Course Name: Design and Analysis of Algorithms Lab
Course Code: T7491
Faculty: Engineering
Course Credit: 1
Course Level: 4
Sub-Committee (Specialization): Computer Science
Learning Objectives:

The students will be able to:
Examine the working of various searching and sorting algorithms.
Formulate the upper and lower bounds on comparison-based search algorithms.
Programs for solving algorithmic problems based on graph traversals.
Solve shortest-path problem using greedy algorithm design strategy.
Formulate a combinatorial optimization problem using algorithm design methods such as greedy method, dynamic programming.
Devise a graph analysis algorithm for solving the all-pairs shortest-path problem.
Devise algorithm to study the optimal way to merge n initially sorted files.
Develop, design and analyze backtracking, branch and bound algorithms.

Books Recommended:

Book	Author	Publisher
Computer Algorithms/C++	Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran	Universities Press, Second Edition, 2007.
Design and Analysis of an MST-Based Topology Control Algorithm	IEEE INFOCOM	NA
Design and Analysis of Optimized Selection Sort Algorithm	International Journal of Electric & Computer Sciences IJECS	NA
Introduction to Algorithms	T. H. Cormen, C. E. Leiserson, R.L.Rivest, and C. Stein	Prentice Hall of India Pvt. Ltd., Second Edition, 2003.
Object Oriented Data Structures using C++	K.S. Easwarakumar	Vikas Publishing House pvt. Ltd., 2000.
The Design and Analysis of Computer Algorithms	Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman	Pearson Education, Fourth Edition, 1999.

Course Outline:

Sr. No.	Topic	Actual Teaching Hours	Contact Hours Equivalence
1	**WAP to search an element using linear search and binary search (recursive).** Analyze its complexity.	**2**	**2**
2	**WAP to perform Quick sort, Merge sort.** Display the partial pass-wise sorting done.	**4**	**4**
3	**Create implementations for the Kruskal's & Prim's Algorithm for finding the Minimum Cost Spanning Tree for a given graph.** You may input the graph from file and use adjacency matrix.	**4**	**4**
4	**Implement Dijkstra's Algorithm to find a path of minimum total weight (cost) from a starting node to all the other nodes in the weighted graph.**	**4**	**4**

5	**WAP to perform Knapsack problem using Greedy strategy and Dynamic Programming.**	**4**	**4**
6	**Implement Floyd's Algorithm to find a shortest path between each pair of vertices in a weighted graph.** Display the results as a distance matrix.	**4**	**4**
7	**WAP to perform Optimal Merge Pattern.**	**4**	**4**
8	**WAP to implement n-queens problem.**	**4**	**4**
	Total	**30**	**30**

Pre Requisites:

None

Evaluation:

Seminar

Viva

Pedagogy:

Practical using Dev C++, Borland C

Mini projects