CS 225 Project Goals

In the project you must load one or more data files as a graph and be able to use that graph to run algorithms of interest to the data set. This must include at least one of the Traversals we covered in class and be able to output that result.

It must also handle two other options one of which must be from the complex or uncovered options listed below.

Your code must work on your chosen data set and on other data that is of the correct format to match the data set you are working on.

Your code must build and run on EWS using only the material in your repo and libraries publicly installed on EWS. All instructions needed to build the project must be in your repo and easy to understand.

You can not use The Boost Graph Library (BGL) to implement this project since it would completely avoid the project.

Traversals

1. BFS (Breadth First Search)
2. DFS (Depth First Search)

Covered Algorithms

1. Shortest Path
    a. Dijkstra's Algorithm
    b. Floyd-Warshall Algorithm
2. Minimum Spanning Tree
    a. Kruskal Algorithm
    b. Prim's Algorithm

Complex or Uncovered Options

1. Landmark Path (shortest path from a to b through c)
2. A* Search (https://en.wikipedia.org/wiki/A*_search_algorithm)
3. Iterative deepening depth-first search (https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search)
4. Delta-stepping SSSP (https://www.sciencedirect.com/science/article/pii/S0196677403000762)
5. Graphic Output of Graph

    a. Project on to map based on data
    b. Force-directed graph drawing (https://en.wikipedia.org/wiki/Force-directed_graph_drawing)
    c. Layered graph drawing (https://en.wikipedia.org/wiki/Layered_graph_drawing)
    d. Other methods
6. Betweenness centrality (https://en.wikipedia.org/wiki/Betweenness_centrality)
7. Graph coloring (https://en.wikipedia.org/wiki/Graph_coloring#Algorithms)
8. Eulerian path / cycle identification (https://en.wikipedia.org/wiki/Eulerian_path)
9. Strongly connected component (https://en.wikipedia.org/wiki/Strongly_connected_component#Algorithms)
10. PageRank (https://en.wikipedia.org/wiki/PageRank)
11. Other ideas with improvement