Course Syllabus
Gyanmanjari College of Computer Application
Semester-3 (MCA)

Subject : Analysis and Design of Algorithm - MCAXX13516

Type of course: Major Core

Prerequisite: Data Structure, C Programming Language.

Rationale:

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 analyze efficient algorithms for various applications.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
				Theory Marks		Practical Marks		CA	
CI	T	P	C	ESE	MSE	V	P	ALA	
3	0	2	4	60	30	10	20	30	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; MSE- Mid Semester Examination; LWA - Lab Work Assessment; V – Viva voce; CCE- Continuous and Comprehensive Evaluation; ALA- Active Learning Activities.

Course Content:

Sr. No	Course content	Hrs	% Weightage
1	**Basic Concepts of Analysis and Design of Algorithms in Computing and Growth of Functions:** Algorithms, Algorithms as a technology, analyzing algorithms, Designing algorithms, Asymptotic notation, Standard notations and common functions.	5	10%
2	**Algorithms Using Divide-and-Conquer**: The 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, Max-Min problem, Merge Sort.	14	25%

3	**Dynamic Programming**: Rod cutting, Matrix-chain multiplication, Elements of dynamic programming, longest common subsequence, Optimal binary search trees.	5	15%
4	**Greedy Algorithms and Amortized Analysis**: An activity-selection problem, Elements of the greedy strategy, The Knapsack Problem, Huffman codes. Aggregate analysis, The accounting method, The potential method and Dynamic tables.	10	25%
5	**Graphs, Minimum Spanning Trees and Single-Source Shortest Paths:** Graphs, traversing Graphs, Depth First Search, Breath First Search, growing a minimum spanning tree, the algorithms of Kruskal and Prim, The Bellman-Ford algorithm, Single source shortest paths in directed acyclic graphs, Dijkstra's algorithm.	11	25%

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**Analysis of Real-life Algorithm application:** Students have to identify the usage of various algorithms in real life scenarios and they have to create a documentation of it, after that they have to upload that on GMIU Web portal.	10
2	**Visualizations:** Students have to create a visualization for asymptotic notations. they can use visual aids or online tools to visualize the differences in time complexity represented by different asymptotic notations. after that they have to upload it on GMIU Web portal.	10
3	**Deep Analysis :** Students have to create an analysis table for the Best case, Average case and Worst case analysis for minimum 10 algorithms from the syllabus and they have to upload analysis document on GMIU Web portal.	10
Total		30

Suggested Specification table with Marks (Theory):60

Distribution of Theory Marks **(Revised Bloom's Taxonomy)**						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage	35%	30%	20%	15%	-	-

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

Course Outcome:

After learning the course the students should be able to:	
CO1	Analyze algorithms by working out for time and space complexity of algorithms, design algorithm and growth of functions.
CO2	Identify algorithm design methodology to solve problems. Analyze divide-and conquer.
CO3	Understand dynamic programming, its elements and analyze existing algorithm of dynamic programming.
CO4	Understand concept of greedy algorithms and amortized analysis.
CO5	Apply knowledge of graphs, minimum spanning trees and design algorithm for shortest paths.

List of Practical

Sr. No	Descriptions	Hrs
1	Determine smallest divisor of an integer.	2
2	For a given value of n, generate prime numbers <= n (more than one algorithms are possible)	2
3	Determine product of 2 integers (a * b) as repeated sums. Iterative and recursive algorithms are possible.	2
4	Find Factorial of n. Iterative and recursive algorithms are possible.	2
5	Generate Fibonacci series up to n terms Iterative and recursive algorithms are possible.	2
6	Program for finding maximum and minimum number using Divide and conquer.	2
7	Breadth First Search (BFS) in a binary tree.	2
8	Depth First Search (DFS) in a binary tree.	2
9	Binary Search of an ordered array. Iterative and Recursive algorithms are possible.	2
10	Sort a given sequence of numbers using Bubble Sort Algorithm.	2
11	Sort a given sequence of numbers using Merge Sort Algorithm.	2
12	Solution of Rod-cutting problem using Dynamic Programming algorithm.	2
13	Prim's algorithm to find minimum cost tree (shortest path in a tree).	2
14	Kruskal's algorithm to find minimum cost tree (shortest path in a tree).	2
15	Implement Bellman-Ford Single Source Shortest Path Algorithm.	2
		30

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 10% topics are suggested for flipped mode instruction.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory

The internal evaluation will be done on the basis of Active Learning Assignment

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

[1] Introduction to Algorithms, Third Edition, MIT Press (2009), Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein
[2] Design and Analysis of Algorithms, Pearson (2014), Parag H Dave, Himanshu B Dave,
[3] Introduction to Algorithms, 2 nd Edition, PHI, Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein
[4] Introduction to Design and Analysis of Algorithms, Pearson (2014), Anany Levitin