USN | | | | | | | | | | 18CS42

Fourth Semester B.E. Degree Examination, Dec.2023/Jan.2024
Design & Analysis of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Explain asymptotic notations BigO, Big Ω and Bigθ that are used to compare the order of growth of an algorithm with example. **(06 Marks)**

b. List two methods to measure the time complexity of algorithm by counting program steps. Apply the same for the algorithm to find the SUM of n numbers. **(06 Marks)**

c. Write an algorithm to search a key using sequential search. Derive its time efficiency for best case, worst case and average case. **(08 Marks)**

OR

2 a. Define an Algorithm. Discuss the criteria's that an algorithm must satisfy with an example. **(06 Marks)**

b. Consider the following algorithm :

Algorithm : GUESS (A[ ][ ])
Method : for i $\leftarrow$ 0 to n − 1
  for j $\leftarrow$ 0 to i
  A[i] [j] $\leftarrow$ 0

(i) What does the algorithm compute?
(ii) What is the basic operation?
(iii) What is the time complexity of this algorithm? **(06 Marks)**

c. Explain the mathematical analysis of non-recursive algorithm. Write an algorithm to check whether all the elements of given array are distinct. Give its worst case time complexity. **(08 Marks)**

Module-2

3 a. Write the algorithm for Merge Sort. Illustrate with an example. **(06 Marks)**

b. Apply the source removal method to obtain topological sort for the graph in Fig. Q3 (b). **(06 Marks)**

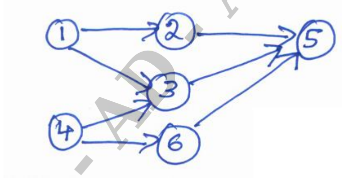

Fig. Q3 (b)

c. Apply quicksort to sort the following list in ascending order :
25, 91, 46, 35, 11, 82, 14, 55
Represent the recursive call in the form of tree. **(08 Marks)**

OR

4 a. What are the 3 variations of decrease and conquer technique. Explain in detail. **(06 Marks)**

b. Solve the following recurrence relation and find the upper bound using substitution method.

$$T(n) = 2.T\left(\frac{n}{2}\right) + n \;;\; T(1) = 2$$ **(06 Marks)**

c. Explain recursive binary search algorithm. Derive its time efficiency for best, worst and average case. **(08 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-3

5 a. What is the solution generated by job sequencing when n = 5
$(P_1, P_2, P_3, P_4, P_5) = (20, 15, 10, 5, 1)$
$(d_1, d_2, d_3, d_4, d_5) = (2, 2, 1, 3, 3)$ **(06 Marks)**

b. Construct Huffman code for the following data :

Symbol :	A	B	C	D	E
Frequency :	0.35	0.1	0.2	0.2	0.15

Also Encode CAB and DAC. **(06 Marks)**

c. Apply Prisms and Kruskals algorithm to get the minimium spanning tree for the graph given in Fig. Q5 (c). **(08 Marks)**

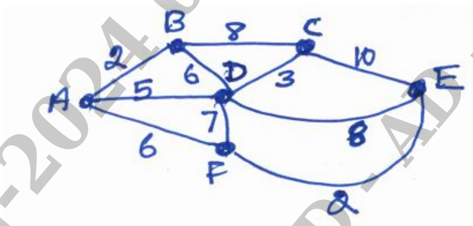

Fig. Q5 (c)

OR

6 a. Obtain the solution for the fractional Knapsack problem using greedy method for n = 3. Capacity m = 20, Values $V_1 = 25$, $V_2 = 24$, $V_3 = 15$ and weights $w_1 = 18$, $w_2 = 15$, $w_3 = 10$ respectively. **(06 Marks)**

b. Sort the array 2, 9, 7, 6, 5,8 by heap sort. Show the intermediate steps. **(06 Marks)**

c. Apply Dijikstra's algorithm to find single some shortest path for the graph given in Fig. Q6 (c). Consider node 6 as source. **(08 Marks)**

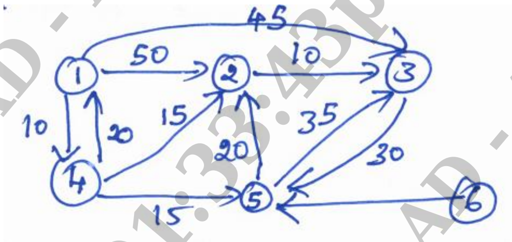

Fig. Q6 (c)

Module-4

7 a. Find all pair shortest path for the graph given in Fig. Q7 (a) using Floyd's algorithm. **(10 Marks)**

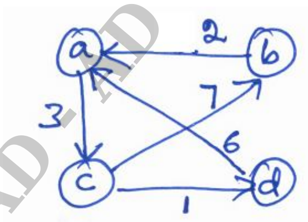

Fig. Q7 (a)

b. Apply bottom up dynamic programming algorithm for the following instance of the knapsack problem. Knapsack capacity M = 10.

Item	Weight	Value
1	7	42
2	3	12
3	4	40
4	5	25

(10 Marks)

OR

8 a. Apply Bellman-ford algorithm to the graph given in Fig. Q8 (a). Find the shortest path to all the vertices from S. **(10 Marks)**

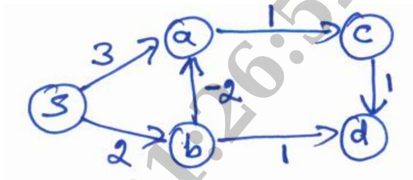

Fig. Q8 (a)

b. Solve the following TSP using dynamic programming,

	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Starting City = 1

(10 Marks)

Module-5

9 a. Let w = {3, 5, 6, 7} and m =15. Find all possible subsets of w that sum to m. Draw the state space tree. **(10 Marks)**

b. With the help of state space tree, solve the following instance of knapsack problem by branch and bound algorithm. Knapsack capacity w = 10.

Item No.	1	2	3	4
Weight	4	7	5	3
Value	40	42	25	12

(10 Marks)

OR

10 a. Apply branch and bound algorithm to solve the TSP for the graph given in Fig.Q10 (a). Consider start city as A. Give the state space tree. **(10 Marks)**

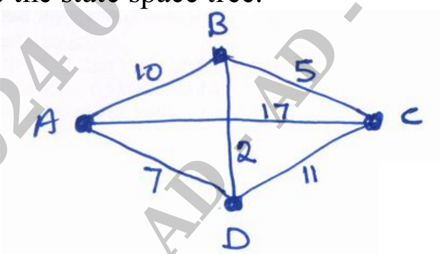

Fig. Q10 (a)

b. Explain the following with example :
(i) Class NP problems
(ii) Class P problems
(iii) NP complete problem
(iv) NP hard problem **(10 Marks)**

* * * * *