

Course Code: CSE3004	Course Title Design and Analysis of Algorithms	TPC	3	2	4
Version No.	1.2				
Course Pre-requisites/ Co-requisites	CSE2001				
Anti-requisites (if any).	None				
Objectives:	<div>1. This course aims to introduce the classic algorithms in various domains, and techniques for designing efficient algorithms.</div> <div>2. Analyze the asymptotic performance of algorithms.</div> <div>3. Write rigorous correctness proofs for algorithms.</div> <div>4. Demonstrate a familiarity with major algorithms and data structures.</div> <div>5. Apply important algorithmic design paradigms and methods of analysis.</div> <div>6. Synthesize efficient algorithms in common engineering design situations.</div>				
COs mapping with POs and PEOs					
Course Outcomes	Course Outcome Statement			POs / PEOs	
CO1	Argue the correctness of algorithms using inductive proofs and analyze running times of algorithms using asymptotic analysis.			PO1, PO2, PO3, PO4, PO5, PO6, PO11	
CO2	Describe the brute force and divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.			PO1, PO2, PO4, PO6, PO7, PO8	
CO3	Learn the Greedy and Dynamic programming Algorithm Paradigms and Recite algorithms that employ this paradigms and its performance analysis			PO1, PO2, PO3, PO4, PO9, PO10, Po11, PO12	
CO4	Understanding the mechanism of iterative-improvement technique which involves finding a solution to an optimization problem by generating a sequence of feasible solutions with improving values of the problem’s objective function.			PO2, PO3, PO4, PO5, PO6, PO11	
CO5	Learn the construction of state-space trees in the depth-first-search fashion using Backtracking and its applications.			PO1, PO2, PO3, PO4, PO5, PO11	
CO6	Familiarize on the concept of Branch and Bound Algorithm Design Technique and gain knowledge in the class of P and Np type problems			PO3, PO4, PO5, PO6, PO9	
TOTAL HOURS OF INSTRUCTIONS: 45					
Module No. 1	Introduction				8 Hours
Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithm Efficiency – Analysis Framework – Asymptotic Notations and its properties – Mathematical analysis for Recursive and Non-recursive algorithms.					
Module No. 2	Divide-And-Conquer				6 Hours
Divide and Conquer methodology – Merge sort – Quick sort –Binary Search.					
Module No. 3	Greedy and Dynamic Programming				12 Hours
Greedy– Fractional Knapsack Problem, Job Sequencing with deadlines, Prim’s algorithm- Kruskal’s Algorithm-Dijkstra’s Algorithm, String Matching Algo. (Naive string matching, Rabin-Karp algorithm)					

<b>Dynamic Programming-</b> Computing Warshall and Floyd algorithm, Optimal Binary Search Trees, Knapsack Problem, Traveling Salesman Problem, Matrix chain multiplication, Longest Common Subsequence (LCS)		
<b>Module No. 4</b>	<b>Iterative Improvement</b>	<b>5 Hours</b>
The Simplex Method-The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs- The Stable Marriage Problem.		
<b>Module No. 5</b>	<b>Backtracking</b>	<b>6 Hours</b>
n-Queens problem (Four & Eight) – Hamiltonian Circuit Problem – Subset Sum Problem –Graph Coloring Problem		
<b>Module No. 6</b>	<b>Branch and Bound</b>	<b>8 Hours</b>
15-puzzle problem, Assignment problem, Knapsack Problem, Job Sequencing, Traveling Salesman Problem.		
<b>NP-completeness:</b> Reduction amongst problems, classes NP, NP-complete, and polynomial time reductions.		
<b>Text Books</b>		
1. <b>Anany Levitin</b> , “Introduction to the Design and Analysis of Algorithms”, Pearson Education, Third Edition, 2017.		
<b>References</b>		
1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, MIT Press/PHI Learning Private Limited, Third Edition, 2012. 2. E. Horowitz, S. Sahni, S. Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press, 2 <sup>nd</sup> Ed, 2008. 2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006. 3. Donald E. Knuth, “The Art of Computer Programming”, Volumes 1 & 3 Pearson Education, 2009. 4. Steven S. Skiena, “The Algorithm Design Manual”, Springer, Second Edition, 2008.		
<b>Lab Exercises</b>		
1. Write a programs to implement the Prim’s algorithm. 2. Write a programs to implement the Kruskal’s algorithm. 3. Write a program to find optimal ordering of matrix multiplication. (Note: Use Dynamic programming method). 4. Consider the problem of eight queens on an (8x8) chessboard. Two queens are said to attack each other if they are on the same row, column, or diagonal. Write a program that implements backtracking algorithm to solve the problem, i.e., place eight non-attacking queens on the board. 5. Write a program to find the strongly connected components in a digraph. 6. Write a program to implement file compression (and un-compression) using Huffman’s algorithm. 7. Write a program to implement dynamic programming algorithm to solve all pairs shortest path problem. 8. Write a program to solve 0/1 knapsack problem using the following: a) Greedy algorithm. b) Dynamic programming algorithm. 9. Write a program to solve 0/1 knapsack problem using the following: a) Backtracking algorithm. b) Branch and bound algorithm. 10. Write a program that uses dynamic programming algorithm to solve the optimal binary search tree problem. 11. Write a program for solving traveling sales person’s problem (TSP) using Dynamic programming algorithm.		

12. Write a program for solving TSP using the backtracking algorithm.
13. Write a program for solving TSP using the Branch and Bound algorithm.
14. Write a program for solving graph coloring problem using backtracking.
15. Write a program to find the Hamiltonian Circuit in a given graph.

<b>Course Type</b>	<b>Embedded Theory and Lab(ETL)</b>		
<b>Mode of Evaluation</b>	<b>Theory</b>		<b>75%</b>
	Continuous Assessment Test-1	15	
	Continuous Assessment Test-2	15	
	Digital Assignments/Quizes (Min)	30	
	Final Assessment Test	40	
	<b>Laboratory</b>		<b>25%</b>
<b>Modified by</b>	<b>Dr. Saroj Kumar Panigrahy</b>		
<b>Recommended by the Board of Studies on</b>	<b>14<sup>th</sup> BoS, 11.05.2024</b>		
<b>Date of Approval by the Academic Council</b>	<b>12<sup>th</sup> Academic Council, 25.05.2024</b>		