|  |  |
| --- | --- |
|  |  |

**Faculty of Technology and Engineering**

**Chandubhai S Patel Institute of Technology**

**Department of Computer Science & Engineering**

**PRACTICAL – 6**

Roll no.: Date:    /    /

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Academic Year | : | 2024-25 | Semester | : | 4 |
| Course code | : | CSE207 | Course name | : | Design and Analysis of Algorithms |

**AIM:  Dynamic Programming Approach**

|  |  |
| --- | --- |
| **6.1** | "You are organizing a music festival and need to pack supplies in a limited number of trucks for transport. Each truck has a maximum capacity, and you have a variety of supplies, each with a specific weight and value. Your goal is to maximize the total value of the supplies transported without exceeding the truck's weight limit.  You are given:  · An integer array weights[] where weights[i] represents the weight of the ith supply item.  · An integer array values[] where values[i] represents the value of the ith supply item.  · An integer capacity representing the maximum weight the truck can carry.  You need to determine the maximum total value of supplies that can be packed into the truck." |

**6.1.1 Comparative Analysis of Time Complexity in Brute Force, Memoization, and Bottom-Up Approaches for above problem**

|  |  |
| --- | --- |
| **Brute force Approach** | **DP approach** |
|  |  |

|  |  |
| --- | --- |
| **6.2** | "In a bioinformatics lab where you need to compare two DNA sequences to find the longest common DNA subsequence. This is essential for identifying similarities and differences between the genetic material of different organisms, which can help in various biological studies.  Given two strings, s1 and s2, representing the DNA sequences, your task is to find the length of the longest common DNA subsequence between them." |

**6.2.1 Comparative Analysis of Time Complexity in Brute Force, Memoization, and Bottom-Up Approaches for above problem**

|  |  |
| --- | --- |
| **Brute force Approach** | **DP approach** |
|  |  |

|  |  |
| --- | --- |
| **6.3** | "An image processing system that applies a series of transformations to a collection of images. Each transformation is represented as a matrix operation. To process an image, these transformations must be applied in sequence, with the output of one operation serving as the input to the next. The efficiency of the image processing pipeline depends heavily on the order in which these matrix operations are performed, as the computational cost of multiplying matrices varies with their dimensions.  Your goal is to determine the optimal sequence of matrix operations that minimizes the total computational cost (number of scalar multiplications) required to process the image.    Given a chain < A1, A2,...,An> of n matrices, where for i=1,2,...,n matrix Ai with dimensions. Implement the program to fully parenthesize the product A1,A2,...,An in a way that minimizes the number of scalar multiplications. Also calculate the number of scalar multiplications for all possible combinations of matrices.    Test Case  n  Matrices with dimensions  1  3  A1: 3\*5, A2: 5\*6, A3: 6\*4  2  6  A1: 30\*35, A2: 35\*15, A3: 15\*5, A4: 5\*10, A5: 10\*20, A6: 20\*25" |

**6.3.1 Comparative Analysis of Time Complexity in Brute Force, Memoization, and Bottom-Up Approaches for above problem**

|  |  |
| --- | --- |
| **Brute force Approach** | **DP approach** |
|  |  |

**Conclusion:**

|  |  |
| --- | --- |
|  |  |

**Faculty of Technology and Engineering**

**Chandubhai S Patel Institute of Technology**

**Department of Computer Science & Engineering**

**Performa for PRACTICAL – 6**

Roll no.: Date:    /    /

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Academic Year | : | 2024-25 | Semester | : | 4 |
| Course code | : | CSE207 | Course name | : | Design and Analysis of Algorithms |

**Answer the following Questions:**

1. Compare and contrast the dynamic programming and greedy approaches for problem-solving.
2. In the 0/1 Knapsack problem, dynamic programming always provides the optimal solution, whereas the greedy approach does not. Justify your answer.
3. List the key domains where matrix multiplication plays a crucial role. Also, explain why dynamic programming is preferred for solving the matrix multiplication problem.

**Grade / Marks Sign of Lab Teacher with Date**