Homework 10: Due April 16 (11:59 p.m.)

Instructions

- **Answer each question on a separate page.**
- **Honors questions are optional. They will not count towards your grade in the course. However you are encouraged to submit your solutions to these problems to receive feedback on your attempts. Our estimation of the difficulty level of these problems is expressed through an indicative number of stars** $('*' =$ **easiest**$)$ **to** $('*****' =$ **hardest**$)$**.**
- **You must enter the names of your collaborators or other sources as a response to Question** 0**. Do NOT leave this blank; if you worked on the homework entirely on your own, please write "None" here. Even though collaborations in groups of up to** 3 **people are encouraged, you are required to write your own solution.**

Question 0: List all your collaborators and sources: ($-\infty$ points if left blank)

Question 1: (10 points)

We present the following **false** claim:

Claim 1. *If a directed graph G contains cycles (i.e., is not acyclic), then topological sort*[1] *produces a vertex ordering that minimizes the number of "bad" edges that are inconsistent with the ordering produced. More precisely, a bad edge is one going from a vertex later in the ordering to an earlier vertex.*

Disprove this claim by providing a counterexample. Briefly justify why your graph fails the claim.

Question 2: (5+5+10=20 points)

Let V be a set of n vertices. We want to test whether a given directed graph $G = (V, E)$ is "weakly connected" in the following sense.

Property 1 (Weakly connected)**.** *A directed graph $G = (V, E)$ is **weakly connected** if for any distinct pairs of vertices $u, v \in V$, we have a path from u to v or v to u (or both).*

1. Suppose $G = (V, E)$ is a directed *acyclic* graph. Design an $O(|V|+|E|)$ time algorithm to determine whether or not G is weakly connected.
2. Suppose $G = (V, E)$ is any directed graph. Design a $O(|V| + |E|)$ time algorithm to determine whether or not G is weakly connected.
3. Prove that your algorithm in Part 2 is correct and runs in time $O(|V| + |E|)$.

(Hint: Recall that any directed graph G can be decomposed into a DAG of strongly connected components)

[1] The algorithm in which we run DFS and output vertices in decreasing order of finish time.

Question 3: (5+5=10 points)

1. How many valid topological sorts does the directed graph G in Figure 1 below have? List all the valid topological orderings of vertices in the following table. One of them has been listed as an example, where node A has the last finish time and D has the first. Note that DFS with a particular tie-breaking rule outputs *some* valid topological ordering, but not *all* of it.

1.	A	B	C	E	F	D
2.						

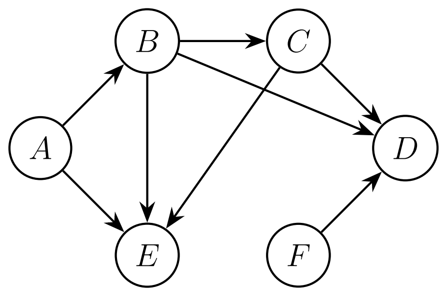

Figure 1: Directed G for topological sort.

2. Find a minimum spanning tree (MST) of the undirected graph G in Figure 2 below. Also, list the cost of your MST.

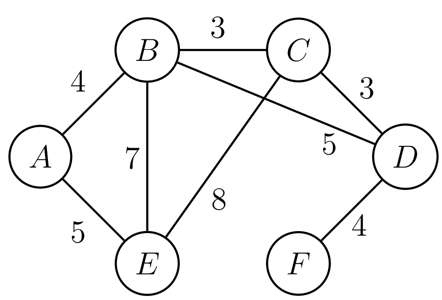

Figure 2: Undirected G for minimum spanning tree.

Question 4: (5+5=10 points)

Suppose we are given both an undirected graph G with weighted edges and a minimum spanning tree T of G. Further we are given an edge e of G and its new weight in the modified graph G'. In each of these cases, prove why T remains an MST of the modified graph G'.

1. The weight of one edge $e \in T$ is decreased.

2. The weight of one edge $e \notin T$ is increased.