CVM UNIVERSITY
Aegis: Charutar Vidya Mandal (Estd.1945)

FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2022-23

Programme: **Bachelor of Technology (Computer Engineering)**

Semester: **V**

Course Code: **202045601**

Course Title: **Design and Analysis of Algorithms**

Course Group: **Professional Core Course**

Course Objectives: This course provides the fundamental knowledge to design and analyse the algorithms. Different algorithm paradigms will be explored. Students will learn how to measure performance of various algorithms.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
				Theory		J/V/P*		Total
Lecture	Tutorial	Practical		Internal	External	Internal	External	
4	0	2	5	50 / 18	50 / 17	25 / 9	25 / 9	150 / 53

* **J**: Jury; **V**: Viva; **P**: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	**Basics of Algorithms and Mathematics:** Definition of Algorithm, Importance of design and analysis of algorithms, Mathematics for Algorithmic Sets, Functions and Relations, Quantifiers, Vectors and Matrices, simple series, basic combinations. **Analysis of Algorithm:** Time complexity, Space complexity, Analysis: average, best and worst case, Asymptotic notations, Limit rules, Conditional asymptotic notations, Analyzing generalize algorithm with control structures: "for", "while" and "repeat" loops. Amortized analysis.	06
2	**Methods to Solve Recurrence:** Substitution, homogeneous Recurrences, Inhomogeneous Recurrences, Change of Variable, Master Theorem, Range Transformations and Recursion Tree. **Sorting Algorithms with analysis:** Bubble sort, Selection sort, Insertion sort, Heap sort. **Sorting in linear time:** Bucket sort and Counting sort.	11
3	**Divide and Conquer Algorithms:** Introduction, multiplying large integers problem, Problem solving using divide conquer algorithm - Binary search, Merge sort and Quick sort algorithms with analysis, Max-Min problem, Matrix multiplication, Exponential.	06

4	**Greedy Algorithms:** General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm- Making change problem, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Single Source Shortest paths (Dijkstra's algorithm, The Bellman-Ford algorithm), The Knapsack Problem, Job Scheduling Problem, Huffman code.	**07**
5	**Dynamic Programming:** Introduction, Comparison with Greedy algorithm and divide & conquer algorithm, Problem solving using dynamic programming – Calculating the binomial coefficient, The principle of optimality, Making change problem, The knapsack problem, All points shortest path (Floyd's algorithm), Chained matrix multiplication, longest common subsequence.	**08**
6	**Exploring Graphs:** Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search. **Backtracking:** Introduction, The Eight queen's problem, The knapsack problem. **Branch and Bound:** The assignment problem, The knapsack problem. Minimax principle.	**07**
7	**String Matching:** Introduction, The naive string-matching algorithm, The Rabin-Karp algorithm, The Knuth-Morris-Pratt algorithm.	**04**
8	**Introduction to NP-Completeness:** The class P and NP, Polynomial reduction, NP-Completeness Problem, NP-Hard Problems, Travelling Salesman problem, Hamiltonian problem.	**03**
	Total	**52**

List of Practicals / Tutorials:

1	Write a program to sort given elements of an array in ascending order using bubble sort. Analyze the time complexity for best, average and worst case.
2	Write a program to sort given elements of an array in ascending order using selection sort. Analyze the time complexity for best, average and worst case.
3	Write a program to implement heap sort.
4	Write a program to search given element from an array using sequential search and binary search. Analyze the time complexity for best, average and worst case.
5	Write a program to sort given elements of an array in ascending order using merge sort. Analyze the time complexity for best, average and worst case.
6	Write a program to sort given elements of an array in ascending order using quick sort. Analyze the time complexity for best, average and worst case.
7	Write a program to implement making change problem using greedy algorithm.
8	Write a program to implement the knapsack problem using greedy algorithm.
9	Write a program to implement making change problem using dynamic programming.
10	Write a program to implement the knapsack problem using dynamic programming.
11	Write a program to implement Floyd's algorithm for finding shortest path using dynamic programming.
12	Write a program to implement chained matrix multiplication using dynamic programming.
13	Write a program to implement longest common subsequence using dynamic programming.

Reference Books:

1	Fundamental of Algorithmics by Gills Brassard and Paul Bratley, PHI.
2	Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.

3	Fundamentals of Computer Algorithms by Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekharan, Galgotia.
4	Design and Analysis of Algorithms by Dave and Dave, Pearson.

Supplementary learning Material:	
1	Lecture Notes
2	NPTEL - Swayam Courses

Pedagogy:

- Direct classroom teaching
- Audio Visual presentations/demonstrations
- Assignments/Quiz
- Continuous assessment
- Interactive methods
- Seminar/Poster Presentation
- Industrial/ Field visits
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						**R**: Remembering; **U**: Understanding;
						A: Applying; **N**: Analyzing; **E**: Evaluating;
						C: Creating

R	**U**	**A**	**N**	**E**	**C**	
10%	30%	10%	20%	20%	10%	

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 Outcomes (CO):

Sr.	**Course Outcome Statements**	**%weightage**
CO-1	To study the asymptotic performance of algorithms.	**20**
CO-2	Apply various complexity measures and find out performance of the algorithm through divide and conquer like searching and sorting.	**30**
CO-3	To generate optimal solutions by applying various Greedy and Dynamic algorithms.	**30**
CO-4	To apply fundamental algorithms to model engineering problem solving using various graph methods or using suitable data structures.	**20**

Curriculum Revision:	
Version:	2.0
Drafted on (Month-Year):	June-2022
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	June-2025