Gyanmanjari Innovative University

Course Syllabus
Gyanmanjari College of Computer Application
Semester-5(BCA)

Subject: Design and Analysis of Algorithm – BCAXX15321

Type of course: Major Core

Prerequisite: Basic Programming Language and basics of Data Structure

Rationale:

In field of computer science, obtaining efficient algorithms is crucial as the demand for faster, scalable, and resource-efficient applications continue to grow. This course is designed to provide a deep understanding of algorithm design and analysis, enabling learners to develop optimal solutions for real-world computational problems. The course also focuses on practical problem-solving through real-world applications.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
				SEE		CCE			
CI	T	P	C	Theory	Practical	MSE	LWA	ALA	
3	0	2	4	75	25	30	20	50	200

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.

3 Credits * 25 Marks = 75 Marks (each credit carries 25 Marks) Theory
1 Credits * 25 Marks = 25 Marks (each credit carries 25 Marks) Practical
SEE 100 Marks will be converted in to 50 Marks
CCE 100 Marks will be converted in to 50 Marks
It is compulsory to pass in each individual component.

Course Content:

Sr. No	Course content	Hrs	% Weightage
1	**Algorithms Basics and Analysis:** Basics of Algorithms, Algorithm Specification, Analysis of Algorithms: Analyzing control Statement, Loop invariants, Asymptotic Notations, Amortized Analysis, Recurrence	6	15%

	equations and methods: substitution method and master method.		
2	**Searching and Sorting:** Searching: Linear search and binary search 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. **Graphs:** An introduction using graphs and games, undirected graphs, directed graphs, traversing Graphs, depth first search, breadth first search, topological sort, connected components.	10	25%
3	**Divide and conquer algorithm:** Introduction, multiplying large integer problem, problem solving using divide and conquer algorithm – binary search, Max-min problem, sorting (Merge sort, Quick Sort), strassen's matrix multiplication, Exponential.	8	15%
4	**Greedy Algorithm:** Characteristics and Problem-Solving Approaches - Activity selection problem, The Knapsack Problem, Job Scheduling Problem, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, Huffman coding. **Dynamic Programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Assembly Line-Scheduling, Knapsack problem, Matrix chain multiplication, LCS, All pair shortest path	12	25%
5	**Backtracking and Branch and Bound:** Introduction, The Eight queens' problem, Knapsack problem, Travelling Salesman problem, Minimax principle **String Matching:** Introduction, The naive string-matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm.	11	20%

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**Review Paper on Algorithmic Topics:** Writing a review paper on selected topic from content with latest research issues in that topic. Submit it in PDF format on the GMIU web portal.	10
2	**Identify a real-time application and determine the algorithm used in it:** Identify a real-time application and determine the algorithm it uses for its core functionality. After selecting the application, upload the document on the GMIU web portal.	10
3	**Strategy Showdown - Comparing Algorithm Design Techniques:** Each group (max of 4 students) is required to conduct in-depth research on an assigned algorithm design strategy and prepare a detailed analysis	10

	covering the following aspects: Definition, Working Principle, Time Complexity, Space Complexity, Real-World Applications, Advantages, Limitations, and Comparison with other strategies. Upon completion, compile the analysis into a PDF file and upload it on the GMIU web portal.	
4	**Real-Life Graph Conversion:** Students have to choose a real-world system (like social networks, road maps, or organizational hierarchy) and represent it as a graph. Demonstrate DFS, BFS, or Topological Sort on it. Submit the diagram and traversal results on GMIU web portal.	10
5	**Design a Huffman Code for Custom Text:** Students may take any paragraph (from a news article, your textbook, or a story). They have to create a Huffman encoding tree, assign binary codes, and calculate compression efficiency. They have to submit full process and results on the GMIU web portal.	10
Total		50

Suggested Specification table with Marks (Theory):75

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage	15%	35%	15%	15%	13%	7%

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 algorithm efficiency using asymptotic notations, recurrence relations, and correctness criteria.
CO2	Construct and evaluate sorting and searching algorithms using divide and conquer strategies.
CO3	Implement graph traversal for complex inputs.
CO4	Design and analyze solutions using greedy and dynamic programming techniques for optimization problems.
CO5	Assess algorithmic solutions for constraint-based problems using backtracking and branch-and-bound approaches.

List of Practical

Sr. No	Descriptions	Unit No	Hrs
1	Implementation and Time analysis of factorial program using iterative and recursive method.	1	2
2	Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort.	2	6
3	Implementation and Time analysis of linear and binary search algorithm.	2	2
4	Implementation of Graph and Depth first search. (DFS)	3	2
5	Implementation of Graph and Breadth first search. (BFS)	3	2
6	Implement prim's algorithm.	3	2
7	Implement kruskal's algorithm.	3	2
8	Implement a program for String Matching using Knuth-Morris-Pratt Algorithm on a text file content.	4	4
9	Implement LCS problem.	4	2
10	Implementation of a knapsack problem using dynamic programming.	4	2
11	Implementation of a knapsack problem using greedy algorithm.	4	2
12	Implement an algorithm based on the Minimax principle.	5	2
		Total	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 etc.

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] Design and Analysis of Algorithms, Dave and Dave, Pearson
[2] Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
[3] Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
[4] Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.
[5] Foundations of Algorithms, Shailesh R Sathe, Penram.