Exercise Session 11

Exercise 1.
Consider a weighted directed graph $G = (V, E)$ with nonnegative weight function $w\colon E \to \mathbb{N}$. Solve the following computational problems assuming you can solve the single-source shortest-paths problem using e.g., Dijkstra's algorithm or the Bellman-Ford algorithm. Analyse the running time of your solutions.

Single-destination shortest-paths problem: Find a shortest path to a given destination vertex t from each vertex $v \in V$.

Single-pair shortest-path problem: Find a shortest path from u to v for given vertices u and v.

All-pairs shortest-paths problem: Find a shortest path from u to v for every pair of vertices u and v.

Exercise 2.
Consider a weighted tree $T = (V, E)$ having weight function $w\colon E \to \mathbb{R}$. The diameter of T is defined as $\max_{u,v \in V} \delta(u, v)$, that is, the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyse the running time of your algorithm.

Exercise 3.
(CLRS 24.5-4) Let $G = (V, E)$ be a weighted, directed graph with source vertex s, and let G be initialised by INITAILIZE-SINGLE-SOURCE(G, s). Prove that if a sequence of relaxation steps sets $s.\pi$ to a non-NIL value, then G contains a negative-weight cycle.

Exercise 4.
(CLRS 24.3-3) Consider the pseudocode for Dijkstra's algorithm.

DIJKSTRA(G, w, s)

1 INITIALIZE-SINGLE-SOURCE(G, s)
2 $S = \emptyset$
3 $Q = G.V$
4 **while** $Q \neq \emptyset$
5 $\quad u =$ EXTRACT-MIN(Q)
6 $\quad S = S \cup \{u\}$
7 $\quad$ **for each** $v \in G.Adj[u]$
8 $\qquad$ RELAX(u, v, w)

Suppose we change guard of the while loop in line 4 as $|Q| > 1$. This causes the while loop to execute $|V| - 1$ times instead of $|V|$ times. Is this proposed algorithm still correct? Motivate your answer.

★ **Exercise 5.**
(CLRS 24-3) Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 U.S. dollar buys 49 Indian rupees, 1 Indian rupee buys 2 Japanese yen, and 1 Japanese yen buys 0.0107 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $49 \cdot 2 \cdot 0,0107 = 1.0486$ U.S. dollars, thus turning a profit of 4.86 percent. Suppose that we are given n currencies $c_1, c_2, \ldots c_n$ and an $n \times n$ table R of exchange rates, such that one unit of currency c_i buys $R[i, j]$ units of currency c_j.

(a) Give an efficient algorithm to determine whether or not there exists a sequence of currencies $\langle c_{i_1}, c_{i_2}, \ldots, c_{i_k} \rangle$ such that $R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_k, i_1] > 1$. Analyse the running time of your algorithm.

(b) Give an efficient algorithm to print out such a sequence if one exists. Analyse the running time of your algorithm.