Design and Analysis of Algorithms

1.1 Course Number: CS341

1.2 Contact Hours: 3-0-2 Credits: 11

1.3 Semester-offered: 3rd Year-Even

1.4 Prerequisite: Computer Programming, Data Structures.

1.5 Syllabus Committee Member: Dr. Sushum Biswas, Dr. Daya Sagar Gupta & Dr. Gargi Srivastava

2. **Objective:** Upon completion of this course, students will be able to do the following:

- Analyze the asymptotic performance of algorithms.
- Demonstrate a familiarity with major algorithms.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

3. **Course Content:**

Unit-wise distribution of content and number of lectures

Unit	Topics	Sub-topic	Lectures
1	Introduction, analysis of algorithms	Example: air travel, xerox shop, document similarity; introduction and motivation; input size, worst case, average case; quantifying efficiency: O(). Omega(), Theta(); Examples: analysis of iterative and recursive algorithms	5
2	Searching and sorting	Arrays and lists; searching in an array; selection sort; insertion sort; merge sort; merge sort - analysis; quicksort; quicksort - analysis; sorting - concluding remarks	5
3	Graphs	Introduction to graphs; representing graphs; breadth first search; depth first search; applications of BFS and DFS; directed acyclic graphs: topological sort, longest paths	5
4	Weighted graphs	Single source shortest paths: Dijkstra's algorithm; analysis; negative edge weights: Bellman-Ford algorithm; all pairs shortest paths; minimum cost spanning trees; Prims algorithm; Kruskal's algorithm	5
5	Data structures: union-find and heaps, divide and conquer	Union-find using arrays; union-find using pointers; priority queues; heaps - updating values, sorting; counting inversions; closest pair of points	5

6	Data Structures: Search trees, greedy algorithms	Binary search trees; balanced search trees; interval scheduling; scheduling with deadlines: minimizing lateness; Huffman codes	5
7	Dynamic Programming	Introduction to dynamic programming; memoization; grid paths; common subwords and subsequences; edit distance; matrix multiplication	5
8	Linear Programming and Network flows, intractability	Linear programming; LP modelling: production planning, bandwidth allocation; network flows; reductions; checking algorithms; P and NP	5
		Total	**40**

4. Readings

4.1 Textbook: Introduction to algorithms by Coreman, PHI, IIIrd Edition.

4.2 Reference books:

1. Computer Algorithms by Hoeowitz, Sahanai, and Rajashekaran
2. Data Structures and Algorithm Analysis in C, by Mark Allen Weiss, 2nd edition, 1997, Addison-Wesley, ISBN 0-201-49840-5, Kleinberg and Tardos, Algorithm Design, 2005.

5 **Outcome of the Course:** Students who complete the course will have demonstrated the ability to do the following: Analyze worst-case running times of algorithms using asymptotic analysis. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them. Explain the major graph algorithms and their analyses.