HW 8

Due on **Wednesday, October 26, 2022 at 9am CST**

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2022 Version: **1.0**

Submission instructions as in previous homeworks.

Extra guidelines for no extra charge. As a reminder, describing an algorithm requires not only describing the algorithm itself, but you also have to explain how and why your algorithm works. You also need to provide an analysis of the running time of your algorithm. Finally, you also need to explain the correctness of your algorithm, and provide a proof if asked to provide one.

15 (100 PTS.) The invasion of the Shire.

In an act of sheer folly, Narnia had invaded the Shire. The Narnian military is now deployed on the roads of the Shire. There are junctions were the roads meet (for simplicity, we assume that a road might have junctions only in its two ends). Since its the rainy season the Narnian army can not leave the roads, and even worst, the gates back to Narnia are closed. The hobbits (i.e., the people living in the Shire), are trying to figure out which roads to destroy in their fight against Narnia. A road is a ***bottleneck*** if destroying it disconnects the (initially connected) road network. The road network has m roads and n junctions.

15.A. (50 PTS.) Describe an algorithm, with running time $O(n + m)$, that computes all the bottleneck roads. (Hint: First think about an algorithm that decides for a single road if it is a bottleneck, and then extend it so that it makes this decision for all roads. **DFS** is your friend.)

15.B. (50 PTS.) The high command of the Shire military conjectures that destroying junctions might work better. Describe a linear time algorithm that will identify every vulnerable junctions. Formally, a junction is ***vulnerable*** if removing it disconnects the road network.

16 (100 PTS.) The package delivery problem.

16.A. (50 PTS.) Let G be a directed acyclic graph with a unique source s and unique sink t. A delivery truck starts at s and arrives at t, but it can travel on at most k edges. Every node v has a profit $p(v) > 0$. Describe an algorithm, as fast as possible, that computes the maximum profit path in G (using at most k edges).

16.B. (50 PTS.) Let G be a directed graph with n vertices, and m edges. Describe an algorithm, as fast as possible, that computes an edge $u \to v$, that its addition to G would create the largest (in number of vertices) strong connected component in the new graph (such an edge might not be unique – you need to output only one such edge).

Let k be the number of strong connected components in G. What is the running time of your algorithm as a function of n, m and k?

For full credit, your algorithm should run in linear time if $k = o(n^{1/4})$.