Jason Chen

DS 210

Prof. Kontothanass

5/4/22

Add a report describing what data set you picked, what interesting thing you discovered, what algorithms you implemented and anything else you consider relevant

The dataset that I picked was the “Pennsylvania road network”. This dataset consists of intersections and endpoints represented by nodes. Then there are the roads connecting to these intersections or endpoints that are represented by undirected edges. In total there are 1088092 nodes and 1541898 edges. Some other inconsequential information included that the largest WCC (weakest connected components) is 1087562, and the SCC (strongest connected components) is also 1087562 - which is surprising. The same applied for the WCC and SCC of edges it is both 1541514. The dataset is huge as it contains over 1000+ vertices, but it only consists of numbers without actual names, so we don't know the roads that the nodes correspond to. The interesting thing that I discovered was that a few of the nodes appeared the same amount of time. Namely Node 859326 (frequency: 9), Node 847932 (frequency: 9), Node 759553 (frequency: 9), Node 674502 (frequency: 9). These are the top most frequent nodes. This means that there isn’t a single ‘most important’ road/nodes to the entire network/graph, but rather that there are 4. These 4 are the ones that connected the rest of the graph/network together. The algorithms that I implemented was breadth first search and a simplified version of betweenness centrality. The breadth first search was implemented because the graph was considered unweighted as each nodes and edges have equal weight. The betweenness centrality was implemented by taking the shortest path that was calculated by BFS. Using the shortest path, I counted the frequency of the nodes that appeared. In the end, the nodes with the highest frequency were return. This is a simplified version of the betweenness centrality as no equation or formula were taken into consideration.