Bryson Reese
CS491 Final Project Design Document

Existing Functionality and Technologies Used

This project is a Network Simulator of sorts that I created in my CPE 400 course last semester. The point of the program is to generate random double-weighted graphs and find the shortest distance between each node to every other node in the graph. This program is not optimized in the slightest, it brute forces calculating each possible path between all nodes and crunches the numbers to find the shortest path. The user can specify the number of simulations they want to run, and the program will create a directory of directories representing each simulation. In each simulation directory exists a .png picture of the graph and a text file that has a list of the shortest path between every node to every other node, along with the weight of taking that path.

The project is written in Python3 and makes use of the matplotlib and network libraries to create double weighted graphs and represent them in a .png format. The program brute forces all its calculations.

Unit Testing Coverage Plans

I refactored the code a fair bit to separate functions such as creating the network, creating the project directory, creating the simulation iteration directory, generating the list of paths, and generating the picture representation of the graph. This modularity will help me test individual behaviors of each function. I will be creating mock graphs that I can run the simulation against and ensure that the results come out the same. I may need to refactor the code a bit more to store these in a list, but that shouldn't be an issue. I will easily be able to reach 75% code coverage with this, and I will use the built-in unit testing library from python measure this using Coverage.py.

Centralizing and Testing Code

This project is hosted on GitHub and will be tested using GitHub Workflows. I chose this because it is rather simple to get up and running. Combining this with the built-in unit testing and code coverage calculating tools, this should be a rather easy task.

Building and Deploying Code

Since this is a rather simple python script, building isn't that big of an issue. I will maybe package this together with a README.md file. To get some experience with obfuscation for a bit more real-world applications, I could automate the generation of a .pyo object file so that the code could be effectively obfuscated. I could also make this into a Windows Executable and publish as well, I just heard about that tool.