VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of the UGC Act, 1956)

Reg. No. : 22BAI1266

Final Assessment Test(FAT) - Nov/Dec 2024

Programme	**B.Tech.**	Semester	**Fall Semester 2024-25**
Course Code	**BCSE204L**	Faculty Name	**Prof. Indira B**
Course Title	**Design and Analysis of Algorithms**	Slot	**A1+TA1**
		Class Nbr	**CH2024250100957**
Time	**3 hours**	Max. Marks	**100**

General Instructions

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.
- Answer all the EIGHT 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: Computation of T(n), Time-complexity.

Course Outcomes

1. Apply the mathematical tools to analyze and derive the running time of the algorithms
2. Demonstrate the major algorithm design paradigms.
3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
5. Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

Section - I
Answer all Questions (4 × 10 Marks)

*M - Marks

Q.No	Question	*M	CO	BL
01.	Let $S = \{d_1, d_2, d_3, \dots d_n\}$, where d_i's are integers of varying values. Concatenation of two integers, denoted as $d_1 \times d_2$, is defined as combining d_1 and d_2 in the same order and the cost of concatenation is $\|d_1\| \times \|d_2\| \times (\|d_1\| - \|d_2\|)$, where $\|d_1\|$ and $\|d_2\|$ represent the number of digits in d_1 and d_2 respectively. 123×2345 is 1232345. Cost of concatenation of $(123 \times 4567) \times 333$ is $4 \times 3 \times (-1) + 7 \times 3 \times 4 = 72$. Cost of concatenation of $123 \times (4567 \times 333)$ is $4 \times 3 \times 1 + 7 \times 3 \times (-4) = -72$. **Given a set S, Design a pseudocode to compute the parenthesization of $d_1 \times d_2 \dots \times d_n$ in such a way that the cost of concatenation is minimum.** You have to illustrate your pseudocode by considering a set S with a minimum of eight elements. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for pseudocode : 4 marks, Pseudocode : 3 marks, Time-complexity :1 mark]	10	2	3

02.	**Consider a string W of length n, design a brute-force pseudocode to compute the maximal repeated substring that can appear in multiple places, but with no overlapping occurrences.** If $W = banana$, your pseudocode should return an and na. You have to illustrate your pseudocode by considering a string W of length at least 7 and there should be at least two maximal repeated substrings. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for pseudocode : 4 marks, Pseudocode : 3 marks, Time-complexity :1 mark]	10	2	3
03.	Consider the problem P: Given a positive integer N, the task is to compute all the even factors of N. For example, 2 is an even factor of 6. **Compute the class-complexity of the problem P (P/NP/NPC) with justification.** [5 marks] [**Rubrics:** Identification of class-complexity (2 marks), Justification (3 marks)] Consider the problem P_1: Given the set $S = \{1, 2, 3, \ldots, n\}$ and $X = \{s_1, s_2, \ldots, s_m\}$, s_i's are the subsets of S, the task is to identify the smallest subset of X whose union is S. For example, given $S = \{s_1 = \{1, 2, 3\}, s_2 = \{2, 4\}, s_3 = \{3, 4\}, s_4 = \{4, 5\}\}$, the smallest subset of X whose union is S is $\{s_1, s_4\}$. **Compute the class-complexity of the problem P (P/NP/NPC) with justification.** [5 marks] [**Rubrics:** Identification of class-complexity (2 marks), Justification (3 marks)]	10	5	4
04.	Consider an one-dimensional line. Consider the set of line segments, $S = \{l_1 = (a_1, b_1), l_2 = (a_2, b_2), \ldots, l_n = (a_n, b_n)\}$, where $l_n = (a_n, b_n)$ means that a_n is the starting point of the line segment l_n and b_n is the end point of the line segment l_n. **Given S, design a pseudocode to compute the maximum length of the line segment which overlaps with all the line segments of S.** Your algorithm should return the maximum length of the required line segment and the starting and the end point of that line segment. If there is no intersection among the segments, your pseudocode should return zero. For $S = \{[1, 4][2, 5][3, 6][7, 8]\}$, your pseudocode should output 1 and [3,4]. You have to illustrate your pseudocode by considering a set S with a minimum of 8 elements and maximum length of the required line segment should atleast be 2. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 4 marks, Pseudocode : 3 marks, Time-complexity :1 mark]	10	3	3

Section - II

Answer all Questions (4 × 15 Marks)

*M - Marks

Q.No	Question	*M	CO	BL
05.	**Given a string S, design two different pseudocodes: Pseudocode 1 , Pseudocode 2, such that the pseudocode 1 applies the divide-conquer-combine strategy, Pseudocode 2 applies a strategy different from divide-conquer-combine, to compute the longest substring of S which is also a palindrome (palindrome is a string that reads the same when read from left to right or vice versa).** For the input S= 'abbadefggfhhfgggg', your pseudocode should output `ggfhhfgg'. For the illustration of your pseudocode, you are supposed to take a string S which has a minimum length of 10 and the length of the longest substring which is a palindrome should be atleast 3. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for Pseudocode 1 : 2 marks, Illustration for the Pseudocode 2 : 3 marks, Pseudocode 1 : 3 marks, pseudocode 2 : 2 marks, Time-complexity of the pseudocode 1 : 2 marks, Time-complexity of the pseudocode 2 : 1 mark]	15	2	6

06. Given an $N \times M$ grid matrix, the cells of the grid may contain either english vowels or english consonants. A path from the cell $(0,0)$ to the cell $(N-1, M-1)$ is a sequence of the coordinates of the cells in the grid, through which one can move to reach $(N-1, M-1)$, starting from $(0,0)$. Two paths are said to be distinct if there is no cell common to both the paths except the starting cell and the destination cell. **Given a grid, Design a pseudocode to compute all the distinct paths from the top-left corner $(0,0)$ to the bottom-right corner $(N-1, M-1)$. Your pseudocode should be in two parts , (i) Pseudocode that computes all the appropriate paths, (ii) Pseudocode that computes all the distinct paths, by taking the output of the earlier pseudocode.** You may only move right, down, up, or left at each step. If a cell contains vowels , movement is not allowed through the cell. Your pseudocode should output the path as a sequence of the cells visited in the traversal. A path from $(0,0)$ to $(3,3)$ in the following grid is given by

$$\begin{bmatrix} f & b & d & i \\ e & a & l & o \\ s & t & t & u \\ d & t & p & b \end{bmatrix}$$

$(0,0)-(0,1)-(0,2)-(1,2)-(2,2)-(3,2)-(3,3)$. Your pseudocode should output all the possible distinct paths. For the illustration of your pseudocode, you are supposed to take a grid with both N and M greater than 4 and the illustration should be in such a way that it yields a minimum of four paths and a minimum of two distinct paths. You are not supposed to use any illustration given in this question as part of your illustration in the answer.

[**Rubrics:** Logic for pseudocode: 2 marks, Illustration for the pseudocode that computes all the appropriate paths: 4 marks, Illustration for the pseudocode that computes all the distinct paths : 2 marks, Pseudocode that computes all the appropriate paths : 3 marks, Pseudocode that computes all the distinct paths : 3 marks, Time-complexity :1 mark]

15 2 6

07. Consider a positive integer d of length n with n-digits and a set S which consists of a few sub-integers of d. Some sub-integers of 234567 are 23, 34, 4567. **Design a greedy based recursive pseudocode and a greedy based iterative pseudocode to compute the maximum number of non-overlapping sub-integers from S such that no two selected sub-integers overlap in their positions in d.** If $d = 12345678$, $S = \{123, 234, 345, 456, 678, 34\}$, maximum number of sub-integers of d in Swhich does not over-lap is 2 and the those sub-integers are $123, 456$. Your pseudocode should output the maximum number of sub-integers as well as those sub-integers. You have to illustrate your pseudocode by considering a set S with a minimum of eight elements. You are not supposed to use any illustration given in this question as part of your illustration in the answer.

[**Rubrics:** Logic for pseudocode: 3 marks, Illustration for greedy-based recursive pseudocode : 4 marks, Greedy based recursive Pseudocode : 3 marks, Greedy based iterative pseudocode : 3 marks, Time-complexity of the greedy based recursive pseudocode :1 mark, Time-complexity of the greedy based iterative pseudocode :1 mark]

15 2 6

08.

Algorithm 1 Find MFV in Flow Network

15 | 1,3 | 4

```
Require: Flow network $G = (V, E, c, s, t)$ with capacities $c$, source $s$, and target $t$
Ensure: Set of MFV in $G$
Initialize maxFlow $\leftarrow 0$
Initialize vertexVisitCount[$v$] $\leftarrow 0$ for each vertex $v$ in $V$
Initialize MF $\leftarrow 0$
Initialize MFV $\leftarrow \emptyset$
while there exists an augmenting path $P$ from $s$ to $t$ in $G$ do
    Find the minimum residual capacity $c_{\min}$ along path $P$
    Update the flow along edges in $P$ by $c_{\min}$
    maxFlow $\leftarrow$ maxFlow + $c_{\min}$
    for each vertex $v$ in path $P$ (excluding $s$ and $t$) do
        vertexVisitCount[$v$] $\leftarrow$ vertexVisitCount[$v$] + 1
    end for
end while
for each vertex $v$ in $V$ do
    if vertexVisitCount[$v$] > MF then
        MF $\leftarrow$ vertexVisitCount[$v$]
        MFV $\leftarrow \{v\}$
    else if vertexVisitCount[$v$] = MF then
        Add $v$ to MFV
    end if
end for
return MFV
```

Understand the Algorithm 1 and answer the following.

a. Describe the functionality of the above algorithm. [3 marks]

b. Compute the time-complexity of the algorithm. [2 marks]

c. Consider a directed flow network $G = (V, E, s, c)$ with: Vertices: $V = \{s, v_1, v_2, v_3, v_4\}$, Edges with capacities c: $\{(s, v_1) : 10, (s, v_2) : 15, (v_1, v_3) : 5, (v_2, v_3) : 10, (v_2, v_4) : 10, (v_3, v_4) : 10\}$, source s. Given G, compute the output of Algorithm 1 with target vertex as v_4 [5 marks]

d. Will there be a change in the output if the target vertex is changed from v_4 to v_3 to the graph described in the question no (3)? If so, what will be the new output? [3 marks]

e. What is the purpose of the vertexVisitCount array in the algorithm, and how is it updated during the execution of the algorithm? [2 marks]

BL-Bloom's Taxonomy Levels - (1.Remembering, 2.Understanding, 3.Applying, 4.Analysing, 5.Evaluating, 6.Creating)