Scientific Computation
Spring 2022
Project 2

Due: Monday March 7th 18:00 GMT

There are three main files for this assignment: 1) the one that you are reading which is the project description, 2) *project2.py*, a Python module which you will complete and submit on Blackboard (see below for details) and 3) *report2.tex*, a template file for your report which will also be submitted on Blackboard. The discussion and figure(s) described below should be placed in this report.

Part 1

1. (3 pts) You have been provided with the function *gSearch* in *project2.py*. This function is an implementation of Dijkstra's algorithm which attempts to use a binary heap data structure to manage the priority queue. Design a timing test to critically assess the efficiency with which the highest-priority node is identified and removed from the queue. Your test should generate one or more figures presenting walltime results, and add a clear and concise discussion of your findings to your report. The code for your test should be placed in the function *test_queue*.

2. (5 pts) You will now assess the capacity of a portion of the internet to transfer data between routers. Each router is represented by a node in a connected, weighted, undirected graph. The edge weights are all positive and the weight for the edge between nodes i and j represents the maximum size of a data packet that can be transferred between the two nodes (in either direction). Network data is provided via an N-element adjacency list, Alist. Nodes are numbered from 0 to $N-1$, and the ith element of Alist is a list of two-element lists. The 1st of these two elements is a neighbor of node i, and the 2nd is the weight for the link connecting nodes i and j. So if len(Alist[i])=m, then node i has m neighbors and Alist[i][j][1] is the weight of the edge connecting i and node Alist[i][j][0] (assuming $j < m$).

   Develop an efficient algorithm to determine *pmax*, the maximum size of a data packet that can be transferred between nodes n_1 and n_1. Here, n_1, n_2, and Alist are all provided as input. Complete the function *find_pmax* so that it finds *pmax* and a feasible route from user n_1 to n_2 with packet size = *pmax*. The function should return both p_{max}‘ and the computed route (see the function documentation). Add a clear description of your implementation to your report.

Note for part 1
You may use networkx, heapq, numpy, matplotlib, time, and timeit for question 1. You may use the collections and heapq modules for question 2. Please do not import/use other modules without permission.

Part 2

In Part 2, you will consider the simulation of initial value problems.

1. (6 pts) The function *solveSDE* computes numerical solutions to the SDE,

$$dX(t) = \lambda X(t)dt + \mu X(t)dW(t)$$

with $\lambda = 2$ and $\mu = 1$. Design a set of numerical tests to critically assess the accuracy, efficiency, and stability of the method implemented in *solveSDE* compared to the Euler-Maruyama method. Your tests should create one or more figures illustrating key trends related to the implemented method and how well it works. These figures and accompanying discussion should be included in your report. Add the code used to generate the figures to the *testSDE* function in *project2.py*. Note: You may find it helpful to create a new function which uses the same method but allows you to easily vary the parameters set within *solveSDE*.

2. (6 pts) Now, consider the following model for food-web dynamics in an ecosystem with $n = 20$ species. The abundance of species i is represented by x_i, and the governing equations are,

$$\frac{dx_i}{dt} = x_i \left(\alpha_i + \gamma \sum_{j=1}^{n} C_{ij} x_j \right) \quad i = 1, 2, .., n \tag{1}$$

$$x_i(t = 0) = 10^{-6} \text{ for all } i$$

The coefficients, α_i, and the coupling matrix C have both been provided and code for loading them is included in the function *model*1.

   (a) Complete the function, *model1*, so that it efficiently and accurately computes a numerical solution for (1) given initial conditions as input. The timespan of the simulation and the number of timesteps to output are provided as input – see the documentation in *project2.py*. The function should return an array containing $x_i(t)$ for each species at each time step (including the initial condition). Add a brief explanation to your report explaining how you have designed your function and why it can be considered to be efficient.

   (b) Now, you will analyze solutions computed by *model1*. You should develop explanations of key trends at early and long times, and you should consider $\gamma = 1$ and at least one other value of γ which will allow you to provide a meaningful comment on the parameter's influence. Add code to the function *analyze1* which generates one ore more figures which support your analysis. Add a discussion of your findings along with your figure(s) to your report. Your analysis should attempt to go beyond a simple qualitative description of the computed results.

Note for Part 2: You may use numpy, scipy, matplotlib, time, and timeit for part2. Please do not use any other modules without permission.

Further guidance

- You should submit both your completed python file and a pdf containing your discussion and figure(s). 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, and include the message, "This is my own work unless indicated otherwise." to confirm the work as your own.

- Please do not modify the input/output of the provided functions without permission. You may create additional functions as needed, and you may use any code that I have provided during the term.

- Marking will be based on the correctness of your work, the soundness of your analysis, and the degree to which your submission reflects a good understanding of the material covered up to the release of this assignment. You should aim to keep the pdf version of your report to less than 5 pages of text with 20 or less figures, however you will not be penalized if you exceed this guidance.

- 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"), using the "Chat" on Teams during a Q&A session, or by arrangement with the instructor.

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

- 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.