Instructions for hw3-programming

A. Jupyter notebook

You should complete this assignment in a Jupyter notebook using:

- Python 3.7 or 3.8
- NetworkX 2.4+

If you installed the Anaconda distribution as instructed on Canvas, you will have Python 3.8 and NetworkX 2.4+. Follow the instructions on Canvas to install Python and familiarize yourselves with the Jupyter notebook environment.

B. The Assignment (70 points)

First, download the Jupyter notebook that appears on Canvas under Programming assignments/hw1p, as well as the data file contiguous-usa.dat. Then, launch the Jupyter notebook, which contains clear instructions on what you need to fill in.

The purpose of the assignment is to use a max flow algorithm to solve flow with demands. **You are only allowed to call functions that solve the maximum flow problem** from NetworkX (e.g., maximum_flow).

IMPORTANT:

1. **Do NOT change any function names or delete any cells or add code outside the functions you are asked to fill in.**
2. **Please comment or delete all of the print functions when submitting any ipynb file. Otherwise your code will fail. Same requirement is applicable to Extra Credit.**

C. Submission

Once you have completed the assignment, you should save your notebook as **hw3psolution.ipynb** and upload it to Gradescope. Gradescope will immediately check that:

- your filename is correct
- you have not renamed or changed the parameters of the required functions

If these checks fail, you **MUST** fix your assignment and resubmit (see next section).

D. Grading

Please follow the instructions below carefully.

- **You should not change any code we give. EXCEPTION**: you **should** delete or comment the single print command in the original notebook before you submit your code (we only include the command so you can test your code)**. You may not import any new libraries.** You should be able to solve this assignment using basic Python and NetworkX functions.
- You **may only use functions that compute max flow** from NetworkX. For example, you may not use functions that compute min-cost flow.
- You should double check that your **file name is correct**.
- Your code should work for **any** input flow network with demands.
- After you submit your solution, you should see some basic test cases. If not, please double check your code and contact the instructor team after you carefully check your code.

You will receive 30 points if your divergence vector is correct for the particular input provided, and another 40 points if it succeeds in all our hidden test cases.

Please see below for a list of common mistakes and how they will be penalized. **Penalties due to the following errors are non-negotiable.**

1. Incorrect filename: -70 points
2. Renaming the required functions or changing their parameters: -70 points
3. Use of additional modules/libraries that are not built in to Python: -70 points
4. Use of additional modules/libraries that are built in to Python: -30 points
5. Modifying the global process (e.g., adding code outside the functions, **introducing/using global variables**): -40 points
6. Returning the flow on a modified graph (added/deleted nodes/edges): -30 points
7. Forgot to comment or delete print function: -30 points

The first three mistakes result in a deduction of all 70 points because they will cause our Gradescope autograder to fail. Fortunately, we have set it up so that if these mistakes are detected upon submission, Gradescope will immediately let you know. You are then responsible for fixing your assignment and resubmitting before the deadline.

Finally, we will check for similarity among submissions. As usual, you may brainstorm with a small number of your classmates but you should write up your code **entirely on your own** to avoid receiving a 0 in this assignment (and possibly further disciplinary actions).