DESIGN AND ANALYSIS OF ALGORITHMS

CREDITS: 03

SEMESTER : 04 **L:T:P – 3:0:0**

IA/UE : 30/70 **CONTACT HRS/WEEK : 03**

Course Objective:

1. Understanding of various algorithm design techniques like Divide and Conquer, Greedy Method, Dynamic Programming, Backtracking and Branch and Bound with several examples rather than individual algorithms

2. Students will be emphasized on asymptotic analysis in measuring algorithms running time. Also enlighten the students to a good program structure and its correctness.

Course Outcomes:

CO 1 Explain the process of designing and analyzing an algorithm through basic problem types and asymptotic notations.

CO 2 Identify the key characteristics of a given problem, suitable design approach and its impact on performance.

CO 3 Solve the given real time problems through an appropriate design strategy.

CO 4 Examine the space and time efficiency of an algorithm and compare with other design strategy.

CO 5 Select the best design strategy suitable for optimization problems.

CO 6 Demonstrate an algorithm using appropriate design strategies for problem solving.

CO-PO MAPPING

Course Outcomes – Program Outcome Mapping (H/M/L : 3/2/1)

	PO1.	PO2.	PO3.	PO4.	PO5.	PO6.	PO7.	PO8.	PO9.	PO10	PO11	PO12	PSO1	PSO2
CO1.	3	2	2	1	1	1	1					3	3	3
CO2.	3	3	2	2	2	1	1					3	3	3
CO3.	3	3	3	3	3	1	1					3	3	3
CO4.	3	3	3	3	3	1	1					3	3	3
CO5.	3	3	3	3	3	1	1					3	3	3

CO6.	3	3	3	3	3	1	1				3	3	3

UNIT – I Introduction to Algorithms and Elementary Data Structures [10 HOURS]

What is an Algorithm? Algorithm Specification, Performance Analysis: Space complexity, Time complexity. Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ). Introduction to Randomized algorithms. Stacks and queues, trees, Dictionaries: Binary search tree, AVL tree, Cost amortization. Priority queue, sets and disjoint set union, graphs.

1. **Explain how you will analyze the performance of an algorithm. Write about the different asymptotic notations.**
2. **Write a brief note on AVL trees and Priority queues.**
3. **Define Stack. Write the pseudo code stack operations.**
4. **Explain the queue linear data structures. Why the linked list representation of the queue is efficient than the array representation justify.**

UNIT – II Basic Traversal, Search Techniques and Divide and Conquer [10 HOURS]

Techniques for binary trees, techniques for graphs: breadth first search and traversal, depth first search and traversal, connected components and spanning trees, biconnected components and DFS.

General method of divide and conquer, Binary search, Finding the maximum and minimum, Merge sort, Quick sort Strassen's matrix multiplication. Advantages and Disadvantages of divide and conquer.

5. **Explain BFS algorithm using some example and write the pseudo code.**
6. **Write a brief note on the DFS algorithm. Using an example show how you will find the graph traversal of a graph. Note: Take any graph G.**
7. **Define spanning tree. Explain how you will find the spanning tree or trees of the given graph using some example.**
8. **Explain Quick sort algorithm and Write the pseudo code for Quick sort algorithm.**
9. **What is binary search? Explain how binary search is efficient than any other sorting technique.**
10. **Explain divide and conquer approach for problem solving using some example.**
11. **Write the Java code to find the minimum and maximum element of the given list of elements.**

UNIT – III Greedy Method [8HOURS]

General method, Knapsack Problem, Job sequencing with deadlines, minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm. single source shortest paths.

Knapsack problem
Job sequencing with deadlines
Kruskal's Algorithm. single source shortest paths
MCST

UNIT- IV Dynamic Programming [9 HOURS]

General method, Multistage Graphs, **Single source shortest paths: Dijkstra's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem**, Bellman-Ford Algorithm, **Travelling Sales Person problem.**

UNIT- V Backtracking, Branch and Bound [9 HOURS]

General method of backtracking, **N-Queens problem, Sum of subsets problem**, **Graph coloring**, Hamiltonian cycles, **Knapsack problem**.

General method of Branch and Bound, 0/1 Knapsack problem: LC Branch and Bound solution, FIFO Branch and Bound solution, Travelling Sales Person problem, NP-Hard and NP-Complete problems: Basic concepts.

TEXT BOOKS

1. Fundamentals of computer Algorithms, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition,
2014, Universities Press
3. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009. Pearson

Unit 3:
Knapsack problem
Job sequencing with deadlines
Kruskal's Algorithm. single source shortest paths
MCST

4. Unit 4:
5. **Single source shortest paths: Dijkstra's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem**, Bellman-Ford Algorithm, **Travelling Sales Person problem**
6. **Unit 5:**
7. **N-Queens problem, Sum of subsets problem**, **Graph coloring**, Hamiltonian cycles, **Knapsack problem**.
8. **General method of Branch and Bound, 0/1 Knapsack problem: LC Branch and Bound solution, FIFO Branch and Bound solution**,

REFERENCES

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).