MIT | Arts, Commerce & Science College

An Autonomous College Affiliated to Savitribai Phule Pune University

Accredited by NAAC with "A" Grade

SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS

DEPARTMENT OF SCIENCE AND COMPUTER SCIENCE

FY M.Sc.(Computer Science)

Semester II

AY 2024-25

Course Code: 2412MJCP204

Course Name: Lab course on Design & Analysis of Algorithm

Name of the Student: _____

Roll Number: _____

MIT | Arts, Commerce & Science College

An Autonomous College Affiliated to Savitribai Phule Pune University

Accredited by NAAC with "A" Grade

SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS

DEPARTMENT OF SCIENCE AND COMPUTER SCIENCE

CERTIFICATE

This is to certify that Mr./ Miss. _____

Of F.Y M.Sc. (Computer Science) Division: _____ Roll Number: _____ Exam Seat Number: _____

has satisfactorily completed his/her practical's in the Course: **Lab course on Design & Analysis of Algorithm** Course Code: **2412MJCP204**. As laid down by the MAEER'S MIT Art's, Commerce and Science College Alandi (D), Pune as an Autonomous, Affiliated to Savitribai Phule Pune University, Pune for the academic year 2024-25

Date:

Course Incharge: **Signature of HOD:**

Signature of Internal **Signature of External**

Index: Assignment Completion Sheet

Sr. No.	Assignment Name	Total Marks	Assigned Marks
1	Basic Algorithm	5	
2	Sorting Algorithm Divide & Conquer Algorithm	5	
3	Greedy Method & Dynamic Programming	5	
4	Graphs	5	
5	Backtracking and B&B	5	

1.Basic Algorithm

Learning Outcome:

- Develop problem-solving skills by breaking down tasks and implementing algorithms.
- Gain proficiency in using arrays and matrices for data manipulation and operations.
- Understand and implement key algorithms like GCD, LCM, Fibonacci, and searching.
- Improve logical thinking through recursion, iteration, and mathematical concepts.
- Strengthen Java programming skills by working with functions, loops, and user input.

Program for Practice:

1. Write a program to find the factorial of a number using a function.
2. Write a program to find the factorial of a number using a recursive function.
3. Write a program to read array from user and print it on the screen
4. Write a program to read a 3X3 matrix from the user and print it.

Assignments:

1. Write a program to read an array from the user and find maximum and minimum numbers using the function.
2. Write a program to read a 3X3 matrix from the user and print its row, column and diagonal sum using functions .
3. Write a program to perform Linear Search using functions.
4. Write a java program to implement Fibonacci series using functions.
5. Write a program to find GCD of 2 numbers using functions.
6. Write a program to find LCM of 2 numbers using functions.
7. Write a program to perform matrix Addition using functions.
8. Write a program to perform matrix Multiplication using Iterative approach.
9. Write a program to perform matrix Multiplication using Recursive Approach.
10. Write a java program to implement Fibonacci series recursive using functions.

2. Sorting Algorithm & Divide & Conquer

Learning Outcome:

- Learned how to sort numbers using different sorting algorithms.
- Understood how binary search works and its efficiency.
- Practiced reading input from the user and files in Java.
- Analyzed the time complexity of sorting and searching methods.
- Implemented Strassen's matrix multiplication for faster computation.

Assignments:

1. Write programs in Java to sort a list of n numbers in ascending order using selection sort. (Use Dynamically initialized array)
2. Write a program to perform Binary Search using functions and determine the time complexity for the same..
3. Write programs in Java to sort a list of n numbers in ascending order using Insertion Sort and determine the time complexity for the same.. (Read the elements from user)
4. Write programs in Java to sort a list of n numbers in ascending order using Bubble Sort and determine the time
5. complexity for the same.. (Read the elements from file)
6. Write programs in Java to sort a list of n numbers in ascending order using Merge Sort and determine the time complexity for the same.. (Read the elements from file)
7. Write programs in Java to sort a list of n numbers in ascending order using Quick Sort and determine the time complexity for the same.. (Read the elements from user)
8. Write a program in Java to implement Strassen"s Matrix multiplication (2X2 Matrix) and determine the time complexity for the same..
9. Write a program in Java to implement Strassen"s Matrix multiplication (NXN Matrix)and determine the time complexity for the same..

Assignment Evaluation		
0. Not Done [ ]	1. Incomplete [ ]	2. Late Complete [ ]
3. Needs Improvement [ ]	4. Complete [ ]	5. Well Done [ ]
Signature of the Instructor		Date:

3. Greedy Method

Learning Outcome:

1. Learned how to solve the **Knapsack Problem** using Greedy and Dynamic Programming approaches.
2. Understood **Job Sequencing with Deadlines** and its importance in scheduling tasks efficiently.
3. Gained knowledge of **Huffman Coding** and how to generate optimal prefix codes for data compression.
4. Implemented **Optimal Merge Pattern** and **Optimal Storage on Tape** to minimize retrieval times.
5. Practiced solving **Matrix Chain Multiplication** and **Longest Common Subsequence (LCS)** using Dynamic Programming.

Assignments:

1. Write Java Program to implement Fractional Knapsack Problem using greedy by Profit
2. Write Java Program to implement Fractional Knapsack Problem using greedy by Weight
3. Write Java Program to implement Fractional Knapsack Problem using greedy by Density
4. Write Java Program to implement Job Sequencing with deadline problem
5. Write a Java Program to implement Optimal Merge Pattern
6. Write a Java Program to implement Huffman Coding.
7. Read a paragraph from a file and write a Java Program to find Huffman code for all the characters.
8. Write Java Program to find the order of programs for which MRT is minimized (Optimal Storage on tape)
9. Write Java Prog to find the order of programs for m tapes for which TRT is minimized (Optimal Storage on tape)
10. Write Java Program to find the order of programs for which ERT is minimized (Optilam st. tp)

Assignment Evaluation		
0. Not Done [ ]	1. Incomplete [ ]	2. Late Complete [ ]
3. Needs Improvement [ ]	4. Complete [ ]	5. Well Done [ ]
Signature of the Instructor		Date:

3. Dynamic Programming

Learning Outcome:

1. Learned how to solve the **Matrix Chain Multiplication** using Dynamic Programming approaches.
2. Learned how to solve the **Knapsack Problem** using Dynamic Programming approaches.
3. Practiced solving **Longest Common Subsequence (LCS)** using Dynamic Programming.
4. Practiced solving **String Editing (LCS)** using Dynamic Programming.

Assignments:

1. Write Java Program to implement 0/1 Knapsack Problem using Dynamic Programming
2. Write a program to implement matrix chain multiplication using Dynamic Programming.
3. Write a Program in Java to find only the length of the Longest Common Subsequence.
4. Write a Program in Java to implement String editing Problem.
5. Write a program to implement Sum of Subset by Backtracking

Assignment Evaluation		
0. Not Done [ ]	1. Incomplete [ ]	2. Late Complete [ ]
3. Needs Improvement [ ]	4. Complete [ ]	5. Well Done [ ]
Signature of the Instructor		Date:

4. Graphs

Learning Outcome:

1. Learned how to implement **Depth First Search (DFS)** and **Breadth First Search (BFS)** to traverse graphs.
2. Understood **Minimum Cost Spanning Tree (MST)** and implemented **Prim's and Kruskal's algorithms**.
3. Gained knowledge of **Dijkstra's algorithm** to find the shortest path in a weighted graph.
4. Implemented the **Traveling Salesman Problem (TSP)** to find the optimal tour.
5. Learned how to check for **Hamiltonian cycles** in a graph.

Assignments:

1. Write a program to implement DFS (Depth First Search) and determine the time complexity for the same.
2. Write a program to implement BFS (Breadth First Search) and determine the time complexity for the same.
3. Write a program to find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm
4. Write a program for the Implementation of Kruskal's algorithm to find minimum cost spanning tree.
5. Write a program for the Implementation of Dijkstra's algorithm to find shortest path to other vertices.
6. Write a program for finding Topological sorting for Directed Acyclic Graph (DAG)