CSPC2206 DESIGN AND ANALYSIS OF ALGORITHMS LAB. (0-0-3)

1. Sort a given set of n integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. 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. Demonstrate how the brute force method works along with its time complexity analysis: worst case, average case and best case.

2. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. 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. Demonstrate how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

3. Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. 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. Demonstrate how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

4. Write a program to solve Knapsack problem using Greedy method.

5. Write a program to find shortest paths to other vertices from a given vertex in a weighted connected graph, using Dijkstra's algorithm.

6. Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.

7. Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.

8. Write a program to solve All-Pairs Shortest Paths problem using Floyd's algorithm.
    Write a program to solve Travelling Sales Person problem using Dynamic programming.
    Write a program to solve 0/1 Knapsack problem using Dynamic Programming method.

9. Design and implement C++/Java Program to 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}. Display a suitable message, if the given problem instance doesn't have a solution.

10. Design and implement a program to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.