22CS3E1: DESIGN & ANALYSIS OF ALGORITHMS

<table>
<tr><th>Course Name</th><th colspan="2">Design & Analysis of Algorithms</th><th>L</th><th>T</th><th>P</th><th>C</th><th>CIA</th><th>SEE</th><th>TM</th></tr>
<tr><td><b>Course Code</b></td><td colspan="2">20CS3E1</td><td>4</td><td>0</td><td>0</td><td>4</td><td>30</td><td>70</td><td>100</td></tr>
<tr><td colspan="1"><b>Year of Introduction:</b> 2005</td><td><b>Year of Offering:</b> 2022</td><td colspan="3"><b>Year of Revision</b>: 2022</td><td colspan="5"><b>Percentage of Revision:</b> NIL</td></tr>
<tr><td colspan="10"><b>L</b>-Lecture, T-Tutorial, <b>P</b>-Practical, <b>C</b>-Credits, <b>CIA</b>-Internal Marks, <b>SEE</b>-External Marks, <b>TM</b>-Total Marks</td></tr>
</table>

Course Description and Purpose: The course is intended to develop proficiency in *Problem Solving and Programming*, perform *Analysis of various Algorithms in regard to Time and Space Complexity,* gain of *good understanding of Applications of Data Structures*, develop base for *Advanced Study* in *Computer Science*, apply *Design Techniques* to solve different types of problems as per their *Complexity* and develop *ability to segregate NP-Hard and NP-Complete problems.*

Course Objective: This course will help the students to understand and learn basic ideas about *Analysis of Algorithms*, *Divide-and-Conquer* and *Greedy Method*, *Dynamic Programing* & *Basic Traversal and Search Techniques*, *Backtracking* and *Branch and Bound* Techniques and *NP-Hard* and *NP-Complete* Problems.

Specific Objectives include:

- To understand Basic Ideas about *Analysis of Algorithms* and the *Concept of Data Structures*.
- To know *Divide and Conquer*, *Greedy Methods* and *Solving Various Problems* by applying them.
- To apply *Dynamic Programming Method* and *Basic Traversal* and *Search Techniques* to solve various Problems.
- To understand *Backtracking and Branch and Bound* Techniques to Design Algorithms.
- To categorize *NP-Har*d and *NP-Complete* Problems.

Course Leaning Outcomes: On successful completion of this course

CO1: The course imparts a foundational understanding of algorithms, data structures, performance analysis, randomized algorithms, and graph theory, enabling students to analyze, design, and implement efficient solutions to a wide array of computational problems.

CO2: Students will have a comprehensive understanding of advanced algorithmic paradigms, including Divide-and-Conquer and Greedy methods, enabling them to apply these techniques to solve a wide range of computational problems efficiently and effectively.

CO3: The course empowers students with a comprehensive understanding of dynamic programming techniques, traversal and search algorithms for binary trees and graphs, equipping them with the skills to solve complex optimization problems efficiently and effectively in diverse domains.

CO4: The course provides students with a comprehensive understanding of backtracking and branch-and-bound algorithms, enabling them to efficiently solve complex combinatorial and optimization problems, such as the 8-Queens problem, graph coloring, and the traveling salesman problem, across various application domains.

CO5: The course equips students with a profound understanding of NP-Hard and NP-Complete problems, enabling them to recognize, analyze, and address computationally challenging problems across various domains, including graph theory, scheduling, code generation, and decision problem solving, while comprehending the theoretical underpinnings and implications of these complexities.

UNIT-I (12 Hours)

Introduction: What is Algorithm, Algorithm Specification Pseudo code Conventions, Recursive Algorithms, Performance Analysis: Space Complexity Time Complexity, Asymptotic Notation, Performance Measurement, Randomized Algorithms: Basics of Probability Theory, Randomized Algorithms Identifying the Repeated Element, Primality Testing: Advantages and Disadvantages.

Elementary Data Structures: Stacks and Queues, Trees: Terminology, Binary Trees, Dictionaries: Binary Search Trees, Priority Queues, Heaps , Heapsort , Sets and Disjoint Set Union: Introduction-Union and Find Operations, Graphs: Introduction, Definitions, Graph Representations.

UNIT-II (12 Hours)

Divide-and-Conquer: General Method, Defective Chess Board, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Selection Problem, Strassen's Matrix Multiplication, Convex Hull: Some Geometric Primitives, The Quick Hull Algorithm, Graham's Scan, An O(nlogn) Divide and Conquer Algorithm.
The Greedy Method: The General Method, Container Loading, Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Optimal Storage on Tapes, Optimal Merge Patterns, Single Source Shortest Paths.

UNIT-III (12 Hours)

Dynamic Programming: The General Method, Multi Stage Graphs, All Pairs Shortest Paths, Single Source Shortest Paths, Optimal Binary Search Trees, String Editing -0/1 Knapsack, Reliability Design, The Traveling Sales Person Problem, Flow Shop Scheduling.
Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs: Breadth First Search and Traversal-Depth First Search, Connected Components and Spanning Trees, Bi-Connected Components and DFS.

UNIT-IV (12 Hours)

Backtracking: The General Method, The 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack Problem.
Branch and Bound: The Method: Least Cost Search, The 15 Puzzle Control Abstractions for LC Search, Bounding, FIFO Branch and Bound , LC Branch and Bound, 0/1 Knapsack Problem, LC Branch and Bound Solution, FIFO Branch and Bound Solution, Traveling Sales person.

UNIT-V (12 Hours)

NP-Hard and NP-Complete Problems: Basic Concepts: Non Deterministic Algorithms, The Classes NP Hard and NP Complex, Cook's Theorem, NP Hard Graph Problems, Clique Decision Problem, Node Cover Decision Problem Chromatic Number Decision Problem, Directed Hamiltonian Cycle, Traveling Sales Person Decision Problem, AND/OR Graph Decision Problem, NP-Hard Scheduling Problems, Scheduling Identical Processors, Flow Shop Scheduling, Job Scheduling, NP-Hard Code Generation Problems, Code Generation With Common Sub Expressions, Implementing Parallel Assignment Instructions, Some Simplified NP-Hard Problems.

Prescribed Text Book:
1. Sartaj Sahni, Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2nd Edition, 2008.

Reference Text Books:
1. Anany Levitin, Introduction to the Design & Analysis of Algorithms, 2nd Edition, Pearson Education, 2007.
2. I.Chandra Mohan, Design and Analysis of Algorithms, PHI, 2nd Edition, 2012.
3. Prabhakar Gupta, Vineet Agrawal, Design and Analysis of Algorithms, PHI, 2nd Edition 2012.
4. Parag Himanshu, Dave, Design and Analysis of Algorithms, Pearson Education, 1st Edition 2008.

Course Focus: Foundation / Skill Development.

Reference Websites:
1. https://epgp.inflibnet.ac.in/Home
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/lecture-notes/
3. https://www.cukashmir.ac.in/cukashmir/User_Files/imagefile/DIT/StudyMaterial/DAA/DAA_UNIT I_6th-Sem_StudyMaterial.pdf

22CS3E1

P.B.SIDDHARTHA COLLEGE OF ARTS & SCIENCE (AUTONOMOUS), VIJAYAWADA-520010

(An Autonomous College in the Jurisdiction of Krishna University, A.P., India.)

M.Sc.(COMPUTER SCIENCE) DEGREE EXAMINATIONS

THIRD SEMESTER

DESIGN & ANALYSIS OF ALGORITHMS

SYLLABUS W.E.F 2022-2023

Time 3 Hours **Max.Marks: 70**

SECTION-A

Answer ALL questions **(5×4 = 20 Marks)**

1.(a) Define *Algorithm*. Explain the algorithm specification briefly.(CO1,L1)

(or)

(b) What are the operations in a *Priority Queue*? (CO1, L1)

2. (a) Explain the Divide and Conquer Algorithms to solve *Convex Hull Problem.* (CO2,L1)

(or)

(b) What is *Tree Vertex Splitting?* (CO2,L1)

3. (a) What is *String Editing?* (CO3,L1)

(or)

(b) Differentiate *DFS and BFS.* (CO3,L1)

4. (a) What is *Graph Colouring?* (CO4,L1)

(or)

(b) What is *Branch and Bound* technique*?*(CO4,L1)

5. (a) Compare *NP hard and NP Complete Classes.* (CO5,L1)

(or)

(b) Explain *flow shop scheduling* in *NP Hard Scheduling Problems.* (CO5,L1)

SECTION - B

Answer all questions. All question carry equal marks. **5 × 10 = 50 Marks**

6. (a) Define Algorithm. Discuss *Performance Analysis of Algorithms* briefly. (CO1,L2)

(or)

(b) Explain *Disjoint Sets*, *Disjoint Set Union* & *Find Operations* with Algorithms. (CO1,L2)

7. (a) Discuss the method for *Divide and Conquer* approach and write algorithm for Quick Sort with an example. (CO2,L6)

(or)

(b) Discuss the general method for *Greedy Method.* Apply it on *Single Source Shortest Path* by writing an algorithm with suitable example. (CO2,L6)

8. (a) Examine algorithm and procedure of finding *Optimal Binary Search Tree* using Dynamic Programming with example. (CO3,L4)

(or)

(b) Examine *Traversal Techniques for Graphs* with an example. (CO3,L4)

9. (a) Explain *Control Abstraction for LC Search.* Solve *0/1-Knapsack Problem* using *Branch and Bound Technique*. (CO4,L5)

(or)

(b) Explain the *Sum of Subsets Problem* using *Back Tracking Technique.* (CO4,L5)

10. (a) Make use of different formulae prove *COOKs Theorem..* (CO5,L3)

(or)

(b) Choose *NP-Hard Graph Problems* and explain. (CO5,L3)