

MASTER OF COMPUTER APPLICATIONS (MCA_NEW)

**ASSIGNMENTS
OF MCA_NEW (2Yrs) PROGRAMME
SEMESTER-I**

(January - 2025 & July - 2025)

MCS-211, MCS-212, MCS-213, MCS-214, MCS-215

MCSL-216, MCSL-217



**SCHOOL OF COMPUTER AND INFORMATION SCIENCES
INDIRA GANDHI NATIONAL OPEN UNIVERSITY
MAIDAN GARHI, NEW DELHI – 110 068**

CONTENTS

| Course Code | Assignment No. | Submission-Schedule | | Page No. |
|-------------|----------------------------------|------------------------------|--------------------------------|----------|
| | | For January-June Session | For July-December Session | |
| MCS-211 | MCA_NEW(I)/211/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 3 |
| MCS-212 | MCA_NEW(I)/212/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 6 |
| MCS-213 | MCA_NEW(I)/213/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 8 |
| MCS-214 | MCA_NEW(I)/214/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 9 |
| MCS-215 | MCA_NEW(I)/215/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 12 |
| MCSL-216 | MCA_NEW(I)/L-216/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 14 |
| MCSL-217 | MCA_NEW(I)/L-217/Assignment/2025 | 30 th April, 2025 | 31 st October, 2025 | 16 |

Important Notes

1. Submit your assignments to the Coordinator of your Study Centre on or before the due date.
2. Assignment submission before due dates is compulsory to become eligible for appearing in corresponding Term End Examinations. For further details, please refer to Programme Guide of MCA (2Yrs).
3. To become eligible for appearing the Term End Practical Examination for the lab courses, it is essential to fulfill the minimum attendance requirements as well as submission of assignments (on or before the due date). For further details, please refer to the Programme Guide of MCA (2yrs).
4. The viva voce is compulsory for the assignments. For any course, if a student submitted the assignment and not attended the viva-voce, then the assignment is treated as not successfully completed and would be marked as ZERO.

| | | |
|----------------------------------|---|---|
| Course Code | : | MCS-211 |
| Course Title | : | Design and Analysis of Algorithms |
| Assignment Number | : | MCA_NEW(I)/211/Assign/2025 |
| Maximum Marks | : | 100 |
| Weightage | : | 30% |
| Last Dates for Submission | : | 30th April 2025 (for January Session) 31st October 2025 (for July Session) |

This assignment has four questions (80 Marks). Answer all questions. The remaining 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding assignments given in the Programme guide for the presentation format.

- Q1:**
- Design and develop an efficient algorithm to find the list of prime numbers in the range 501 to 2000. What is the complexity of this algorithm? **(2 Marks)**
 - Differentiate between Cubic-time and Factorial-time algorithms. Give example of one algorithm each for these two running times. **(2 Marks)**
 - Write an algorithm to multiply two square matrices of order $n \times n$. Also explain the time complexity of this algorithm. **(2 Marks)**
 - What are asymptotic bounds for analysis of efficiency of algorithms? Why are asymptotic bounds used? What are their shortcomings? Explain the Big O and Big Θ notation with the help of a diagram. Find the Big O-notation and Θ -notation for the function:
$$f(n) = 100n^4 + 1000n^3 + 100000$$
 (4 Marks)
 - Write and explain the Left to Right binary exponentiation algorithm. Demonstrate the use of this algorithm to compute the value of 3^{29} (Show the steps of computation). Explain the worst-case complexity of this algorithm. **(4 Marks)**
 - Write and explain the Bubble sort algorithm. Discuss its best and worst-case time complexity. **(3 Marks)**
 - What are the uses of recurrence relations? Solve the following recurrence relations using the Master's method **(3 Marks)**

- $T(n) = 4T\left(\frac{n}{4}\right) + n^1$
- $T(n) = 4T\left(\frac{3n}{4}\right) + n^1$

- Q2:**
- What is an Optimisation Problem? Explain with the help of an example. When would you use a Greedy Approach to solve optimisation problem? Formulate the Task Scheduling Problem as an optimisation problem and write a greedy algorithm to solve this problem. Also, solve the following fractional Knapsack problem using greedy approach. Show all the steps. **(4 Marks)**

Suppose there is a knapsack of capacity 20 Kg and the following 6 items are to be packed in it. The weight and profit of the items are as under:

$$(p_1, p_2, \dots, p_6) = (30, 16, 18, 20, 10, 7)$$

$$(w_1, w_2, \dots, w_6) = (5, 4, 6, 4, 5, 7)$$

Select a subset of the items that maximises the profit while keeping the total weight below or equal to the given capacity.

- b) Assuming that data to be transmitted consists of only characters 'a' to 'g', design the Huffman code for the following frequencies of character data. Show all the steps of building a Huffman tree. Also, show how a coded sequence using Huffman code can be decoded. (4 Marks)
a:5, b:25, c:10, d:15, e:8, f:7, g:30
- c) Explain the Merge procedure of the Merge Sort algorithm. Demonstrate the use of recursive Merge sort algorithm for sorting the following data of size 8: [19, 18, 16, 12, 11, 10, 9, 8]. Compute the complexity of Merge Sort algorithm. (4 Marks)
- d) Explain the divide and conquer approach of multiplying two large integers. Compute the time complexity of this approach. Also, explain the binary search algorithm and find its time complexity. (4 Marks)
- e) Explain the Topological sorting with the help of an example. Also, explain the algorithm of finding strongly connected components in a directed Graph. (4 Marks)

Q3: Consider the following Graph:

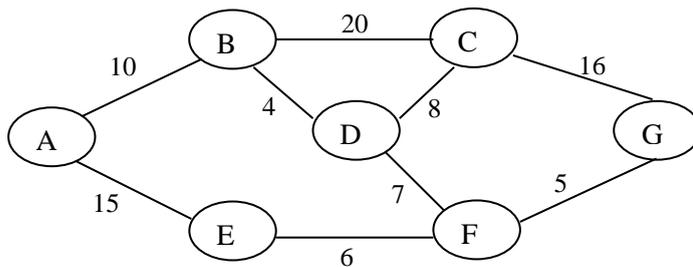


Figure 1: A sample weighted Graph

- a) Write the Prim's algorithm to find the minimum cost spanning tree of a graph. Also, find the time complexity of Prim's algorithm. Demonstrate the use of Kruskal's algorithm and Prim's algorithm to find the minimum cost spanning tree for the Graph given in Figure 1. Show all the steps. (4 Marks)
- b) Write the Dijkstra's shortest path algorithm. Also, find the time complexity of this shortest path algorithm. Find the shortest paths from the vertex 'A' using Dijkstra's shortest path algorithm for the graph given in Figure 1. Show all the steps of computation. (4 Marks)
- c) Explain the algorithm to find the optimal Binary Search Tree. Demonstrate this algorithm to find the Optimal Binary Search Tree for the following probability data (where p_i represents the probability that the search will be for the key node k_i , whereas q_i represents that the search is for dummy node d_i . Make suitable assumptions, if any) (6 Marks)

| i | 0 | 1 | 2 | 3 | 4 |
|----------|-------------|-------------|-------------|-------------|-------------|
| p_i | | 0.10 | 0.15 | 0.20 | 0.10 |
| q_i | 0.05 | 0.10 | 0.10 | 0.10 | 0.10 |

- d) Given the following sequence of chain multiplication of the matrices. Find the optimal way of multiplying these matrices: **(2 Marks)**

| Matrix | Dimension |
|---------------|------------------|
| A1 | 10×15 |
| A2 | 15×5 |
| A3 | 5×20 |
| A4 | 20×10 |

- e) Explain the Rabin Karp algorithm for string matching with the help of an example. Find the time complexity of this algorithm. **(4 Marks)**

- Q4:** a) Explain the term Decision problem with the help of an example. Define the following problems and identify if they are decision problem or optimisation problem? Give reasons in support of your answer. **(4 Marks)**
- (i) Travelling Salesman Problem
 - (ii) Graph Colouring Problem
 - (iii) 0-1 Knapsack Problem

- b) What are P and NP class of Problems? Explain each class with the help of at least two examples. **(4 Marks)**

- c) Define the NP-Hard and NP-Complete problem. How are they different from each other. Explain the use of polynomial time reduction with the help of an example. **(4 Marks)**

- d) Define the following Problems: **(8 Marks)**
- (i) SAT Problem
 - (ii) Clique problem
 - (iii) Hamiltonian Cycle Problem
 - (iv) Subset Sum Problem