Course Title	Design and Analysis of Algorithms	**Semester**	IV
Course Code	23CS252	**CIE**	50
Total No. of Contact Hours	50	**SEE**	50
No. of Contact Hours/week	3:2:0:2	**Total**	100
Credits	4 3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

Course objective is to:

1. Enable students to learn the methods for analysing algorithms and evaluating their performance.
2. Enable students to demonstrate the efficiency of algorithms.
3. Prepare students to solve problems using various algorithm design methods
4. Appreciate the concepts of P and NP complexity classes.

PREREQUISITES : This course requires that the students are familiar with programming language and Data Structures and Applications. Graph Theory is desirable

Module-1: Introduction and Overview	**RBT Levels: L1,L2,L3**	8 ~~10 Hours~~

Introduction and Examples: What is an Algorithm? Algorithm Specification, Examples from real life: Air Travel, Xerox Shop, Document Similarity and types of algorithms.

Motivation for Performance Analysis using Examples: Bubble Sort, Selection Sort, Insertion Sort, String Pattern Matching. Contrast performance analysis versus actual runs.

Performance Analysis Framework: Space complexity, Time complexity. **Asymptotic Notations:** *Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ) - recursive and recursive Algorithms with Examples.* ation (□), Mathematical analysis of non

Text Book 1: Chapter 1.1,2.1-2.4,3.1,3.2 Digital Resource: D1

Module-2: Decrease and Conquer, Divide and Conquer, Greedy Method	**RBT Levels: L1,L2,L3**	8 ~~10 Hours~~

Decrease and Conquer Approach: Insertion sort**,** Topological Sort**.**

Divide and Conquer: General method, Recurrence equation for divide and conquer, Finding the maximum and minimum. Merge sort, Quick sort

Greedy Method: General method, Knapsack Problem, Job sequencing with deadlines,

Text book 1: Chapter 4.1,4.2,5.1,5.2,6.4 Text book 2:4.1,4.3,4.5

Module-3: Greedy Method, Transform and Conquer, Dynamic Programming	**RBT Levels:L1,L2,L3**	8 ~~10 Hours~~

Minimum cost spanning trees: Prim"s Algorithm, Kruskal"s Algorithm, union find method

Single source shortest paths: Dijkstra's Algorithm. **Optimal Tree problem:** Huffman Trees and Codes. **Transform and Conquer Approach:** Heaps and Heap Sort.

Dynamic Programming: General method with Examples, **Transitive Closure:** Warshall"s Algorithm, **All Pairs Shortest Paths:** Floyd's Algorithm

Text book 1: 9.1,9.2,9.3,9.4,8.1,8.4		
Module-4: Dynamic Programming, Backtracking	**RBT Levels:L1,L2,L3**	**10 Hours**
Dynamic Programming (cont…): Multistage Graphs, Travelling Sales Person problem, Knapsack problem, Bellman-Ford Algorithm, **Backtracking:** General method, N-Queens problem, Sum of subsets problem, Hamiltonian cycles. **Text book 1: 8.2,12.1 text book 2: 5.2, 5.4, 5.9**		
Module-5: Space and Time Trade-offs, Branch and Bound and Case Studies	**RBT Levels:L1,L2,L3**	**10 Hours**
Counting Sort Branch and Bound: Assignment Problem, Travelling Salesperson problem, 0/1 Knapsack **NP-Complete and NP-Hard Problems concept.** Case Studies: Efficient Route calculation application in GPS navigation system: **The navigation system calculates the optimal route from the user's current location to the destination. This calculation considers factors such as distance, estimated travel time, traffic conditions.** **Summarization of all modules.** **Text book 1: 7.1, 12.2, 12.3**		
Follow on Courses: Advanced Algorithms, Machine Learning		
Resources for Advanced Learners: 1. Data Structures and Algorithms Made Easy – Data Structures & Algorithmic Puzzles Author: Narasimha Karumanchi (M.Tech IIT Bombay, Founder- CareerMonk.com) 5 th Edition 2. *Algorithms* - Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani published by Mc Graw Hill		

Suggested Learning Resources:	
Textbooks:	
1.	Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009. Pearson.
2.	Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press
Reference Books:	
1.	Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
2.	Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education).
Web links and Video Lectures (e-Resources): 1. Design and Analysis of Algorithms course by Madhavan Mukund	

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Real world problem solving and puzzles using group discussion. E.g., Fake coin identification,Peasant, wolf, goat, cabbage puzzle, Konigsberg bridge puzzle etc.,
2. Demonstration of solution to a problem through programming.

Course outcomes:	
CO1	Apply asymptotic notational method to analyse the performance of the algorithms in terms of time complexity
CO2	Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems
CO3	Use transform & conquer and dynamic programming methodologies to solve the given real world computational problems
CO4	Explain various classes (P, NP and NP Complete) of problems and demonstrate backtracking, branch & bound and approximation methods.
CO5	Apply appropriate algorithm design strategies to a given case study/use case.

CO-PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO2	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO3	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO4	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO5	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-

High-3, Medium-2, Low-1

Question Paper Pattern:

1. **CIE Assessment Pattern:**

2. **SEE Assessment Pattern:**