**Template for DAA PCC-CS494 Lab Softcopy**

1. **Make a handwritten copy for your lab copy, where pages will be A4 size and put all in a clip file.**
2. **Handwritten Lab Copy is mandatory for all the students. Submit your lab copy during every class day, failing this attendance will not be considered.**
3. 1st page is your Top Sheet Given in page no. 2 (You all already knew about the Top Sheet format)
4. 2nd page is the assignments list given in page no. 3 to 7
5. From the 3rd page onwards place the solves for all assignments one by one sequentially in order (Assignment 1 to Assignment 12)
6. After assignment list, write the 1st assignment in next page where mention the Assignment no. at the Top of the Page then write the sub question part no. like(1.1, 1.2… if required) and mention the corresponding part question then paste corresponding solve there. Do same for rest of the assignments.
   1. Draw the Flowchart. (1st)
   2. Next Place Time complexity analysis, all the cases. (2nd)
   3. Next Place your ‘C’ Code related to the given program. (3rd)
   4. Place your outputs after that. (4th)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Exp. No.** | **List of Experiments** | **Date** | **CO- Specific Marks** | | | | | **Total Marks** | **Remark & Signature** |
| **CO1** | **CO2** | **CO3** | **CO4** | **CO5** | **40** |  |
| **1.** | Assignment 1 |  |  |  |  |  |  |  |  |
| **2.** | Assignment 2 |  |  |  |  |  |  |  |  |
| **3.** | Assignment 3 |  |  |  |  |  |  |  |  |
| **4.** | Assignment 4 |  |  |  |  |  |  |  |  |
| **5.** | Assignment 5 |  |  |  |  |  |  |  |  |
| **6.** | Assignment 6 |  |  |  |  |  |  |  |  |
| **7.** | Assignment 7 |  |  |  |  |  |  |  |  |
| **8.** | Assignment 8 |  |  |  |  |  |  |  |  |
| **9.** | Assignment 9 |  |  |  |  |  |  |  |  |
| **10.** | Assignment 10 |  |  |  |  |  |  |  |  |
| **11.** | Assignment 11 |  |  |  |  |  |  |  |  |
| **12.** | Assignment 12 |  |  |  |  |  |  |  |  |

Lab Execution Top Sheet for CSE, Sec-A

Student Name :

Roll No :

**Subject Name :**

**Subject Code :**

**Session : 2022 - 2023**

|  |  |
| --- | --- |
| **NAME OF THE PROGRAM: *CSE*** | **DEGREE: *B. Tech*** |
| **COURSE NAME: *Design & Analysis of Algorithm*** | **SEMESTER: *4th*** |
| **COURSE CODE: *PCC-CS494*** | **COURSE CREDIT: *2*** |
| **COURSE TYPE: *LAB*** | **CONTACT HOURS: *4P*** |
| **SESSION: *2022-2023*** |  |

|  |  |  |
| --- | --- | --- |
| **Assignment No.** | **List of Experiments** | **Week No.** |
| Assignment 1 | I) Implement Binary Search using Divide and Conquer approach.  II) Implement Merge Sort, Quick Sort and Heap Sort using Divide and Conquer approach. | **Week 1** |
| Assignment 2 | I) Find Maximum and Minimum element from a array of integer using Divide and Conquer approach as well as dynamic programming approach. | **Week 2** |
| II) WAP to multiply two 2x2 matrices using Strassen’s Matrix Multiplication.  The first matrix is  5 6  1 7  The second matrix is  6 2  8 7 |
| III) WAP to multiply two 4x4 matrices using Strassen's Matrix Multiplication. |
| Assignment 3 | Find the minimum number of scalar multiplications needed for a chain of matrices whose sequences are <5, 10, 3, 12, 5, 50, 6> using the dynamic programming technique. | **Week 3** |
| Assignment 4 | I) WAP using the single-source-shortest-path problem to find out the shortest path from the source vertex ‘1’, using the Dijkstra’s algorithm, using the dynamic programming technique. | **Week 4** |
| II) WAP using the single-source-shortest-path problem to find out the shortest path from the source vertex ‘s’ using the Bellman-Ford’s algorithm, using the dynamic programming technique. |
| Assignment 5 | I) WAP to find out the all-pair-shortest-path for the given graph using the Floyd-Warshall’s algorithm by the dynamic programming technique. | **Week 5** |
| II) WAP using dynamic programming to find the optimal route for the given graph using TSP and taking vertex ‘1’ as source. |
| Assignment 6 | I) WAP to implement BFS on the given graph starting at ‘a’. | **Week 6** |
| II) WAP to implement DFS on the given graph starting at ‘A’. |
| Assignment 7 | I) WAP using greedy method to find the MST for the given graph using the Prim’s Algorithm. | **Week 7** |
| II) WAP using greedy method to find the MST for the given graph using the Kruskal’s Algorithm. |
| Assignment 8 | I) Consider the following knapsack problem where n = 3, W = 20 Kgs, (v1, v2, v3) = (25, 24, 15), and (w1, w2, w3) = (18, 15, 10). WAP to find the optimal solution by fractional knapsack (greedy method) as well as 0 / 1 knapsack (Dynamic programming method). | **Week 8** |
| II) Given a set of ‘10’ jobs with their si and fi, find the optimal sequence of mutually compatible jobs using the greedy method: A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, si = {3, 4, 5, 6, 7, 8, 9, 10, 11, 12} and fi = {5, 7, 8, 10, 11, 12, 13, 14, 15, 16}. |
| Assignment 9 | Write a procedure using B & B technique to solve the 15-puzzle problem. (Take an initial state of your choice that will converge in the goal state.) | **Week 9** |
| Assignment 10 | I) WAP to implement the 4-Queen’s problem using the method of backtracking. | **Week 10** |
| II) WAP using backtracking to find the chromatic number for the given graph, using the graph-coloring problem. |
| Assignment 11 | The Maximum Sub-array Sum problem is to find the sub-array (A sub-array of the array A [0…N-1] is A[i…j] where 0 ≤ i ≤ j < N ) for which the sum of its elements is maximum. For example, given an array {12, -13, -5, 25, -20, 30, 10}, the maximum sub-array sum is 45 for sub-array {25, -20, 30, 10}. The naive solution to this problem is to calculate the sum of all sub-arrays starting with every element and returning the maximum of all. WAP to find the maximum sub-array given below:  (1) Divide the given array in two halves.  (2) Return the maximum of the following:  i. Maximum sub-array sum in the left half (Make a  recursive call).  ii. Maximum sub-array sum in the right half (Make  a recursive call). | **Week 11** |
| Assignment 12 | 1. Find the only 10 digit number where the digit at the ith position (i = 0, 1, 2, 3 etc.) from left specifies the number of occurrences of digit "i" in the given number. | **Week 12** |
| 1. Find the numbers from 1 to N that contains exactly k non-zero digits   Given two integers **N** and **K**. The task is to find the number of integers between **1** and **N** (inclusive) that contains exactly **K** non-zero digits when written in base ten.  Example:***Input:****N = 100, K = 1* ***Output:****19* ***Explanation:*** *The digits with exactly 1 non zero digits between 1 and 100 are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100*  ***Input:****N = 25, K = 2* ***Output:****14* ***Explanation:*** *The digits with exactly 2 non zero digits between 1 and 25 are: 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25* |