Scientific Computation
Spring 2023
Project 2

Due: Friday November 24th, 1pm

In addition to this project description, there are 3 files for this assignment:

- project2.py: a Python file which you will complete and submit on Blackboard (see below for details)
- report2.tex: a Latex template file for the short report which you will submit. The discussion and figure(s) described below should be placed in this report.
- project2.npy: a datafile needed for part 2

## Part 1

You have been provided the functions, *searchGPT*, generated by chatGPT3.5, and *searchPKR*, generated by chatPKR1.0. Each function has the same input: a weighted NetworkX graph, and two integers corresponding to nodes in the graph. Each function also returns an integer, $d$; *searchGPT* also returns a list. For the same input, the two functions should return the same values for $d$.

1. (2 points) Carefully analyze the functions, and in your report, describe the problem that they are attempting to solve and the strategy that *searchGPT* is using to solve this problem.
2. (6 pts) Unfortunately, chatPKR premium is needed to obtain the code for constructing the list returned by *searchGPT*, so you will have to add this code yourself. Complete *searchPKR2* so that it efficiently constructs a list which is either the same as the list returned by *searchGPT* (for the same input) or is an equally valid solution to the problem being solved. In your report, carefully explain what tangible differences, if any, there are between *searchGPT* and *searchPKR2* with respect to: (a) correctness and (b) computational cost.

   **Note:** You should not use any NetworkX functions, though you may use methods and attributes of the input graph as is already done in the provided code. You should not use built-in functions where the primary purpose is to improve the wall-time of the code; the focus here should be on the time complexity.

## Part 2

Consider the following system of $n$ nonlinear ODEs:

$$\frac{dy_0}{dt} = \alpha y_0 - y_0^3 + \beta\left(y_{n-1} + y_1\right) \tag{1}$$

$$\frac{dy_i}{dt} = \alpha y_i - y_i^3 + \beta\left(y_{i-1} + y_{i+1}\right),\ i = 1, 2, ..., n-2 \tag{2}$$

$$\frac{dy_{n-1}}{dt} = \alpha y_{n-1} - y_{n-1}^3 + \beta\left(y_0 + y_{n-2}\right) \tag{3}$$

$$. \tag{4}$$

Here, $\alpha$ and $\beta$ are model parameters that will be set for you, and initial conditions must be specified for $y_i(t = 0)$, $i = 0, 1, 2, ..., n-1$.

1. (3 pts) The function *part2q1* attempts to compute solutions to this problem, however the code is highly inefficient (see the function documentation for details of the code input/output). Complete *part2q1new* so that solutions are computed efficiently. Here, efficiency corresponds to a low wall-time for a given desired level of accuracy, and you should aim for solutions with an error of around $10^{-6}$ or smaller. In your report, briefly summarize the changes you have made to *part2q1*.

2. (6 pts) You have been provided with two numpy arrays in the file, *project2.npy*. Each array contains a set of $n = 1000$ initial values for this system. Analyze the evolution of the system from these initial conditions. Your analysis should go beyond a simple qualitative description of simulation results and should provide explanations of the observed behavior supported with quantitative results obtained from efficient, appropriately designed computations. You should assume that each initial condition is "close" to an equilibrium solution, and you are not required to consider times greater than $t = 40$. Place your analysis in your report and add the accompanying code to the function, *part2q2*. This function contains code for setting up the initial conditions. Note: code used to "process" results (e.g. code for making figures) will not be assessed for efficiency.

3. (3 pts) The function *part2q3* contains code for simulating the time evolution of a small system. Examine the code, and in your report, explain the problem that the code is solving and the approach it is using to solve it. Make 1-2 well-designed plots which illustrate the effect of the parameter *mu*, and briefly describe what is shown in the plots in your report. Add the code used to generate the figures to *part2q3analyze*. You should include results for at least one positive value of *mu*, and you should go beyond simply saying that increasing *mu* makes the results "noisier". You are not required to use the default values of the input parameters in *part2q3*, and it may be helpful to create a modified version of the code suitable for running many simulations.

**Further guidance**

- You should submit both your completed python file (*project2.py*),a pdf containing your discussion and figure(s) (*report2.pdf*). You are not required to use the provided latex template, any well-organized pdf is fine. To submit your assignment, go to the module Blackboard page and click on"Project 2". There will be an option to attach your files to your submission. (these should be named *project2.py* and *report2.pdf*). After attaching the files, submit your assignment. **Jupyter notebook submissions will not be accepted.**

- Please do not modify the input/output of the provided functions without permission. You may use numpy, scipy, matplotlib, networkx, time, timeit, collections, and heapq as needed **except** as noted above for part 1 question 2. Otherwise, please do not import any modules without permission. You may create additional functions as needed, and you may use any code that I have provided during the term. Some functions have input variables with specified default values. You are not required to use these values unless the relevant question explicitly states that you are.

- Your final .py file should not run any code when imported. No figures should be generated when it is imported.

- Marking will be based on the correctness of your work and the degree to which your submission reflects a good understanding of the material covered up through the lecture 11 slides. Excluding figures, you should aim to keep the pdf version of your report to less than 6 pages.

- 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 Ed (which is distinct from "anonymous") or by arrangement with the instructor.

- Please regularly backup your work. For example, you could keep an updated copy of your files on OneDrive, and use Overleaf to prepare your report.

- 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.

- You have been asked to submit code in Python functions, but it may be helpful to initially develop code outside of functions so that you can easily check the values of variables in a Python terminal.

- The weightings for assignments can be found in the lecture 1 slides.