[redacted]

VIT

Vellore Institute of Technology

Continuous Assessment Test II - July 2023

Programme	B.Tech.(CSE)	Semester	Fall Inter 2023-24
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr M Janaki Meena, Dr. Joe Dhanith P R, Dr. Pandiyaraju V, Dr. Senthil Kumar A M, Dr. Suguna M, Dr. Dominic Savio M	Slot/Class No.	D2 + TD2 / CH2022232500782, CH2022232500783, CH2022232500784, CH2022232500785, CH2022232500786, CH2022232500781
Time	90 Minutes	Max. Marks	50

Instructions:

- Answer all the FIVE 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.

---

1. Let m and n be any two even positive integers. A common sequence Z_K of two strings X_m and Y_n is said to be a common end-point-sequence if either ($z_1 = x_1$ or y_1) or ($z_k = x_m$ or y_n). A common end-point sequence Z_K is said to be a common middle-end-point-sequence of X_m and Y_n if $z_{\lfloor \frac{k}{2} \rfloor} = x_{\lfloor \frac{m}{2} \rfloor}$ or $y_{\lfloor \frac{n}{2} \rfloor}$ Here, the notations Z_k, X_m, Y_n, z_i, x_i, y_i, for any k, m, n, i have the usual meaning. Given two sequences X_m and Y_n, design a dynamic programming pseudocode to compute the longest common middle-end-point-sequence of X_m and Y_n.

   [**Rubrics:** Logic : 3 marks, Illustration: 3 marks , Pseudocode: 3 marks, Time-complexity 1 mark]

2. You are given an $n \times n$ grid with n^2 cells, in which the cells in the top row, rightmost column, bottom row and the leftmost column are filled with the digits from $\{0, 1, 2, ..., 9\}$. That is, all the boundary cells have a digit in them. A sample grid of size 4×4 is given here for clarity.

1	7	3	4
9	4	2	0
8	1	6	5
6	2	1	3

   Given an $n \times n$ grid with the digits filled in the $4n - 4$ cells, and ten digits $\{0, 1, 2, 3, .., 9\}$, design a pseudocode that uses backtracking strategy, to fill the remaining empty cells with the given 10 digits in such a way that

   - No digit get repeated in any row
   - No digit get repeated in any column

   [**Rubrics:** Logic : 4 marks, Illustration: 2 marks , Pseudocode: 3 marks, Time-complexity 1 mark]

3. The set of all real numbers that lie between the positive integer a and the positive integer b (a and b inclusive) is denoted by $[a, b]$, read as the 'closed interval a,b'. Here, we call a and b as the end-points of the interval $[a, b]$ and size of the closed interval is $b - a$. $[2, 7]$ is a closed interval of size 5. Two closed intervals $[a_i, b_i]$ and $[a_j, b_j]$ is said to be nested-compatible if $a_i \leq a_j$ and $b_i \geq b_j$. A set S is said to be a nested compatible set of closed intervals if all the pairs of elements of S are nested-compatible. For example, for $S = \{[1, 10], [2, 8], [3, 5], [12, 17,], [13, 15], [20, 21]\}$, the nested compatible subsets of S are $\{[1, 10], [2, 8], [3, 5]\}$ and $\{[12, 17,], [13, 15]\}$. In fact, $\{[1, 10], [2, 8], [3, 5]\}$ is the maximum-size nested-compatible subset of S. Given n closed intervals $S = \{[a_1, b_1], [a_2, b_2], ...[a_n, bn]\}$, design a greedy pseudocode to compute the Maximum-size nested-compatible subset of S. Your pseudocode should compute both the maximum-size of the nested-compatibel subset as well the corresponding subset.

   [**Rubrics:** Logic : 3 marks, Illustration: 3 marks , Pseudocode: 3 marks, Time-complexity 1 mark]

4. Let m and n be any two positive integers. A pattern P[1,...,m] is said to be a 3-phase-pattern if P occurs in T[1,...,n] in three phases, with different shift values. For example, P=ABEFGHI is a 3-phase-pattern in T=XYZABTTEFBAAGHI with $s_1 = 3, s_2 = 7, s_3 = 12$. Given a pattern P[1,...,m], T[1,...,n], design a pseudocode to decide whether the given pattern is 3-phase-pattern or not. If P is a 3-phase-pattern, your pseudocode should retrun all the three shift values. [10 marks]

[Rubrics: Logic : 3 marks, **Illustration:** 3 marks , Pseudocode: 3 marks, Time-complexity 1 mark]

5. Consider the following algorithm

string matching rabin karp.

Algorithm 1 : $F_1(S_1, S_2)$

```
0: Input : S1, S2
1: /S1, S2 are strings of same length and " indicates empty string/
2: n = S1.length()
3: d_table[1,...,n] = "
4: d_table[1] = S1
5: d_table[2] = S2
6: return F2(n, d_table)
7: ---------------------------------------------
8: Algorithm : F2(n, d_table)
9: if d_table[n] ≠ " then
10:     return d_table[n]
11: end if
12: Sx = F3(F2(n − 2, d_table), F2(n − 1, d_table))
13: d_table[n] = Sx
14: return Sx
15: ---------------------------------------------
16: Algorithm : F3(S1, S2)
17: Sx = "
18: m = S1.length()
19: for i = 1 to m do
20:     if i %2 ≠ 0 then
21:         Sx = Sx + S1[i]
22:     else
23:         Sx = Sx + S2[i]
24:     end if
25: end for
26: return Sx
```

Understand the above algorithm and answer the following.

(a) For the input $ABCD, DEFG$, Compute the output of the algorithm. [2 marks]

(b) Describe clearly the functionality of the algorithm. [2 marks]

(c) Compute the time-complexity of the algortihm. [2 marks]

(d) Describe the design-strategy used in the algorithm. [2 marks]

(e) The above algorithm is modified by deleting the line:13. Compute the time-complexity of the modified algorithm. [2 marks]

⟺ ⟺ ⟺