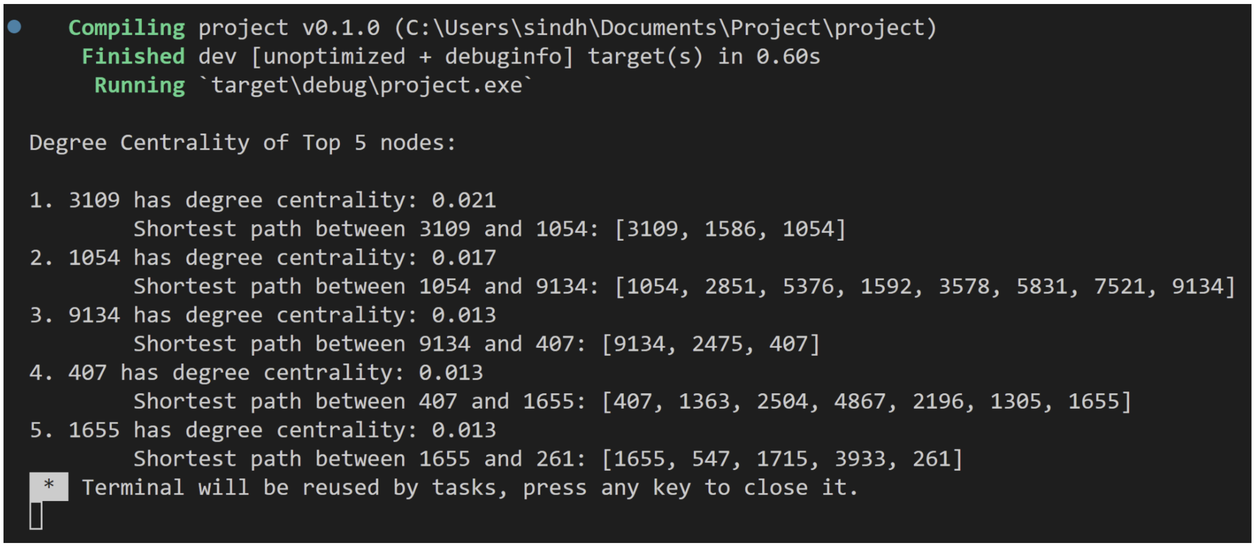
Sindhuja Satheesh Kumar
DS210

DS 210 Final Project Report

For my project, I was interested in exploring the “Gnutella peer-to-peer network, August 4 2002” dataset outsourced from Stanford’s Large Network Dataset Collection. The dataset contains a sequence of 9 snapshots of the Gnutella peer-to-peer file-sharing network from August 2002. The nodes represent hosts in the Gnutella network topology and the edges represent connections between the Gnutella hosts. The dataset can be downloaded as a GNU zip of a text file. Moreover, the dataset has 10876 nodes and 39994 edges, producing a directed acyclic graph. The goal of this project is to find the degree of centrality of all the nodes and output the Top 5 nodes that have the highest degree of centrality. By doing so, we can identify the 5 most influential and significant hosts of the Gnutella network that play an important role in file sharing. This implies that these nodes get high traffic and could be potential points of inspection in handling node failure in the network. In addition to that, the shortest path between two nodes was calculated and implemented on pairs of the Top 5 nodes with the highest degree of centrality.

The degree of centrality of a node is basically its degree which is the number of incoming and outgoing edges. Therefore, the higher the degree the more central the node is in the graph. For each node, it is calculated by dividing the degree of the node by the number of nodes minus 1. In addition, the shortest path is calculated by implementing Dijkstra’s algorithm. The GitHub repository contains a p2p-Gnutella04.txt, *src* folder, Cargo.lock, and Cargo.toml. The p2p-Gnutella04.txt contains the dataset. The *src* folder contains the files main.rs, graph.rs, utils.rs, and tests.rs. The file graph.rs contains the implementation to represent the dataset as a graph using a struct. The file utils.rs contains a function that reads the text file and tests.rs contains all the unit tests for the functions defined in the graph.rs file.



```
   Compiling project v0.1.0 (C:\Users\sindh\Documents\Project\project)
    Finished dev [unoptimized + debuginfo] target(s) in 0.60s
     Running `target\debug\project.exe`

Degree Centrality of Top 5 nodes:

1. 3109 has degree centrality: 0.021
        Shortest path between 3109 and 1054: [3109, 1586, 1054]
2. 1054 has degree centrality: 0.017
        Shortest path between 1054 and 9134: [1054, 2851, 5376, 1592, 3578, 5831, 7521, 9134]
3. 9134 has degree centrality: 0.013
        Shortest path between 9134 and 407: [9134, 2475, 407]
4. 407 has degree centrality: 0.013
        Shortest path between 407 and 1655: [407, 1363, 2504, 4867, 2196, 1305, 1655]
5. 1655 has degree centrality: 0.013
        Shortest path between 1655 and 261: [1655, 547, 1715, 3933, 261]
 *  Terminal will be reused by tasks, press any key to close it.
```

Figure 1. Output

Figure 1 above shows the output when the source code is compiled. The Top 5 nodes with the highest degree of centrality are printed along with the shortest path between 2 consecutive nodes found in the Top 5 nodes. As seen above, the node with the highest degree of centrality is node 3109 with a centrality of 0.021, and the shortest path between 3109 and the node with the second highest degree of centrality which is 1054 is [3109, 1586, 1054].

I was very interested in peer-to-peer networks as they are the main architectures used in Tor for anonymous internet browsers and Bitcoin's decentralized transaction ledger. Therefore, I was curious to explore this dataset and try to learn more about how peer-to-peer networks work.