Enrollment No: _____ Exam Seat No:_____

C. U. SHAH UNIVERSITY

Winter Examination-2019

Subject Name : Design and Analysis of Algorithms

Subject Code : 4TE05DAA1 **Branch: B.Tech (CE)**

Semester : 5 **Date : 21/11/2019** **Time : 10:30 To 01:30** **Marks : 70**

Instructions:

(1) Use of Programmable calculator & any other electronic instrument is prohibited.
(2) Instructions written on main answer book are strictly to be obeyed.
(3) Draw neat diagrams and figures (if necessary) at right places.
(4) Assume suitable data if needed.

Q-1 **Attempt the following questions:** **(14)**

a) Arrange following rate of growth in increasing order.
2^N, n log n, n^2, 1 , n, log n, n!, n^3

b) What is memorization?

c) What is space complexity of an algorithm?

d) Define Θ notation.

e) What is time complexity of fun()?

```
int fun(int n)
{
  int count = 0;
  for (int i = n; i > 0; i /= 2)
     for (int j = 0; j < i; j++)
        count += 1;
  return count;
}
```

(i).O(n^2) (ii). O(nLogn) (iii). O(n) (iv).O(nLognLogn)

f) What is principal of optimality?

g) What is amortized analysis?

h) Is $2^{n+1} = O(2^n)$? Explain.

i) Give big theta (Θ) notation for $f(n) = 14 * 7 + 83$.

j) List out characteristics of Greedy algorithm.

k) Give best case and worst case time complexity of linear search algorithm.

l) What is backtracking?

m) Give big omega (Ω) notation for $f(n) = 83n^3 + 84n$.

n) Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove following.
$f(n) + g(n) = \Theta(\min(f(n), g(n)))$

Attempt any four questions from Q-2 to Q-8

Q-2 **Attempt all questions**

(a) Using recurrence tree method solve the following recurrences: **(04)**

(i) $T(n) = T(n/3) + T(2n/3) + O(n)$

(ii) $T(n) = 3T(n/4) + cn^2$

(b) What is an algorithm? Explain various properties of an algorithm. **(04)**

(c) Write an algorithm for quick sort and also give its best case, worst case and average case time complexity. **(06)**

Q-3 **Attempt all questions** **(14)**

(a) Explain master theorem and solve the following recurrence equation with master method

1. $T(n)= 9T(n/3) + n$
2. $T(n)= 3T(n/4) + n\lg n$.

(b) Explain Binary search algorithm with divide and conquer strategy and use the recurrence tree to show that the solution to the binary search recurrence $T(n)= T(n/2) + \Theta(1)$ is $T(n) = \Theta(\lg n)$.

Q-4 **Attempt all questions** **(14)**

(a) Write equations for finding shortest path using Floyd-Warshall algorithm. Find out shortest path for below mentioned all pairs of graph.

	A	B	C	D
A	0	∞	3	∞
B	2	0	∞	∞
C	∞	7	0	1
D	6	∞	∞	0

(b) Explain merge sort with suitable example. Also give its recurrence equation and its best case, worst case and average case time complexity.

Q-5 **Attempt all questions** **(14)**

(a) Write down Kruskal's algorithm for finding minimum spanning tree. Give one example. Also give its worst case and best case running time complexity.

(b) Solve following knapsack problem using dynamic programming algorithm with capacity of knapsack W=5, Weight and Value are given as: (2, 12), (1, 10), (3, 20), (2, 15).

Q-6 **Attempt all questions** **(14)**

(a) Explain spurious hits in Rabin-Karp string matching algorithm with example. With working modulo q=13, how many spurious hits does the Rabin-Karp matcher encounter in the text T = 2359023141526739921 when looking for the pattern P = 31415?

(b) Using greedy algorithm find an optimal solution for knapsack instance where n=7, M = 15, $(P1, P2, P3, P4, P5, P6, P7)=(10, 5, 15, 7, 6, 18, 3)$ and $(w1, w2, w3, w4, w5, w6, w7) = (2, 3, 5, 7, 1, 4, 1)$

Q-7 **Attempt all questions** **(14)**

(a) Explain N-Queen problem with an example of 8-queens problem. Give at least four possible solutions of 8-queens problem.

(b) Explain how to find out Longest Common Subsequence of two strings using Dynamic Programming method. Find any one Longest Common Subsequence of given two strings using Dynamic Programming.
S1=abbacdcba
S2=bcdbbcaac

Q-8 **Attempt all questions** **(14)**

(a) Explain the class P and NP, polynomial time reduction, NP-hard problem and NP-complete problem with an example of each.

(b) Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is <4, 10, 3, 12, 20, 7>.