Name: **Enrolment No:**	

UPES
End Semester Examination, December 2023

Course: Design and Analysis of Algorithm **Time:** 03 hrs.
Semester: III
Program: B.Tech (CSE-H+NH)-All Spec
Course Code: CSEG2021 **Max. Marks:** 100

Instructions:

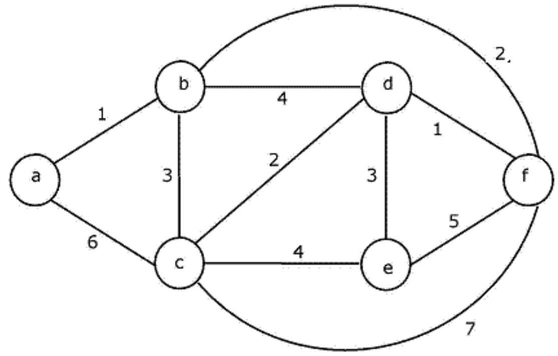
SECTION A
(5Qx4M=20Marks)

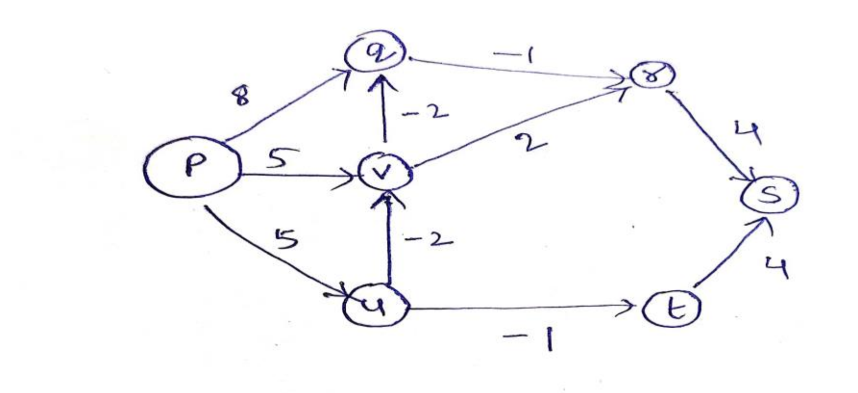
S. No.		Marks	CO

Q 1)	Consider the following function:
	int unknown(int n) {
	   int i, j, k = 0;
	   for (i = n/2; i <= n; i++)
	      for (j = 2; j <= n; j = j * 2)
	         k = k + n/2;
	   return k;
	}
	What is the returned value of the above function?
		4	**CO1**

Q 2)	Arrange the following asymptotic complexity of functions f1, f2, f3 and f4 in increasing order. f1(n) = 2^n f2(n) = n^(3/2) f3(n) = nLogn f4(n) = n^(Logn)	**4**	**CO1**
Q 3)	Among, Merge sort, Insertion sort, Bubble sort which sorting techniques is the best in the worst case. Support your argument with an example and analysis.	**4**	**CO2**
Q 4)	Consider the equality $\sum_{i=0}^{n} i^3 = X$ and the following choices for X: I. $\Theta(n^4)$ II. $\Theta(n^5)$ III. $O(n^5)$ IV. $\Omega(n^3)$	**4**	**CO1**

	Which of the above choices are correct in replacement of X to make the equality correct?		
Q 5)	Consider the recurrence function and calculate T(n) in terms of Θ notation using Master's theorem. $$T(n) = \begin{cases} 2T(\sqrt{n}) + 1, & n > 2 \\ 2, & 0 < n \leq 2 \end{cases}$$	**4**	**CO2**
SECTION B **(4Qx10M= 40 Marks)**			
Q 6)	Discuss N-queen problem. Illustrate the solution of 4-Queens problem using backtracking. What is the time complexity of the algorithm? OR Is counting sort a comparison based sorting technique? Justify your answer with the help of an example. Discuss its time and space complexity.	**10**	**CO5**
Q 7)	Explain the P, NP, NP-hard, NP-complete classes? Give relationship between them?	**10**	**CO6**
Q 8)	Suppose you are choosing between the following of three algorithms: a. Algorithm A solves problems by dividing them into five sub problems of half the size, recursively solving each problem, and then combining the solution in linear time. b. Algorithm B solve the problems of size n by recursively solving two sub-problems of size n-1 and then combining the solutions in constant time. c. Algorithm C solves problems of size n by dividing them into nine sub problems of size n/3, recursively solving each problem, and then combining the solutions in $O(n^2)$ time. What are the running times of each of three algorithms (in big-oh notation), and which would you choose?	**10**	**CO2**
Q 9)	Consider the following graph:	**10**	**CO3**

	 Find the minimum spanning tree using Kruskal's Algorithm.		
SECTION-C **(2Qx20M=40 Marks)**			
Q 10)	Assume that multiplying a matrix G_1 of dimension p×q with another matrix G_2 of dimension q×r requires pqr scalar multiplications. Computing the product of n matrices $G_1G_2G_3$ ….. G_n can be done by parenthesizing in different ways. Define G_iG_{i+1} as an explicitly computed pair for a given paranthesization if they are directly multiplied. For example, in the matrix multiplication chain $G_1G_2G_3G_4G_5G_6$ using parenthesization $(G_1(G_2G_3))(G_4(G_5G_6))$, G_2G_3 and G_5G_6 are only explicitly computed pairs. Consider a matrix multiplication chain $F_1F_2F_3F_4F_5$, where matrices F_1,F_2,F_3,F_4 and F_5 are of dimensions 2×25,25×3,3×16,16×1 and 1×1000, respectively. In the parenthesization of $F_1F_2F_3F_4F_5$ that minimizes the total number of scalar multiplications, calculate the explicitly computed pair(s). Also, find out the minimum number of scalar multiplications required to solve the given problem and optimal parenthesis sequence of matrices. What is the time and space complexity of matrix chain multiplication using dynamic programming? **OR** Consider two strings X=abacc and Y=babcacb. Let P be the longest common subsequence (not necessarily contiguous) between X and Y and let Q be the number of such longest common subsequences between X and Y. Calculate 10P+Q using dynamic programming. Discuss its time and space complexity.	**20**	**CO4**
Q 11)	Illustrate the working of Bellman Ford Shortest Path Algorithm on a given graph. Discuss the drawback of Bellman Ford Single Source Shortest Path Algorithm. What is the time complexity of Bellman Ford, if the given graph is complete and consists of n vertices?	**20**	**CO3**

q
r
p
v
s
u
t
8
-1
-2
5
2
4
5
-2
4
-1