Indira Gandhi Delhi Technical University for Women

(Established by Govt. of Delhi vide Act 09 of 2012)

(Formerly Indira Gandhi Institute of Technology)

Kashmere Gate, Delhi - 110006

LABORATORY MANUAL

[2013-2014]

for

DESIGN AND ANALYSIS OF ALGORITHMS

MCA – 252

TABLE OF CONTENTS

OBJECTIVES OF THE LAB

This manual is intended for the Fourth Semester students of MCA in the subject of Design and Analysis of Algorithms.

This manual typically contains practical/Lab Sessions related to the subject to enhance understanding.

The subject deals with the concepts of designing and analyzing computer algorithms.

1. The “design” deals with the description of algorithm at an abstract level by means of a pseudo language, and Proof of correctness which states that the algorithm solves the given problem in all cases.

2. The “analysis” deals with performance evaluation (complexity analysis).

RULES TO BE FOLLOWED IN LAB

1. Plagiarism in any form will not be tolerated in the lab work and project.

2. The students must submit the files for checking strictly as advised by the instructor in the lab without fail.

3. The students are supposed to mandatorily appear for the internal assessment on the day as announced in the lab.

4. Any student found indulged in malpractices, disobedience will be debarred from the lab without any warning for a part/whole semester.

5. It is expected and strongly encouraged that students utilize their Lab hours efficiently.

6. Every effort should be made to complete the lab work in the assigned lab only.

7. Every student work on assigned system only. Any problem in the system must be reported to the lab in charge in advance. Reporting and ensuring the physical condition of the system is the responsibility of the students. No exercise in this regard will be entertained.

Syllabus

Code No.: MCA 202	L T C
Paper: Design and Analysis of Algorithms	3 1 4

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVE: In this course, students will learn how:

- *To design new algorithms based on standard algorithm-design strategies.*
- *To analyze the time and space usage and correctness of new algorithms based on standard algorithm-analysis techniques.*
- *To apply and adapt fundamental algorithms (sorting, searching, order statistics, graph algorithms) to new situations.*
- *To solve problems and to express your solutions using the language and concepts of algorithms and its mathematical tools.*

PRE-REQUISITES

- Programming in C
- Data Structure in C
- Discrete Mathematics

UNIT - I

Notion of Algorithm, Growth of functions, Summations, Recurrences: The substitution method, The iteration method, Asymptotic Notations and Basic Efficiency Classes. Use of Big O , θ, Ω in analysis .Mathematical Analysis of few Non-recursive and Recursive Algorithms , Proof of Correctness. [No of Hrs.: 10]

UNIT - II

Sorting and Searching Techniques , Selection Sort , Bubble Sort , Insertion Sort , Sequential Search Binary Search , Depth first Search and Breadth First Search. , Balanced Search trees , AVL Trees , Red-Black trees , Heaps and Heap sort , Hash Tables, disjoint set and their implementation , Divide and conquer Paradigm of Problem solving , complexity analysis and understanding of Merge sort , Quick Sort , Binary Search Trees, Sorting in linear time, Medians and Order statistics. [No of Hrs.: 10]

UNIT - III:

Greedy Techniques, Prim's Algorithm, Kruskal's Algorithm , Dijkstra's and Bellman Ford Algorithm , Huffman trees. Knapsack Problem , Dynamic Programming paradigm , Warshall 's and Floyd's Algorithm , Optimal Binary Search trees , Matrix multiplication Problem , 0/1 Knapsack Problem , maximum network flow problem , naive string matching algorithm , string matching with finite automata Knuth morris Pratt algorithm , The Rabin-Karp Algorithm.

[No of Hrs.: 10]

UNIT - IV

Backtracking, n-Queen's Problem, Hamiltonian Circuit problem, Subset-Sum problem, Branch and bound, Assignment problem, travelling salesman problem. Introduction to Computability, Polynomial-time verification, NP-Completeness and Reducibility, NP-Completeness Proof, NP-Complete problems, Proof of cook's theorem. [No of Hrs.: 10]

TEXT BOOKS

1. Jon Kleinberg and Eva Tardos, "Algorithm Design", Pearson Edition, 2006.
2. Richard Neapolitan and Kumarss Naimipour, "Foundations of Algorithms", Jones & Bartlett, 2004.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms" PHI, 3rd Ed., 2009.

REFERENCES:

1. Johnsonbaugh , "Algorithms", Pearson, 2004.
2. Anany Levitin, "Introduction to the Design and Analysis of Algorithm", Pearson Education, 2003.
3. Sara Baase and Allen Van Gelder, "Computer Algorithms - Introduction to Design and Analysis", Pearson Education, 2003.
4. A.V. Aho, J. E. Hopcroft and J.D.Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education, 2003.
5. R. S. Salaria, Khanna, "Data Structure & Algorithms", Book Publishing Co. (P) Ltd., 2002.
6. R. Panneerselvam, "Design and Analysis of Algorithm", PHI, 2007.
7. Steven S. Skiena , "Algorithm Design Manual", Springer, 1998.
8. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamental of Computer Algorithms", OrientLongman , 2006.

GUIDELINES

1. In file, every lab experiment will consist of the following four components (each on separate page):
    - One page of description of the theoretical concepts of the experiment.
    - Algorithm for the problem.
    - Program based on the algorithm designed by the student.
    - Output + Aysmptotic complexity analysis of the algorithm.

2. The students also have to save their programs in a directory on the machine they are assigned in the lab (details will be given in lab).

3. The students must ensure that the schedule for completion of the lab experiments is strictly adhered to. Delay in performing practicals or submission of file will suggest underperformance by the student and will inevitably lead to penalty in marks.

4. Students are expected to read the lab manual carefully and come prepared with the theoretical concepts of the experiment they are going to perform in the lab so as to ensure effective utilization of lab hours to finish the experiment timely and allow students to work on more complicated cases based on the experiment which will be discussed in lab once the students have completed the basic experiment.

5. It must be noted that the every lab experiment once completed will be followed by implementing the same on complex data sets and problems. The same will be announced in lab in detail.

6. Students may write programs in any language, support for which is available on their machines in lab. It is however advisable to do the programming implementation of the experiments using C language on Linux.

LIST OF EXPERIMENTS

1. Write a program to perform Binary Search for maximum and minimum in a list using divide and conquer strategy.

2. Write a program to demonstrate Masters Theorem.

3. Sort a list of numbers in ascending order using :
   (a) Selection Sort
   (b) Insertion Sort
   (c) Heap Sort
   (d) Merge Sort
   (e) Quick Sort

4. Implement Depth First Search and Breadth First Search, compare their performance.

5. Write a program to implement Tower of Hanoi.

MINOR - I

6. Write a program to find minimum cost of spanning tree using Prims Algorithm.

7. Write a program to demonstrate Kruskal's Algorithm.

8. Write a program to find the shortest path in a given graph using Dijkstra algorithm.

9. Write a program to find Adjacency matrix for a given graph.

10. Write a program to perform All pairs shortest path problem.

11. Write a program to perform Matrix Chain Multiplication.

12. Write a program to find Optimal solution for a Knap Sack Problem using Greedy Method.

MINOR - II

13. Write a program to solve Eight Queens problem using Back Tracking Technique.

14. Write a program to implement traveling salesperson problem using dynamic programming.

15. Write a program to demonstrate Longest Common Subsequence.

Evaluation Policy

1. The will be regular supervision and assessment of the students during the lab hours throughout the semester.

2. The student must complete the lab work and get the files submitted and checked as advised in the lab by the instructor.

3. Internal Assessment will be done before the Minor exams which will include viva, practical and project status report.

4. The students are expected to come up with novice approaches in problem solving especially while working on the project which will earn them marks accordingly.

Project

The students will be required to work in group of n (2<n<6) students for the project work which is primarily going to be designed to evaluate the student's understanding of the subject and also inculcate the qualities of applying the theoretical knowledge to the practical real world applications. The students are expected to not only implement the use of subject knowledge but also think out of the box and look for proposing new issues, problems and their solutions in the subject domain.

The details of the same will be discussed in the lab. The projects will be assigned before Minor I and evaluated throughout the semester in the lab for progress.
Final evaluation will be done 1 week after the Minor II.

The students are allowed and encouraged to refer the study material in books, refrences, internet etc but Plagiarism from any possible source is strictly prohibited.