DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
FIRST SESSIONAL
SUBJECT: (CE-423) DESIGN & ANALYSIS OF ALGORITHMS

Examination	**: B.Tech Semester-IV**	**Seat No**	**: _____**
Date	**: 01/01/2024**	**Day**	**: Monday**
Time	**: 01:00 P.M.-02:15 P.M.**	**Max. Marks**	**: 36**

INSTRUCTIONS:

1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 Do as directed. **[12]**

CO1 U (a) State true/false and explain your answer: "Greedy algorithm does not guarantee an optimal solution always". **[2]**

CO2 N (b) In case of dense graph among Kruskal and Prim, which MST finding algorithm is faster? Justify. **[2]**

CO2 E (c) Let T be a tree constructed by Dijkstra's algorithm in the process of solving the single-source shortest-paths problem for a weighted connected graph G. **[2]**

i. True or false: "T is a spanning tree of G".
ii. True or false: "T is a minimum spanning tree of G".

CO2 A (d) Solve following using Master's Theorem. Show necessary steps of computation. **[2]**

$$T(n) = 9T(n/3) + n^3, n>1$$
$$= c, n=1$$

CO1 U (e) Identify the **correct options only** and show their correctness by obtaining constant c and threshold n_0: **[2]**

i. $2^n = O(n^2)$ **iii.** $n^2 + 5n + 6 = \theta(n^2)$
ii. $n^2 = O(2^n)$ **iv.** $n^2 + 5n + 6 = \theta(n^3)$

CO2 N (f) Arrange following functions in **increasing order** of their **asymptotic complexity** with necessary justification. **[2]**

$f1(n)=2^n$ **$f2(n)=n^{1.5}$** $f3(n)=n*\log(n)$ **$f4(n)=n^{\log n}$**

Q.2 Attempt *Any Two* of the following questions. **[12]**

CO1 C (a) Consider the scenario of scheduling talks or sessions for a conference. Each talk or session has a fixed start time and finish time, and the goal is to schedule as many talks as possible without any overlap. The objective is to maximize the number of talks or sessions held during the conference. Write a greedy algorithm to solve the problem and analyze the time complexity of your algorithm. **[6]**

CO1 N (b) Write an algorithm for solving the problem of Job Sequencing with deadline which has asymptotic time complexity O (n*lg n). Also show the trace of the algorithm on the below given problem instance. **[6]**

Job ID	**A**	**B**	**C**	**D**	**E**
Deadline	2	3	1	3	1
Profit	55	75	100	30	80

CO1 A (c) Show the creation of binary min-heap for the below given array. Show **each and every** step using binary tree and explain it. **[6]**

16	14	10	8	7	9	3	2	4	1

Q.3 Attempt the following questions. **[12]**

CO1 C (a) i. Write **Partition ( )** function of the Quick sort to find the proper position of the Pivot. What is its Time Complexity? **[3]**

ii. Write algorithm to find the **K^{th} smallest Element** from the Array A of n elements **without** sorting the array. Use **Partition ( )** to obtain the K^{th} smallest Element. **[3]**

CO2 A (b) i. Solve following using **Recurrence Tree Method**. Show every necessary step of Computation. **[4]**

$$T(n)=4T(n/2) + cn, \quad n>1$$
$$=c, \quad n=1$$

ii. Give example of the divide and conquer based recurrence, to solve which the Master's Theorem is not useful. Justify the reason for the same. **[2]**

OR

CO1 C (a) Given an array containing **n** integers consisting of both positive and negative values. Write **Divide and Conquer based** algorithm to find the Sub-Array having **Minimum Sum**. Derive its recurrence and solve it using Master's Theorem. **[6]**

CO2 A (b) i. Write the recurrence for getting the output of following function f. Solve the recurrence using any appropriate Method. Show every necessary step of Computation. **[3]**

```
int f(int n)
{
   if(n= =1)
     return 1;
   else
   {  x=4*f(n-1)+1;
      return x;
   }
}
```

ii. Let us say the problem is divided in to 8 problems, each of which is of size **n/2**. Time required to divide the problem is **O (n)** and to Merge the sub-problems is **O (n^2).** Write the recurrence for the same and solve it using Master's Theorem. **[3]**