Reg. No. : [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Question Paper Code : 40030

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2021.

Third Semester

Artificial Intelligence and Data Science

AD 8351 — DESIGN AND ANALYSIS OF ALGORITHMS

(Common to : Computer Science and Business System)

(Regulations 2017)

Time : Three hours | Maximum : 100 marks

Answer ALL questions.

PART A — ($10 \times 2 = 20$ marks)

1. Give an algorithm to search for an element 'x' in an array $[1:n]$. What is its time complexity?

2. Derive the average-case time complexity for successfully searching in a Binary Search Tree.

3. Give an algorithm that merges three sorted arrays $A[1:m], B[1:n]$ and $C[1:k]$ and yields the output as a single sorted array having $m+n+k$ elements.

4. List down the properties of Minimum Cost Spanning Trees pertaining to a Graph.

5. Outline the characteristic features of State- Space approach.

6. What is Memorization? List down its advantage over traditional recursion technique while generating Fibonacci Series $F(n) = F(n-1) + F(n-2)$.

7. What is the philosophy behind Branch and Bound technique?

8. Illustrate a state space that provides all the possible positions of placing queens in a 4×4 chessboard to solve a 4-Queen Problem.

9. What is Euler's path and Euler's circuit in a graph? What are the conditions for a graph to have an Euler circuit?

10. What are NP-Complete Problems? Give example.

PART B — (5 × 13 = 65 marks)

11. (a) (i) Find the time complexity of the below recurrence relation:

$$T(n) = \begin{cases} 2T(n-1) - & if \ n > 0 \\ 1 & otherwise \end{cases} \quad (7)$$

(ii) Find the time complexity of the following recurrence relation:

$$T(n) = \begin{cases} 2T(n/2) + 1 & if \ n > 1 \\ 1 & otherwise \end{cases} \quad (6)$$

Or

(b) Assume the following preposition holds. (13)

(i) $f(n) = O(g(n))$ and $g(n) \neq O(f(n))$

(ii) $g(n) = O(h(n))$ and $h(n) \neq O(g(n))$

With proper analysis, prove or disprove the following statements.

(1) $h(n) = O(f(n))$

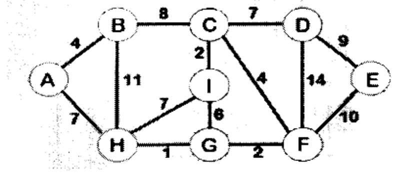
(2) $f(n) + g(n) = \theta(h(n))$

(3) $g(n) + h(n) \neq \theta(h(n).h(n))$

12. (a) Devise an algorithm for Quick Sort and derive its time complexity. For the above devised quick sort algorithm, derive the time complexity if all the elements are arranged in ascending order. Illustrate a Recurrence Tree for the same. (13)

Or

(b) Give an algorithm to find minimal cost spanning tree (MCST) using prim's algorithm. Analyze its complexity by considering priority queue implementation using mere sophisticated data structures. (13)



13. (a) Given an undirected weighted graph, devise an algorithm that finds all pairs shortest path. What would the expected time complexity for your algorithm? (13)

Or

(b) (i) Given an undirected unweighted graph (G) give algorithms that perform Breadth First Search (BFS) and Depth First Search (DFS) strategy to visit all the vertices of G. List down the order of visiting the vertices in each traversal method. (10)

(ii) What would be expected time complexity of the prescribed algorithm? (3)

14. (a) (i) Design a Recursive algorithm (pseudocode) for solving 0/1 Knapsack problem using backtracking approach. (6)

(ii) Give an algorithm to check whether the given graph is bipartite or not. (7)

Or

(b) Devise an algorithm that computes the minimum number of modification operations needed to convert one string into another string. The following are the operations that are allowed:

(i) Add a Character (3)

(ii) Delete a Character (3)

(iii) Change a Character (3)

(iv) Twiddle operation. (4)

Also outline the Recurrence relation for the same.

15. (a) (i) Give a brief note about SAT and 3-SAT problems with example for each. (7)

(ii) Write short notes on P. NP and NP-Hard problems with example for each. (6)

Or

(b) Outline the two broad categories of Randomized algorithms. Give a few examples. Give an algorithm that performs randomized quick sort by considering an arbitrary element of an input array as the pivot element. (13)

PART C — (1 × 15 = 15 marks)

16. (a) Solve the following 0/1 Knapsack problem using Branch and Bound Strategy. (15)

Weight (Wi)	2	3.14	1.98	5	3
Price (Pi)	40	50	100	95	30

Or

(b) Solve the following Travelling Salesman Problem of the given weighted matrix M. The value stored in $M[i, j]$ represents the distance from city 'i' an city 'j'. (15)

	C1	C2	C3	C4	C5
C1	0	9	5	5	6
C2	8	0	19	13	17
C3	10	4	0	7	1
C4	11	6	9	0	15
C5	8	18	7	17	0