Jayden Chung

DS210 A1

Final Project Writeup

The dataset I decided to use is a .txt file that contains a road network of California which I found on Stanford's lage network data collection website. It has over 1 million nodes and 2 million edges. As a Californian, I wanted to see how well connected the roads were in my home state.

My functions file consists of a struct Graph which has an adjacency list that stores numbers for the nodes and vectors for the edges that correspond to the nodes. Because the .txt file has a short description in the beginning, I have to make it so that my read_graph function ignores any line that starts with "#" in order to incorporate only the data. Next, there is the distance function which calculates the distance between two nodes using breadth first search. Additionally, the average distance function finds the average distance between a random sample of nodes by running the distance function for each node and then dividing the total distance by the number of pairs. Also, there is a six_degrees_of_separation function which uses breadth first search to determine if two nodes are within six edges of each other. Furthermore, there is the percentage_within_six_degrees function which uses the six_degrees_of_separation function to calculate how many nodes are within six edges of each other and divides that number by the total number of nodes in the dataset in order to get a percentage.

In my main file, there are two tests. Firstly, I import my functions module and use the functions from that file. My test functions consist of a test for the six_degrees_of_separation function and the distance function. Both tests create a very simple graph and execute the functions to see if they line up with the predetermined answers using assert! or assert_eq! My main function starts off with using the read_graph function to create a hashmap of the dataset. If it does not work, it will return “Failed to read graph.” If it reads the graph successfully, then it will display the percentage of nodes within six edges of each other using the percentage_within_six_edges function. Moreover, because there are so many entries in the California road network dataset, I decided to take a random sample of 100 nodes and find each of their average distances from all the other nodes in the graph using the average_distance function. Afterwards, I get the overall average of the sampled nodes adding all the average distances together and dividing it by the number of randomly sampled nodes.

Upon entering “cargo run” into the terminal, the program will take a while to run due to the sheer number of nodes and edges. It should return that the percentage of nodes within 6 degrees of each other is 0.00234%, and the average distance is approximately 260.3674. However, the average distance can vary due to the nodes being randomly sampled. The test_six_degrees function is supposed to return true, true, and false. The test_distance() function is supposed to return 1 and 2.

Appendix

Dataset: https://snap.stanford.edu/data/roadNet-CA.html

Sources:

https://youtu.be/_lHSawdgXpI?si=wKF1QVd4AytFSF53

https://youtu.be/9PHkk0UavIM?si=DkNggogzrdlMqUBB

https://youtu.be/obWXjtg0L64?si=r1j5GTZHwfgQVf1Q

https://youtu.be/HZ5YTanv5QE?si=tzzfj1Efno2mNDqK

https://youtu.be/Urx87-NMm6c?si=q-EgMgB3CnXV6mAU https://youtu.be/sTK8fagTsMk?si=4tzYJPaN5B3eTIg8

https://youtu.be/puQ3w0FtCcQ?si=0Brw2fCET5vr3sgz

https://youtu.be/6cfcWzsvLrA?si=nCer24eVVptRuk28

https://doc.rust-lang.org/std/index.html