Roll No. _____

Pandit Deendayal Petroleum University, Gandhinagar

School of Technology

End-Semester Examination

B. Tech. (Computer + ICT) **Semester - IV**

Date: 22/05/2019 **Time: 10.00 am to 1.00 pm**

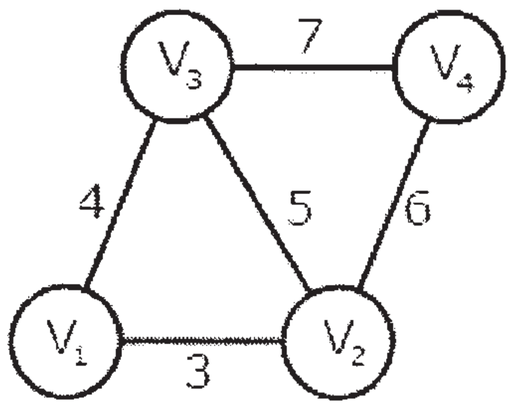
Course Name : Design & Analysis of Algorithms **Course Code : 17CP211T**

Max. Marks: 100

Instructions to students:

1. Do not write anything other than your roll number on the question paper.
2. Assume suitable data wherever required and mention it clearly.
3. You are required to answer all the questions in sequence as given in the question paper.

Q.1 Answer the following questions. (I to VIII : 2 marks each. IX : 4 marks) **(20)**

(I)	An undirected graph $G(V,E)$ contains $n(n > 2)$ nodes named $V_1, V_2, \ldots, V_n$. Two nodes V_i, V_j are connected if and only if $0 < \|i - j\| <= 2$. Each edge (V_i, V_j) is assigned a weight $i + j$. A sample graph with n = 4 is shown below.  What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes? A. $1/12(11n^2 - 5n)$ C. $n^2 - n + 1$ D. $6n - 11$ B. $3n + 1$
(II)	Write full form of P and NP.
(III)	What are the sizes of chess board for which solution using N-queen is not available?
(IV)	Explain the use of approximation algorithm.
(V)	The complexity of following code is $\theta(\_\_)$ int fun(int n) { int count = 0; for (int i = 0; i< n; i++) for (int j = i; j > 0; j--) count = count + 1; return count;}
(VI)	Consider $f(n) = 3n^{\sqrt{n}}$, $g(n) = 2^{\sqrt{n}\log_2 n}$, $h(n) = n!$. Which of following statement(s) true. A. $h(n) = O(f(n))$ D. $h(n) = O(g(n))$ B. $g(n) = O(f(n))$ C. $f(n) = O(g(n))$
(VII)	Arrange complexities in non-decreasing order. $1, n^2, nlgn, n^{1.5}, n!, 2^n, n^{lgn}$
(VIII)	In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed. Consider the following for loops: A) for(i = 0; i< n; i++) B) for(i = 0; i< n; i += 2) C) for(i = 1; i< n; i *= 2) D) for(i = n; i> -1; i /= 2) If n is the size of input(positive), which function is most efficient? Why?
(IX)	Design iterative and recursive function FUN(C,N) having complexity $\theta(C^N)$ without using any other in-built functions. Also prove the complexity. Consider C is positive constant i.e. $C > 0$.

Q.2 Attempt Following (Any Five): **(40)**

(I)	Explain the best case, average case and worst case with example in each
(II)	Explain three cases of master theorem with example in each
(III)	Write an algorithm for MST using Prim and derive its complexity
(IV)	Write an algorithm for LCS using dynamic programming and derive its complexity.
(V)	Write a recursive algorithm for exponent using divide & conquer and derive its complexity
(VI)	Explain Branch and Bound technique using Travelling Salesman Problem (TSP).

(VII)	Suppose you are given an array A with **n** entries, with each entry holding a distinct number. You are told that the sequence of values A[1], A[2], . . . , A[n] is *unimodal*: For some index **p** between **1** and **n**, the values in the array entries increase up to position **p** in A and then decrease the remainder of the way until position **n**. (So if you were to draw a plot with the array position **j** on the x-axis and the value of the entry A[j] on the y-axis, the plotted points would rise until x-value **p**, where they'd achieve their maximum, and then fall from there on.) You'd like to find the "peak entry" **p** without having to read the entire array—in fact, by reading as few entries of A as possible. Write an efficient algorithm to find the entry **p** from A. Prove the complexity for same i.e. best case is $\theta(1)$, worst case $\theta(\log_2 n)$.

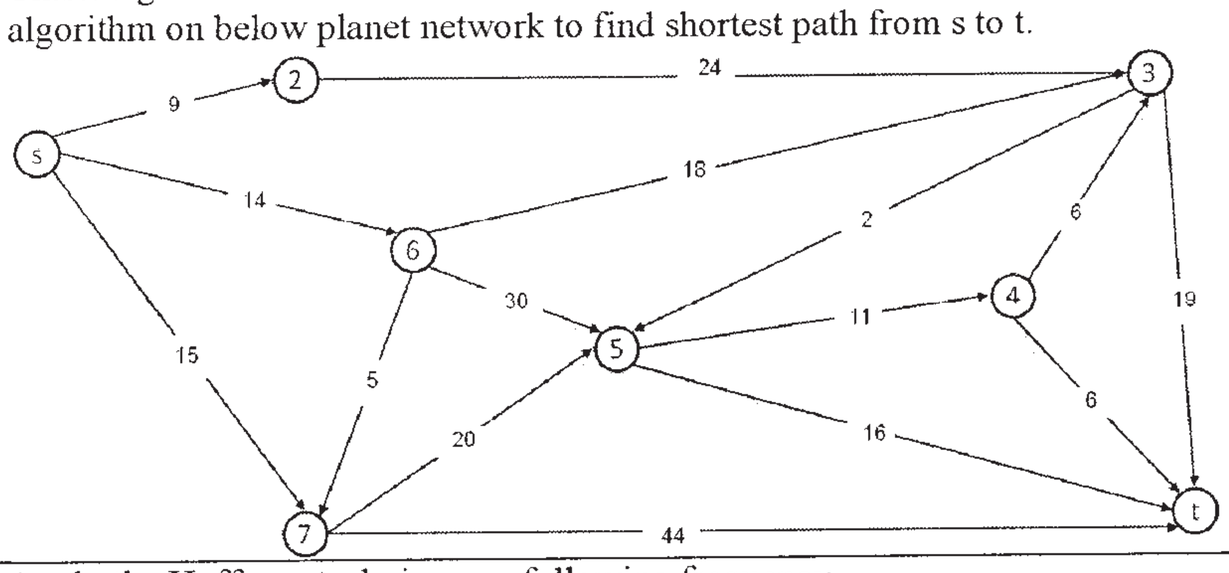
Q.3 Attempt following (Any Four): **(20)**

(I) Apply Merge sort on {7,4,12,3,90,5,2,67,14,1,45,13,80}

(II) Solve following 0/1 knapsack using dynamic programming for knapsack capacity W=5

Value	3	4	5	6
Weight	2	3	4	5

(III) Thanoz gets the information that one of stone is at t and wants to reach by shortest distance from s. Show Dijkstra's algorithm on below planet network to find shortest path from s to t.



(IV) Apply the Huffman technique on following frequency

Char	N	I	S	H	A	n	T	D	O	s	h	i
Freq.	1	8	2	1	4	1	2	2	2	1	1	1

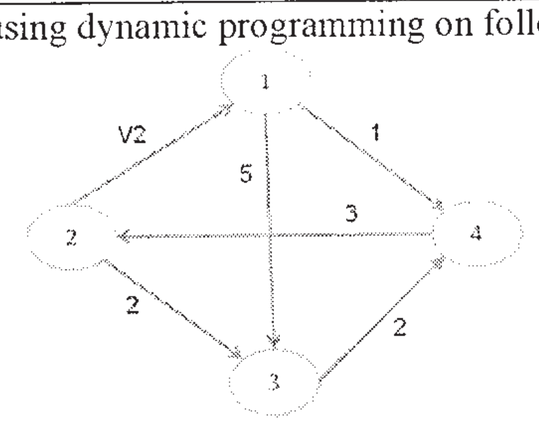
(V) Solve $T(n) = 2T(\sqrt{n}) + \log_2 n$ using suitable method. Assume suitable data if required.

Q.4 Attempt Following (Any Four): **(20)**

Consider V1= 5th smallest element of Q.3 (I). V2=Profit from Q.3 (II). V3=Shortest path length from s to t of Q.3 (III). V4=No of bits (as per Huffman technique) of Q.3 (IV). Take integer part only i.e. if 123.95 than 123.

(I) Multiply A and B matrix using Strassen's Matrix multiplication algorithm $A = \begin{bmatrix} V1 & 6 \\ 7 & 8 \end{bmatrix}$ $B = \begin{bmatrix} 9 & V1 \\ 4 & 5 \end{bmatrix}$

(II) Apply the All pair shortest path algorithm using dynamic programming on following city network



(III) Solve the following Fractional knapsack using greedy technique for knapsack capacity W = V4

Value	15	16	6	3	16	19	14	7	18	5
Weight	13	5	2	2	14	13	19	8	12	9

(IV) Apply Heap sort using max heap on following{V3,7,4,12,3,90,5,2,67,14,1,45,13}

(V) Find the big theta complexity of following using suitable method $a_n = 2a_{n-1} - a_{n-2} + 2^n$ with $a_0 = 1$ and $a_1 = 2$

Best Wishes