# Network Science
# Autumn 2020
# Project 2 (version 1.0)

Due: December 18th, 12:00pm GMT (noon)

---

There are three files for this assignment: 1) the one that you are reading which is the project description, 2) *project2.ipynb*, a Jupyter notebook which you will complete and submit on Blackboard (see below for details) and 3) *project2.dat*, a data file needed for part 2.

## Part 1

1. (3 pts) The notebook *project2.ipynb* contains a complete Python function named *Graph1* which analyzes a simple, undirected NetworkX graph provided as input (see the function documentation for further details on the input). Provide an explanation of what the code does including a careful description of the function output. Also provide a critical assessment of the code's efficiency. This assessment should include an estimate of the code's running time. Place your discussion where indicated in the markdown cell below the function.

2. (7 pts) Consider the source edge centrality for a simple, connected, undirected $N$-node graph defined as,

$$w_{\alpha\beta a} = \sum_{b=1}^{N} \gamma_{\beta b} m_{\alpha\beta}(a,b)/p_{ab},$$

   where $m_{\alpha\beta}(a,b)$ is the number of shortest paths between nodes $a$ and $b$ passing through edge $\alpha-\beta$, and $p_{ab}$ is the number of shortest paths between nodes $a$ and $b$ as defined in lecture. Additionally, $\gamma_{\beta b} = \gamma_0$ if $\beta = b$ and 1 otherwise. Here, $\gamma_0$ is a parameter that must be specified.

   (a) Complete the function *Graph2* so that it efficiently computes $w_{\alpha\beta a}$ for each edge in the simple, connected, undirected NetworkX graph $G$ provided as input. The source node, $a$, and $\gamma_0$ are also provided as input. The nodes in the graph are numbered from 1 to $N$, and the links are numbered from 1 to $L$ where $L$ is the number of links in the graph. The function should return an $L$-element list, Lf, where Lf[i] contains the source edge centrality for link $i+1$ (see the function documentation for further details on input/output). The function includes code that creates a Python dictionary, *Dlink*, that provides the link number given the nodes that the link connects. For example, if link 6 connects nodes 4 and 23, then Dlink[4][23]=D[23][4]=6. You are not required to analyze or modify the code given to create Dlink. Furthermore, operations like x=Dlink[4][23] are O(1), i.e. the cost does not depend on the size of Dlink.

(b) Provide a brief, clear description of your code, along with a careful and concise analysis of its running time. Place your discussion in the markdown cell below the function.

**Note:** For part 1, you may use numpy and the collections modules. Use of NetworkX is restricted: you may use methods for the graph, G, such as G.nodes(), but you should not use general Networkx functions such as nx.degree(). Please do not import/use other modules or functions without explicit permission.

# Part 2

In part 2, you will simulate and analyze the spread of an infectious disease in a real-world network. You have been provided with code that loads *project2.dat* and creates a Numpy array containing the edges for a simple undirected graph.

1. (4 pts) You will now add code to the function, *SInet*, so that it simulates the naive-network SI model with time-periodic transmission on a simple undirected N-node NetworkX graph provided as input:

$$d\langle x_i\rangle/dt = \beta\left[1+\delta\cos(\omega t)\right]\sum_{j=1}^{N} A_{ij}\langle s_i\rangle\langle x_j\rangle\,.$$

Here, $\beta$, $\delta$, and $\omega$ are parameters that are also provided as input, and $0 \le \delta \le 1$.

   (a) First complete the function *RHS* so that it efficiently computes the RHS of the model for each node with $\langle x_i\rangle$ and $t$ provided as input. Note that variables defined "above" *RHS* within *SInet* will be available within *RHS*. See the function documentation for details on its input and output.

   (b) Now, complete *SInet* so that it accurately and efficiently simulates the model stated above. The solution should be returned at $Nt+1$ equispaced times from $t=0$ to $t=T$ (with $Nt$ and $T$ provided as input). The initial condition should be an infection-free state except for node $i_0$ where $\langle x_{i_0}(t=0)\rangle >= x_0$. The parameters $i_0$ and $x_0$ are also provided as input. The function should return $\langle x_i(t)\rangle$ for each node and time specified above. See the function documentation for further information.

   (c) Briefly (in a few sentences) explain why your code should be considered to be accurate and efficient.

2. (3 pts) Investigate the spread of infectious disease on the given network using this model and your simulation code for cases where $\delta = 1$, and $\omega = 80\beta$. The initial condition should set $i_0$ to be the node with largest degree and $x_0 = 0.001$. You should focus on the initial spread of the disease, and it is up to you to design this focus in a sensible and meaningful manner. Add your analysis and accompanying code in the appropriate markdown cells in your notebook. You should include 1-3 well-made figures illustrating key trends, and your discussion should explain these trends and why they are important.

3. (3 pts) Keeping the model parameters the same as in 2.2 (though you may change the initial condition if you would like), does the community structure of the network have a substantive influence on the spread of the disease? Provide a concise explanation of your conclusion supported by 1-3 well-made figures. Include your code and discussion where indicated in your notebook

   **Note:** For part 2, you may use numpy, scipy, matplotlib, itertools, and networkx as needed. Please do not import/use other modules without explicit permission.

**Further guidance**

- You should submit both your completed Jupyter notebook and a pdf version of your notebook (generated using File — Download as). If you cannot generate a pdf, try installing latex first, or submit an html version instead. To submit your assignment, go to the module Blackboard page and click on"Project 2". There will be an option to attach your completed Jupyter notebook and pdf files to your submission. (these should be named *project1.ipynb* and *project1.pdf*). After attaching the notebook, submit your assignment, and include the message, "This is my own work unless indicated otherwise." to confirm that the submission represents your individual work.

- Marking will be based on the correctness of your work, the efficiency of your codes in parts 1.2 and 2.1, and the degree to which your submission reflects a good understanding of the material covered up to the release of this assignment. For open-ended questions, we are particularly interested in your ability to identify and explain important properties and trends, and exhaustive descriptions are not needed. While creative ideas based on class material is welcome, you are not expected to base your work on new ideas/concepts/methods that have not been covered (and it is unlikely that credit will be given for such work). Excluding figures and code, you should aim to keep the pdf version of your notebook to less than 1 page.

- Open-ended questions require sensible time-management on your part. Do not spend so much time on this assignment that it interferes substantially with your other modules. If you are concerned that your approach to the assignment may require an excessive amount of time, please get in touch with the instructor.

- Questions on the assignment should be asked in private settings. This can be a "private" question on Piazza (which is distinct from "anonymous"), using the "Chat" on Teams during a Q&A session, or by arrangement with your Problem class instructor.

- We will not closely examine the efficiency of your code for questions 2.2 and 2.3, but it should not be ludicrously slow (e.g. it should not take more than an hour to run the code in the Notebook on a non-ancient computer).

- Please regularly backup your work. For example, you could keep an updated copy of your notebook on OneDrive.

- In order to assign partial credit, we need to understand what your code is doing, so please add comments to the code to help us.

- I suggest initially developing your code in a Python module (outside of a function) and running it in a qtconsole (or similar terminal) so that you can readily access the values of the variables you are using.
- Feel free to use/modify codes that I have provided during the term.