Amira Zuniga

Professor Kontothanassis

CDS DS 210

May 5, 2024

Final Project Report

1. **Introduction**

For my final project, I was inspired by Professor Kontothanassis’ suggestion in-class to test the six degrees of separation principle. This principle assumes all people are six or fewer social connections away from each other. In the age of social media, it is reasonable to assume things may have changed. Not only are millions of connections now at the touch of a button, but there is now a way to keep track of an individual’s social connections. I decided to test this principle on the social network Facebook, implementing various methods to find the average distance between friends, the average number of friends per user, and the most and least popular users.

1. **Dataset and Processing**

I decided to analyze a dataset from SNAP representing social circles on Facebook, importing the “facebook\_combined.txt” file into my program. This dataset represents an undirected graph and contains 4,039 nodes and 88,234 edges.

1. **Module Design**

I used three modules for my program: reader.rs, graph.rs, and analysis.rs.

The reader.rs module is used to import and read data from the given text file. This module contains the function read\_txt, which takes as input a string representing a text file and returns a list of edges from the dataset.

The graph.rs module is used to create a graph from a list of edges, imported from the dataset. This module initializes a struct Graph, which contains the number of vertices in the graph and the graph’s outer representation. This module contains several functions to manipulate the graph. This module is used to reverse the edges in a graph, add directed edges to a graph, sort the adjacency lists, create a new directed graph, and create a new undirected graph.

The analysis.rs module contains several algorithms to analyze the properties of the graph I believed would be relevant to social media. There is a function implementing the BFS algorithm, which is used in another function to compute the average distance between all pairs of vertices (i.e., degrees of separation). This module also contains a function to compute the average number of friends by iterating through all vertices and their neighbors and adding the degree of the current vertex. This module also contains functions that return the vertices with the maximum and minimum degrees. These vertices represent the most and least popular users, respectively.

1. **Main Functionality**

In my main function, I began by importing my reader, graph, and analysis modules. I then imported my dataset through my reader function and created a new undirected graph using the data. I computed and printed the average distance between friends, the most and least popular users, and the average number of friends per user.

To run the main function, ensure all modules are included in the source folder and the dataset is in the project folder. Using cargo run, the program should take less than a minute to finish running.

1. **Testing**

I included three tests in my program: for the average distance calculation, the average degree calculation, and for finding the nodes with the highest and lowest degree.

To test the average\_distance function, I created a new undirected graph with four nodes and three unique edges and calculated the average distance. Because the edges are unique, the average distance should be greater than zero, which is confirmed by the test function.

To test the average\_friends function, I created a new undirected graph with four nodes and four unique edges, where each node has a singular connection. If the average\_friends function is working correctly, the average number of friends per user should be one since each node has one connection and this is confirmed by the test function

To confirm the most popular and least popular nodes are computed correctly, I created the test function test\_popularity, which creates a new undirected graph with four nodes and four unique edges. In this graph, node zero has three connections while nodes one and three have two connections, and node two has one connection. The test function confirms the node with the most “friends” is node zero and the node with the least “friends” is node two.

1. **Results and Limitations**

From running my program, the average distance between Facebook friends is 3.69 ~ 4 friends. This is significantly less than the hypothesized six degrees of separation, so this data provides some support for the claim that social media increases levels of connectivity. This would logically make sense, as one is able to communicate with more people online than in person. It is also possible to be Facebook friends with unknown users, so Facebook friends may not always equate to true friendship.

The most popular user is represented by node #107, meaning it is the node in the graph with the highest degree of connectivity. The least popular user is represented by node #11, meaning it is the node in the graph with the lowest degree of connectivity. The same considerations mentioned in the previous paragraph should be taken with these results.

The average number of Facebook friends per user is 21 friends. This number is difficult to compare, as it depends on how a user interacts with others. For example, some users become Facebook friends with every known acquaintance and even unknown users. For these users, an average friend count of 21 may be low. But other users may be extremely inactive on Facebook, adding only their closest friends. For them, such an average may be high.

Ultimately, my main takeaway from these results is that there are more connections between users in online interactions than between people in face-to-face interactions.

1. **Final Output**

A computer screen shot of a black screen

Description automatically generated