Algorithm Design and Analysis
Assignment 2
Deadline: April 10, 2023

1. (25 points) Given a directed weighted graph $G = (V, E, w)$ where edges can be negatively weighted and a vertex s, consider the execution of Bellman-Ford algorithm. Recall that the algorithm starts by initializing $\text{dist}(s) = 0$ and $\text{dist}(u) = \infty$ for each $u \neq s$; and, at each iteration, the algorithm updates $\text{dist}(v) \leftarrow \min\{\text{dist}(v), \text{dist}(u) + w(u, v)\}$ for each edge (u, v).

   (a) (10 points) In the class, we have proved that G contains a negatively weighted cycle if $\text{dist}(u)$ is updated for some $u \in V$ at the $|V|$-th iteration. In this subquestion, you are to complete the correctness proof of Bellman-Ford algorithm by proving the converse of this statement: if G contains a negatively weighted cycle, then there exists $u \in V$ such that $\text{dist}(u)$ is updated at the $|V|$-th iteration.

   (b) (5 points) Give a counterexample to disprove the following claim: for a vertex t, if there is a path from s to t that contains a negatively weighted cycle, then $\text{dist}(t)$ must be updated at the $|V|$-th iteration.

   (c) (10 points) Adapt Bellman-Ford algorithm to solve the following problem. Give a vertex s and a vertex t, decide if there is an s-t path that contains a negatively weighted cycle. You need to prove the correctness of your algorithm. Your algorithm must run in time $O(|V| \cdot |E|)$.

2. (25 points) Given a directed graph $G = (V, E)$, a starting vertex $s \in V$ and a destination vertex $t \in V$. Design a polynomial time algorithm to decide if you can walk from s to t such that every vertex is visited. You are allowed to visit a vertex or an edge more than once. Prove the correctness of your algorithm and analyze its time complexity.

3. (25 points) Let $G = (V, E)$ be a graph and s be a vertex such that there is a path from s to each $u \in V$. We say G is a *good graph* if there exists a tree $T = (V, E')$ that share the same vertex set V with G such that T is both a depth-first search tree and a breadth-first search tree.

   (a) (10 points) If G is an undirected graph, prove that G is a good graph if and only if G is a tree.

   (b) (10 points) If G is a directed acyclic graph, prove or disprove that the vertex set V can be sorting in an array $\mathcal{L}$ such that $\mathcal{L}$ is both an ascending order of the distances from s and a topological order.

   (c) (5 points) Prove or disprove the converse of (b). That is, if G is a directed acyclic graph where the vertex set V can be sorting in an array $\mathcal{L}$ such that $\mathcal{L}$ is both an ascending order of the distances from s and a topological order, then G is a good graph.

4. (25 + 5 bonus points) Given an undirected simple graph $G = (V, E)$ and a vertex s, you need to find a cycle with the minimum length that contains s. A cycle cannot contain an edge more than once.

   (a) (25 points) If G is unweighted, design an algorithm for this problem that runs in time $O(|V| + |E|)$.

   (b) (5 bonus points) If G is edge-weighted such that the weights are positive, design an algorithm for this problem that runs in time $O(|V|^2)$.

   For both parts, you need to prove the correctness of your algorithms and analyze their time complexities.

5. How long does it take you to finish the assignment (including thinking and discussing)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Write down their names here.