COCSC06/CACSC06: Design and Analysis of Algorithms

Course Coordinator: Dr. Abhinav Tomar

Course Structure: 4 Credits

Prerequisites: Data Structure

Learning Objectives: After completing this course, students will be able to:

1. To be able to analyze a problem in terms of processing steps, time and space complexity.
2. To be able to design and implement the algorithms for any given application.
3. To be able to develop software applications using various programming languages in collaborative groups.
4. To apply the principles in solving problems encountered in career or real life situations.

Suggested Readings:

- Berman. Paul, "Algorithms, Cengage Learning".
- Richard Neopolitan, Kumar SS Naimipour, "Foundations of Algorithms"
- T .H . Cormen, C .E .Leiserson, R .L .Rivest "Introduction to Algorithms", PHI.
- E. Horowitz, S. Sahni, and S. Rajsekaran, "Funadmentals of Computer Algorithms," Galotia Publication

S. No.	Practical Name
Practical 1	Dr. Johnson has collected the blood pressure rate of a tuberculosis infected patient. He has stored the collected data for observing the blood pressure behaviour of a given patient. During the operation of the patient one of the doctors has asked Dr. Johnson to tell the maximum and minimum blood pressure of the patient. So, this is the routine work of Dr. Johnson. So, he wishes to have a divide conquer based approach which takes lesser comparison to achieve the desire results. Perform the following sorting algorithms-Merge sort, Quick sort, Bubble sort, insertion sort and selection sort.
Program 2	Perform the following sorting algorithms-Radix sort, Count sort, Bucket sort, and Shell sort.

Practical 3	Perform searching algorithm-Linear and Binary search.
Practical 4	Perform tower of Hanoi.
Practical 5	Write a program for inserting elements in: i. AVL tree ii. Red-Black Tree iii. BST
Practical 6	Write a program for deleting elements in: i. AVL tree ii. Red-Black Tree iii. BST
Practical 7	Given the root of a binary tree, return the maximum height of the tree. A binary tree's maximum height is the number of nodes along the longest path from the root node down to the farthest leaf node.
Practical 8	Find maximum and minimum of array using the dynamic programing.
Practical 9	Given a set of N jobs where each job i has a deadline and profit associated with it. Each job takes 1 unit of time to complete and only one job can be scheduled at a time. We earn the profit associated with the job if and only if the job is completed by its deadline. Find the number of jobs done and the maximum profit.
Practical 10	Given weights and values of N items, we need to put these items in a knapsack of capacity W to get the maximum total value in the knapsack. Note: Unlike 0/1 knapsack, you are allowed to break the item.
Practical 11	Write a program for the fractional and dynamic knapsack problem
Practical 12	Implement Minimum Spanning trees: Prim's algorithm and Kruskal's algorithm Input a directed graph G = (V, E) where vertices V are represented as alphabetical numbers. Run the DFS-based topological ordering algorithm on the graph. Whenever you have a choice of vertices to explore, always pick the one that is alphabetically first. Let G = (V, E) be a directed graph. Vertices u and v are strongly connected if there are u v and v u paths in G. A strongly connected component is a set

	of vertices C ⊆ V such that u, v is strongly connected for all u, v ∈ C (and no other vertices are strongly connected to a vertex u ∈ C.) Design an algorithm to identify all strongly connected components in G.
Practical 13	Given a sequence of matrices, find the most efficient way to multiply these matrices together. The efficient way is the one that involves the least number of multiplications. The dimensions of the matrices are given in an array arr[] of size N (such that N = number of matrices + 1) where the ith matrix has the dimensions (arr[i-1] x arr[i]).
Practical 14	Write a program to implement Strassen's Matrix Multiplication. Implement Matrix chain multiplication (MCM) using dynamic programming, you need to estimate the minimum number of operations and assign the parentheses for multiplying multiple matrices. Input: 6 (Number of matrices, followed by matrix size) 2 4 4 3 3 6 6 5 5 2 2 1 Output: 23XX22 (Number of operations) (A1 ((A2 A3) (A4 (A5 A6))))
Practical 15	Write a program to implement Longest Common Subsequence.
Practical 16	Implement Travelling Salesman Problem.

Evaluation Scheme:

PCA	15
PES	15
Total	**30**

Dr. Abhinav Tomar