Registration Number : 22BCE5150

VIT
Vellore Institute of Technology
CHENNAI

Continuous Assessment Test I - August 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024-25
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.Srinivasa Rao U, Dr. Jayaram B, Dr.Revathi A R	Slot/Class No.	D1/CH2024250101206 /CH2024250101204/ CH2024250101208
[illegible]	90 Minutes	Max. Marks	50

[illegible]ions:

- Answer all the FOUR questions.
- If any assumptions are required, assume the same and mention those assumptions in the answer script.
- Use of intelligence is highly appreciated.
- Your answer for all the questions should have both the 'design' component and the 'analysis component'
- The 'Design' component should consist: understanding of the problem, logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Proof-of-Correctness, Computation of T(n), Time-complexity.

---

1. An equation is said to be a *line in two variables* if it is written in the form of $L(x, y) = ax + by + c = 0$, where $a, b, \&c$ are real numbers and the coefficients of x and y are $a \neq 0$ and $b \neq 0$ respectively. A point $P = (x_1, y_1)$ is said to be on a line $L(x, y)$, if $ax_1 + by_1 + c = 0$. For example, $L(x, y) = 10x - 2y + 4 = 0$ is a linear equation and $P(x_1 = 1, y_1 = 7)$ is a point on the line $L(x, y)$. That is, $10 \times 1 - 2 \times 7 + 4 = 0$.

   Closest Pair problem: Given a line $L(x, y) = 0$, and assume $P_1 = (x_1, y_1), P_2 = (x_2, y_2), \cdots, P_n = (x_n, y_n)$ are n points on the line $L(x, y) = 0$. Find the pair of points which are closest (in the sense Euclidean distance) among all such pairs. [Hint: The Euclidean distance of $P_1 = (x_1, y_1)$ and $P_2 =$ [illegible]s $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

   [illegible]o different algorithms using two different design techniques to solve the **Closet Pair problem**. [illegible]ult, justify which of the two design techniques is more efficient.

   [15 marks]

   [Rubrics: Logic's :2 marks, Illustrations :2 marks, Algorithms: 8 marks, Time-complexities: 3 marks]

2. A subarray is an array that is a contiguous part of an array. In any given array, the maximum sum subarray is a subarray with maximum sum. For example : Input: $A = [1, 2, 7, -4, 3, 2, -10, 9, 1]$. The subarray yielding the maximum sum is [1, 2, 7, -4, 3, 2] and output is 11.

   2D Maximum Diagonal Subarray Sum(2MDSS) problem: A diagonal matrix A is a square matrix in which all of the elements except the principal diagonal elements are zeroes. It is both upper and lower triangular, as all the elements except the main diagonal elements are zeros. For example, a diagonal matrix of size 5×5 is shown in Figure 1. Your task is to find the maximum sum subarray of the principal diagonal elements of A.

   Design two different algorithms using two different design techniques to solve the **2MDSS** problem. As a result, justify which of the two design techniques is more efficient.

   [Rubrics: Logic's:2 marks, Il

Diagonal Matrix

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix}_{5\times 5}$$

Figure 1: 5×5 Diagonal Matrix

Algorithm 1 CC(A[ ])

```
Input: Array A[1, ··· , n] of n integers.
Output: Array S[1, ··· , n] of elements.
for i = 1 to n do
    count[i] ← 0
end for
for i = 1 to n − 1 do
    for j = i + 1 to n do
        if A[i] < A[j] then
            count[j] = count[j] + 1
        else
            count[i] = count[i] + 1
        end if
    end for
end for
for i = 1 to n do
    S[count[i]] ← A[i]
end for
return S
```

3. Consider the following algorithm: [10 marks]

Understand the functionality of the above algorithm and answer the following.

(a) Write the output when, $A = [60, 35, 81, 98, 14, 47]$. [2]

(b) Describe the functionality of CC algorithm and also write the proof of correctness for the algorithm. [1+4]

(c) Compute the time-complexity of the algorithm. [3]

4. Let $A[1 \ldots n]$ be an array of n integers such that the number zero does not belong to A. Write an algorithm, using the **divide and conquer** design technique to segregate positive and negative integers in the array A and output the segregated array A. Additionally, the elements of the array A after segregation should satisfy the following criteria: positive integers should appear in the same order as they originally occured in A and the negative integers should appear in the same order as they originally occured in the A. For e.g. If A=[8, 3, -2, 10, 9, -1, -2, 2] then the algorithm should output A = [-2, -1, -2, 8, 3, 10, 9, 2]. [10 marks]

[Rubrics: Logic : 2 marks, Illustration: 3 marks, Algorithm :3 marks and Time-complexity :2 marks]

⟺ ⟺ ⟺ ⟺