Algorithms, Winter 2010-11, Homework 7
due Friday 2/11/11, 4:00pm

For Problems 2 and 3 include the pseudocode of your algorithm and a short verbal description. Briefly argue your algorithm's correctness and explain its running time.

Problem 1

Consider the following BFS-inspired algorithm for the single source shortest path problem. Let $G = (V, E, w)$ be a positively weighted undirected graph and let $s \in V$ be one of its vertices.

BFSshortestPath($G = (V, E, w)$, s)

For $v \in V$ do
    Let $found[v] = false$ and $dist[v] = \infty$.
Let $Q[0] = s$ and $found[s] = true$ and $dist[s] = 0$.
Let $beg = 0$ and $end = 1$.
While ($beg < end$) do
    Let $v = Q[beg]$.
    For every neighbor u of v do
        If not $found[u]$ then
            Let $found[u] = true$ and let $dist[u] = dist[v] + w(v, u)$.
            Let $Q[end] = u$ and increment end by 1.
        Else
            If $dist[u] > dist[v] + w(v, u)$ then let $dist[u] = dist[v] + w(v, u)$.
    Increment beg by 1.
Return $dist[]$.

Does the algorithm work? If yes, argue why it works. If not, find a counterexample and compare the correct answer with the output of the algorithm.

Problem 2

We say that an undirected graph $G = (V, E)$ is *positive-vertex-weighted* if we have a positive weight function w on G's vertices, i.e., $w : V \to \mathbf{R}^+$. Give an $O(n^2)$ algorithm for the single-source shortest path problem on positive-vertex-weighted undirected graphs.

Problem 3

Let $G = (V, E, w)$ be an edge-weighted directed graph with all weights positive, i.e., $w : E \to \mathbf{R}^+$. Give an $O(n^3)$ algorithm that finds the minimum distance from u to v for every pair of vertices $u, v \in V$. Moreover, output the number of distinct paths from u to v of length equal to the shortest distance.