REG.NO.:

SLOT: F1+TF1

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

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

Programme Name & Branch : **BTech**
Course Code and Course Name : **BCSE204 L - Design and Analysis of Algorithms**
Faculty Name(s) : DEBI PRASANNA ACHARJYA ,UMADEVI K S ,GAYATHRI P,SRIVANI A,SHARIEF BASHA S,ILANTHENRAL K P S K,SATYAJIT DAS ,JOSHVA DEVADAS T,ATHIRA K,DURGESH KUMAR,IYAPPAN P,MALINI S,KARTHIK G M,VASANTHI P,ANKAREDDY RAJESH ,THIRUNAVUKKARASAN M,PADMAVATHY T,DINESH P,RANJITH R,DIVIYA M,SHARMILA C,MUKHTAR AHMAD SOFI,VANI RAJASEKAR,HARAPRIYA KAR,SYED RIYAZUL HAQ,MONESH S,EVELIN NISSY THOMAS,NAVEEN KUMAR S

Class Number(s) : **ALL Batches**
Date of Examination : **1-02-2026 (FN)**
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):
Answer All Questions – **Provide the necessary steps wherever required**
M - Max mark; CO – Course Outcome; BL – Bloom's Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyze, 5 – Evaluate, 6 – Create)

Course Outcomes

1. Apply the mathematical tools to analyse and derive the running time of the algorithms
2. Demonstrate the major algorithm design paradigms.

<table>
<tr><th>Q. No</th><th>Question</th><th>Module</th><th>Marks</th><th>CO</th><th>BL</th></tr>
<tr><td>1.</td><td>a) Consider the following recursive procedure to solve the Tower of Hanoi problem for <i>n disks</i>:
The rules of Tower of Hanoi are:
1. Only one disk can be moved at a time.
2. A disk can only be placed on an empty peg or on top of a larger disk.
3. All disks must be moved from the source peg to the destination peg.
<pre>
Procedure TOH (n, A, B, C):
  if n == 1:
    Move disk from A to C
  else:
    TOH (n-1, A, C, B)
    Move disk from A to C
    TOH (n-1, B, A, C)
</pre>
Define a suitable recursion invariant for this algorithm and use it to prove that the algorithm correctly moves all <i>n disks</i> from the source peg to the destination peg while following the rules. <i>(4 Marks)</i>
b) Consider the recurrence relation:
$$T(n) = 2T(n-2) + n, \quad T(2) = \Theta(1)$$
Explain why the Master Theorem cannot be applied to this recurrence, and solve it using the back-substitution (substitution) method to obtain a tight Θ-bound. <i>(6 marks)</i></td><td>1</td><td>4+6</td><td>1</td><td>3</td></tr>
</table>

REG.NO.:

SLOT: F1+TF1

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

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

<table>
<tr><td>2.</td><td>A data file uses symbols from an alphabet Σ, where one symbol occurs in more than 50% of the total frequency. The symbols and their frequencies (in hundreds) are:
<table><tr><td>Symbol</td><td>p</td><td>q</td><td>r</td><td>s</td><td>t</td><td>v</td><td>w</td></tr><tr><td>Frequencies</td><td>16</td><td>5</td><td>25</td><td>12</td><td>9</td><td>45</td><td>13</td></tr></table>Construct an optimal Huffman code for the above symbols and draw the Huffman tree. Determine the percentage of saving in file size compared to fixed-length encoding. Explain why a perfectly balanced Huffman tree cannot be constructed for this frequency distribution, and discuss how the skewness of symbol frequencies affects the codeword lengths of other symbols and the overall compression efficiency.</td><td>1</td><td>10</td><td>1</td><td>3</td></tr>
<tr><td>3.</td><td>Consider the array
A= [13, −3, −25, 20, −3, −16, −23, 18, 20, −7,12, −5, −22,15,]
Find the maximum subarray using the divide-and-conquer method, clearly showing the left subarray sum, right subarray sum, and crossing subarray sum at each recursive step. Briefly explain why the overall time complexity of this algorithm is $O(n\log n)$.</td><td>1</td><td>10</td><td>1</td><td>3</td></tr>
<tr><td>4.</td><td>A salesman must visit 4 cities (A, B, C, D) exactly once and return to the starting city. The distances (in km) between the cities are given in the table below:
<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><td>A</td><td>0</td><td>10</td><td>15</td><td>20</td></tr><tr><td>B</td><td>10</td><td>0</td><td>35</td><td>25</td></tr><tr><td>C</td><td>15</td><td>35</td><td>0</td><td>30</td></tr><tr><td>D</td><td>20</td><td>25</td><td>30</td><td>0</td></tr></table>Using the Dynamic Programming method, determine the minimum cost tour starting and ending at city A. Show the DP state representation and state transitions, and explain why storing only the minimum cost for each state preserves optimality. Suppose the distance between cities B and C is incorrectly recorded as 5 instead of 35; explain how this change affects the DP computation and the resulting optimal tour.</td><td>2</td><td>10</td><td>2</td><td>3</td></tr>
<tr><td>5.</td><td>A monitoring system needs to place 4 sensors on a 4 × 4 grid, one sensor per row, such that the following constraints are satisfied:
No two sensors are placed in the same row.
No two sensors are placed in the same column.
No two sensors are placed at grid positions (i, j) and $(i+2, j+1)$or $(i+2, j-1)$. . Here (i-row, j-column)
i. Model this problem as a backtracking problem by clearly defining the state representation, decision variables, and constraints. (4 Marks).
ii. Using the backtracking approach, determine one valid placement of the sensors. Clearly indicate the points where backtracking occurs (4 Marks)
iii. Analyse the time complexity of this approach and briefly explain how pruning reduces the number of explored configurations compared to brute-force placement. (2 Marks)</td><td>2</td><td>10</td><td>2</td><td>3</td></tr>
</table>

****