CS1105: Design and Analysis of Algorithms

Course Title and Code: Design and Analysis of Algorithms: CS1105	
Hours per Week	**L-T-P: 3-0-4**
Credits	**4 (CSE)**

Course Objective:

This course introduces an understanding of the design and analysis of algorithms. The course aims to develop a familiarity with important algorithms and data structures and an ability to analyze the asymptotic performance of algorithms. It will equip the students to apply important algorithmic design paradigms and methods of analysis to develop efficient algorithms in common engineering design situations.

Course Weightage: Theory 65%, Practical – 35%

Course Outcome:

On successful completion of this course, the students should be able to:

- CS1105.1. Understand the importance of time complexity and evaluate the complexity of algorithms using asymptotic notations.
- CS1105.2. Understand and implement different tree data structures
- CS1105.3. Understand and differentiate between different algorithm design paradigms: Divide and Conquer, Greedy, Dynamic Programming and Backtracking.
- CS1105.4. Develop energy-efficient algorithms and programs using Greedy approach to solve various computing problems, e.g., Minimum Spanning Trees, Shortest Path, fractional Knapsack, etc.
- CS1105.5. Apply Dynamic Programming technique to solve various computing problems, e.g., 0/1 Knapsack, All pairs shortest path, Matrix Chain Multiplication, Longest common subsequence.
- CS1105.6. Understand various types of graph problems and their complexity: Traveling Salesperson Problem, Hamiltonian cycle, Graph Coloring problem, Vertex Cover problem, Maximum bipartite matchings etc.
- CS1105.7. Differentiate between P, NP, NP-Complete, and NP-Hard problems.

Prerequisites: Nil

Sr. No	**Specifications**	**Marks**
1	Attendance	5
2	Assignment	10
3	Class Participation	Nil
4	Quiz	10
5	Theory Exam-I	20
6	Theory Exam-II	Nil

7	Theory Exam-III	30
8	Report-I	Nil
9	Report-II	Nil
10	Report-III	Nil
11	Project-I	Nil
12	Project-II	Nil
13	Project-III	Nil
14	Lab Evaluation-I	10
15	Lab Evaluation-II (Test)	10
16	Course Portfolio	5
17	Presentation	Nil
18	Viva	Nil
	Total (100)	100

Retest Evaluation Scheme		
1	Theory Exam–III	30
	Total (30)	**30**

Syllabus (Theory):

Unit 1:

Introduction: Algorithms, Analyzing algorithms, Designing algorithms, Algorithm Paradigms

Characterizing Running Times: Asymptotic notations: formal definition, O-notation, Y-notation, and , Theta notation, Recurrence relations, The substitution method for solving recurrences, The master method for solving recurrences.

Divide-and-Conquer: Multiplying square matrices, Strassen's algorithm for matrix multiplication, The recursion-tree method for solving recurrences, Analysing Merge sort and Quicksort.

Unit 2:

Revision of Binary Search Trees, Balanced Binary search trees: Definition of an AVL Tree, Insertion and Deletion in AVL trees using rotation, Introduction to Red-black trees, Introduction to Splay trees

Multiway Search Trees: Introduction to Multiway Search Trees and types of MST, Definition of B-Trees, Insertion and deletion in B-trees, Introduction to B+ trees.

Unit 3:
Elementary Graph Algorithms: Graphs and basic theorems, Breadth-First search, Depth-First search, Topological sort, Strongly connected components.

Unit 4:
Greedy Algorithms: Elements of Greedy Strategy, Fractional Knapsack problem, Huffman codes, Minimum spanning tree: Kruskal and Prims algorithm, Dijkstra's algorithm

Unit 5:
Dynamic Programming: Scheduling Problem, Matrix-chain multiplication, Elements of dynamic programming, Traveling Salesman Problem, Bellman-Ford Algorithm, The Floyd-Warshall Algorithm

Unit 6:
Maximum Flow: Flow networks, The Ford-Fulkerson method, Maximum bipartite matching

Unit 7:
NP and NP-Completeness: Polynomial time complexity, understanding NP, Polynomial-time verification, NP-completeness and reducibility, Undestanding NP-complete problems

Approximation Algorithms: Approximate solutions to NP Complete problems like Traveling Salesman Problem, Hamiltonian cycle, Vertex Cover problem, etc.

Textbook (s)

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, “Introduction to Algorithms”, Prentice Hall of India. 2002.
2. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, “Algorithms”, McGraw-Hill Education, 2006.

Reference Book(s)

1. RCT Lee, SS Tseng, RC Chang and YT Tsai. Introduction to the Design and Analysis of Algorithms. Mc Graw Hill, 2005.
2. E. Horowitz & S Sahni. Fundamentals of Computer Algorithms. 1984
3. Berman, Paul. Algorithms. Cengage Learning. 2002
4. Aho, Hopcraft, Ullman, The Design and Analysis of Computer Algorithms. Pearson Education, 2008.

NPTEL Swayam Course:

1. https://nptel.ac.in/courses/106/106/106106127/
2. https://nptel.ac.in/courses/106/102/106102064/
3. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html

Course Articulation Matrix: (Mapping of COs with POs)

Course Outcomes	PO1	PO2a	PO2b	PO2c	PO3a	PO3b	PO3c	PO4a	PO4b	PO4c	PO5a	PO5b	PO6	PO7a	PO7b	PSO1	PSO2
CS1105.1	2		1		2						1			1		2	2
CS1105.2	2		1		2				1							2	2
CS1105.3	2		1		2				1						2	2	2
CS1105.4	2		1		1				1							2	2
CS1105.5	1		1		1				1					1		2	2
CS1105.6	1					2			1						1	2	2
CS1105.7	1		1		1				2							2	2

1- Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation

1 Introduction 2L
1.1. Algorithms
1.2. Analysing algorithms : Insertion sort
1.3 Designing algorithms
1.4. Algorithm Paradigms

2 Characterizing Running Times – 3L.
2.1 Asymptotic notation: formal definition
2.2 O-notation, Y-notation, and , Theta notation
2.3 Recurrence relations
2.4. The substitution method for solving recurrences.
2.5. The master method for solving recurrences.

3. Divide-and-Conquer - 3L
3.1. Multiplying square matrices.
3.2. Strassen's algorithm for matrix multiplication
3.3. The recursion-tree method for solving recurrences.
3.4. Analysing Merge sort and Quicksort.

Trees Data Structures

4. Binary Search Trees - 1L
4.1 What is a binary search tree?
4.2 Searching a binary search tree.

4.3 Insertion and deletion in BST

5. Balanced Binary search trees -4L
5.1. Definition of an AVL Tree
5.2. Insertion and Deletion in AVL trees using rotation
5.3. Introduction to Red-black trees
5.4. Introduction to Splay trees

6. Multiway Search Trees - 5L
6.1. Introduction to Multiway Search Trees and types of MST
6.2. Definition of B-Trees
6.3. Insertion and deletion in B-trees
6.4. Introduction to B+ trees.

Graph Algorithms

7. Elementary Graph Algorithms -3L
7.1. Basics of Graphs and Theorems
7.2. Breadth-First search
7.3. Depth-First search
7.4. Topological sort
7.5. Strongly connected components.

Advanced Algorithms and Optimization Problems

8. Greedy Algorithms - 5L
8.1. Elements of Greedy Strategy
8.2. Fractional Knapsack problem
8.3. Huffman codes
8.4. Minimum spanning tree: Kruskal and Prims algorithm
8.6. Dijkstra's algorithm

9. Dynamic Programming – 5L

9.1. Scheduling Problem
9.2 Matrix-chain multiplication
9.3 Elements of dynamic programming
9.4 Traveling Salesman Problem
9.5. Bellman-Ford Algorithm
9.6. The Floyd-Warshall Algorithm

10 Maximum Flow - 3L
10.1 Flow networks
10.2 The Ford-Fulkerson method
10.3 Maximum bipartite matching

Selected Topics

11 NP-Completeness - 3L
11.1 Polynomial time
11.2 Polynomial-time verification
11.3 NP-completeness and reducibility
11.4 NP-complete problems

12 Approximation Algorithms - 3L

Learning Activities

Sl No	Learning Activities	Evaluation Components Used	CO
LA1	Calculation of Algorithm complexity – using sorting and searching Algorithms	Lab Evaluation Assignment	CS1105.1
LA2	Learning Divide and conquer algorithms using Multiplication of Square Matrix and Strassen's algorithm for matrix multiplication	Lab Evaluation Assignment	CS1105.3
LA3	Application of recurrance using Heapsort	Lab Evaluation	CS1105.1
LA4	Tree data structures and its application (AVL and B trees)	Assignment	CS1105.2
LA5	Graph Theory , related algorithms and application	Lab Evaluation Assignment	CS1105.6
LA6	Greedy algorithms – Minimum spanning tree using Prim and Kruskal Algorithms, Djakstra's Algorithm	Lab Evaluation Assignment	CS1105.3 CS1105.4
LA7	Dynamic Programming - Matrix-chain multiplication , Traveling Salesman Problem ,Bellman-Ford Algorithm	Lab Evaluation Assignment	CS1105.5
LA8	Application of Flow networks – Ford-Fulkerson method	Lab Evaluation	CS1105,4
LA9	NP – Completeness , Approximation Algorithm	Assignment	CS1105.7

Activities	Correlation with course objectives						
	CS1105.1	**CS1105.2**	**CS1105.3**	**CS1105.4**	**CS1105.5**	**CS1105.6**	**CS1205.7**
LA1	2						
LA2			2				
LA3	1						
LA4		1					
LA5						2	
LA6			2	2			
LA7					2		
LA8				1			
LA9							1