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 recommended for the first programming assignment, you will have NetworkX version 2.3.)

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. **So you are only allowed to call functions that solve the maximum flow problem** from NetworkX (e.g., maximum_flow).

IMPORTANT: Do NOT change any function names or delete any cells or add code outside the functions you are asked to fill in.

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

Once you have completed the assignment, you should save your notebook as hw3psolution.ipynb and then do the following.

1. Open a Terminal in Mac OSX or Linux and run the following commands in order. Replace <myUNI> below with **your UNI**.
    - cd
    - ls // *should list* hw3psolution.ipynb *as one of the files in the directory*
    - mkdir <myUNI_hw3>
    - cp hw3psolution.ipynb <myUNI_hw3>/
    - tar -czvf <myUNI_hw3.tgz> <myUNI_hw3>/
2. You should now have the file <myUNI_hw3>.tgz in your home directory. Upload this file on Canvas (under Assignments, hw3p).
    - For example, if your UNI is ab1234, you should upload the file ab1234_hw3.tgz after running the following commands:
        - cd
        - ls //*should list* hw3psolution.ipynb *as one of the files in the directory*
        - mkdir ab1234_hw3
        - cp hw3psolution.ipynb ab1234_hw3/
        - tar -czvf ab1234_hw3.tgz ab1234_hw3/

Windows users: If you create a tar.gz file (e.g., using 7zip), simply rename it to tgz and submit the resulting file. Alternatively, the commands above should work essentially as they are in a windows command prompt (just replace **cp** by **copy** and **don't use slashes after directory name)** or you can use **Windows PowerShell.**

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, <myUNI_hw3> should now contain two files, hw3psolution.ipynb and myFF.ipynb. You should submit one tgz file.

D. Grading

Please follow the instructions below carefully.

- **You should not change any code we give. 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.

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

Note that the largest score for this assignment will appear as 150 on courseworks. This is just for technical reasons (we need to register your total score). The max score for this assignment is 70 points; excess points will be used as extra credit.

Finally, we will use software to detect 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).