DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

IV Semester: CSE / CSIT / CSE (CS)

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
ACSC15	**Core**	0	0	3	1.5	30	70	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 36**				**Total Classes: 36**		
Prerequisites: Programming for Problem Solving, Data Structures								

I. COURSE OVERVIEW:

Design and analysis of algorithms is the process of finding the computational complexity of algorithms. It helps to design and analyze the logic on how the algorithm will work before developing the actual code for a program. It focuses on introduction to algorithm, asymptotic complexity, sorting and searching using divide and conquer, greedy method, dynamic programming, backtracking, branch and bound. NP-hard and NP-complete problems. The applications of algorithm design are used for information storage, retrieval, transportation through networks, and presentation to users.

II. COURSE OBJECTIVES:

The students will try to learn :
I.   The problem analysis and design the solution for the given problem.
II. The suitable algorithm for the given real world problem.

III. COURSE SYLLABUS:

WEEK-1: QUICK SORT

Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

WEEK – 2: MERGE SORT

Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

WEEK – 3: KNAPSACK PROBLEM

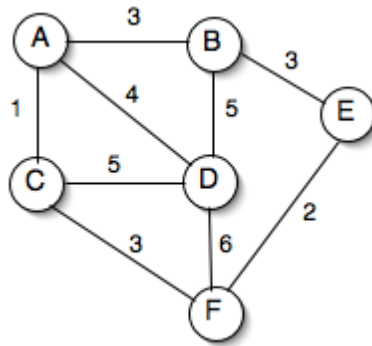Implement 0/1 Knapsack problem using Dynamic Programming.

WEEK – 4: SHORTEST PATHS ALGORITHM

From a given vertex in a weighted connected graph, find shortest paths from 0 to other vertices using Dijkstra's algorithm.
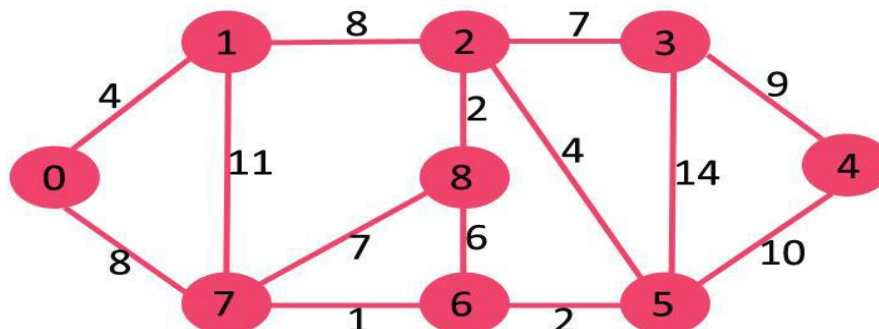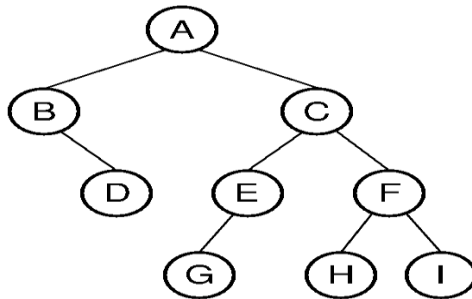
WEEK – 5: MINIMUM COST SPANNING TREE
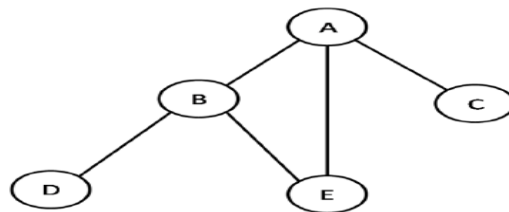Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

WEEK – 6: TREE TRAVESRSALS
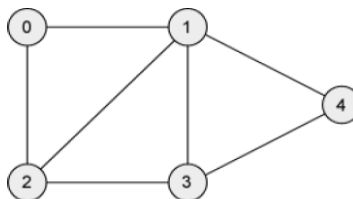Perform various tree traversal algorithms for a given tree.

WEEK – 7: GRAPH TRAVERSALS
a. Print all the nodes reachable from a given starting node in a digraph using BFS method.

b. Check whether a given graph is connected or not using DFS method.

WEEK – 8: SUM OF SUB SETS PROBLEM
Find a subset of a given set S = {sl, s2,....., sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2, 6} and {1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.

WEEK – 9: TRAVELLING SALES PERSON PROBLEM
Implement any scheme to find the optimal solution for the Traveling Sales Person problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
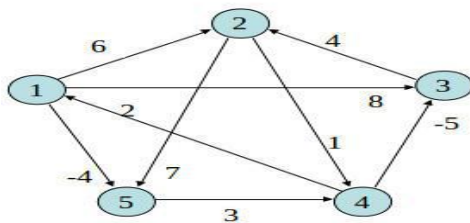
WEEK 10: MINIMUM COST SPANNING TREE
Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
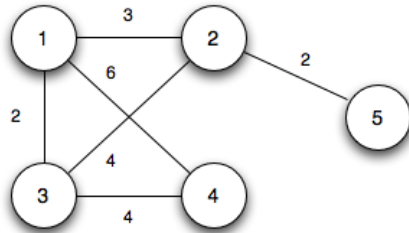
WEEK – 11: ALL PAIRS SHORTEST PATHS

Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.

	1	2	3	4	5
1	0	6	8	∞	–4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	–5	0	∞
5	∞	∞	∞	3	0

WEEK – 12: N QUEENS PROBLEM
Implement N Queen's problem using Back Tracking.

IV. REFERENCE BOOKS:
1. Levitin A, "Introduction to the Design and Analysis of Algorithms", Pearson Education, 2008.
2. Goodrich, M.T. R Tomassia, "Algorithm Design foundations Analysis and Internet Examples", John Wiley and Sons, 2006.
3. Base Sara, Allen Van Gelder, "Computer Algorithms Introduction to Design and Analysis", Pearson, 3rd Edition, 1999.

V. WEB REFERENCE:
1. http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html
2. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
3. http://www.facweb.iitkgp.ernet.in/~sourav/daa.html