Assignment 6

Due: July 28th, 2024

(Ethics: Any behavior on any homework or exam that could be considered copying or cheating will result in an immediate zero on the assignment for all parties involved and will be reported to academichonesty@iit.edu. See the IIT Code of Academic Honesty, https://web.iit.edu/student-affairs/handbook/fine-print/code-academic-honesty)

Use pseudo-code in questions that need you to present your algorithms, but your pseudo-code can be "English-like" if the operation is obvious. Make sure that each line in your pseudo-code is numbered, and the indentation is correct.

1. A **bottleneck spanning tree** T of a graph G is a spanning tree of G whose largest edge weight is minimum over all spanning trees of G.
   a) Prove that a minimum spanning tree is also a bottleneck spanning tree. (Hint: you can prove this statement by contradiction.)
   b) Prove that a bottleneck spanning tree might not be a minimum spanning tree. (Hint: show a bottleneck spanning tree that is not a minimum spanning tree.)

2. Let $G = (V, E)$ be a weighted directed graph and we are trying to look for the shortest path from s to t in graph G. We are given that, $(u, v) \in E$ is the **only one** arc that has negative weight: $w(u, v) < 0$ and s, u, v, t are all different vertices; but we don't know whether (u, v) yields a negative-weighted cycle. Present an algorithm that uses the $\mathbf{Dijkstra}$ algorithm to find the shortest path from s to t.

   Hint 1: All edges in G other than (u, v) have non-negative weights; Dijkstra's algorithm can only be used on a graph with no negative-weighted edges.
   Hint 2: The shortest path from s to t either contains (u, v) or doesn't contain (u, v).

3. Consider a firm that trades shares in n different companies. For each pair $i \neq j$, they maintain a trade ratio r_{ij}, meaning that one share of company i trades for r_{ij} shares of company j. Here we allow the rate r to be fractional; for example, $r_{ij} = \frac{2}{3}$ means that you can trade 1 share of i to get $2/3$ shares of j.

   A trading cycle for a sequence of shares $i_1, i_2, \ldots, i_k$ consists of successively trading shares in company i_1 for shares in company i_2, then shares in company i_2 for shares i_3, and so on, finally trading shares in i_k back to shares in company i_1. After such a sequence of trades, one ends up with shares in the same company i_1 that one starts with. Trading around a cycle is usually a bad idea, as you tend to end up with fewer shares than you started with. But occasionally, for short periods of time, there are opportunities to increase shares. We will call such a cycle an opportunity cycle, if trading along the cycle increases the number of shares. This happens exactly if the product of the ratios along the cycle is above 1. In analyzing the state of the market, a firm engaged in trading would like to know if there are any opportunity cycles.

   Given an $n \times n$ sized chart R of trade ratios where $R(i, j) = r_{ij}$, present an efficient algorithm to find whether such an opportunity cycle exists.

   Hint 1: Construct the above problem into a graph problem; the $\mathbf{Bellman - Ford}$ algorithm should be helpful for the graph problem you constructed.
   Hint 2: Let $a_1, a_2, \ldots, a_k$ be k positive numbers, then:
$$a_1 \times a_2 \times \cdots \times a_k > 1 \Leftrightarrow (-\lg a_1) + (-\lg a_2) + \cdots + (-\lg a_k) < 0$$