Instructions for hw3-programming

A. IPython notebook

You should complete this assignment in an IPython Notebook, using Python 3.7 and NetworkX version 2.0+. (If you installed the Anaconda distribution, as we recommend on the course website, you will have NetworkX version 2.3 and Python 3.8; these will be accepted.)

B. The Assignment (70 points)

You should download the IPython notebook that appears on Canvas under Programming assignments/hw3p on your machine, as well as the data file contiguous-usa.dat. You should launch the IPython notebook. The assignment 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. Extra Credit (80 points)

After completing the basic assignment, you will have the opportunity to get an **extra credit** of 80 points if you complete the same assignment **but implement Ford-Fulkerson to compute max flow entirely from scratch**. In other words, you have to implement every subroutine of the Ford-Fulkerson algorithm, such as construct the residual graph, find an s-t path P and Augment(f,P), **on your own**. For example, you should write your own BFS or DFS routine to find a simple s-t path (you may reuse your DFS from your hw1p). If you call any functions from NetworkX to find augmenting paths or build the residual graph or compute max flow, you will not receive any extra credit.

D. Your submission

You will submit this assignment on Gradescope. Once you have completed the assignment, you should save your notebook as `hw3psolution.ipynb` and then upload it on Gradescope.

Extra credit submission: If you implement Ford-Fulkerson on your own to receive extra credit, you should also submit a second IPython notebook. (You must still submit your solution to the basic assignment.) Follow the instructions below to submit this second notebook:

- Name the second notebook myFF.ipynb.
- myFF.ipynb should have exactly the same cells as our original notebook but should compute max flow using your own implementation of Ford-Fulkerson.
- Follow the instructions for hw3psolution.ipynb above to submit myFF.ipynb as well. That is, submit both hw3psolution.ipynb and myFF.ipynb on Gradescope.

E. 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.
- For the basic assignment, you **may only use functions that compute max flow** from NetworkX. (For example, you may not use functions that compute min-cost flow, see also problem 3 in your homework). See Section C regarding which functions are allowed in the extra credit solution.
- 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: -30 points
2. Use of additional packages/libraries: -30 points
3. Renaming of our functions: -40 points
4. Modifying the global process (e.g., adding code outside the functions, **introducing/using global variables**): -40 points
5. Returning the flow on a modified graph (added/deleted nodes/edges): -30 points
6. Forgot to comment or delete print function: -30 points

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).