VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Final Assessment Test - April 2023

Programme	B.Tech.(CSE)	Semester	Winter 2022-23
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.L.Jeganathan, Dr B Jayaram, Dr M Janaki Meena, Dr T Kalai priyan, Dr U Srinivasa rao, Dr L K Pavithra	Slot/Class No.	B2/CH2022235000286/ CH2022235000287/ CH2022235000288/ CH2022235000289 /CH2022235000291/ CH2022235000292
Time	180 Minutes	Max. Marks	100

Instructions:

- 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: logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Computation of T(n), Time-complexity.
- Rubrics for question No. 1,2 : Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)
- Rubrics for question No.3(a), 3(b): Identification of class-complexity (2 marks), Justification (3 marks)
- Rubrics for question No. 5,6,7: Logic(3 marks), Illustration (5 marks), Pseudocode (5 marks), Running time & Time-complexity (2 marks)
- Rubrics for question No. 8 : Logic for A and B (4 marks), Illustration for A and B (4 marks), Pseudocodes : A and B (4 marks), Running time & Time-complexity of A and B (3 marks)

---

1. The set of all real numbers that lie between an integer a and an integer b (a and b are also included in the set), $a < b$, is represented as $[a, b]$ and is called as interval a,b. Given $[a_1, b_1], [a_2, b_2], ...[a_n, b_n]$, design a greedy pseudocode to identify the maximum number of intervals (among the given n intervals) such that the intersection of any two intervals (among the intervals identified by you) is always empty. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [10 marks]

2. Consider the 2-dimensional plane where the points are represented by a pair of integers. A line with the end-points p_i and p_j is represented as $l(p_i, p_j)$. A polygon $P = \{p_1, p_2, ..., p_m\}$ is a set of points that forms a closed figure with the edges $p_1 - p_2, p_2 - p_3, p_3 - p_4, ..., p_{(m-1)} - p_m, p_m - p_1$. Edges of the polygon can also be called as the line connecting the points. For example, the edge $p_1 - p_2$ can be called as a line l whose end-points are p_1 and p_2. A polygon is said to be a simple polygon if none of its edges cross itself. Given a polygon $P = \{p_1, p_2, ..., p_m\}$, design a pseudocode to decide whether the polygon is a simple polygon or not. [10 marks]

3. (a) Consider the problem P: Given a positive integers N_1 and N_2, the task is to compute the digit(s) of $N_1 * N_2$ (Here $*$ is the usual multiplication operation) that occur in both N_1 and N_2. For example, given two numbers 21 and 12, your algorithm should return 2. Compute the class-complexity of the problem P (P/NP/NPC) with justification. [5 marks]

   (b) Consider the problem P_1: Given a set S of positive integers and another integer k, Does there exist a subset S' of S such that the sum of the elements of S equals t. For example, for the input $S = \{3, 13, 24, 45, 102\}$ and 72, the solution to the problem is 'yes' since there exists the subset $S' = \{13, 24, 45\}$ whose sum of the elements is 72. Compute the class-complexity of the problem P_1 (P/NP/NPC) with justification. [5 marks]

4. Understand the Algorithm and answer the following questions .

- f2(B):
  - q = B mod 10
  - If (q mod 2) == 0 then return B
  - return B
  - else return 0
- f1(A, l, h)
  - if l==h then return f2(A[l])
  - m = (l+h)/2
  - return f1(A, l, m) + f1(A, m+1, h)
- Algorithm XXXX(A)
  - n = A.length
  - f1(A,1,n)

(a) Compute the output of the Algorithm when [123, 45, 52, 61] is given as an input. [2 marks]

(b) Describe the functionality of the Algorithm [2 marks]

(c) Is it possible to remove any lines in functions f1 or f2 such that the functionality of the algorithm does not change. If so, identify the lines. [2 marks]

(d) Compute the time complexity of the algorithm [2 marks]

(e) If the algorithm returns zero for an input array L, comment on the elements of L? [2 marks]

5. A common subsequence Z of two strings X and Y is said to be a 'Distinct Common Subsequence (DCS)' X & Y if Z has no symbols repeated in it. If $X = ABCDEF$, $Y = BADEAF$, then $ADEF$ is a DCS of X and Y. EF is also another DCS of X & Y. Given two strings X of length n and Y of length m, Design a pseudocode to compute all the distinct common subsequences of X and Y, of length greater than $\lfloor \frac{m+n}{2} \rfloor$. Here, $\lfloor . \rfloor$ is the usual floor function. If there are no DCS possible with the sepcified length, your pseudocode should return -1. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [15 marks]

6. Given string of symbols $X = x_1x_2...x_n$, where $x_1, x_2, ..., x_n \in \{a, b, c, ..., x, y, z\}$. We define the navigation through the digits of X as follows.

- **LR-navigation with stride k, RL-navigation with stride k** : You can Start from any symbol of X, with a stride k (step of size k, k ¡ n), in the left-to-right direction and reach the k^{th} symbol (symbol which occurs in the k^{th}-position starting the count from the symbol where you started the navigation). In the string abcdfegh, if you start with the symbol a with a stride 1 in the LR-directions, symbols traversed by you (called as traversal-sequence) is : a-b-c-d-f-e-g-h. If you start from the symbol b with a stride 2 in the LR-direction digits traversed by you is b-d-e-h. Starting from a with a stride of 6 in the LR-direction, symbols traversed is : g. Similarly, we define the RL-navigation with stride k. In the LR-navigation, the direction of movement is from left to right where as in the RL-navigation is from from right to left.
- **Phases of the navigation** : During any navigation, you can start a new phase when no further symbols can be traversed, by changing the direction (from LR to RL or from RL to LR) or by changing the stride to the next higher value or both. In the string abcdfegh, navigation in the LR-direction with the stride 1, the traversal sequence is a-b-c-d-f-e-g-h, navigation after h is not

possible because there are no symbols available. Now, we can change the direction fro LR to RL. Tranversing from h in the RL-direction with stride h , is h-e-d-b. The total-symbol-sequence(tss) is the combination of the traversal-sequence of both the phases of navigation. If the navigation has many phases, the tss of the navigation is the combination of all the traversal sequences of each phase.

Given a string $X = x_1x_2...x_n$, design a pseudocode to navigate through the symbols of X that satisfies following conditions.

- you have to start from the first symbol of N.
- You have to start a new phase if you happen to traverse any symbol that is already in the traversal-sequence.
- You can have any number of phases for the navigation
- The total-symbol-sequence should have all the distinct symbols of N.

Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. [15 marks]

7. Given a graph $G = (V, E, s, c, t)$, where V is the set of vertices, E is the set of edges, $s \in V$ is a vertex designated as a source vertex, c is the set of capacities of all the edges of G, t is a vertex designated as the target vertex. Let $|f|$ denote the maximum flow of the network G with s as the source vertex and t as the target vertex. Given the graph $G = (V, E, s, c, t)$, design a pseudocode to identify the edges (u, v) for which $[c[(u, v)] - f[(u, v)]] \neq 0$, when the flow across the network is $|f|$. For the purpose of the illustration, you have to take a graph with a minimum of six vertices and a minimum of eight edges. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. [15 marks]

8. A cyclic rotation of a string is the process of shifting the last character of the string to the first position and shifting all the other characters one positionto the right. For example, the cyclic rotation of the string *abcd* is *dabc*. *bcda*, *cdab* are also the cyclic rotations of *abcd*. This process of shifting can be repeated any number of times to generate a sequence of cyclic rotations.

   Given two strings S_1 and $S2$, design a brute-force algorithm A to check if $S2$ can be generated from S_1 by a sequence of cyclic rotations or vice versa. Also design a string-matching algorithm B to compute whether S_2 can be generated from S_1 by a sequence of cyclic rotations or vice-versa. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. (15 marks)

$\Longleftrightarrow \Longleftrightarrow \Longleftrightarrow$