



# VIT

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

REG.NO.:

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS  
CONTINUOUS ASSESSMENT TEST - II  
WINTER SEMESTER 2025-2026

SLOT: A1+TA1

Programme Name & Branch : BTECH - Computer Science and Engineering (AI and Data Engineering),  
BTECH - Computer Science and Engineering (Cyber Security)

Course Code and Course Name : BCSE204L - Design and Analysis of Algorithms  
Faculty Name(s) : Dr. Angulakshmi M, Dr. Kuruva Lakshmana, and  
Dr. Yusuf Akhtar

Class Number(s) : VL2025260503559, 3561, and 3585

Exam Duration : 90 minutes

Maximum Marks: 50

**General instruction(s):**

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes: (Type the CO statements covered in this question paper. Use the CO number as per the syllabus copy)
  2. Demonstrate the major algorithm design paradigms.
  3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.

Q. No	Question	M	CO	BL
1.	Given the set: $S = \{5, 10, 12, 13, 15, 18\}$ Target sum = 30  1. Draw the state-space tree using backtracking. 2. Clearly indicate the bounding condition used to prune branches. 3. List all possible subsets that produce the target sum. 4. Compare this approach with Dynamic Programming in terms of space and time complexity.	10	CO 2	BL 3
2.	Given: Pattern $P = "ABABACABA"$ Text: $T = "ABABACABABACABA"$ 1. Construct the LPS (Longest Prefix Suffix) array step by step. 2. Show how KMP avoids redundant comparisons during mismatch. 3. Trace the complete matching process. 4. Compare the number of comparisons with the naïve method for this example. 5. Prove why KMP runs in $O(n+m)$ time.	10	CO 3	BL 3
3.	Consider the directed graph: Vertices: $\{A, B, C, D, E\}$ Edges with weights: $A \rightarrow B (6), A \rightarrow C (7), B \rightarrow C (8), B \rightarrow D (5), B \rightarrow E (-4), C \rightarrow D (-3), C \rightarrow E (9), D \rightarrow B (-2), E \rightarrow D (7)$ 1. Apply Bellman–Ford from source A. 2. Show all relaxation steps (iteration-wise table). 3. Identify whether a negative cycle exists. Justify. 4. Explain why Bellman–Ford works even with negative edges but Dijkstra fails. 5. Analyze time complexity.	10	CO 3	BL 3



# VIT

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

REG.NO.:

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS  
CONTINUOUS ASSESSMENT TEST - II  
WINTER SEMESTER 2025-2026

SLOT: A1+TA1

4.	<p>Consider a small flow network:  <math>S \rightarrow A</math> (16), <math>S \rightarrow B</math> (13)  <math>A \rightarrow B</math> (10), <math>B \rightarrow A</math> (4)  <math>A \rightarrow C</math> (12), <math>B \rightarrow D</math> (14)  <math>C \rightarrow B</math> (9), <math>D \rightarrow C</math> (7)  <math>C \rightarrow T</math> (20), <math>D \rightarrow T</math> (4)</p> <ol style="list-style-type: none"> <li>1. Initialize preflow.</li> <li>2. Show height and excess values after initialization.</li> <li>3. Perform push and relabel operations step-by-step.</li> <li>4. Compute maximum flow.</li> <li>5. Compare theoretical complexity with Edmond-Karp.</li> </ol>	10	CO 3	BL 3
5.	<p>Given the following line segments:</p> <ul style="list-style-type: none"> <li>• <math>L1 = (1,2) \rightarrow (6,5)</math></li> <li>• <math>L2 = (2,5) \rightarrow (7,1)</math></li> <li>• <math>L3 = (3,3) \rightarrow (8,3)</math></li> </ul> <ol style="list-style-type: none"> <li>1. Define orientation of ordered triplet of points.</li> <li>2. Compute orientation values required to determine whether: <ul style="list-style-type: none"> <li>o <math>L1</math> intersects <math>L2</math></li> <li>o <math>L1</math> intersects <math>L3</math></li> </ul> </li> <li>3. Explain special cases (collinearity and overlapping).</li> <li>4. Derive the time complexity of checking intersection of two segments.</li> <li>5. Extend the approach to check intersection among <math>n</math> segments using brute force.</li> </ol>	10	CO 3	BL 2

\*\*\*\*\*