Homework 11: Due December 4 (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: (5+5=10 points)

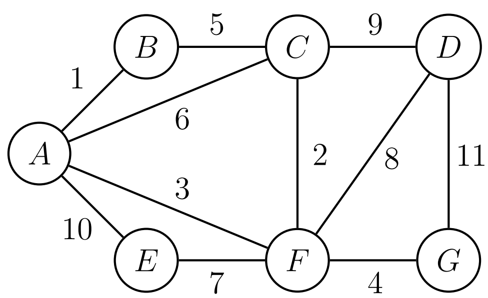

Figure 1: **Undirected graph G.**

Consider the undirected weighted graph $G = (V, E)$ in Figure 1 above.

1. Illustrate a run of Kruskal's algorithm on this graph. State at each step which edge is added to the tree. We have filled the first step in for you. Here, we only want the edge added to the tree and not every edge that is considered.

Step	1	2	3	4	5	6
Edge Added	AB					

2. Illustrate a run of Prim's algorithm on this graph starting from vertex A. State at each step which edge is added to the tree. We have filled the first step in for you. Here, we only want the edge added to the tree and not every edge that is considered.

Step	1	2	3	4	5	6
Edge Added	AB					

Question 2: (5+5=10 points)

Prove or disprove the following statements. If true, give a short explanation. If false, give a counterexample.

1. Let $G = (V, E)$ be a connected, undirected graph with a distinct cost $c(e)$ on every edge e. Suppose e^* is the cheapest edge in G; that is, $c(e^*) < c(e)$ for every edge $e \neq e^*$. Then there is a minimum spanning tree T of G that contains the edge e^*.

2. In the above setting, every MST of G contains the edge e^*.

Question 3: (10 points)

Let $G = (V, E)$ be a connected, undirected graph that contains a cycle $C \subseteq E$. Let $e \in C$ be a maximum-weight edge on this cycle. Prove that there is a minimum spanning tree of $G' = (V, E \setminus \{e\})$ that is also a minimum spanning tree of G. That is, there is a minimum spanning tree of G that does not include e.

Question 4: (10+5=15 points)

1. Assume that all edge weights of the given connected undirected graph $G = (V, E)$ are promised to be 1. Design the fastest algorithm you can to compute the minimum spanning tree (MST) of G. Argue the correctness of the algorithm and state its run-time. Your algorithm should be faster than the $O(|E| \log |V|)$ run-time of Prim and Kruskal.

2. Suppose instead that all edge weights are 1 *except* for a single edge $e_0 = (u_0, v_0)$ whose weight is w_0 (note, w_0 might be either larger or smaller than 1). Show how to modify your solution in part 1 to compute the MST of G. What is the running time of your algorithm and how does it compare to the run-time you obtained in part 1 (or to standard Prim)?

Question 5: (10 points)

Suppose you are given an algorithm P that solves the minimum spanning tree (MST) problem on undirected graphs with positive edge weights. Show how to use P to solve the general MST problem, in which edge weights are allowed to be negative or 0. In other words, *reduce* the general MST problem to the special case where the graph has only positive weights. Your algorithm for MST should take only $O(|V|+|E|)$ time, excluding the execution time of P. Prove the correctness of your reduction.

Honors Questions

(***) Suppose we are given both an undirected graph G with weighted edges and a minimum spanning tree T of G. Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge $e \notin T$ is decreased. Of course, we could just recompute the minimum spanning tree from scratch in $O(|E| \log |V|)$ time, but you can do better. Also justify the correctness of your algorithm.

(****) Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T. Show that for any other minimum spanning tree T' of G, the list L is also the sorted list of edge weights of T'.

Extra Practice Problems:

Question

Kruskal's algorithm can return different spanning trees for the same input graph G, depending on how it breaks ties when the edges are sorted into order. Show that for each minimum spanning tree T of G, there is a way to sort the edges of G in Kruskal's algorithm so that the algorithm returns T.

Question

Show that if all edge weights are positive, then any subset of edges that connects all vertices and has a minimum total weight must be a tree. Give a counterexample to this statement if we allow some weights to be non-positive.