QP Code: R262B025 | Reg. No | | | | | | | | | | | AY 25

Gandhi Institute of Engineering and Technology University, Odisha, Gunupur
(GIET University)

M.Tech. (Second Semester - Regular) Examinations, May – 2026

24MCSPE12001 – Advanced Algorithms

Time: 3 hrs | Maximum: 60 Marks

Answer ALL questions

(The figures in the right hand margin indicate marks)

PART – A **(2 x 5 = 10 Marks)**

Q.1. Answer ***ALL*** questions	CO #	Blooms Level
a. Arrange the following functions in increasing order of growth rate (as $n \to \infty$): n, logn, 100, nlogn, n^3, n^n, 2^n, n!	CO1	K4
b. How does dynamic programming differ from divide-and-conquer in handling overlapping subproblems?	CO2	K2
c. Why does Dijkstra's algorithm fail with negative edge weights?	CO3	K3
d. Why is the N-Queens problem suitable for backtracking?	CO4	K3
e. What is polynomial-time reducibility?	CO5	K2

PART – B **(10 x 5 = 50 Marks)**

Answer ***ALL*** the questions	Marks	CO #	Blooms Level
2. a. Explain the significance of limit theorem in comparing growth of functions.	5	CO1	K3
b. Solve the recurrence relation using master method: $T(n)=4T(n/2)+n^2\log n$	5	CO1	K3
(OR)			
c. Solve the recurrence relation using substitution method: $T(n)=2T(n/2)+n$.	5	CO1	K3
d. Explain the concept of amortized analysis. Using a suitable example, demonstrate how amortized cost differs from worst-case cost. Also, compare amortized analysis with worst-case analysis.	5	CO1	K3
3.a. Write the algorithms for Quick Sort and its partition procedure. Also specify the best-case and worst-case time complexities of Quick Sort.	5	CO2	K3
b. Explain the Matrix Chain Multiplication problem using dynamic programming. Consider the following 6 matrices with their dimensions:	5	CO2	K4

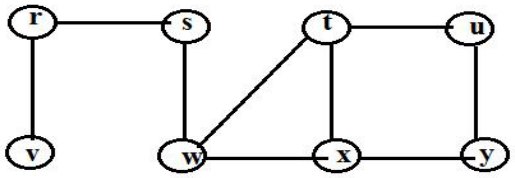
MATRIX	DIMENSION
A1	30 X 35
A2	35 X 15
A3	15 X 5
A4	5 X 10
A5	10 X 20
A6	20 X 25

(OR)

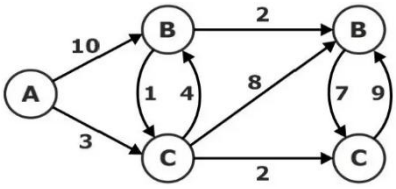
c. Define the Longest Common Subsequence (LCS). Write an algorithm to compute the length of an LCS. Also, find the LCS for the sequences. X={A,B,C,B} and Y={B,D,C,A,B}.	5	CO2	K4
d. Construct a max heap from the following list of elements:	5	CO2	K4

{16,14,10,8,7,9,3,2,4,1}. Using the constructed heap, perform heap sort to arrange the elements in sorted order. Also, write the algorithm for heap sort.

4.a.	Find the BFS of the following graph by considering **u** as the start node.	5	CO3	K3



b.	Write Dijkstra's Algorithm and find the shortest path from the source vertex to all other vertices.	5	CO3	K3



(OR)

c.	Write the algorithm for Kruskal's method to find a minimum spanning tree. Also analyze its running time.	5	CO3	K3
d.	Explain the Floyd–Warshall algorithm based on dynamic programming with a suitable example. Derive and analyze its time complexity.	5	CO3	K3
5.a.	Explain the backtracking approach in detail. Using the 8-Queens problem as an example, illustrate how the state space tree is generated and how pruning is applied to reduce the search space. Also analyze the time complexity of the approach.	5	CO4	K4
b.	Consider a string matching problem with text T="aaabaadaabaaa" and pattern P="aabaaa". Apply the Knuth–Morris–Pratt (KMP) pattern matching algorithm to find all occurrences of the pattern in the text. Show the construction of the prefix function (failure function) and illustrate each step of the matching process.	5	CO4	K4

(OR)

c.	Describe the branch and bound technique for solving the Subset Sum problem. Explain how the search space is pruned using bounding functions. Also, discuss whether the Subset Sum problem can be solved in polynomial time, justifying your answer.	5	CO4	K4
d.	Explain the naïve string-matching algorithm. Illustrate its working with a suitable example and analyse its best-case and worst-case time complexities.	5	CO4	K3
6.a.	Describe an approximation algorithm for the Travelling Salesman Problem (TSP). Explain its working with a suitable example and analyse its approximation ratio and time complexity.	5	CO5	K3
b.	Describe the branch and bound method for the assignment problem. Explain how bounding helps in reducing the search space.	5	CO5	K3

(OR)

c.	Differentiate between NP-complete and NP-hard problems with suitable examples.	5	CO5	K4
d.	Explain Cook's Theorem and discuss its significance in complexity theory.	5	CO5	K3

--- End of Paper ---