SEMESTER- IV	
Name of the course: **Design and Analysis of Algorithms**	
Course code: **UGCMSMAC08**	
Total Class Hours: **120**	Credit: **4+2 (Theory & Lab)**

Course Objectives:

1. Understating of design aspects of different computational algorithms.
2. To gain the capability of estimating cost of devising an algorithm.
3. Capability to apply the knowledge of algorithm design for practical problem solving.

SYLLABUS

A Theory (60 Hours) 4 Credits

1. **Introduction** (7
   L)
   Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.
2. **Algorithm Design Techniques** (7
   L) Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.
3. **Sorting and Searching Techniques** (20
   L) Asymptotic notations and complexity analysis- substitution and recurrence tree. Elementary sorting techniques– Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics.
4. **Lower Bounding Techniques** (3
   L) Decision Trees
5. **Balanced Trees** (7
   L)
   Red-Black Trees
6. **Advanced Analysis Technique** (3
   L) Amortized analysis
7. **Graphs** (8

L)

Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees- Kruskal’s, Prim’s.

8. **String Processing** (5 L)

String Matching, KMP Technique

B Practical (60 Hours) 2 Credits

1. Implement Insertion Sort (The program should report the number of comparisons)
2. Implement Merge Sort (The program should report the number of comparisons)
3. Implement Heap Sort (The program should report the number of comparisons)
4. Implement Randomized Quick sort (The program should report the number of comparisons)
5. Implement Radix Sort
6. Create a Red-Black Tree and perform following operations on it:
7. Insert a node
8. Delete a node
9. Search for a number & also report the color of the node containing this number.
10. Write a program to determine the LCS of two given sequences
11. Implement Breadth-First Search in a graph
12. Implement Depth-First Search in a graph
13. Write a program to determine the minimum spanning tree of a graph
14. For the algorithms at S. No. 1 to 4 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.

Course Outcomes:

CO No.	**Course Outcomes**	**Cognitive Level**	**PO Addressed**	**PSO Addressed**
CO1	Ability to define how to analyse algorithms and estimate their worst-case and average-case behaviour.	R(1)	PO1	PSO1
CO2	Ability to understand good principles of algorithm design.	U(2)	PO2	PSO1 PSO2
CO3	Ability to analyse and be accustomed to the description of algorithms in both functional and procedural styles.	An(4)	PO4	PSO4

CO4	Ability to apply their theoretical knowledge in practice and design algorithms for problem solving.	Ap(3), C(6)	PO6	PSO6

R= remembering, U = understanding, Ap = applying, An = analysing, E = evaluating, and C = creating

Reference Books

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sarabasse & A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999