COMP 200 & COMP 130
Assignment 5: Recursion, Graph Searching

Be sure to read the course policies, as posted on the course web site:

http://www.clear.rice.edu/comp200/policies.shtml

Work in assigned pairs on this assignment. Each pair should put all assignment answers in one Python file called *netid1_netid2_*`5.py`, substituting in the students' NetIDs. Put the answers to non-programming problems in comments. When submitting, only one of the member should turn in the pair's answers.

Total points: 100 for COMP 200, 150 for COMP 130.

Wherever reasonable, use functions that have been defined previously on this assignment, on a previous assignment, or in class.

Recursion – COMP 200 & COMP 130 (30 points)

1. (30 points)

   In class, we presented two functions to sum a list of numbers. One used a loop, the second used recursion.

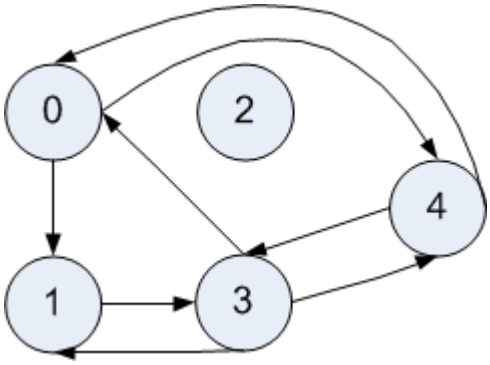
   For this problem, our data will be in a list of lists of lists of numbers. For example, `[[[4, 2], [3]], [], [[7, 1, 9], [], [8, 3]]]`.

   Define two functions that each sums a list of lists of lists of numbers. One should use looping, and the other should use recursion. Do not use the built-in function `sum`. Define and use helper functions as needed to decompose the problem.

Graph Searching – COMP 200 & COMP 130 (70 points)

We will use the following graph as an example in the following problems. This is represented in Python as

```
directedGraph = {0: [1,4], 1: [3], 2: [], 3: [0,1,4], 4: [0,3]}
```



2. (20 points)

   In class, we presented pseudo-code for breadth-first search. Define a Python function `BFS` that implements this algorithm.

3. (40 points total)

   The versions of DFS and BFS described in class search for the desired node. As originally described, we typically want not just this result, but the path from the starting node to the found node. This path describes how the two nodes are related. (As we have previously mentioned, there could be multiple ways that the nodes are related. This path is the one we discovered first.)

   We will represent such a path as a list of nodes, where the first is the starting node, and the last is the found node. A path can also be `None`, representing that there is no such path.

   (a) (20 points)

   Define a function `pathDFS(graph,source,target)` that returns the discovered path to the target node. This should be a modified version of `DFS` presented in class.

   For example, `pathDFS(directedGraph,0,4)` could return `[0,4]` or `[0,1,3,4]`, depending on the order in which we traverse each node's edges. Meanwhile, `pathDFS(directedGraph,3,3)` would return `[3]`, since the target node is the same as the source node, and `pathDFS(directedGraph,0,2)` would return `None`, since there is no such path.

   (b) (20 points)

   Define a function `pathBFS(graph,source,target)` that returns the discovered path to the target node. This should be a modified version of `BFS` from the previous problem.

   Since BFS is guaranteed to find a shortest path, `pathBFS(directedGraph,0,4)` would return `[0,4]`, while `pathBFS(directedGraph,0,3)` could return `[0,1,3]` or `[0,4,3]`. Meanwhile, `pathBFS(directedGraph,0,2)` would return `None`, since there is no such path.

4. (10 points)

   If you have a graph of n nodes and e edges, what is the maximum length of a shortest path between two nodes? Briefly explain your answer.

   For example, in the above graph, `[3,0]` is a shortest path between 3 and 0, while `[4,3,1]` is a shortest path between 4 and 1. The latter is among several shortest paths of length 2, and there are no longer shortest paths in this graph. The problem asks the more general question of what is the longest possible shortest path in any graph of the given size?

   Hint: Don't try to generate every graph of the given size to figure this out. There is a simple answer with a simple explanation.

Graph Data Mining – COMP 130 ONLY (50 points total)

Outside of school, you will be asked to solve problems, rather than demonstrating knowledge of some algorithm or technique. *How* you solve them is often not as important as your ability to convince someone else of your technique's appropriateness and viability. That is, the computational strategy is generally less important than getting useful results. But, you need to argue that your chosen approach is a good solution on grounds of some sort of metrics such as accuracy, speed, ease of implementation, maintainability, and extensibility.

This problem, or more aptly, mini-project, is similar. The directions are deliberately vague and without specific direction on how you should approach the problem and what code you should write. The point of this exercise is for you and your partner to come up with a plan on how to analyze the problem and the code to produce results consistent with that analysis plan.

Your job in these exercises is to convince the staff that you have devised a valid analysis technique for the data and to show code and results based on your described analysis technique.

There are no specific "right" or "wrong" answers for these exercises!

Do *not* rely only on the class lecture notes for inspiration and guidance. You should actively search books and the Internet, use class and other resources, consult on-line code and library documentations, and seek advice from other students, TAs, and professors. **As with any proper scientific endeavor, *any* ideas that you receive from *any* source should be properly acknowledged and cited in your discussions.** Along with writing your own code, you are free to use any pre-existing library functions or classes you wish, so long as those libraries are part of the Enthought Python Distribution the class is using.

Suggestion: Run your analysis ideas past a COMP 130/200 staff member *before* committing large amounts of time to it!

Strong suggestion: *Start Early!* Expect to have to adjust your techniques and code many times during these exercises, perhaps even completely ditching your original ideas and starting over. Science is full of far, far more failures than successes. However, always remember that every failure is really part of the path to understanding, and in such, is never a waste of time. A scientist learns nothing from instant success.

The data used for these exercies is real Facebook data for Rice undergraduate students, which has been "anonymized" to protect identities. We have been granted special permission use this data, so please adhere to our license agreement for the data and do not use the data for anything except this course, and do not distribute the data to anyone outside of COMP 130.

The data and some utility functions for these exercises can be found in the Resources section of the COMP 130 OWL-Space site. Note that a small (10%) subset of the full data is also being supplied to make it easier and faster to test your analysis. This subset only contains students from 3 colleges, not the 9 in the full dataset. (This data is a few years old). Be sure to let the staff know if you encounter problems or you need assistance creating specialized subsets of the data.

5. (25 points total) The Most Inter-Connected College

   The raw data, after being read in from the disk, is in the form of an undirected `networkx Graph` of `Student` objects connected by "friend" relationships. See the supplied utility code for documentation on the `Student` class.

   While there is certainly much to be learned just from the "friends" graph, there is actually another graph buried within the friends graph. This hidden graph is a weighted graph where the nodes are the colleges to which the students belong and the edges represent friend connections between people in one college with people in another. That is, we are looking at the "inter-connections" or connections *between* colleges. The edge weights are the number of friend connections between any two colleges. *Be careful about double-counting the number of friend connections between any two colleges!*

   (a) (10 points) Analysis Plan Description:

   Clearly and completely describe how you define "most inter-connected college" and how you would process the friends graph data to determine that college.

   Careful! This may not be as simple as you think – be sure to consider all angles, situations, pros/cons and possibilities! Likewise, in words, not code, describe how you intend to analyze the friends graph data to arrive at your "most connected college" determination. Address whether and to what extent your proposed techniques are applicable to analyzing other aspects of the data.

   You will be graded on the strength of your arguments, and the completeness and clarity of your descriptions and discussions. Two points will be reserved for "creativity" in your solution.

   Write your discussions as a comment in your submitted code.

   (b) (10 points) Analysis Code:

   Write functions to support your analysis plan. When your code is run, the analysis should be performed with no operator intervention. As always, all functions should be fully documented.

We will grade your code not only on how well it executes your plan of analysis, but also how well-written your code is. You should observe the practices we have covered, including decomposing your code, having no hard-coded values, and reusing code where appropriate.

One question you should be sure to address is "How do you know that your code actually produces the results you desire?" That is, do you have proof of the operational veracity of your code? How will you deal with the fact that the Facebook dataset is too large for you to simply look at it and glean the correct answers to algorithms?

(c) (5 points) Analysis Results:

As previously mentioned, your Python file should automatically perform your proposed analysis with no user intervention. Print labels to your data to help the reader understand each processing step and to emphasize your results and conclusions. Create plots and data printouts of raw and processed data to both illustrate your analysis process and support your conclusions.

6. (25 points total) The Most Intra-Connected College

In the same vein as the previous exercise, consider the notion of the most "intra-connected" college, that is the college that is most connected *internally*.

(a) (10 points) Analysis Plan Description:

Clearly and completely describe how you define "most intra-connected college" and how you would process the friends graph data to determine that college. Follow the same directions as the corresponding section in the previous exercise.

(b) (10 points) Analysis Code:

Write functions to support your analysis plan. Follow the same directions as the corresponding section in the previous exercise.

(c) (5 points) Analysis Results:

Your Python file should have code to automatically perform your proposed analysis with no user intervention. Print labels to your data to help the reader understand each processing step and to emphasize your results and conclusions.

General Suggestions:

START EARLY!!

Do not underestimate the amount of time that these exercises could take! There are many pitfalls and difficulties that simply cannot be predicted ahead of time.

Have *LOTS* of discussions with your partner.

There are lots of ideas that need to be hammered out. Don't rely on a single person's opinion.

Talk to the course staff early and often.

The staff's advice will help you navigate the sea of possibilities, keep you from going down blind alleys, help you devise algorithms and generally save you tons of time.

Use a good functional breakdown of the problem to decouple the parts of your analysis process.

Separating the pieces of your analysis process will enable you to separately test each small piece, making it much easier to debug and have confidence in your total solution. Do not attempt to debug your total solution without having tested its components first!

Write down a list of tasks that need to be accomplished.

This is a technique to help you get a hold on the scope of what you are trying to accomplish and aid you in allocating your time and human resources. This also goes hand in hand with your functional breakdown.

Don't pound your head against the wall alone in your room!

These exercises are designed to be accomplished as a partner project in close cooperation with other people. If you're not sure, *ASK!!* Isolation is a sure recipe for disaster here.

Feedback – COMP 200 & COMP 130 (0 points)

1. Roughly how many hours did you spend on the two sets of finger exercises?
2. On a scale of 1 (very easy) to 5 (very difficult), how difficult were the finger exercises?
3. Roughly how many hours did you spend on this homework?
4. On a scale of 1 (very easy) to 5 (very difficult), how difficult was this homework?
5. Which material did you find most challenging?
6. Did you feel that the class material adequately prepared you for the homework?