README

The tsp_node class creates nodes with data which acts as a label for the TSP problem.

dsa_final.ipynb: This file contains the methods that generate and test the graphs for the Brute Force algorithm, Nearest Neighbor algorithm, and Hungarian algorithm. The tsp_node class creates nodes with data which acts as a label for the TSP problem.

More specifically, the node_generator function is given a graph and adds n nodes to it and the edge_generator function, given the same graph with n nodes, generates edges of random length between 10-50 between each node and every other node in the graph. Additionally, it returns a matrix representing the grab and edge lengths for the Hungarian algorithm. The node labeler makes a dictionary with labels for each node, which makes the networkx graph look cleaner and makes it more clear which node is which. The brute_force_solver is given a graph with nodes and edges, and generates every possibility for the TSP on that graph, returning the route with the lowest combined edge weights.

nearest_neighbor_finished.ipynb: This file contains the implementation of Nearest Neighbor algorithm before it is run with the edge and node generators. It takes in two values: the starting city coordinate, and a list of all cities that need to be visited. Further explanations of methods are documented within the file.