Lab 6 - Scientific Computation

All of the steps below should be recorded in a lab report in your lab repository

- Graphs and Networks are ubiquitous in Scientific Computations. Networkx is an open source python package located here https://networkx.github.io/
- A nice tutorial may be found in https://networkx.github.io/documentation/stable/tutorial.html
- Your task for this class is to use networkx to do graph theoretic/network computations.

Please start by doing the following:

- Download and Install networkx and matplotlib (See http://matplotlib.org/)

```
sudo apt-get install python-networkx python-matplotlib
```

or

```
pip install networkx matplotlib
```

or

```
pip3 install networkx matplotlib
```

or

```
conda install networkx matplotlib
```

- Investigate networkx using examples in https://networkx.github.io/documentation/stable/tutorial.html, https://networkx.github.io/documentation/stable/auto_examples/index.html and https://networkx.github.io/documentation/stable/reference/index.html (use this last site as a reference as you continue). You do not need to show anything in your lab report from this stage, but spending 10 or 15 minutes reviewing these resources will make your life easier.
- Run the example from https://networkx.org/documentation/stable/auto_examples/drawing/plot_degree.html#sphx-glr-auto-examples-drawing-plot-degree-py. If you get the image shown on that page, then your installation is working correctly.
- Stanford Graphbase is a book written by Prof. Donald Knuth and contains many interesting examples on graph algorithms and implementations of programs (written in literate programming style). Here is an abstract of that book. Dr. Goldschmidt Master's thesis is a Java implementation. These problems are also implemented in networkx.
- Download https://github.com/rcos/networkx/blob/master/examples/graph/plot_words.py. This is a python implementation of the word ladder game courtesy of networkx. See also http://wordplay.blogs.nytimes.com/2013/06/19/climb-the-ladder/. Our ***fork*** is just a convenience so that we

can rely on stable locations for the duration of the course. The algorithmic idea behind this problem is the shortest path algorithm implemented in networkx. Look through the code and understand it. **This will be the basis for the rest of the lab. You are not required to start from scratch for any of the following exercises.**

- Download words lists for words of length five (or direct link) and words of length four

Once you have installed the tools and familiarized yourself with the networkx code, you are ready to begin recording your lab in your lab report. We will start with the problem of words of length 5 and words of length 4.

- The word game implementation you downloaded already takes words of length 5. Modify the base code to test the following words:

    1. `chaos` to `order`
    2. `plots` to `graph`
    3. `moron` to `smart`
    4. `flies` to `swims`
    5. `mango` to `peach`
    6. `pound` to `marks`

    Include the code and a copy of your results in your lab report.

- Now generate a new version that takes words of length 4 and test with:

    4. `cold` to `warm`
    5. `love` to `hate`
    6. `good` to `evil`
    7. `pear` to `beef`
    8. `make` to `take`

    Include the code and a copy of your results in your lab report.

- Next implement a variation where we consider two words (nodes) to be adjacent if there is a one letter difference without regard to ordering. You will need to change the edit_distance_one function to disregard letter position. Test with:

    1. `chaos` to `order`
    2. `plots` to `graph`
    3. `moron` to `smart`
    4. `flies` to `swims`
    5. `mango` to `peach`
    6. `pound` to `marks`

There are several ways to attack this. One way is to use multisets (Counter) from the collections module, another is to use permutations from itertools. Of the two, I ***highly*** recommend itertools. The multiset implementation is far more difficult to get correct.

An example:

```
Shortest path between chaos and order is
  chaos
  echos
  chore
  coder
  order
```

Your path may be different.

Include the code and a copy of your results in your lab report.

- ***(Optional)*** An interesting variation on word ladder is suggested in http://rexwordpuzzle.blogspot.com/2017/02/actress-form-mixed-martial-arts.html - Your task is to find words that precede SLID, DOTE, HERD and OMEN and the words that follow (immediately) NINE, SELL, STAT and WHAT. To do this, create a function so that given the word SLID you print out all of the words that are 1 letter away from it (ie SLIT, SAID etc.). Do this for all all 8 of these words.

  Include the code and a copy of your results in your lab report.

Document your program in a lab 6 writeup and then submit the report via Submitty by inputting your repository information into the gradeable. Your lab report should contain:

```
- Your results for the 6 five letter pairs
- Your code for the four letter solution
- Your results for the 5 four letter pairs
- Your code for the unordered solution
- Your results for the 6 five letter pairs using the unordered implementation
- (OPTIONAL) your code for precede/follow
- (OPTIONAL) precede/follow results
```

If you get done early, please work on your project. And don't forget to blog!