UNIVERSITY OF ENGINEERING & MANAGEMENT

Good Education, Good Jobs

UNIVERSITY : UNIVERSITY OF ENGINEERING & MANAGEMENT

DEPARTEMNT : CSE Department

COURSE : B.TECH

SEMESTER : 4TH SEMESTER

SUBJECT : Design & Analysis of Algorithm Lab (PCCCSE494)

Pre-requisites:

Data Structures

Course Objectives:

- To familiarize the students with the designing of an algorithm.
- To familiarize the students with the analysis of an algorithm.

Course Outcomes:

- Able to design an algorithm
- Able to analyze an algorithm

All programs to be implemented in C & Python Language

Instruction:

1. Mention the following thing using A4 pages:
   a. Program Objective
   b. Algorithm
   c. Program Code.
   d. Output

2. Handwritten assignment solutions are required. **No printout is allowed.**
3. Submit all the assignments using the channel file.
4. One Index page must be attached there.
5. **Late submission will not be entertained. Maintain the deadline.**

N.B- Write the Algorithm of the function only, not the algorithm of whole program.

WEEK 1

Searching & Sorting(With Analysis)

Lab Assignments:

2. 1. Write a program to search an element in an Array using dynamic memory allocation. Apply different algorithm separately. Show the running time complexity w.r.t different input cases (best/average/worst). Finally comment that which one is better and why.

3. Write a program to perform insertion sort using dynamic memory allocation. Show the running time complexity w.r.t different input cases. Finally comment that whether it is matching with the expected complexity of O(n^2) or not.

4. Write a program to perform selection sort using dynamic memory allocation. Show the running time complexity w.r.t different input cases. Finally comment that whether it is matching with the expected complexity of O(n^2) or not.

5. Write a program to perform bubble sort using dynamic memory allocation. Show the running time complexity w.r.t different input cases. Finally comment that whether it is matching with the expected complexity of O(n^2) or not.

6. Critically comment that if in all the above program instead of dynamic array, if we use static array then whether that would effect the complexity or not.

WEEK 2

Divide & Conquer 1(sorting algorithms)

Lab Assignments:

1. 1. Write a program to sort a list of element in an Array using Quick Sort. Show the running time complexity w.r.t different input cases (best/average/worst).

2. Write a program to sort a list of element in an Array using Merge Sort. Show the running time complexity w.r.t different input cases (best/average/worst).

3. Write a program to sort a list of element in an Array using Heap Sort. Show the running time complexity w.r.t different input cases (best/average/worst).

WEEK 3

Divide & Conquer II

Lab Assignments:

1. 1. Write a program to find maximum and minimum element present in a list of element using different approach. Show the running time complexity w.r.t different input cases (best/average/worst). Finally comment which one is better & why.

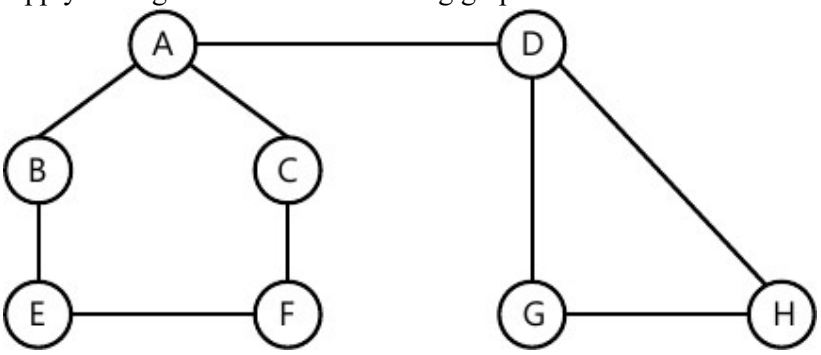
2. Write a program to implement Strassen's matrix multiplication.

WEEK 4

Greedy Method I

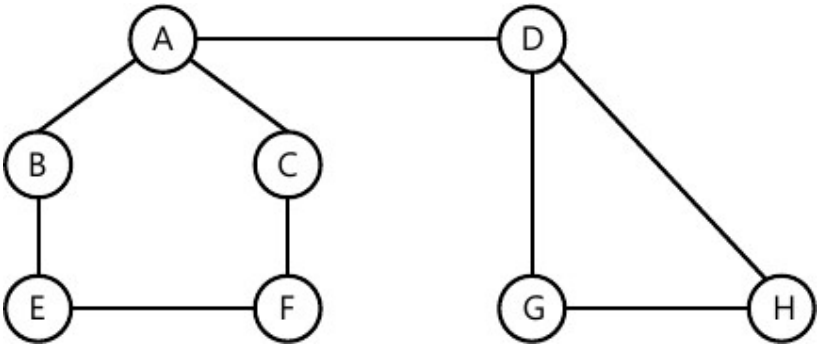
Lab Assignments:

1. 1. Write a program to Implement the concept of Depth First Search algorithm. Apply the algorithm on the following graph:



Show the running time complexity w.r.t different input cases (best/average/worst).

2. Write a program to implement the concept of Breadth First Search algorithm. Apply the algorithm on the following graph:



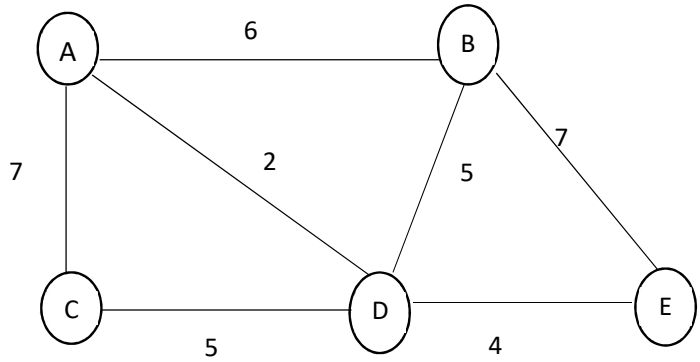
Show the running time complexity w.r.t different input cases (best/average/worst).

WEEK 5

Greedy Method II (Spanning Tree, Knapsack Problem)

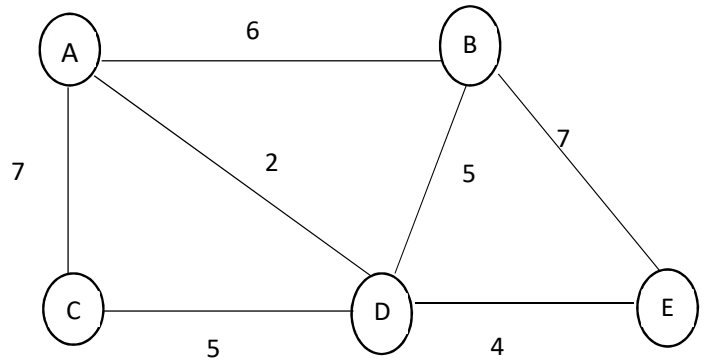
Lab Assignments:

1. 1. Write a program to implement the concept of Kruskal's algorithm. Apply it to find the MST for the following graph.



Show the running time complexity w.r.t different input cases(best/average/worst).

2. Write a program to implement the concept of Prim's algorithm. Apply it to find the MST for the following graph.



Show the running time compıexity w.r.t different input cases (best/average/worst).

3. Solve the following instances of the 0/1 Knapsack problem given the /Knapsack capacity W=7

Item	Weight	Value
1	3	10
2	2	12

Implement a program to solve the above problem. Confirm the running time of the algorithm with proper analysis.

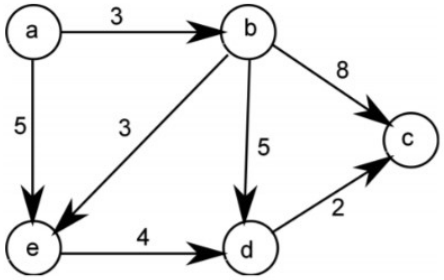
4. Implement a greedy algorithm and find the optimal solution to the fractional Knapsack instance n=7, m=15, (p1 p2 p3 p4 ……p7)=(1,5,10,10,5,7,8) and (w1,w2…….w7)=(3,2,5,7,2,8, 4). Confirm the running time of the algorithm with proper analysis.

WEEK 6

Greedy Method III (Job Sequencing with Deadline, Shortest Path)

Lab Assignments:

1. Solve the following instance of "job scheduling with deadlines" problem: n = 7, profits (p1, p2, p3, p4, p5, p6, p7) = (3, 5, 20, 18, 1, 6, and 30) and deadlines (d1, d2, d3, d4, d5, d6, d7) = (1, 3, 4, 3, 2, 1, and 2). Schedule the jobs in such a way to get maximum profit. (Here only one machine is available).
2. Apply Dijkstra's Algorithm to find out the shortest path from node 'a' to all other nodes of the graph.



WEEK 7

Dynamic Programming1 (Matrix Chain Multiplication)

Lab Assignments:

1. Find out the minimal number of scaler Multiplication needed to multiply these matrices. The dimension of the matrices are 8 x 12, 12 x 13, 13 x 14, and 14 x 19. Apply Dynamic Programming (Bottom-up) approach. Here input is an array storing dimensions of the matrices like

p = | 8 | 12 | 13 | 14 | 19 |

WEEK 8

Dynamic Programming-II (0/1 Knapsack and Longest common Subsequence)

Lab Assignments:

1. For the given set of items and knapsack capacity = 5 kg, find the optimal solution for the 0/1 knapsack problem making use of dynamic programming approach.

Item	Weight	Value
1	2	3
2	3	4
3	4	5
4	5	6

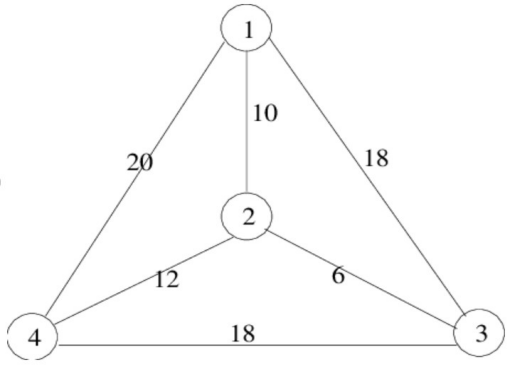
2. Using dynamic programming determine longest sequence of (ABCBADCDF) and (CBAF).

WEEK 9

Dynamic Programming-III (Travelling Sales Man problem, Shortest Path)

Lab Assignments:

1. A circuit which passes through every vertex exactly once is called a Hamilton circuit. A minimum weight Hamilton circuit is a Hamilton circuit that has the smallest possible weight of all Hamilton circuits. In graph theory terms, the *TSP* is the problem of finding a minimum weight Hamilton circuit. Apply dynamic programming to find out the minimum weight Hamiltonian circuit.



2. Apply Floyd Warshall algorithm to find out shortest path between every pairs of vertices of the graph shown below. The graph is stored in adjacency matrix.

$$\begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

WEEK 10

Backtracking

Lab Assignments:

1. N - Queens problem is to place n - queens in such a manner on an n x n chessboard that no queens attack each other by being in the same row, column or diagonal. Apply Backtracking to generate all possible valid assignment of a 4- Queen problem.

2. We are given a graph, we need to assign colors to the vertices of the graph. In the graph coloring problem, we have a graph and m colors, we need to find a way to color the vertices of the graph using the m colors such that any two adjacent vertices are not having the same color.
   Apply Backtracking to generate all possible valid coloring of the following graph. Here m = 3.