Part A

For our graph implementation, we created vertex and edge classes. Each vertex has a key, value, and list of neighboring vertices. Each edge has a list of two vertices, and a weight. The vertices and edges that make up a graph are stored in lists. We chose lists because they are simple to use, are built into Python, and can easily add or remove elements. A tutorial on how to use our graph implementation is included in the graph_demo.py file.

We represent obstacles in our maps by setting the weights of any edges connected to the obstacle to infinity. When printing a visualization of a map, the obstacles are represented by an ‘@’. A tutorial on how to use our map implementation is included in the graph_demo.py file.

Part B

The tutorial/demo for the four functions from part B is included in the part_b_demo.py file.

Part C

The tutorial/demo for the two functions from part C is included in the part_c_demo.py file.

Bonus Improvements

We added Whatever First Search (WFS), a print function for WFS, modified Dijkstra and WFS to make SPSP algorithms, and added a print function for the modified functions. A demo of these can be found in the bonus_demo.py file.