COL 351

TUTORIAL SHEET 8

1. [**KT-Chapter6**] Suppose you are given a directed graph $G = (V, E)$ with length l_e on edges (which could be negative, but no negative cycles), and a sink vertex t. Assume you are also given finite values $d(v)$ for all the vertices $v \in V$. Someone claims that for each node $v \in V$, the quantity $d(v)$ is the cost of the minimum-cost path from node v to t. (i) Give a linear time algorithm which verifies whether the claim is correct, (ii) Assuming that all the distances $d(v)$ are correct, and that all $d(v)$ values are finite, you now need to compute distances to a different sink vertex t'. Give an $O(m \log n)$ time algorithm for computing these distances $d'(v)$ for all the vertices $v \in V$.

2. [**KT-Chapter6**] Suppose we are given a directed graph $G = (V, E)$, with costs on edges – the costs may be positive or negative, but every cycle in the graph has positive cost. We are also given two nodes v and w in the graph G. Give an efficient algorithm to compute the number of shortest $v - w$ paths in G (the algorithm should NOT list the paths; it should just output the number of such paths).

3. You are given a directed graph G where all edge lengths are positive except for one edge. Given a source vertex s, give $O(m \log n)$ time algorithm for finding a shortest path from a vertex s to a vertex t. Now assume there are a constant number of edges in G which have negative weights (rest have positive weights). Give an $O(m \log n)$ time algorithm to find a shortest path from s to t.

4. Describe an efficient algorithm to find the second minimum shortest path between vertices u and v in a weighted graph without negative weights. The second minimum weight path must differ from the shortest path by at least one edge and may have the same weight as the shortest path.

5. Given a directed acyclic graph that has maximal path length k, design an efficient algorithm that partitions the vertices into $k+1$ sets such that there is no path between any pair of vertices in a set.