Vivekanand Education Society's Institute of Technology
Department of Computer Engineering
(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Since 1962

COURSE NAME: DESIGN AND ANALYSIS OF ALGORITHMS

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NCMPC 41	Design and Analysis of Algorithms (Theory)	03	---	---	02	---	---	02
NCMPC L41	Design and Analysis of Algorithms (Lab)	---	02	---	---	01	---	01

Design And Analysis Of Algorithms (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NCMPC 41	Design and Analysis of Algorithms (Theory)	03	-	-	02	-	-	02

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPC 41	Design and Analysis of Algorithms (Theory)	20	20	60	02	-	-	100

Course Prerequisite: Data Structures, Discrete Structures & Graph Theory	
Course Objectives	
1	To provide mathematical approaches for the Analysis of Algorithms
2	To understand and solve problems using various algorithmic approaches
3	To analyze algorithms using various methods
4	To understand and solve string-matching algorithms

Course Outcomes	
1	Analyze the running time and space complexity of algorithms and describe P and NP Algorithms.
2	Describe, apply, and analyze the complexity of the Divide and Conquer strategy.
3	Describe, apply, and analyze the complexity of the Greedy strategy.
4	Describe, apply, and analyze the complexity of the Dynamic Programming strategy.
5	Explain and apply Backtracking, Branch and Bound.
6	Explain and apply string-matching techniques.

Module	**Detailed Contents**		**Hours**
1	**Introduction to Design and Analysis of Algorithms**		10
	1.1	Performance analysis, space, and time complexity Growth of function, Big-Oh, Omega Theta notation Mathematical background for algorithm analysis; Analysis of selection sort, insertion sort.	
	1.2	Recurrences: The substitution method, Recursion tree method, Master method	
	1.3	Complexity Classes: Definition of P, NP, NP-Hard, NP-Complete	
2	**Divide and Conquer Strategy**		06
	2.1	General method, Min-Max Algorithm, Merge sort, Quick sort, Analysis of Binary search, Strassen's Matrix Multiplication.	
3	**Greedy Method Approach**		06
	3.1 -	General Method, Single source shortest path: Dijkstra Algorithm Fractional Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees: Kruskal and Prim's algorithms	
4	**Dynamic Programming Approach**		09
	4.1	General Method, Multistage graphs, Single source shortest path: Bellman Ford Algorithm, All pair shortest path: Floyd Warshall Algorithm, Matrix Chain Multiplication, Longest common subsequence, Optimal Binary Search Trees, 0/1 knapsack Problem.	
5	**Backtracking and Branch and bound**		05
	5.1	Backtracking: N-queen problem, Sum of subsets, Graph coloring	
	5.2	Branch and Bound: 15 Puzzle problem, Traveling Salesperson problem.	
6	**String Matching Algorithms**		03
	6.1	Naïve string-matching algorithm, Rabin Karp algorithm, Knuth-Morris-Pratt algorithm	

Module	Detailed Contents		Hours
1	**Introduction to Design and Analysis of Algorithms**		10
	1.1	Performance analysis, space, and time complexity Growth of function, Big-Oh, Omega Theta notation Mathematical background for algorithm analysis; Analysis of selection sort, insertion sort.	
	1.2	Recurrences: The substitution method, Recursion tree method, Master method	
	1.3	Complexity Classes: Definition of P, NP, NP-Hard, NP-Complete	
2	**Divide and Conquer Strategy**		06
	2.1	General method, Min-Max Algorithm, Merge sort, Quick sort, Analysis of Binary search, Strassen's Matrix Multiplication.	
3	**Greedy Method Approach**		06
	3.1 -	General Method, Single source shortest path: Dijkstra Algorithm Fractional Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees: Kruskal and Prim's algorithms	
4	**Dynamic Programming Approach**		09
	4.1	General Method, Multistage graphs, Single source shortest path: Bellman Ford Algorithm, All pair shortest path: Floyd Warshall Algorithm, Matrix Chain Multiplication, Longest common subsequence, Optimal Binary Search Trees, 0/1 knapsack Problem.	
5	**Backtracking and Branch and bound**		05
	5.1	Backtracking: N-queen problem, Sum of subsets, Graph coloring	
	5.2	Branch and Bound: 15 Puzzle problem, Traveling Salesperson problem.	
		Total	**39**

Textbooks	
1	T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms", 2nd Edition, PHI Publication 2005.
2	Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms',' University Press.
References	
1	Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw Hill Edition.
2	S. K. Basu, "Design Methods and Analysis of Algorithm", PHI.
3	J. Kleinberg and E. Tardos, Algorithm Design, Pearson International Edition, 2005.

Useful Links	
Resources	
1	https://nptel.ac.in/courses/106/106/106106131/
2	https://swayam.gov.in/nd1_noc19_cs47/preview
3	https://www.coursera.org/specializations/algorithms
4	https://www.mooc-list.com/tags/algorithms
AI Tools	
5	Algorithmia: https://algorithmia.com/
6	TensorFlow: https://www.tensorflow.org/
7	VisuAlgo: https://visualgo.net/
8	Algorithm Visualizer: https://algorithm-visualizer.org/
9	Pathfinding Visualizer: https://bengavrilov.github.io/Path-Finding-Visualizer/
Industry articles	
10	Artificial intelligence (AI) algorithms: a complete overview : https://www.tableau.com/data-insights/ai/algorithms
11	What Is an Algorithm? http://bit.ly/3RndUg6
12	Algorithmic bias detection and mitigation: Best practices and policies to reduce consumer harms https://bit.ly/4b1Rw31
13	Code-Dependent: Pros and Cons of the Algorithm Age : https://pewrsr.ch/3Ro3P2H
Case Studies	
14	A Case Study in Algorithm Analysis https://ics.uci.edu/~goodrich/teach/cs161/notes/MaxSubarray.pdf
15	An Introduction to the Analysis of Algorithms https://sedgewick.io/books/analysis-of-algorithms/
16	Parallel MCMC Algorithms: Theoretical Foundations, Algorithm Design, Case Studies https://ar5iv.org/abs/2209.04750

Internal Assessment		
Assessment consists of one Mid Term Test of 20 marks and Continuous Assessment of 20 marks. The Mid Term test is to be conducted when approximately 50% syllabus is completed and its duration will be one hour.		
Continuous Assessment		
Continuous Assessment is of **20 marks.** The rubrics for assessment will be considered on approval by the subject teachers. It should be minimum 2 or maximum 4 from the following table.		
Sr. No	**Rubrics**	**Marks**
1	Multiple Choice Questions (Quiz)	5

2	Literature review of papers/journals	5
3	Participation in event/ workshop/ talk / competition followed by small report and certificate of participation relevant to the subject	5
4	Wins in the event/competition/hackathon pertaining to the course	10
5	Case study, Presentation, group discussion, technical debate on recent trends in the said course	10
6	Project based Learning and evaluation / Extra assignment / Question paper solution	10
7	NPTEL/ Coursera/ Udemy/any MOOC Certificate course for 4 weeks or more	10
8	Content beyond syllabus presentation	10
9	Creating Proof of Concept	10
10	Mini Project / Extra Experiments/ Virtual Lab	10
11	Peer Review and participation	5/10
12	GATE Based Assignment tests/Tutorials etc	10
*For sr.no.7, the date of certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.		
Indirect Assessment		
1	Mock Viva/Practical	
2	Skill Enhancement Lecture	
3	Extra Assignments/lab/lecture	
End Semester Theory Examination:		
1	Question paper will be of 60 marks	
2	Question paper will have a total of five questions	
3	All questions have equal weightage and carry 20 marks each	
4	Any three questions out of five need to be solved.	

Design and Analysis of Algorithms (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NCMPCL41	Design and Analysis of Algorithms (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPCL41	Design and Analysis of Algorithms (Lab)	-	-	-	-	25	25	50

Lab Prerequisite: Basic knowledge of programming and data structure	
Lab Objectives	
1	To introduce the methods of designing and analyzing algorithms
2	Design and implement efficient algorithms for a specified application
3	Strengthen the ability to identify and apply a suitable algorithm for the given real-world problem.
4	Analyze the worst-case running time of algorithms and understand fundamental algorithmic problem
Lab Outcomes	
1	To introduce the methods of designing and analyzing algorithms
2	Design and implement efficient algorithms for a specified application
3	Strengthen the ability to identify and apply a suitable algorithm for the given real-world problem.
4	Analyze the worst-case running time of algorithms and understand fundamental algorithmic problems.

Suggested Experiments: Students are required to complete **at least 10 experiments.** Implementation can be in any programming language.	
Sr. No.	**Name of the Experiment**
1	**Introduction to Design and Analysis of Algorithms** ● Implement Selection Sort and compare the sorting time based on step count. ● Implement Insertion Sort and compare the sorting time based on step count.
2	**Introduction to Design and Analysis of Algorithms** ● Write a case study on Complexity Classes: P, NP, NP-Hard, NP-Complete
3	**Divide and Conquer Approach:** ● Implement and analyze Merge sort ● Implement and analyze Quick sort
4	**Divide and Conquer Approach:** ● Implement and analyze Binary search
5	**Greedy Method** ● Single source shortest path- Dijkstra
6	**Greedy Method** ● Implement and analyze Fractional Knapsack problem
7	**Greedy Method** ● Implement and analyze Job sequencing with deadlines
8	**Greedy Method** ● Implement and analyze Minimum cost spanning tree using Kruskal algorithm ● Implement and analyze Minimum cost spanning tree using Prim's algorithm
9	**Dynamic Programming Approach (any one)** ● Single source shortest path- Bellman-Ford ● All pair shortest path- Floyd Warshall ● Implement and analyze 0/1 knapsack ● Implement and analyze Matrix Chain Multiplication ● Implement and analyze Longest common subsequence ● Implement and analyze Optimal Binary Search Tree
10	**Backtracking and Branch and bound (any one)** ● Implement and analyze N-queen problem using Backtracking design strategy ● Implement and analyze Sum of subsets using Backtracking design strategy ● Implement and analyze Graph coloring ● Implement and analyze 15 Puzzle Problems using Branch and Bound design strategy.
11	**String Matching Algorithms (any one)** ● Implement Naïve string-matching Algorithms ● Implement Rabin Karp algorithm ● Implement Knuth-Morris-Pratt algorithm

Useful Links	
1	https://cse01-iiith.vlabs.ac.in/exp/sorting/
2	https://nptel.ac.in/courses/106/106/106106131/
3	https://swayam.gov.in/nd1_noc19_cs47/preview
4	https://www.coursera.org/specializations/algorithms
Tools and Articles	
5	Algorithm Visualizer: https://algorithm-visualizer.org/
6	Pathfinding Visualizer: https://bengavrilov.github.io/Path-Finding-Visualizer/
7	Design and Analysis of Algorithms by Stanford University: https://online.stanford.edu/courses/cs161-design-and-analysis-algorithms
8	MIT OpenCourseWare - Design and Analysis of Algorithms: https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/

Term Work	
1	Term work should consist of 10 experiments.
2	The journal must include at least 2 assignments.
3	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)