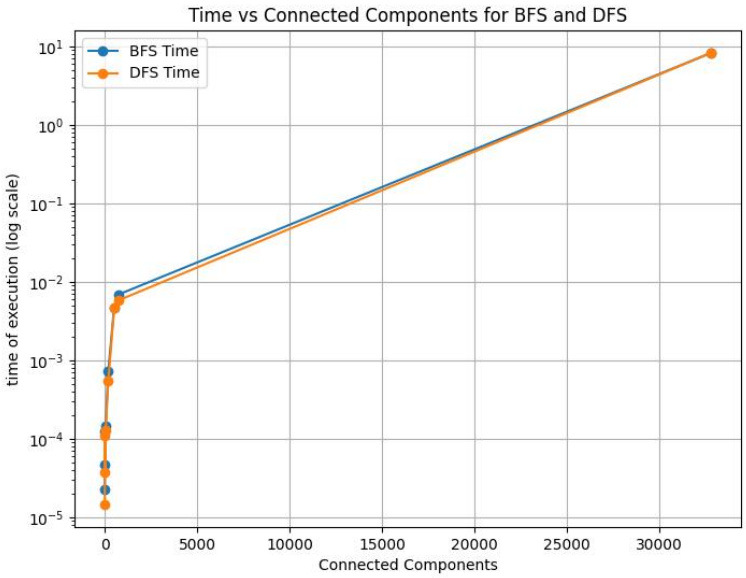
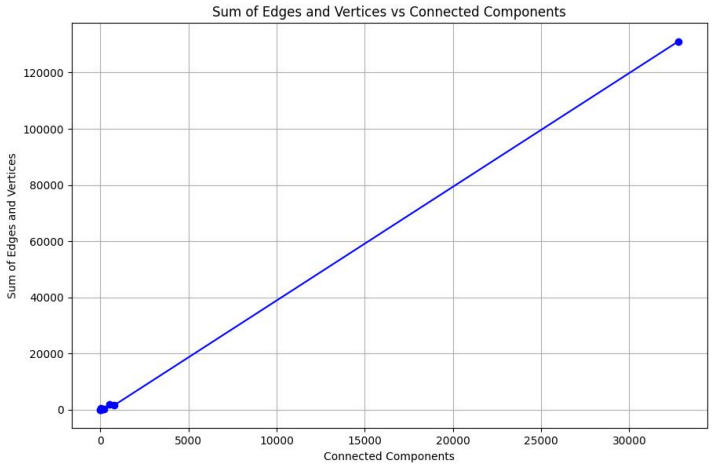
Analysis Name: Yuqi Jin

The first graph is Time vs Connected Components for BFS and DFS, shown in Graph 1.
The connected components are sorted from largest to smallest due to most of the running time for BFS and DFS being very small, taking the log scale of time values to visualize the difference better. In the graph, the y-axis is the number of times in seconds, and the x-axis is the number of connected components. The number of connected components is the same for BFS and DFS methods because both visited all the nodes. The graph shows that the time values of BFS are slightly higher than the time of DFS; in other words, DFS runs slightly faster than BFS. Except in the last case, when the connected components are 32774, the DFS (8.28385s) running time is slower than the BFS (8.26218s).

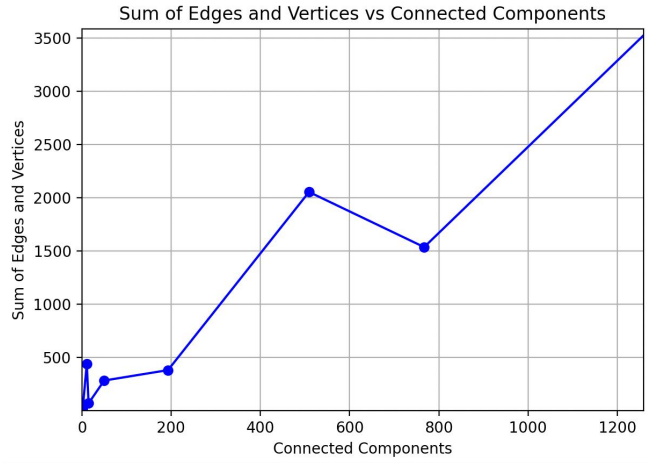


Graph 1: Time vs Connected Components for BFS and DFS

The second graph is the Sum of Edges and Vertices vs Connected Components shown in graph 2. Graph 3 is a zoom-in version by cutting the range of [0, 1200] in the connected components.
The connected components are sorted from largest to smallest, and each edges + vertices corresponds to the connected components. Based on graphs 2 and 3, to know that there is an increasing trace when the graph size becomes more complicated and more extensive, the more connected components come out.



Graph 2: Sum of Edges and Vertices vs Connected Components



Graph 3: Closer Version

Conclusion

This experiment aims to use BFS and DFS to traverse all nodes from the input txt files to find their time executions and connected components. The result shows that using DFS is slightly faster than using BFS. The reason is that DFS is good at traversing at the beginning node and proceeding the nodes as far as possible until it reaches the node with no unvisited nearby nodes. So, DFS would traverse more edges each time to reach a destination vertex than BFS. However, BFS is better at finding the shortest path and more suitable for searching vertices closer to the given source.