Design & Analysis of Algorithms Laboratory			
Course Code	**23CS253**	**Semester**	**IV**
L:T:P:S	**0:0:2:1**	**CIE Marks**	**50**
Hrs. / Week	**2**	**SEE Marks**	**50**
Credits	**1**	**Total Marks**	**100**
Exam Hours	**03**		

Course objectives : This laboratory course enables students to	
1.	Train students to design and implement various algorithms in JAVA/C/C++
2.	Assist students to employ various design strategies for problem solving.
3.	Enable students to measure and compare the performance of different algorithms.

Pgm. No.	**List of Experiments / Programs**	**Hours**	**COs**
	Prerequisite Experiments / Programs / Demo		
This course requires that the students are familiar with programming language and Data Structures & Applications. Graph Theory is desirable.			
	PART-A		
1	**Title:** Program that uses simple technique to a) search a key b) to sort n elements **Problem Description:** Run the program for varied values of n>5000 and record the time taken to search/sort. Plot a graph of the time taken v/s n. The elements can be read from a file or can be generated using the random number generator. **Method:** A Brute Force- Non recursive algorithm implementation. Sequential Search and Selection Sort. **Theory Reference:** Module 1	2	CO1,CO 3
2	**Title**: Sorting elements based on their value. **Problem Description:** Run the program for varied values of n to demonstrate the behaviour of the algorithm in the Worst, Best and Average Cases. Record the time taken to sort. Plot a graph of the time taken versus *n* on graph sheet. The elements can be	2	CO1,CO3

	read from a file or can be generated using the random number generator for large values of n. **Method:** A Divide & Conquer approach partitioning the data set on a pivot- Recursive algorithm implementation. **Theory Reference:** Module 2		
3	**Title:** Sorting elements in a list by breaking them into sub-lists. **Problem Description:** Run the program for varied values of n to demonstrate the behaviour of the algorithm in the Worst, Best and Average Cases. Record the time taken to sort. Plot a graph of the time taken versus *n* on graph sheet. The elements can be read from a file or can be generated using the random number generator for large values of n. **Method:** A Divide & Conquer- Recursive algorithm implementation- Merge Sort **Theory Reference:** Module 2	2	CO1,CO3
4	**Title:** Urban planning of water supply networks **Problem Description:** Design an optimized water distribution network that minimizes the total cost of the infrastructure while ensuring reliable water supply to all areas in the given region. **Method:** Represent the problem in the form of a graph. Use the greedy Technique for a spanning tree which has a total minimum weight. **Theory Reference:** Module 3	2	CO2,CO3
5	**Title:** Optimal Road Network **Problem Description:** A local government wants to improve the road network between towns to enhance connectivity and reduce travel costs. The region consists of five towns that need to be connected by the most efficient road network possible, minimizing the total length of the roads while ensuring that each town is accessible from any other town. **Method:** A greedy Technique for a spanning tree which has a total minimum weight. Use Union-Find Data Structure to detect cycles during the execution of the algorithm. **Theory Reference:** Module 3	2	CO2,CO3

	PART-B		
6	**Title:** Optimal Road Routes **Problem Description:** You are the chief planner for a transportation department in a country with several cities. The cities are connected by a network of roads, each with a specific travel time (in hours). Your task is to determine the shortest travel time from a designated source city to all other cities. **Method:** Represent the problem in the form of a graph using adjacency matrix or adjacency list. Using the Single Source Shortest Path Algorithm find the shortest path from the source city to all other cities. **Theory Reference:** Module 3	2	CO2,CO3
7	**Title:** Metropolitan Subway System Optimization **Problem Description:** A metropolitan city is planning to optimize its subway system to ensure efficient travel for its commuters. The subway system consists of various stations connected by subway lines, and each line has a different travel time. The city's transportation authority wants to find the shortest travel time between all pairs of stations. **Method:** Use Dynamic programming technique to find the shortest travel time between all pairs of stations using the Floyd-Warshall algorithm **Theory Reference:** Module 3	2	CO2,CO3
8	**Title:** Optimal Product Selection for a Limited Shelf Space **Problem Description:** You are a store manager responsible for stocking products on the shelves. The store has limited shelf space, and you want to maximize the total profit by selecting the most valuable products to display. Each product has a specific weight and profit associated with it. **Method:** Solve the 0/1 Knapsack Problem using dynamic programming or other optimization techniques. The key idea is to build a table that stores the maximum value achievable for different weights and items. **Theory Reference:** Module 4	2	CO2,CO3

	Title: Event Budget Allocation **Problem Description:** An event planner is organizing a conference and has a fixed budget to allocate across various categories such as venue, catering, speakers, and marketing. Theplanner needs to ensure that the total expenditure does notexceed the budget while maximizing the quality of the event. Setof Expenses (with costs): Venue rental-$5000, Catering-$2000, Keynote Speaker-$3000, Marketing-$1500, Audio/Visual Equipment-$1000. Given Fixed budget-$8000 **Method:** Use backtracking technique to solve the problem. **Theory Reference:** Module 4	2	CO2,CO3
10	**Title:** Delivery Route Optimization. **Problem Description:** Imagine a delivery company that operates in a city with five key delivery locations. The company needs to design a route for their delivery vehicle that visits each location exactly once and returns to the starting point without retracing its steps. **Method:** Given a graph representing the locations as vertices and the roads between them as edges, determine whether a Hamiltonian Circuit exists. If it does, find the circuit. **Theory Reference: Module 4**	2	CO2,CO3

PART-C

Beyond Syllabus Virtual Lab Content

1. N-Queens problem
2. Topological Sorting
3. Heap Sort

Suggested Learning Resources:

Textbooks:

1	Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, 2009.Pearson.
2	Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press

Web links and Video Lectures (e-Resources):

1. http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS43.html
2. https://nptel.ac.in/courses/106/101/106101060/
3. http://elearning.vtu.ac.in/econtent/courses/video/FEP/ADA.html
4. http://cse01-iiith.vlabs.ac.in/
5. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms

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	Choose appropriate algorithm design techniques like divide and conquer, Decrease and conquer, Transform and conquer to develop solutions to the computational and complex problems.
CO2	Develop programs to solve computational problems using greedy method, Dynamic programming, Backtracking algorithm design strategies.
CO3	Compare algorithm design strategies by developing equivalent programs and observing running times for empirical analysis.

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	3	-	-	-	-	-	-	1	3	-	-
CO2	3	3	3	3	3	-	-	-	-	-	-	1	3	-	-
CO3	3	3	3	3	3	-	-	-	-	-	-	1	3	-	-

Question Paper Pattern:

1. **CIE Assessment Pattern:**

2. **SEE Assessment Pattern:**