B505/I500: Applied Algorithms

HW5 (Due: **Apr. 26 Friday 5pm**)

https://iu.instructure.com/courses/1771436

(This homework contains 40 bonus points)

1. (10 pts) Show how to find the maximum spanning tree of a graph, which is the spanning tree of the largest total weight.

2. (20 pts) Consider an undirected graph G = (V, E) with nonnegative edge weights w_e. Suppose that you have computed a minimum spanning tree of G, and that you have also computed shortest paths to all nodes from a particular node s. Now suppose each edge weight is increased by 1; the new weights are $w'_e = w_e + 1$.

(a) Does the minimum spanning tree change? Give an example where it changes or prove it cannot change.

(b) Do the shortest paths change? Give an example where they change or prove they cannot change.

3. (15 pts) Consider the following algorithm.

   Input: a weighted graph G=(V, E) and w.

   Maybe-MST(G, w)

   Sort the edges into non-increasing order of edge weights w;

   T $\leftarrow$ E

   for each edge e $\in$ E, taken in non-increasing order by weight

       if T-{e} is a connected graph

           T $\leftarrow$ T-{e}

   Output T

   Either prove that the output of the algorithm is a minimum-spanning tree or give a counterexample when the algorithm does not output a minimum-spanning tree.

4. (20 pts) The diameter of a tree T=(V, E) is defined as the largest length of all shortest path (i.e., the path with the smallest number of edges) between pairs of vertices. Devise a O(|V| + |E|) algorithm to compute a diameter for a given tree.

5. (20 pts) Let G be an arbitrary connected, undirected graph with a distinct positive weight on each edge. Let e=(u,v) and e'= (x, y) are two arbitrary edges connecting four distinct vertices u, v, x and y. Devise a polynomial time algorithm to find a cycle in G with the lowest total weight that contains both e and e'. What is the running time of your algorithm.

6. (20 pts) Decide whether you think each of the following statements are true or false. Justify your answers.
   a. Let G be an arbitrary flow network with a source s, a sink t and a positive

integer capacity c(e) on each edge. If f is a maximum s-t flow in G, then f saturates every edge out of s with flow, i.e., for all edges e out of s, we have f(e)=c(e).

b. Let G be an arbitrary flow network with a source s, a sink t and a positive integer capacity c(e) on each edge. Let (A, B) be a minimum s-t cut w.r.t. these capacities. Now suppose we add 1 to every capacity, then (A,B) is still a minimum s-t cut w.r.t. these new capacities, i.e., c'(e)=c(e)+1 for every edge e.

7. (15 pts) Consider the following scenario. Due to large-scale flooding in a region, parametics have identified a set of *n* injured people distributed across the region who need to be rushed to hospitals. There are *k* hospitals in the region and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).

   At the same time, one does not want to overload any one of the hospitals by sending too many patients its way. The parametics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospital is *balanced*: each hospital receives at most *n/k* people. Give a polynomial-time algorithm that takes the given information about the people's and the hospitals' location, and determines whether this is possible.

8. (20 pts) Here's a problem that occurs in automatic program analysis. For a set of variables $x_1, \ldots, x_n$, you are given some equality constraints, of the form "$x_i=x_j$" and some disequality constraints, of the form "$x_i \neq x_j$". Is it possible to satisfy all of them? For instance, the constraints $x_1=x_2$, $x_2=x_3$, $x_3=x_4$, and $x_1 \neq x_4$ cannot be satisfied. Devise an efficient algorithm that takes m constraints over n variables and decides whether the constraints can be satisfied.