Prof. Arvind Kumar

JAYPEE UNIVERSITY OF INFORMATRION TECHNOLOGY, WAKNAGHAT

END SEMESTER EXAMINATION-2015

B.Tech, IV Semester

COURSE CODE: 10B11CI411 MAX. MARKS: 45

COURSE NAME: Fundamentals of Algorithms

COURSE CREDITS: 04 MAX. TIME: 3 HRS

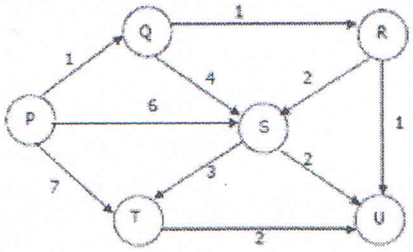
***Note:** All questions are compulsory. Carrying of mobile phone during examinations will be treated as case of unfair means.*

Section A Marks[5x2=10]

1. If the DFS finishing time f[u] < f[v] for two vertices u and v in a directed graph G, and u and v are in the same DFS tree in the DFS forest, Determine the category of edge u->v.
2. Define NP Hard and NP Complete classes.
3. Prove that f(ax+by) ≐ O(f(x) +f(y))
4. Let G be an undirected connected graph with distinct edge weight. Let *emax* be the edge with maximum weight and *emin* the edge with minimum weight. Then, No minimum spanning tree contains *emax*. True/False.
5. Why do you use dynamic programming?

Section B Marks[5x3=15]

6. Consider the strings A = "pqprqrpqrp". Extract the largest palindromic substring of A using dynamic programming.
7. Assume that you are given a chain of matrices < A1 A2 A3 A4 A5>, with dimensions 30 × 5, 5 × 40, 40 × 10, 10 x 15 and 15 ×50 respectively. Compute the optimal number of multiplications required to calculate the chain product and also indicate what the optimal order of multiplication should be using parentheses.
8. Run Bellman-Ford's shortest-path algorithm on the following edge weighted directed graph with vertex P as the source.



Section C Marks[5x4=20]

9. A Directed Acyclic Graph (DAG) is a graph without any cycles. The Longest Path problem on a graph is concerned with computing the length of the longest simple path between each pair of vertices in that graph. Given a DAG D, argue that the Longest Path problem does satisfy the optimal substructure property demanded by dynamic programming. Derive a recurrence relation for this problem

10. You need to search for a given number in an n X n matrix in which every row and every column are sorted in increasing order. Can you design a O(n) algorithm for this problem? Express the algorithm in pseudocode.

11. Give an example where the greedy algorithm fails to optimize but can be solved correctly using dynamic programming. Write both the algorithms with complexity.

12. Consider a person X has a car, which can travel *n* km when petrol tank is full. There are *k* no. of petrol stations between two cities, Mumbai and Delhi. Write two algorithms one greedy and other dynamic programming to determine how to stop at fewer no. of stations.