Printed Page:-5 **Subject Code:- BCSBS0501**

Roll. No:

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA

(An Autonomous Institute Affiliated to AKTU, Lucknow)

B.Tech

SEM: V - THEORY EXAMINATION (2025 - 2026)

Subject: Design and Analysis of Algorithms

Time: 3 Hours **Max. Marks: 100**

General Instructions:

IMP: *Verify that you have received the question paper with the correct course, code, branch etc.*

1. *This Question paper comprises of* ***three Sections -A, B, & C.*** *It consists of Multiple Choice Questions (MCQ's) & Subjective type questions.*

2. *Maximum marks for each question are indicated on right -hand side of each question.*

3. *Illustrate your answers with neat sketches wherever necessary.*

4. *Assume suitable data if necessary.*

5. *Preferably, write the answers in sequential order.*

6. *No sheet should be left blank. Any written material after a blank sheet will not be evaluated/checked.*

SECTION-A 20

1. Attempt all parts:-

1-a. Select the algorithm that follows the Divide and Conquer approach: CO1,K2 — 1

(a) Bubble Sort
(b) Selection Sort
(c) Heap sort
(d) Merge Sort

1-b. The complexity of the following code is? CO1,K3 — 1

```
int a = 0, i = n;
while (i > 0)
{
   a += i;
   i /= 2;
}
```

(a) O(N)
(b) O(Sqrt(N))
(c) O(N / 2)
(d) O(log N)

1-c. Recognize the property of a problem that involves breaking it into subproblems reused multiple times.CO2,K1 — 1

(a) Overlapping subproblems
(b) Prim's algorithm
(c) Kruskal algorithm
(d) Bellmen Ford Shortest path algorithm

1-d. Choose a problem that is well-suited for solving with dynamic programming. — 1

CO2,K2

(a) Merge sort
(b) Binary search
(c) Longest common subsequence
(d) Quicksort

1-e. List algorithms commonly applied to solve the single-source shortest path problem: CO3,K1 — 1

(a) Kruskal's Algorithm
(b) Prim's Algorithm
(c) Dijkstra's Algorithm
(d) Flyod-Warshal Algorithm

1-f. Select problems that can be addressed using longest subsequence approaches. CO3,K2 — 1

(a) Longest increasing subsequence
(b) Longest palindromic subsequence
(c) Longest bitonic subsequence
(d) Longest decreasing subsequence

1-g. Identify the class of decision problems solved by non-deterministic polynomial algorithms. CO4, K1 — 1

(a) NP
(b) P
(c) Hard
(d) Complete

1-h. Complete phrase: Randomized algorithm uses random bits to achieve _____ good performance over all bit choices. CO4,K2 — 1

(a) worst case
(b) best case
(c) average case
(d) none of the mentioned

1-i. List the necessary condition for a problem in NP to be NP-complete.CO5,K3 — 1

(a) It can be reduced to the 3-SAT problem in polynomial time
(b) The 3-SAT problem can be reduced to it in polynomial time
(c) It can be reduced to any other problem in NP in polynomial time
(d) Some problems in NP can be reduced to polynomial time.

1-j. State the matching time of Rabin-Karp algorithm when expected valid shifts are few and modulus large. CO5,K2 — 1

(a) Theta(m)
(b) Big-Oh(n+m)
(c) Theta(n-m)
(d) Big-Oh(n)

2. Attempt all parts:-

2.a. Explain the concept of time and space trade-offs in algorithm design. CO1,K2 — 2

2.b. Differentiate between the memoization approach and the tabulation (table) method used in dynamic programming. CO2,K3 — 2

2.c. Specify the data structure commonly used to implement the Breadth-First Search (BFS) algorithm. CO3,K1 — 2

2.d. If a problem X can be reduced to an NP-complete problem Y in polynomial time, explain whether X is NP-complete or NP-hard, and justify your answer. CO4,K4 — 2

2.e. Describe the relationship between NP-complete and NP-hard problems, and illustrate it with a suitable example. CO5,K4 — 2

<u>SECTION-B</u> 30

3.a. Answer any <u>one</u> of the following:-

3.a.(i) Find the Time Complexity CO1,K3 — 6

```
A( )
{
   int n=22;
   for (i=1; i<=n; i++)
   {
      j=2;
      while (j<=n)
      {
         j=j^2;
         printf (“CSBS”)
      }
   }
}
```


3.a.(ii) Write the big-O notation for the following function: CO1,K3 — 6

f(n) = (nlogn + n^2)(n^3 + 2) is?

3.b. Answer any one of the following:-

3.b.(i) Write down Algorithm to compute Knapsack problem. With Example also find its complexity. CO2,K3 — 6

where n =50.

Objects	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Profits	**10**	**5**	**15**	**7**	**6**	**18**	**3**
Weights	**2**	**3**	**5**	**7**	**1**	**4**	**1**

3.b.(ii) State the objective of Job Sequencing with deadlines with example. CO2,K3 — 6

Jobs	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9
Profits	15	20	30	18	18	10	23	16	25
Deadlines	7	2	5	3	4	5	2	7	3

3.c. Answer any one of the following:-

3.c.(i) Explain the steps of Huffman Coding algorithm with a suitable example. Construct Huffman codes for the symbols {A:5, B:9, C:12, D:13, E:16, F:45} and calculate — 6 the average code length. CO3,K4

3.c.(ii) Explain Prim's Algorithm for finding the Minimum Spanning Tree (MST). Apply the algorithm to the following weighted graph and derive the MST. Show all steps clearly. CO3,K3 — 6

Graph (Edge Weights):

Edge	Weight
A–B	4
A–C	3
B–C	1
B–D	2
C–D	4
C–E	5
D–E	7

3.d. Answer any one of the following:-

3.d.(i) Outline the steps involved in computing approximate solutions to NP-hard optimization problems using approximation algorithms, and provide an illustrative example. CO4,K3 — 6

3.d.(ii) Describe a string matching algorithm used for pattern finding and analyze its time complexity. CO4,K4 — 6

3.e. Answer any one of the following:-

3.e.(i) Define the following problems related to NPC: CO5,K3 — 6
(i) Vertex Cover
(ii) Clique
(iii) SAT and its variants

3.e.(ii) Define class NP in computational complexity theory. List five problems for which no polynomial-time algorithm has been discovered. CO5,K2 — 6

SECTION-C — 50

4. Answer any one of the following:-

4-a. An algorithm with time complexity O(f(n)) and processing time T(n) = cf(n), where f(n) is a known function of n, spends 10 seconds to process 1000 data items. How much time will be spent to process 100,000 data items if f(n) = n and f(n) = n ^3 ? CO1,K3 — 10

4-b. 1. Solve this recurrence Relation 2T(n/2) + 2^n. CO1,K3 — 10
2. Find the Time Complexity

```
int a = 0;
for (i = 0; i < n; i++)
{
   for (j = n; j > i; j--)
      {
         a = a + i + j;
      }
}
```

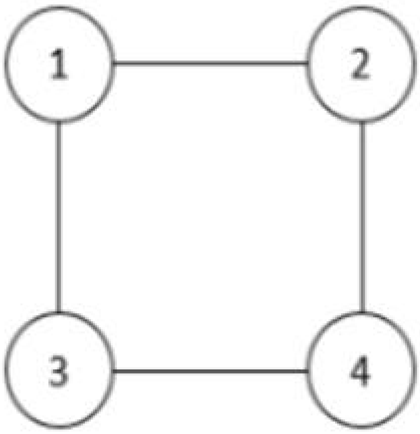
5. Answer any one of the following:-

5-a. You are given a knapsack that can carry a maximum weight of 60. There are 4 items with weights {20, 30, 40, 70} and values {70, 80, 90, 200}. Define the maximum value of the items you can carry using the knapsack? CO2,K3 — 10

5-b. Write algorithm for the Huffman code algorithm and find the Huffman code for the character of given text file — 10
Character: <45, 5, 20, 15, 10, 25> CO2,K3
Frequency:

6. Answer any <u>one</u> of the following:-

6-a. Differentiate between planar and non-planar graph. Graph coloring problem can apply on them, Explain for Bipartite graph how many colors required to color the graph. Apply backtracking on the following instance of graph coloring problem of 4 nodes and 3 colors. — 10
CO3,K3



6-b. Implement Kruskal's algorithm to find minimum spanning tree. Explain the algorithm step by step. Analyze its time complexity. Find Minimum Spanning Tree of the given graph using Kruskal's algorithm. CO3,K3 — 10

7. Answer any <u>one</u> of the following:-

7-a. Outline the procedure for solving the 3-SAT problem by reducing it to the Independent Set problem, explaining the construction steps and reasoning involved. CO4,K3 — 10

7-b. Demonstrate how reductions are used to prove that a problem is complete for a particular complexity class, supporting your explanation with a suitable illustrative example. CO4,K5 — 10

8. Answer any <u>one</u> of the following:-

8-a. Explain the concept of approximation algorithms and randomized algorithms. Provide one illustrative example for each, including a discussion of performance guarantees, approximation ratio, or expected runtime. CO5,K5 — 10

8-b. Define the class of problems beyond NP, specifically PSPACE, and describe its significance in computational complexity theory. Provide an example of a PSPACE-complete problem. CO5,K5 — 10