


Final Assessment Test – April 2026

 Course: **BCSE204L - Design and Analysis of Algorithms**

 Class NBR(s): **3559/3561/3585**

 Slot: **A1+TA1**

 Time: **Three Hours**

 Max. Marks: **100**

- > KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
 > DON'T WRITE ANYTHING ON THE QUESTION PAPER

COs	CO Statements
CO1	Apply the mathematical tools to analyze and derive the running time of the algorithms
CO2	Demonstrate the major algorithm design paradigms.
CO3	Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
CO4	Articulating Randomized Algorithms.
CO5	Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)

Answer ALL Questions

(10 X 10 = 100 Marks)

1. a) Design an algorithm for the Fractional Knapsack problem using the greedy [5] CO1 BL3 approach. Given two lists of equal length:
- Profits: [5, 10, 15, 7, 8, 9]
- Weights: [1, 3, 5, 4, 1, 3]
- Determine the maximum profit that can be stored in a Knapsack of capacity $W = 15$ Kg. You must use the Fractional Knapsack approach. Show the calculation of the profit/weight ratio for each item and state the order in which items are selected. Why is the Fractional Knapsack problem solved using a Greedy approach, while 0/1 Knapsack cannot be?
- b) Obtain an upper bound for the following recurrence equation using [5] recursive tree method and verify using Master's theorem

$$T(n) = \begin{cases} 1 & \text{when } n = 1 \\ 3T\left(\frac{n}{4}\right) + n^2 & \text{when } n > 1 \end{cases}$$

2. Design the pseudocode for Divide and Conquer Maximum Subarray and Apply algorithm to find the contiguous subarray with the largest sum in the following array: CO1 BL3

$A = [-2, 5, -1, 3, -4, 7, -2, 1]$

Show the left sum, right sum, and cross sum for each recursive division. Clearly state the starting and ending indices of the maximum subarray and its sum .

- 3.(a) Consider the N-Queens problem for $N = 6$. Construct the state-space tree up to level 3. Clearly indicate promising and non-promising nodes. Show the complete backtracking process to obtain one valid solution. Modify the traditional backtracking approach to reduce unnecessary safety checks. CO2 BL4

Analyze the worst-case time complexity and compare it with brute-force permutation generation

OR

- 3.(b) You are attending a workshop with various sessions happening throughout the day. CO2 BL4

You'd like to attend as many sessions as possible without any conflicts.

Session	Start Time	End Time
A	9:00	10:00
B	9:30	11:30
C	11:00	12:00
D	12:30	14:00
E	13:00	14:30
F	14:30	15:00

Find a greedy approach to find the maximum number of sessions you can attend without attending any part of two sessions simultaneously. Describe the algorithm and implement it to obtain the solution

4. Solve the Travelling Salesman Problem for the following graph using Dynamic Programming approach : CO2 BL3

Cost Matrix (Distance between cities):

	A	B	C	D
A	∞	20	42	35
B	20	∞	30	34
C	42	30	∞	12
D	35	34	12	∞

Starting city: A

Show all calculations in detail. Determine the optimal tour and its cost. List the order of cities visited. Trace back to reconstruct the optimal path. Explain why Dynamic Programming becomes increasingly more efficient than Brute Force as the number of cities increases.

5. Given:

CO3 BL4

- Text T: "14159265358979323846" (first 20 digits of π)
- Pattern P: "59265"

Assume: Base $d=10$ and Prime $q=13$ (Hint: $H = (\text{digit}_0 * d^{(m-1)} + \text{digit}_1 * d^{(m-2)} + \dots + \text{digit}_{(m-1)}) \bmod q$)

Compute the initial hash values for the pattern and first window. Show rolling hash computation for next three shifts. Identify spurious hits (if any). Explain why Rabin–Karp performs well for multiple pattern matching. Analyse average and worst-case time complexity.

6. Given weighted directed graph:

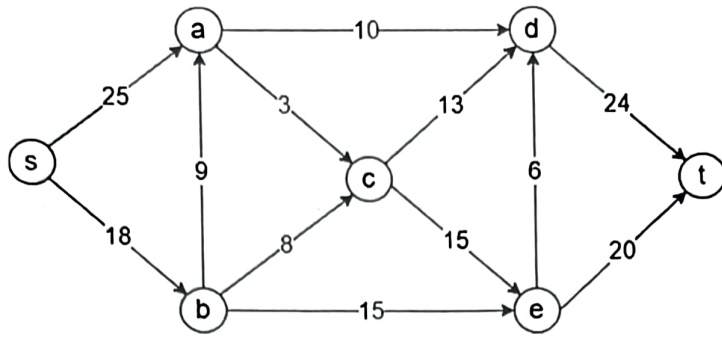
CO3 BL3

- B — A: Weight 10
- A—C : Weight 12
- C — B: Weight 20
- C — D: Weight 32
- A — D: Weight 60
- E — A: Weight 7

Find all pairs shortest path for the given weighted digraph using Floyd Warshall's Algorithm. Analyse its time complexity. Illustrate the tracing of the algorithm. Detect whether a negative cycle exists using diagonal entries.

7. Apply Ford Fulkerson's algorithm on the following network flow graph to find the maximum flow from start vertex 's' to sink vertex 't'. In each iteration, the graph traversal performed to find the augmenting path, the flow value to be augmented and the residual flow network obtained after augmenting the flow should be drawn and described.

CO3 BL3



8. Consider the following set of points 'p' in a XY plane. Construct a smallest polygon enclosing all the points. Design a $O(n \log n)$ time algorithm to find the same and trace it using the following points. Elaborate the step-by-step process of constructing the polygon.

CO3 BL3

P1(10,3), P2(6,1), P3(8,4), P4(2,2), P5(3,5), P6(1,6), P7(7,3), P8(6,7)

9. Apply Randomized quicksort algorithm by selecting the random pivot to trace the steps to sort in detail. Discuss how the average time complexity of randomized quicksort is better than the worst-case time complexity of deterministic quicksort. The elements are given below:

CO4 BL3

99, 34, 46, 22, 60, 23, 15, 14, 33, 20

10.a) Provide a summarized explanation of the complexity classes P, NP, NP-Hard, and NP-Complete, highlighting their role in assessing the hardness of real-world computational problems concerning algorithmic efficiency.

CO5 BL5

OR

10.b) Prove that the Clique Decision Problem is NP-Complete using 3-CNF SA

CO5 BL5

$$(a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee b \vee c) \wedge (a \vee b \vee c)$$

⇔⇔⇔ U/D/TZ ⇔⇔⇔