Teaching Plan (2021-2022)

Name of the Faculty: Ms. Maulein Pathak

Name of the Course: BSc (H) Computer Science

Semester : IV Sec (if any): B

Title of the Paper : Design and Analysis of Algorithms

Course Learning Outcomes:

On successful completion of this course, the student will be able to:

1. Given an algorithm, identify the problem it solves.
2. Write algorithms choosing the best one or a combination of two or more of the algorithm design techniques: Iterative, divide-n-conquer, Greedy, Dynamic Programming using appropriate data structures.
3. Write proofs for correctness of algorithms.
4. Re-write a given algorithm replacing the (algorithm design) technique used with a more appropriate/efficient (algorithm design) technique.

Month	Topics Covered	References	Practicals
January	Graph Algorithms - Ch 3 Divide and Conquer Techniques – Ch 7 Assignment	[2] [1]	3,7,8
February	Iterative Techniques and Divide and Conquer Techniques - Ch 2 Heapsort - Ch 6 Linear Time Sorting Algorithms – Ch 8 Test	 [1] [1] [1]	1,2,4,5
March	Median Order Statistics – 9.1, 9.2 Greedy Algorithms 4.1, 4.2, 4.4, 4.5(except reverse delete), 4.6	[1] [2]	6,9
April	Dynamic Programming 6.1, 6.2, 6.4 Amortized Analysis 17.1, 17.2, 17.3 Test Assignment	[2] [1]	10,11

References:

[1] Introduction to Algorithms, Cormen, Leiserson, Rivest & Stein, 3rd edition, PHI

[2] Algorithm Design, Kleinberg and Tardos, Pearson Publication

Practical List

1. i. Implement Insertion Sort (The program should report the number of comparisons)
   ii. Implement Merge Sort (The program should report the number of comparisons)
2. Implement Heap Sort(The program should report the number of comparisons)
3. Implement Randomized Quick sort (The program should report the number of comparisons)
4. Implement Radix Sort
5. Implement Bucket Sort
6. Implement Randomized Select
7. Implement Breadth-First Search in a graph
8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph (Both Prims and Kruskals)
10. Write a program to solve the weighted interval scheduling problem
11. Write a program to solve the 0-1 knapsack problems

For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.