Introduction

Assignment 1: Graphs

This project provides implementation and analysis of Erdős-Rényi random graphs using NetworkX and Matplotlib packages in Python. The program graphs, stores, and reads graphs from a file. First, the user will input a set of parameters into the terminal. These parameters include values for input and output, constants, nodes, etc. The program also computes the shortest path using Breadth First Search (BFS) when the user specifies a node. Additionally, it calculates the probability value based on the users given constant.

Pre-Requisites

I recommend running the program in the terminal on Windows.

You must have the latest version of Python installed. For installation, visit:
https://www.python.org/downloads/

Additionally, ensure that the NetworkX and Matplotlib packages are installed.

To Run the Program:

1. On windows, start by opening your Powershell or Command Prompt window.
2. Use the 'cd' command to navigate to the directory where the python file is located.
   ```
   cd path/to/directory
   ```
   You should now be located in the directory where the python file is saved.
3. To run the python file, enter the following command in terminal and hit enter:
   ```
   python graph.py
   ```
4. `graph.py` will now execute.
5. Command inputs follow this format:
   ```
       python ./graph.py --input out_graph_file.gml
   --create_random_graph --nodes n --constant c --plot
   --BFS a --output out_graph_file.gml
   ```
   Note: input and output names need to match

.
   Note: gml file must reside in the same location as .py
6. Example inputs:
   - For generating a random graph:
   ```
       python ./graph.py --create_random_graph --nodes
   100 --constant 1.1 --plot --output out_graph_file.gml
   ```

   - For generating a BFS graph.

```
python ./graph.py --input out_graph_file.gml --BFS 1
--plot
```

Note: When the BFS graph is displayed, the nodes may appear clustered together. To view them more clearly, try zooming in.

Author

Karla Chuprinski