BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(END SEMESTER EXAMINATION)

CLASS: BTECH
BRANCH: CSE/AIML

SEMESTER : IV/ADD
SESSION : SP/2025

SUBJECT: CS241 DESIGN AND ANALYSIS OF ALGORITHMS

TIME: 3 Hours

FULL MARKS: 50

INSTRUCTIONS:
1. The question paper contains 5 questions each of 10 marks and total 50 marks.
2. Attempt all questions.
3. The missing data, if any, may be assumed suitably.
4. Before attempting the question paper, be sure that you have got the correct question paper.
5. Tables/Data hand book/Graph paper etc. to be supplied to the candidates in the examination hall.

			CO	BL
Q.1(a)	Consider the following recursive function	[5]	1	4

```
foo(n)
{
        if (n<=1)
                return 1;
        else
        {
                sum = 0;
                for(I = 0; I < n; i++)
                {
                        sum = sum +i;
                }
                return (foo(n-2)+foo(n-2)+foo(n-2)+foo(n-2)+foo(n-2));
        }
}
```

- Derive the recurrence relation that represents the time complexity of the above function.
- Analyze the time-complexity using either the substitution method or the recursion tree method.

Q.1(b) (1) Given the following functions, arrange them in asymptotically increasing Order of growth (sufficient calculations are to be provided): [5] 1 3

$f1 = 10^n$, $f2 = n^{\log n}$, $f3 = n^{\sqrt{n}}$

(2) Solve the recurrence relation using the Master Theorem if possible:

$T(n) = 0.5 \times T(n/2) + 1/n$

Q.2(a) Given an unsorted array A[1..n], where odd-indexed elements are sorted in ascending order and the even-indexed elements are sorted in descending order. Design an Algorithm to sort this array in O(n) worst-case time, in ascending order. [5] 2 4

Q.2(b) Write the MERGE-SORT(A, p, r) procedure where at each step it divides the array/sub-array into two parts such that the second part contains elements twice of the first part instead of dividing at middle. Here A, p, and r correspond to the array, the index of the first element of the array/sub-array, and the element of the last element of the array/sub-array respectively. [5] 2 3

For array A = {10, 45, 15, 40, 10, 20, 40, 25, 35}, the above MERGE-SORT(A, 1, 9) is applied to sort the array in ascending order.

Show in diagram how this procedure is applied to this array.

PTO

Q.3(a) The fractional Knapsack problem, when solved using some suitable greedy approach, is guaranteed to yield the optimal solution for all the instances. However, it may not be true for 0/1 Knapsack problem. Demonstrate it using a suitable example.
Suppose all items have the same value-to-weight ratio, how does the greedy algorithm behave, in terms of performance, for these two problem types? [5] 2 4

Q.3(b) Does a graph with negative edge weights affect Minimum Spanning Tree algorithms? If yes, then modify the algorithm to handle this situation and analyze the change in time complexity. If no, then explain the working of the Prim's algorithm (no need to specify the algorithm) in case of negative edge weights by means of an example. [5] 3 3

Q.4(a) Given: [5] 2 3
Keys: A, B, C; Probabilities: p = [0.2, 0.5, 0.3]
Construct the optimal BST and calculate the minimum cost.

Q.4(b) Given the following distance matrix, determine the optimal TSP path and cost. [5] 2 3

	A	B	C	D
A	0	12	3	45
B	12	0	17	8
C	3	17	0	23
D	45	8	23	0

Q.5(a) Consider the classes P, NP, NP-complete, and NP-hard. Suggest a diagrammatic comparison between the following two possible cases: [5] 5 3
Case 1: The classes P and NP are not equal.
Case 2: The classes P and NP are equal.
Briefly argue over the possible correctness of the solution provided by you.

According to Don Knuth, the name "NP-complete" was popularized by Alfred Aho, John Hopcroft, and Jeffrey Ullman in their celebrated textbook "The Design and Analysis of Computer Algorithms". State the notion of the term "complete" in the aforementioned statement.

Given that the theory of NP Completeness applies directly not to optimization problems, however, but to decision problems, can a decision problem be NP Hard? If yes, give an example (only the statement of the problem is to be stated).

Q.5(b) Given that the MAX-CLIQUE problem is at the intersection of the problem classes NP and NP-hard, show that the VERTEX-COVER problem is computationally at least as hard as the MAX-CLIQUE problem. [5] 5 3
Following your proof, is it correct to say that the MAX-CLIQUE problem is computationally no harder to solve than the VERTEX-COVER problem? Justify your answer through proper argument.

::::::28/04/2025::::::M