Yana Pathak

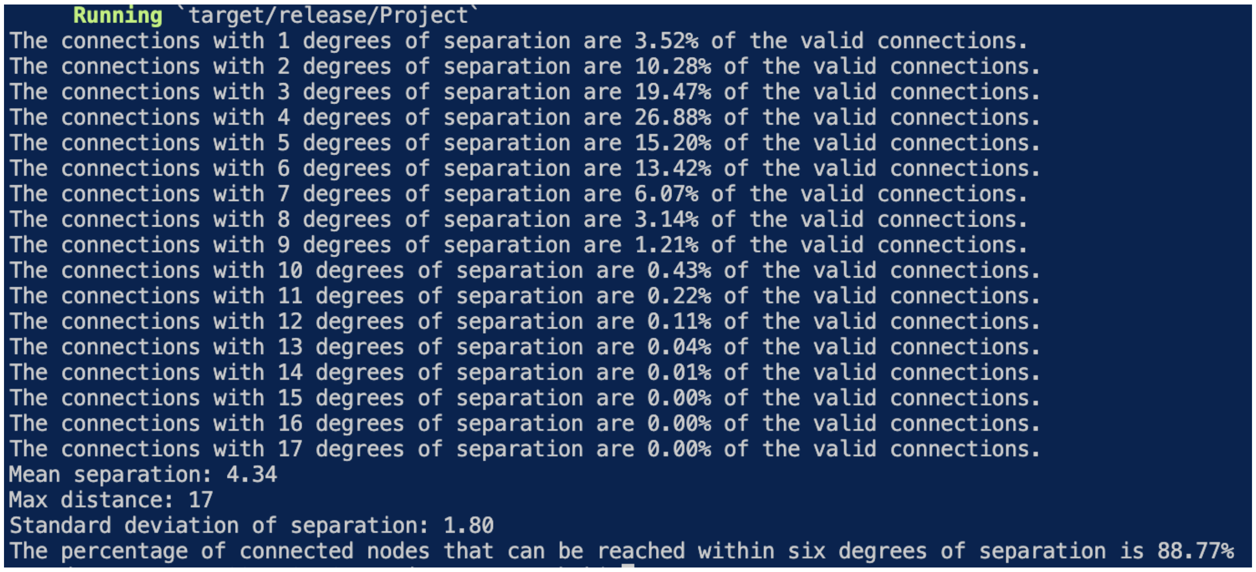
DS210 Final Project Report

I chose to use an undirected dataset showing Facebook connections. The nodes on the "facebook_combined.csv" are a user and the edges are the people whom that user is connected to. There are almost 90,000 vertices in this CSV so when running the code in release mode, it takes almost 10 minutes. For the project, I wanted to look at the degrees of separation between the nodes, or the users of Facebook.

There are 3 files within this project. The main file, a module that processes the "facebook_combined.csv", and a module that implements Breadth-First Search algorithm. The BFS file performs the Breadth-First Search algorithm on a graph to find the shortest path between two nodes. The bfs function takes a graph (represented as an adjacency list), a starting node index, and an ending node index. It returns the shortest distance (in terms of the number of edges) between the start and end nodes, or None if no path exists. The function uses a queue (VecDeque) to traverse the graph and a vector to keep track of visited nodes, ensuring each node is processed only once. The processor module is responsible for reading and processing a CSV file to convert it into a graph representation. It reads the CSV file line by line, skips the header, and then extracts node and edge information from each line. It constructs a list of unique nodes and a map that associates each node with its adjacent nodes (edges). The function returns a tuple containing the list of nodes and the graph's adjacency list. The main file has multiple functions. The main function is the entry point of the program. It uses the processor module to read a CSV file and construct a graph. The function then iterates over all pairs of nodes in the graph, using the bfs function to calculate the distances between them. It maintains a count of the number of connections at each distance, calculates the mean distance, the maximum distance, and the standard deviation of the separations. It also calculates the percentage of connections within six degrees of separation. This information is then printed out to give insights into the network's structure. I created 4 different tests. The test suite contains four tests to validate the BFS implementation. check_self_connection tests whether the BFS function correctly identifies a self-connection (distance of 0). check_one_degree checks if the BFS can correctly find a direct connection (one-degree separation) between two nodes. check_no_connections ensures that the

BFS returns None when there is no path between two nodes. check_valid_connections_count calculates and verifies the number of valid connections in a predefined graph.

This is the output of my code:



```
     Running `target/release/Project`
The connections with 1 degrees of separation are 3.52% of the valid connections.
The connections with 2 degrees of separation are 10.28% of the valid connections.
The connections with 3 degrees of separation are 19.47% of the valid connections.
The connections with 4 degrees of separation are 26.88% of the valid connections.
The connections with 5 degrees of separation are 15.20% of the valid connections.
The connections with 6 degrees of separation are 13.42% of the valid connections.
The connections with 7 degrees of separation are 6.07% of the valid connections.
The connections with 8 degrees of separation are 3.14% of the valid connections.
The connections with 9 degrees of separation are 1.21% of the valid connections.
The connections with 10 degrees of separation are 0.43% of the valid connections.
The connections with 11 degrees of separation are 0.22% of the valid connections.
The connections with 12 degrees of separation are 0.11% of the valid connections.
The connections with 13 degrees of separation are 0.04% of the valid connections.
The connections with 14 degrees of separation are 0.01% of the valid connections.
The connections with 15 degrees of separation are 0.00% of the valid connections.
The connections with 16 degrees of separation are 0.00% of the valid connections.
The connections with 17 degrees of separation are 0.00% of the valid connections.
Mean separation: 4.34
Max distance: 17
Standard deviation of separation: 1.80
The percentage of connected nodes that can be reached within six degrees of separation is 88.77%
```

The output shows multiple different analyses of the CSV. The percentages shown for each degree of separation (from 1 to 17) represent the proportion of all valid connections at that specific distance. For example, "3.52% of the valid connections" for 1 degree of separation means that about 3.52% of all pairs of nodes in the network are exactly one step away from each other. The mean separation indicates the average number of steps between any two connected nodes in the network. A mean separation of 4.34 suggests that, on average, it takes about 4 to 5 steps to travel from one node to another. The code also displays the maximum number of steps found between any two nodes in the network. Even though there are connections at 17 degrees of separation, their proportion is extremely small, rounding down to 0.00% of the total. The standard deviation measures the amount of variation in the separation degrees. A standard deviation of 1.80 indicates a moderate variation in the number of steps between pairs of nodes. The percentage within six degrees tells me that most of the network (88.77%) can be connected within six degrees of separation.