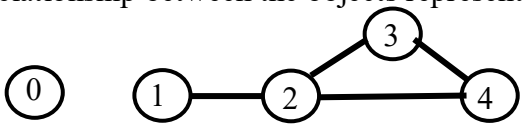
BUAI 492
Individual Homework #3

INSTRUCTIONS: For all programs, you are expected to use good programming style (including good variable names and some data validation). Before class on the due date listed in the syllabus, upload to *Canvas* ONE Jupyter Notebook file with your solutions to the following exercises. The solution to each exercise should be stored, in order, in a separate cell whose first line is a comment with the exercise number (for example, #EXERCISE 1) and is followed by your solution). For any exercise whose solution is not a computer program, write your answer as a comment. For any program that involves writing a function, validate the inputs in the function and also include a main program that calls the function and displays the results. Failing to follow these instructions will result in a loss of points. DO NOT COPY SOMEONE ELSE'S ANSWERS!

Exercise 1. A ***network*** consists of a collection of p circles, called **nodes** that represent objects in solving certain problems. The network also has q lines, called **arcs** that connect pairs of nodes to express a relationship between the objects represented by the connected nodes, for example:



Write functions to perform each of the following tasks.

a. Write a function called *GetData* to get the data for a network from an EXCEL file in which values of p and q are given in cells A1 and B1. Then get the "arc list" starting in row 2 of the file, in which each arc's two connected nodes are in given in column A and B (see the file *ArcList.xlsx* with the data of the network above). The arc list should be returned to main program as a list of lists with element 0 being the list [p, q]. (10 points)

b. Write a function called *Degrees* that returns a list of the "degrees" of all the nodes, where the degree of a node is the number of arcs coming out of that node (for the example above, the degree of Node 1 is 1 and the degree of Node 3 is 2). (10 points)

c. Write a function called *DeleteNode* that deletes a node from a network that is stored as an arc list by performing the following operations:
   1) Delete all of the k arcs coming out of the deleted node.
   2) Reduce by 1 the number of all nodes greater than the number of the node being deleted.
   3) Reduce the total number of nodes by 1 and the number of arcs by k.

   For example, if you want to delete Node 3 from the network above, then you must also remove the k = 2 arcs 2 – 3 and 4 – 3. Node 4 would then become Node 3 and the number of nodes and arcs in the new network would be 4 and 2, so the new arc list would be: [(4, 2), (1, 2), (2, 3)]. (15 points)

Exercise 2. Write the three functions in Exercise 1, but this time, the data for the network are stored in an "adjacency matrix", say A, in which $A_{ij} = 1$ if $i - j$ is an arc in the network, and 0 otherwise. For example, the (5 x 5) adjacency matrix for the network in Exercise 1 is given in the file *AdjacencyMatrix.xlsx* (an empty cell indicates a value of 0 for the adjacency matrix). (**Hint:** Use the numpy method "delete" to delete a row and column of A.) (35 points)

Exercise 3. Write a program to perform each of the following tasks using Numpy arrays.

a. Write a function called *GetAMatrix* that creates a (p x n) matrix A of random real numbers between –10 and 10 in the form of a numpy array, where $p < n$ are both obtained from the user. (5 points)
b. Write a function called *GetBasics* that, given the integers p and n, returns a list, say *basics*, of p different randomly-chosen values from 0, 1, …, $n - 1$ with the remaining values in 0, 1, …, $n - 1$ being stored in the list *nonbasics*. Do this by starting with *basics* = [ ] and *nonbaiscs* = [0, 1, …, $n - 1$]. Then sequentially use the "pop" method to move a random element of *nonbasics* to *basics* exactly p times. (5 points)
c. Given the lists *basics* and *nonbasics* from part (b), write a function called *GetBandN* that returns the submatrix B consisting of the ***columns*** of A indexed by the list *basics* together with the submatrix N consisting of the remaining ***columns*** of A. For example, if $n = 5$ and the basics are [2, 4] and the nonbasics are [1, 5, 3], then B consists of columns 2 and 4 of A while N consists of columns 1, 5, and 3 of A. (5 points)
d. Write a function called *inverse* that determines whether the B matrix has an inverse and, if so, returns that inverse matrix. The function should also return a Boolean variable indicating whether the B matrix has an inverse (True) or not (False). (**Hint:** Use the subpackage linalg of numpy.) (5 points.)

Exercise 4. Fill out the course evaluations at https://weatherhead.case.edu/courseevaluations/.