RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B.Tech) DEGREE COURSE
SEMESTER: V (C.B.C.S.)
BRANCH: COMPUTER SCIENCE AND ENGINEERING

Fifth Semester:-

S. N.	Subject	Teaching Scheme			Evaluation Scheme			Credits	Category
		L	T	P	CA	UE	Total		
1	Artificial Intelligence	3	1	-	30	70	100	4	PCC-CS
2	Artificial I Intelligence-Lab	-	-	2	25	25	50	1	PCC-CS
3	Design & Analysis of Algorithms	3	1	-	30	70	100	4	PCC-CS
4	Design & Analysis of Algorithms –Lab	-	-	2	25	25	50	1	PCC-CS
	Software Engineering & Project Management	3	-	-	30	70	100	3	PCC-CS
5	Elective-I	3	-	-	30	70	100	3	PEC-CS
6	Effective Technical Communication	2	-	-	15	35	50	2	HSMC
7	Profesional Skills Lab I			2	25	25	50	1	ESC
8	Yoga and Meditation (Audit Course)	2	-	-	50	-	-	Audit	MC
	Total	**16**	02	**06**			**600**	**19**	

Elective-I: 1. TCP/IP 2. Design Patterns 3. Data Warehousing and Mining

[Mrs. B. P. Dharaskar]

[Ms. Mona Nilchandani]

Dr. S. V. Sonekar
chairman

RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B.Tech.) DEGREE COURSE
SEMESTER: V (C.B.C.S.)
BRANCH: COMPUTER SCIENCE AND ENGINEERING

Subject: **Design and Analysis of Algorithms** Subject Code: **BTECH_CSE-502T**

Load [Th+Tu]	Credits [Th+Tu]	College Assessment Marks	University Evaluation	Total Marks
[36 + 12]=48 Hrs	**3+1=4**	**30**	**70**	**100**

Course Objectives:

1	Analyze the asymptotic performance of algorithm
2	Apply important algorithmic design paradigms and methods of analysis
3	Solve simple to moderately difficult algorithmic problems arising in applications.
4	Able to demonstrate the hardness of simple NP-complete problems

Course Outcome:

At the end of this course students are able to:

CO1	Illustrate different approaches for analysis and design of efficient algorithms and Analyze performance of various algorithms using asymptotic notations.
CO2	Determine and Apply various divide & conquer strategies and greedy approaches for solving a given computational problem
CO3	Demonstrate and Solve various realtime problems using the concepts of dynamic programming
CO4	Make use of backtracking and graph traversal techniques for solving real-world problems
CO5	Recall and Classify the NP-hard and NP-complete problems

SYLLABUS:

UNIT-I

Definition of algorithms and brief explanation about the basic properties of algorithms Recurrence relations, solutions of recurrence relations using technique of characteristic equation, master theorem ,Asymptotic notations of analysis of algorithms, worst case, average case and best case analysis of insertion sort, selection sort and bubble sort, amortized analysis, application of amortized analysis, Biotonic sorting network.

UNIT-II

Divide and conquer strategies: Binary search, quick sort, merge sort, heap sort, Stressen's matrix multiplication algorithm, min-max algorithm.

Greedy Approach: Application to job sequencing with deadlines problem, knapsack problem, optimal merge pattern, Huffman code.

UNIT-III

Dynamic Programming: Basic Strategy, Multistage graph (forward and backward approach), Longest Common Subsequence, matrix chain multiplication, Optimal Binary Search Tree, 0/1 Knapsack problems, Travelling Salesman problem, single source shortest path using Bellman-Ford algorithm, all pair shortest path using Floyd- Warshall algorithm.

UNIT-IV

Basic Traversal and Search Techniques: Breadth first search and depth first search, connected components.

Backtracking: Basic strategy, N-Queen Problem and their Analysis (4 & 8-Queen), graph coloring, Hamiltonian cycles .

UNIT-V

NP-Hard and NP-complete problems, basic concepts, non-deterministic algorithms, NP Hard and
NP-complete, Cook's theorem, decision and optimization problems, graph based problems on NP Principle.

Text Books:-

1. "Introduction to Algorithms", Thirs Edition, Prentice Hall of India by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
2. The Design and Analysis of Computer Algorithms", Pearson education by Alfred V. Aho, John E. Hopcraft, Jeffrey D. Ullman.
3. "Fundamentals of Computer Algorithms", Second Edition, University Press By Horowitz, Sahani, Rajsekharam
4. Fundamentals of Algorithms", Prentice Hall by Brassard, Bratley
5. "Design and Analysis of Algorithms",Pearson Education, II nd Edition, Parag Dave, Himanshu Dave

Reference Books:

1. Computer Algorithms: Introduction to Design and analysis, 3rd Edition, By Sara Baase and Gelder Pearson Education.

RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B.Tech.) DEGREE COURSE
SEMESTER: V (C.B.C.S.)
BRANCH: COMPUTER SCIENCE AND ENGINEERING

Subject: **Design and Analysis of Algorithms LAB** Subject Code: **BTECH_CSE-502P**

Load	Credits	College Assessment Marks	University Evaluation	Total Marks
2Hrs /Week (Practical)	1	25	25	50

Course Objectives:

1	To learn the importance of designing an algorithm in an effective way by considering space and time complexity
2	To learn graph search algorithms.
3	To study network flow and linear programming problems
4	To learn the dynamic programming design techniques.
5	To develop recursive backtracking algorithms.

Course Outcome:

At the end of this course students will be able to:

CO1	Calculate the time complexity of algorithm.
CO2	Sort the given numbers using various sorting algorithms.
CO3	Develop programs for the problems using Divide and Conquer and greedy methods.
CO4	Develop programs for the problems using Dynamic programming.
CO5	Students will be able to write programs for the problems using Backtracking.

Expected experiments to be performed (Not limited to):

1. To find Time complexity of an algorithm.
2. To find Space complexity of an algorithm.
3. To find HCF and LCM of two numbers
4. Code and analyses to find median element in an array of integers.
5. Code and analyse to find majority element in an array of integers.
6. Code and analyse to sort an array of integers using merge sort
7. Code and analyse to sort an array of integers using quick sort
8. To implement maximum and minimum problem using divide and conquer strategy
9. To implement binary search using divide and conquer strategy
10. To implement program of Heap Sort.
11. WAP of minimum spanning tree using Kruskal algorithm.
12. WAP of minimum spanning tree using Prim's algorithm.
13. WAP to implement matrix chain multiplication
14. Code to find the shortest path in graph using Dijkstra's algorithm.
15. Code to find the shortest path using Bellman-Ford algorithm.
16. To implement LCS problem using Dynamic Programming.
17. To implement matrix chain multiplication problem using dynamic programming.
18. Code and analyze to find the minimum spanning tree in a weighted, undirected graph.
19. Code and analyze to find all occurrences of a pattern P in a given string S.
20. Code and analyze to do a depth-first search (DFS) on an undirected graph. Implementing an application of DFS such as:
    (i) to find the topological sort of a directed acyclic graph.
    (ii) to find a path from source to goal in a maze.
21. Code and analyze to do a breadth-first search (BFS) on an undirected graph. Implementing an application of BFS such as
    (i) to find connected components of an undirected graph.
    (ii) to check whether a given graph is bipartite.