Reg. No. : E N G G T R E E . C O M

Question Paper Code : 30009

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2023.

Third Semester

Artificial Intelligence and Data Science

AD 3351 – DESIGN AND ANALYSIS OF ALGORITHMS

(Common to: Computer Science and Business Systems)

(Regulations 2021)

Time : Three hours | Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is an algorithm?
2. Compare between Worst-Case, Best-Case, and Average-Case Efficiencies of an algorithm.
3. Outline the algorithmic steps for the string-matching problem using brute-force technique.
4. How different is the traveling salesman problem from the problem of finding a Hamiltonian circuit?
5. State the principle of optimality in dynamic programming.
6. What is Dijkstra algorithm used for?
7. Construct a feasible solution space of below LPP

   Maximize $3x + 5y$

   Subject to $x + y \leq 4$

   $x + 3y \leq 6$

   $x \geq 0, y \geq 0$
8. How can you solve a maximum flow problem of a flow network with multiple sources and or multiple sinks?
9. What is meant by N-Queen problem?
10. How does a branch and bound algorithm work?

PART B — (5 × 13 = 65 marks)

11. (a) (i) Illustrate in detail about asymptotic notations and basic efficiency classes of algorithm (7)

(ii) Compare the orders of growth following time complexity functions.

- $n(n-1)/2$ and n^2
- $\log n$ and $\sqrt{n}$
- $n!$ and 2^n (6)

Or

(b) (i) Devise a non-recursive algorithm for the element uniqueness problem: check whether all the elements in a given array of n elements are distinct. Analyze its efficiency. (7)

(ii) Compute the factorial function $F(n) = n!$ for an arbitrary nonnegative integer n with the recursive algorithm and analyze its efficiency. (6)

12. (a) (i) Consider the assignment problem

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

Use exhaustive search technique to find the solution. (6)

(ii) Construct a presorting based algorithm to compute the mode of a given list of numbers. For example, for 5, 1, 5, 7, 6, 5, 7, the mode is 5. (7)

Or

(b) (i) Elaborate in detail about heap construction and heap sort algorithm. (7)

(ii) Apply Strassen's algorithm to compute

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$

exiting the recursion when n = 2, i.e., computing the products of 2-by-2 matrices by the brute-force algorithm. (6)

13. (a) (i) Several coins are placed in cells of an n x m board, no more than one coin per cell. A robot, located in the upper left cell of the board, needs to collect as many of the coins as possible and bring them to the bottom right cell. On each step, the robot can move either one cell to the right or one cell down from its current location. When the robot visits a cell with a coin, it always picks up that coin. Design an algorithm to find the maximum number of coins the robot can collect and a path it needs to follow to do this. (7)

(ii) Apply Warshall's algorithm to find the transitive closure of the digraph defined by the following adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$ (6)

Or

(b) (i) Write pseudo code of the bottom-up dynamic programming algorithm for the knapsack problem. (6)

(ii) Write pseudo code of the algorithm that finds the composition of an optimal subset from the table generated by the bottom-up dynamic programming algorithm for the knapsack problem (7)

14. (a) (i) Prove that a matching M is a maximum matching if and only if there exists no augmenting path with respect to M. (7)

(ii) Write down the algorithm for maximum matching in a bipartite graph by a BFS-like traversal. (6)

Or

(b) (i) Write down the steps of simplex method to solve LPP. (6)

(ii) Explain in detail about maximum flow problem. (7)

15. (a) Compare and contrast between P, NP, NP-complete and NP Hard Problems with examples. (13)

Or

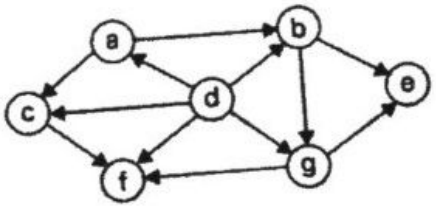
(b) (i) Explain in detail about approximation algorithms. (7)

(ii) Prove that the twice-around-the-tree algorithm is a 2-approximation algorithm for the traveling salesman problem with Euclidean distances. (6)

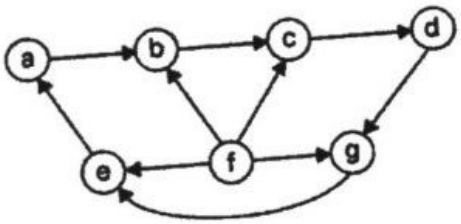
PART C — (1 × 15 = 15 marks)

16. (a) Apply the DFS-based algorithm to solve the topological sorting problem for the following digraphs:

(i)



(ii)



Or

(b) Find a stable marriage matching for the instance defined by the following ranking matrix:

	A	B	C	D
α	1,3	2,3	3,2	4,3
β	1,4	4,1	3,4	2,2
γ	2,2	1,4	3,3	4,1
δ	4,1	2,2	3,1	1,4