Networks Assignment

ACADEMIC HONESTY

As usual, the standard honor code and academic honesty policy applies. We will be using automated **plagiarism detection** software to ensure that only original work is given credit. Submissions isomorphic to (1) those that exist anywhere online, (2) those submitted by your classmates, or (3) those submitted by students in prior semesters, will be detected and considered plagiarism.

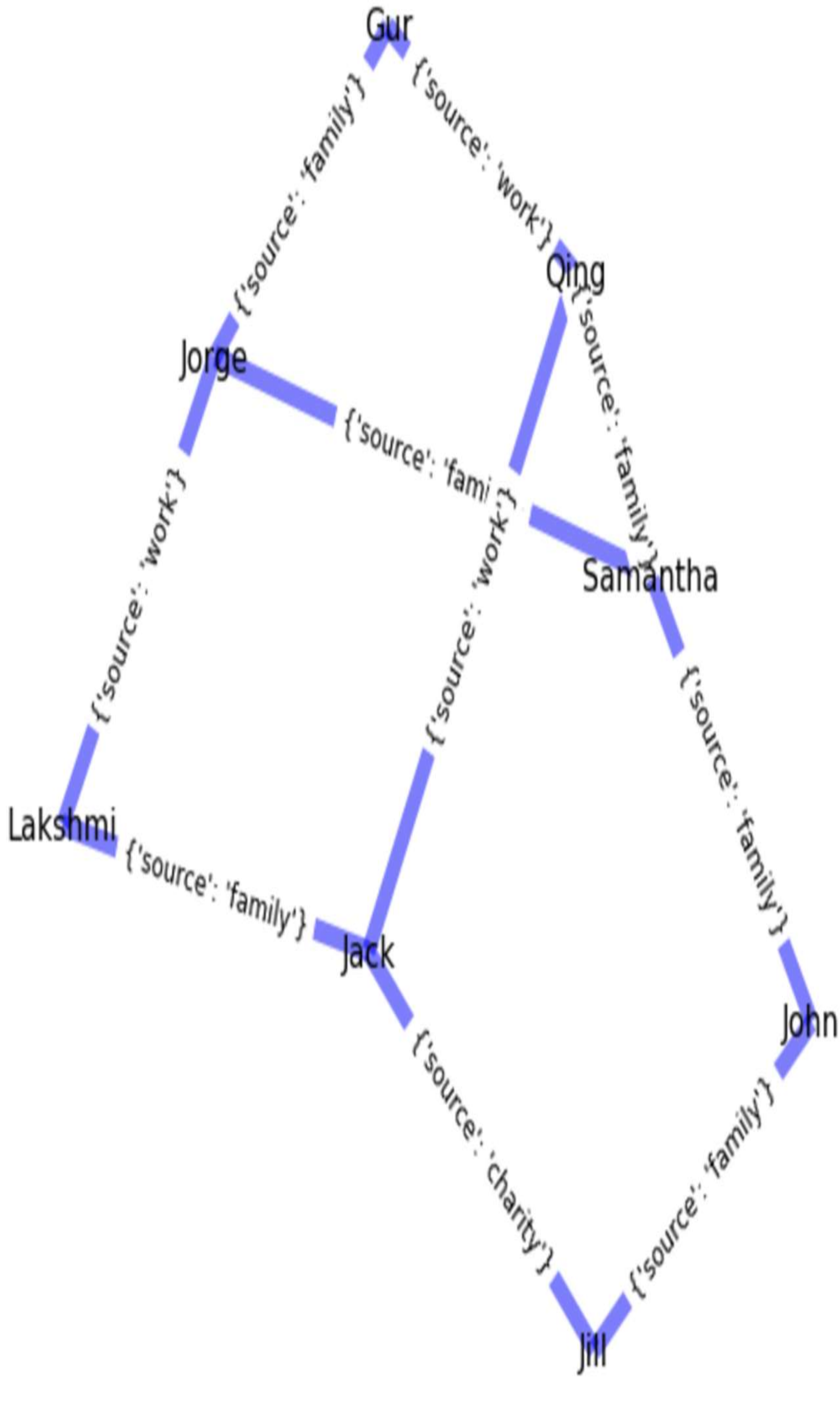
INSTRUCTIONS

This assignment tests whether you have a basic level of competency in Python. The assignment has one required part.

The question can be answered using the material we've covered in class and with the help of the functions listed in the Notes to the question.

I. Networks

The nodes in a network are the names of people and the edges codify the relationship (’family’,’charity’,’work’,’neighbor’) between two persons. An example of this graph is below


Gur
{'source': 'family'}
{'source': 'work'}
Qing
Jorge
{'source': 'family'}
{'source': 'fami
{'source': 'work'}
{'source': 'work'}
Samantha
{'source': 'family'}
Lakshmi
{'source': 'family'}
Jack
John
{'source': 'charity'}
{'source': 'family'}
Jill

Write a function - **get_connections()** - that takes a network, a person, and a relationship as arguments and returns the list of nodes that are connected to the person by the relationship.

For instance, the example graph:

get_connections(G,John,’family’)

Should return **[John, Jill, Samantha, Qing, Jorge, Gur]**.

Note that all the nodes in this returned list have a path of ‘family’ edges from John. Also, note that Jack and Lakshmi are not in the returned list because there is no complete ‘family’ path from John to either Jack or Lakshmi.

NOTE:

The function **G.edges(nbunch=nodelist,data=True)** returns the edges (in the network G) that are directly connected to the nodes in nodelist. For example **G.edges(nbunch=[john,gur],data=True)** will return

```
[(John, Samantha, {'source': 'family'}),
 (John, Jill, {'source': 'family'}),
 (Gur, Jorge, {'source': 'family'}),
 (Gur, Qing, {'source': 'work'})]
```

You can use the output of this function to pick only those nodes that have a ‘family’ edge. Then call G.edges again with this new set of nodes iteratively or recursively and, when you’re done, you’ll have a list of all nodes that have a ‘family’ path from the initial node.

What To Submit. Edit the starter code provided to solve this problem.

USE OF VOCAREUM

This assignment uses Vocareum for submission and grading. Vocareum comes equipped with an editing environment that you may use to do your development work. You are **NOT** required to use the editor. In particular, you are free to choose your favorite editor / IDE to do your development work on. When you are done with your work, you can simply upload your files onto Vocareum for submission and grading.

However, your assignments will be graded on the platform, so you **MUST** make sure that your code executes without error on the platform. To do so, you may use the **RUN** button to make sure that your code runs properly. In particular, we do not recommend using any third-party libraries and packages beyond **numpy**, **matplotlib**, and **scikit-learn**. We do not guarantee that they will work on the platform, even if they work on your personal computer. For the purposes of this project, the standard Python library and the aforementioned packages should be more than sufficient.

Python Assignment #4 Networks (External resource)

(100.0 points possible)

Your email address will be used to identify your submission entry.

Launch Project

Learn About Verified Certificates