Faculty of Computer Science & Applications
Master of Computer Application (MCA)
(W. E. F.: 2023-24)

Document ID: SUTEFCAM-01

Name of Faculty	:	Faculty of Computer Science & Applications
Name of Program	:	Master of Computer Application (MCA)
Course Code	:	2MCA01
Course Title	:	Design and Analysis of Algorithms
Type of Course	:	Professional Couse
Year of Introduction	:	2023-24

Prerequisite	:	Programming (C or C++), Data and file structure
Course Objective	:	Obtaining efficient algorithms is very important in modern computer engineering as the world wants applications to be time and space and energy efficient. This course enables to understand and analyse efficient algorithms for various applications.
Course Outcomes	:	At the end of this course, students will be able to:
	CO 1	Analyze the asymptotic performance of algorithms.
	CO 2	Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
	CO 3	Find optimal solution by applying various methods.
	CO 4	Apply pattern matching algorithms to find particular pattern.
	CO 5	Differentiate polynomial and nonpolynomial problems
	CO 6	Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.

Teaching and Examination Scheme

Teaching Scheme (Contact Hours)			Credits	Examination Marks				
				Theory Marks		Practical Marks		Total Marks
L	**T**	**P**	**C**	**SEE**	**CIA**	**SEE**	**CIA**	
2	0	2	3	70	30	30	20	150

*Legends: **L**-Lecture; **T**–Tutorial/Teacher Guided Theory Practice; **P**–Practical, **C** – Credit, **SEE** – Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.))*

Course Content

Unit No,	Topics	Teaching Hrs.	Weightage	Mapping with CO
1	**Basics of Algorithms and Mathematics:** What is an algorithm?, Mathematics for Algorithmic Sets, Functions and Relations, Vectors and Matrices, Linear Inequalities and Linear Equations.	**02**	**2%**	**CO1**

2	**Analysis of Algorithm:** The efficient algorithm, Average, Best and worst case analysis, Amortized analysis , Asymptotic Notations, Analyzing control statement, Loop invariant and the correctness of the algorithm, Sorting Algorithms and analysis: Bubble sort, Selection sort, Insertion sort, Shell sort Heap sort, Sorting in linear time : Bucket sort, Radix sort and Counting sort	**05**	**10%**	**CO1**
3	**Divide and Conquer Algorithm:** Introduction, Recurrence and different methods to solve recurrence, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential.	**04**	**15%**	**CO2**
4	**Dynamic Programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence.	**04**	**20%**	**CO2**
5	**Greedy Algorithm:** General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm Activity selection problem, Elements of Greedy Strategy, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem, Job Scheduling Problem, Huffman code.	**03**	**20%**	**CO2** **CO3**
6	**Exploring Graphs:** An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort, Connected components,	**03**	**10%**	**CO3** **CO4**
7	**Backtracking and Branch and Bound:** Introduction, The Eight queens problem , Knapsack problem, Travelling Salesman problem, Minimax principle	**03**	**5%**	**CO5**
8	**String Matching:** Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm.	**03**	**8%**	**CO5**

9	**Introduction to NP-Completeness:** The class P and NP, Polynomial reduction, NP-Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Hamiltonian problem, Approximation algorithms	**03**	**10%**	**CO6**

Suggested Distribution of Theory Marks Using Bloom's Taxonomy						
Level	Remembrance	Understanding	Application	Analyse	Evaluate	Create
Weightage	**40%**	**40%**	**20%**	**-**	**-**	**-**

NOTE: This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Suggested List of Experiments/Tutorials

Sr. No.	Name of Experiment/Tutorials	Teaching Hours
1.	Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort	02
2.	Implementation and Time analysis of linear and binary search algorithm.	02
3.	Implementation of max-heap sort algorithm	02
4.	Implementation and Time analysis of factorial program using iterative and recursive method	02
5.	Implementation of a knapsack problem using dynamic programming.	02
6.	Implementation of chain matrix multiplication using dynamic programming.	02
7.	Implementation of making a change problem using dynamic programming	02
8.	Implementation of a knapsack problem using greedy algorithm	02
9.	Implementation of Graph and Searching (DFS and BFS).	02
10.	Implement prim's algorithm	02
11.	Implement kruskal's algorithm.	02
12.	Implement LCS problem.	02
13.	Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort	02
14.	Implementation and Time analysis of linear and binary search algorithm.	02
15.	Implementation of max-heap sort algorithm	02

Major Equipment/ Instruments and Software Required

Sr. No.	Name of Major Equipment/ Instruments and Software
1	TURBO C++ or Any other C IDE

Reference Books

Sr. No.	Name of Reference Books
1	Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
2	Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
3	Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.
4	Foundations of Algorithms, Shailesh R Sathe, Penram
5	Design and Analysis of Algorithms, Dave and Dave, Pearson.