CSCI 3104: Algorithms
Homework 5

Due at **11:00am on Wednesday, October 14, 2015**. Submit your solutions electronically at moodle (name file as **LastName_FirstName_HW5.pdf**) or submit in paper before class. Also, submit your python source code electronically at moodle (name your file as **LastName_FirstName_HW5.py**). Make sure to include your name and student ID. Digital submission should also include the Honor Code Pledge (`http://honorcode.colorado.edu/about-honor-code`), and paper submission should include your signature indicating adherence to the Honor Code Pledge.

1. Professor F. Lake suggests the following algorithm for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become positive, then run Dijkstra's algorithm starting at node s, and return the shortest path found to node t. Is this a valid method? Either prove that it works correctly, or give a counterexample.

2. There is a network of roads $G = (V, E)$ connecting a set of cities V. Each road in E has an associated length l_e. There is a proposal to add one new road to this network, and there is a list E' of pairs of cities between which the new road can be built. Each such potential road $e' \in E'$ has an associated length. As a designer for the public works department you are asked to determine the road $e' \in E'$ whose addition to the existing network G would result in the maximum decrease in the driving distance between two fixed cities s and t in the network. Give an efficient algorithm for solving this problem and analyze its time complexity.

3. A chain of words is a list of words where the i-th word is the $(i-1)$st word with one extra character and some reordering of letters. For example, AN, TAN, RANT, TRAIN, RETINA, NASTIER is a chain of length 6. Write a python program that reads a wordlist filename via command line and finds the longest chain in the wordlist file. A sample wordlist file is provided at moodle via a separate link. Print out three example chains you can find in this file that have the maximum length (there will be a TON... just give three chains).

   HINT: In order to do this, first build a DAG. The DAG will consist of a node for each word (you might want to collapse words into a single node when it makes sense to), and an edge from word x to word y if y can follow x in a chain. Then run DFS from each source node in the DAG and keep track of the maximum depth you reach.