Let $G = (V, E)$ be a graph with strongly connected components (SCCs) $C_1, \ldots, C_k$, and consider an execution of \textsc{Strongly-Connected-Components}(G). Recall from the lecture that for $U \subseteq V$,\footnote{As in CLRS, $u.d$ and $u.f$ refer to the discovery and finishing times, respectively, of $u$ during the first DFS in an execution of \textsc{Strongly-Connected-Components}(G).}

- $d(U) := \min_{u \in U} u.d$ is the time the first vertex in $U$ is discovered, and
- $f(U) := \max_{u \in U} u.f$ is time the last vertex in $U$ finishes.

Let $C_i$ and $C_j$ be two SCCs connected by an edge $(u, v)$ with $u \in C_i$ and $v \in C_j$. From the lecture you know that in such a case $f(C_i) > f(C_j)$. Define additionally $f'(U) := \min_{u \in U} u.f$, i.e., the time the first vertex in $U$ finishes.

(a) (4 points) Assume $d(C_i) < d(C_j)$. Prove or disprove that $f'(C_i) > f'(C_j)$.

\textbf{Solution: } INSERT YOUR SOLUTION HERE

(b) (4 points) Assume $d(C_i) > d(C_j)$. Prove or disprove that $f'(C_i) > f'(C_j)$.

\textbf{Solution: } INSERT YOUR SOLUTION HERE
Devise an efficient algorithm to count the total number of paths (of any positive length) in a directed acyclic graph $G$. Analyze your algorithm’s correctness and running time.

(Hint: Sort $G$ topologically first.)

Solution: INSERT YOUR SOLUTION HERE
Recall from the lecture that the minimum-spanning-tree (MST) problem is to find, in some undirected graph $G$, a spanning sub-tree $T$ with the smallest possible sum $\sum_{e \in T} w(e)$ of edge weights. Consider the related problem of finding a spanning sub-tree $T'$ with the least maximum edge weight $\max_{e \in T'} w(e)$.

(a) (4 points) Assume the edge weights are unique. Argue why Kruskal’s algorithm outputs a spanning tree $T$ that is also a (optimal) solution to the modified problem. You may use the fact that Kruskal is a correct algorithm.

Solution: INSERT YOUR SOLUTION HERE

(b) (2 (+2) points) Find a graph that contains a non-MST spanning tree $T$ with least maximum edge weight. That is $T$ is a (optimal) solution to the modified problem but not an MST. For extra credit, the edge weights must be unique.

Solution: INSERT YOUR SOLUTION HERE

(c) (6 points) Consider now graphs with edge weights that are not necessarily unique. Show that any (optimal) solution $T$ to the MST problem is also an (optimal) solution to the modified problem.

(Hint: Do not use a specific algorithm’s (such as Kruskal’s or Prim’s) correctness since they output some specific MST out of possibly many. Give a general argument instead.)

Solution: INSERT YOUR SOLUTION HERE
Solutions to Problem 4 of Homework 11 (9 Points)

Name: Name (NetID)  Due: Tuesday, December 6
Collaborators: NetID1, NetID2

(a) (5 points) Let $e$ be the maximum-weight edge of some cycle $C$ in a connected, undirected graph $G = (V, E)$. Prove that there exists an MST of $G$ that does not include $e$.

**Solution:** INSERT YOUR SOLUTION HERE

(b) (4 points) Find a connected, undirected graph $G = (V, E)$ with a unique heaviest edge $e$ such that some MST of $G$ contains $e$ and $\deg(v) \geq 2$ for all $v \in V$.

**Solution:** INSERT YOUR SOLUTION HERE
A directed graph $G = (V, E)$ is called semi-connected if for any two vertices $u, v \in V$, there is a path from $u$ to $v$ or a path from $v$ to $u$ (or both).

(a) (4 points) Let $G' = (V, E')$ be an undirected version of $G$ with $E' = \{(u, v) \mid (u, v) \in E \text{ or } (v, u) \in E\}$. Disprove the claim that $G$ is semi-connected if and only if $G'$ is connected.

**Solution:** INSERT YOUR SOLUTION HERE

(b) (12 points) Devise an algorithm to test if a graph is semi-connected. You may assume that you know how to construct the component graph of $G$ in time $O(V + E)$ (i.e., pretend you solved CLRS 22.5-5). Be sure to justify why your algorithm is correct.

**Hint:** Apply topological sort to the component graph $G^{\text{sc}} = (V^{\text{sc}}, E^{\text{sc}})$ and proceed from there.

**Solution:** INSERT YOUR SOLUTION HERE