Design and Analysis of Algorithms

HW5

Total Marks = 26

Q1) You are given a weighted undirected graph G(V, E) and its MST T(V, E'). Now suppose an edge (a, b) ϵ E' has been deleted from the graph. You need to devise an algorithm to update the MST after deletion of (a, b).

Describe an algorithm for updating the MST of a graph when an edge (a, b) is deleted from the MST (and the underlying graph). It's time complexity must be better than running an MST algorithm from scratch. State and explain the time complexity of your algorithm. Analyze time complexity of your algorithm.

You can assume that all edge weights are distinct, that the graph has E edges and V vertices after the deletion, that the graph is still connected after the deletion of the edge, and that your graph and your MST are represented using adjacency lists. [6 Marks]

Q2) Given a directed graph where every edge has weight as either 1 or 2, give an algorithm to find the shortest path from a given source vertex 's' to a given destination vertex 't'. Expected time complexity is O(V+E). [5 Marks]

Q3) Let G = (V, E) be a connected undirected graph. Let w_{min} and w_{max} denote the minimum and maximum weights, respectively, of the edges in the graph. Do not assume that the edge weights in G are distinct or nonnegative. The following statements may or may not be correct. In each case, either prove the statement is correct or give a counter example if it is incorrect. [2+ 2+ 2+2+2 = 10 Marks]

(a) If the graph G has more than |V|−1 edges and there is a unique edge having the largest weight w_{max}, then this edge cannot be part of any minimum spanning tree.
(b) Any edge e with weight w_{min}, must be part of some MST.
(c) If G has a cycle and there is unique edge e which has the minimum weight on this cycle, then e must be part of every MST.
(d) If the edge e is not part of any MST of G, then it must be the maximum weight edge on some cycle in G.
(e) Suppose the edge weights are nonnegative. Then the shortest path between two vertices must be part of some MST.

Q4) Can you use the DFS algorithm to compute the number of distinct paths between two given nodes, s and t? Two paths are considered distinct if they differ by 1 or more edges. If you think the answer is yes, then provide the pseudo-code for a DFS based algo that computes this number in O(|V|+|E|). If you think the answer is no, draw a graphwith eight vertices in total,with two of its vertices labeleds and t, in which DFS will fail to compute the number of distinct pathsbetween s and t. [5 Marks]