LNMIIT, Jaipur
Department of Computer Science & Engineering

Programme: B. Tech. (CSE, CCE)	Course Title: Design and Analysis of Algorithms			Course Code: CSE325
Type of Course: Program Core	**Prerequisites:** Data Structures and Algorithms			**Total Contact Hours:** 40 +20
Year/Semester: 2/Odd	**Lecture Hrs/Week:** 3	**Tutorial Hrs/Week:** 0	**Practical Hrs/Week:** 2	**Credits:** 4

Learning Objective:

The objective of this course is to introduce the notion of algorithm, how to describe an algorithm and model of computation for which an algorithm for a problem is to be designed. Time and space complexities will be introduced to express the resource needed by an algorithm and to compare different algorithms for a problem and to classify an algorithm as efficient or non-efficient. Algorithm design paradigms such as Divide and Conquer, Greedy and Dynamic programming will be introduced that will enable a student to apply one of the paradigms to design algorithm for a given problem. The last part of the course will introduce the notion of NP-completeness to capture the hardness of a problem and will show several fundamental problems in Computer Science and engineering to be NP-complete.

Course outcomes (COs):

On completion of this course, the students will have the ability to:		Bloom's Level
CO-1	**Analyze** the complexity of algorithms in terms of asymptotic notations and **implement** the algorithms	3
CO-2	Perform **Analysis** of important algorithmic design paradigms and **implement** one algorithm designed using each paradigm	4
CO-3	**Design** efficient algorithmic solutions to graph related problems and **implement them**	6
CO-4	**Understand** the concept of Computational Intractability	2

Course Topics	Lecture Hours	CO Mapping
UNIT – I (Introduction)	**14**	**CO1**
1.1 What is algorithm? How to describe an algorithm? Why analyze algorithms? RAM Model of Computation.	**1**	

LNMIIT, Jaipur
Department of Computer Science & Engineering

1.2 Asymptotic **notation: Definitions of** big-Oh, big-Omega, big-Theta, small-oh and problem solving	**2**	
1.3 Recurrence relations: iterative method, substitution method, recursion-tree method, Master method with problem solving	**3**	
1.4 Sorting Algorithms: Bubble sort, Insertion sort, and Selection sort with complexity analysis and proof of correctness	**3**	
1.5 Lower bound of sorting, Counting Sort, Radix Sort, Bucket Sort with complexity analysis and proof of correctness	**3**	
1.6 **Concepts of Graph:** Graph Representation, Applications of graphs, Hand-Shaking Lemma, Different Graph types like connected, strongly connected, etc.	**2**	
UNIT – II (Design Paradigm)	**13**	**CO2**
2.1 **Divide and Conquer:** Stassen's Matrix Multiplication, Merge Sort, Quick Sort; Complexity Analysis and proof of Correctness of all the algorithms.	**3**	
2.2 **Greedy Algorithms:** Huffman Coding, Fractional Knapsack, Kruskal's Algorithm, Prim's Algorithm and its implementation using heap data structure. Complexity Analysis and Correctness proof of all the algorithms.	**4**	
2.3 **Dynamic Programming:** Introduction to Dynamic Programming; Top down versus Bottom Up approach of Problem solving; Computing Fibonacci Number; Matix Chain Multiplication Algorithms; Longest Common subsequence, Complexity Analysis and proof of Correctness of all the algorithms	**4**	
2.4 **Backtracking:** 8 Queen problem with complexity analysis and Correctness proof.	**1**	
2.5 **Branch and Bound**: 0/1 Knapsack problem with complexity Analysis and Correctness proof.	**1**	
UNIT-III (Graph Algorithms)	**7**	**CO3**
3.1 **Breadth First Search (BFS)**: BFS Algorithm, BFS edge classification for directed and undirected graphs, Applications of BFS: Shortest Path on un-weighted graph, testing bipartiteness	**2**	
3.2 **Depth First Search (DFS):** DFS Algorithm, DFS edge classification for directed and undirected graphs, Application of DFS like Topological Sort, Cycle Detection.	**2**	
3.3 Shortest Path Algorithms: Dijkstra's Single source Algorithm; All pair Shortest paths: Floyd-Warshall and Bellman Ford;	**3**	

LNMIIT, Jaipur
Department of Computer Science & Engineering

UNIT-IV (NP-Completeness and Computational Intractability)	**6**	**CO4**
4.1 Optimization and Decision Problems, Decision Problems: Boolean Satisfiability Problem, Hamiltonian Cycle, Independent Set, Clique, Vertex Cover, and Vertex Cover; Definitions of P and NP; Karp and Cook reduction;	**3**	
4.2 NP-complete and NP-hard; Cook-Levin Theorem; Tractability and Intractability problems; Proving NP-completeness of Clique Decision Problem. Independent Set Decision Problem, Vertex Cover Decision Problem;	3	

Design and Analysis of Algorithm Lab (20 Hrs.)

Lab No	Description of problem	Co mapping	Hours
1	Compute the nth Fibonacci Number using recursive and non-recursive program and compute the time taken by each program using time () function and estimate the time complexity	**CO 1**	**2**
2	Implementations of Sorting Algorithms: Bubble sort, Insertion sort, and Selection sort and compare the running time	**CO 1**	**2**
3	Implementations of Counting Sort, Radix Sort, and Bucket Sort, Merge Sort, and Quick Sort	**CO 1, CO 2**	**2**
4-5	Implementation of Kruskal's Algorithm using Union Find Data structures	**CO 2**	**4**
6-7	Implementation of Prim's Algorithm and its implementation using heap data structure	**CO 2**	**4**
8	Implementation of Dynamic programming-based algorithm for Longest Common subsequence	**CO 2**	**2**
9	Implementation of backtracking algorithm to find all possible solutions of 8 Queen problem	**CO 2**	**2**
10	Implementation of (i) DFS based algorithm to test connectedness of a graph and testing acyclicity of a graph (ii) BFS based algorithm to test whether a graph is bipartite or not	**CO 3**	**2**

Textbook References:
Text Book:
1. Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein, Clifford. *Introduction to Algorithms*. 4th Edition: The MIT Press, 2001.

Reference books:
1. Ellis Horowitz, Sartaj Sahani, Sanguthevar Rajasekaran. *Fundamental of Computer Algo-*

rithm. 1st Edition: University Press, 2008.
2. Skiena Steven, S. The algorithm design manual, 2008
3. Kleinberg, J., & Tardos, É. *Algorithm design*. 1st Edition: Pearson Education India, 2006

Additional Resources:

1. ***https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/***
2. ***https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/***
3. ***https://nptel.ac.in/courses/106/106/106106131/***
4. nptel.ac.in/courses/106/101/106101060/

Evaluation Method	
Item	**Weightage (%)**
Assignments (Theory)	3 x 3 = 9
Midterm (Theory)	26
Practical Lab	5 (attendance) + 8 (mid-term) + 12 (end-term) = 25
End Term (Theory)	40

*Please note, as per the existing institute's attendance policy the student should have a minimum of 75% attendance. Students who fail to attend a minimum of 75% lectures will be debarred from the End Term/Final/Comprehensive examination.

LNMIIT, Jaipur
Department of Computer Science & Engineering

CO and PO Correlation Matrix (For CSE)

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1		1					3	3		3	2		
CO2	1	1	3	1					3	3		3	3	2	2
CO3	3	3	3	1					3	3		3	3	2	2
CO4	3	3		3					3	3		3	3	3	3

CO and PO Correlation Matrix (For Others)

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1		1					3	3		3
CO2	1	1	3	1					3	3		3
CO3	1	1	3	1					3	3		3
CO4	3	3		3					3	3		3

Last Updated On:

Updated By:

Approved By: