Problem 1 (20 points, Problem 23.1-9). Let $T$ be a minimum spanning tree of a graph $G = (V, E)$, and let $V'$ be a subset of $V$. Let $T'$ be the subgraph of $T$ induced by $V'$, and let $G'$ be the subgraph of $G$ induced by $V'$. Show that if $T'$ is connected, then $T'$ is a minimum spanning tree of $G'$.

Solution. We can prove it by contradiction. Assume $T''$ is a minimum spanning tree of $G'$ such that $wt(T'') < wt(T')$, then we can get a spanning tree of $G' \cup \{T - T'\}$ with weight smaller than that of $T$, a contradiction.

Problem 2 (30 points, Problem 24.1-5). Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \rightarrow R$. Given an $O(VE)$-time algorithm to find, for each vertex $v \in V$, the value $\delta^*(v) = \min_{u \in V} \{\delta(u, v)\}$. Let $w(s, v) = 0$ for all $v \in G$. Then run the BELLMAN-FORD($G', w, s$). The pseudocode is as follows:

Algorithm 1 BELLMAN-FORD($G', w, s$)

1: for each vertex $v \in G'$ do
2: let $\delta(v) = \infty$;
3: end for
4: let $\delta(s) = 0$;
5: for $i = 1$ to $|V'| - 1$ do
6: for each edge $(u, v) \in E'$ do
7: if $\delta(v) > \delta(u) + w(u, v)$ then
8: let $\delta(v) = \delta(u) + w(u, v)$;
9: end if
10: end for
11: end for
12: for each edge $(u, v) \in E'$ do
13: if $\delta(v) > \delta(u) + w(u, v)$ then
14: return FALSE
15: end if
16: end for

It’s obvious that $\delta^*(v) = \delta(v)$ because $w(s, u) = 0$ for each $u \in G$. The correctness is ensured by the correctness of BELLMAN-FORD. And the complexity is $O(VE)$.

Problem 3 (50 points, Problem 24-3 Arbitrage). 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 $49 \times 2 \times 0.0107 = 1.0486$ 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 \( < c_{i_1}, c_{i_2}, \ldots, c_{i_k} > \) such that \( R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1 \). Analyze the running time of your algorithm.

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

Solution. (a). If there is a sequence \( < c_{i_1}, c_{i_2}, \ldots, c_{i_k} > \) such that
\[
R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1,
\]
then we have
\[
- \sum_{j=1}^{k-1} \log R[i_j, i_{j+1}] - \log R[i_k, i_1] < 0.
\]

We create a directed complete graph \( G = (V, E) \), where each currency \( c_i \) corresponds to a vertex \( v_i \) of \( G \). If \( R[i, j] > 0 \), let the weight of this edge \( w(i, j) \) be \( - \log R[i, j] \); otherwise, let the \( w(i, j) = 0 \). Run Bellman-Ford algorithm starting from an arbitrary vertex on this graph. If there is a negative cycle, then there exists such a sequence. The time complexity is \( O(|V||E|) \).

(b). To find such a sequence. We first create a graph \( G = (V, E) \) as part (a). Then we relax all the edges \( |V| - 1 \) times, as in the Bellman-Ford algorithm. Then we record all of the \( d \) values of the vertices. Then we relax all the edges \( |V| \) more times. Then we check which vertices had their \( d \) values decrease since we record them. All of these vertices must lie on some set of negative weight cycles. Call \( S \) this set of vertices. Run DFS on the induced graph by \( S \) to find a cycle. The time complexity is \( O(|V||E|) \).