: Representations of graphs, Breadth-first search, Depth-first search, Topological sort, strongly connected components.

Minimum Spanning Trees: Growing a minimum spanning tree, the algorithms of Kruskal and Prim.

Single-Source Shortest Paths: The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths, Proofs of shortest-paths properties.

Text book:

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, MIT press, 3rd edition, 2009.
2. Anany Levitin, Introduction to the design and analysis of algorithms, 3rd edition, 2012.

References:

1. J. Kleinberg and E. Tardos, *Algorithms Design*, Pearson Education, 2006.
2. S. Baase, *Computer Algorithms: Introduction to Design and Analysis*, Addison Wesley, 1999.
3. A.V. Levitin, *Introduction to the Design and Analysis of Algorithms*, Pearson Education, 2006.

Practicals: Design and Analysis of Algorithms

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

Sorting Algorithm:

1. To analyze time complexity of insertion sort
2. To analyze time complexity of Quick sort
3. To analyze time complexity of merge sort

Dynamic Algorithm:

4. To implement largest common subsequence
5. To implement optimal binary search tree
6. To implement martrix chain multiplication

Divide And Conquer:

7. Implement Binary Search Algorithm.
8. Implement Merge Sort Algorithm.
9. Implement Quick Sort Algorithm.
10. To implement strassen's martrix multiplication algorithm

The Greedy Method:

11. Implement activity selection problem
12. Implement fractional Knapsack Problem Algorithm.
13. Implement Job Sequencing with Deadlines Algorithm.

14. Implement Minimum-Cost Spanning Trees: Prim's Algorithm.

15. Implement Single Source Shortest Paths: Dijkstra's Algorithm.

Dynamic Programming:

16. Implement Single-Source Shortest Paths: Bellman-Ford's Algorithm.
17. Implement All-Pairs Shortest Paths: Floyd & Warshall's Algorithm.

Graphs:

18. Implement Dijkstra's algorithm

19. Implement Warshall algorithm

20. Implement Bellman Fords algorthim

21. Implement depth first search algorithm

22. Implement depth first search algorithm

String Matching Algorithm:

23. Implement Naïve string matching algorithm

24. Implement Rabin Karp string matching algorithm

Spanning Trees:

25. Implement prim's algorithm

26. Implement Kruskal's algorithm