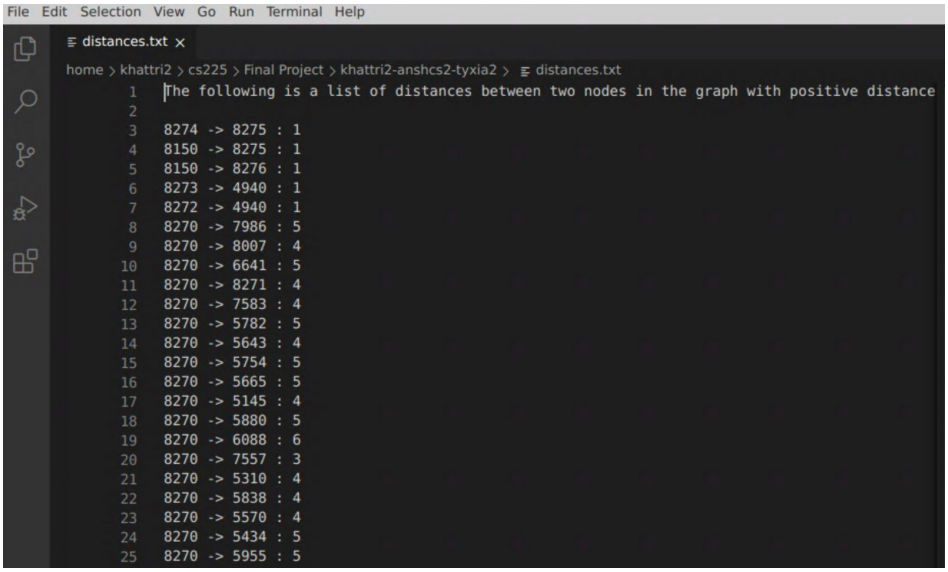
The results of our project, though an achievement in their own right, were different from that which we set out to complete when we had first begun. Due to our usage of the betweenness centrality algorithm, which also required utilizing the Floyd-Warshall algorithm, our original pick for our dataset was not feasible; as one fellow student amusing found over discussing the topic, calculating the runtime of our 800,000 node dataset with the aforementioned formuli would have taken nearly 486 years. Whether or not this is truly accurate, the point was clear to us that we either needed to switch datasets. This endeavour did cost us time, but in the end we believe we delivered a great project.

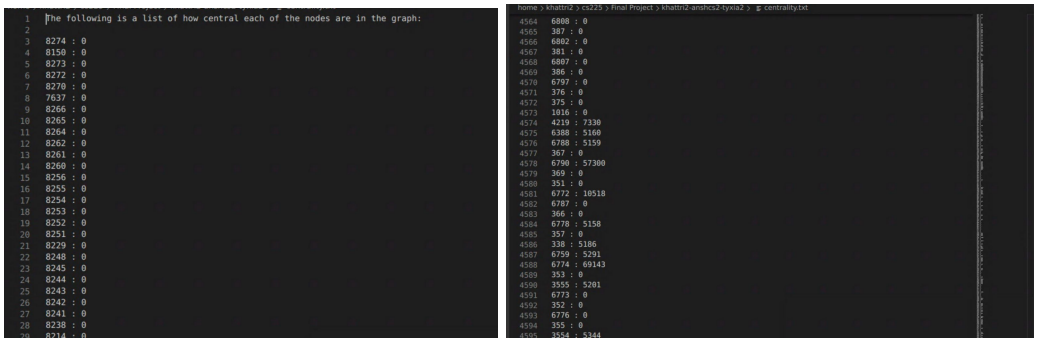
Our program utilizes two main objects, user and connection. Each node in our dataset is representative of a user while each edge is a connection between users. Our program first uses a BFS traversal algorithm, used both to search for a dataset for specific users as well as displaying the total number of users and edges in the data. The next part of our project uses the Floyd-Warshall algorithm for calculations in our Betweenness Centrality algorithm.



Output of Floyd-Warshall distance array

As these parts of the project were completed before Floyd-Warshall was covered in class, it did not come to our attention until implementation that both of these algorithms have very, very bad runtimes, with Floyd-Warshall having a time complexity of O(n^3) and Betweenness Centrality, depending on which algorithm you encounter, having one O(n^3) to O(n*E) (where E is the number of edges). This is the main reason we decided to change datasets. In the end, the program outputs a text document, each line detailing the user, their number of connections, and their centrality value. The running time is also much more bearable; rather than waiting till the heat death of the

universe before we get any returns, our current smaller dataset runs Floyd-Warshall and Betweenness Centrality in about 10-15 minutes.



Output of Betweenness Centrality for each node. Note some nodes contain a value of zero due to no directed edges leading toward them

One of our original goals was to implement a force directed graph, utilizing the centrality value of each node in the algorithm for determining placement. However, this task, coupled with the productivity loss from switching datasets, was deemed too demanding for us to accomplish in the scope of this project; our initial research concluded that we would need to not only utilize our centrality values but also implement either a Hooke's Law or Coulomb's Law . Nevertheless, we did implement all the other algorithms we hoped to achieve and accomplished all our other goals. Overall, we believe we produced a well developed program that runs with reasonable runtime and outputs the correct valuable data.

Based on this data, we can draw a couple conclusions about the behaviour of Wikipedia contributors. There are many users who do not vote on any other user's changes and instead just contribute to wiki pages; this is evidenced by the number of users with a centrality value of 0. The most likely conclusion to be drawn from this is that most of these users can be attributed to people who spotted flaws in existing pages and created accounts solely to fix them. There's also a portion of the user base that are very active, with centrality values in the thousands for some of them.