/\*CSC 172

\* Author: Tianyi Ma

\* NetID: tma8

\* Lab time: MW 615-730

\* Lab number: Project04StreetMapping

\*/

Using command line in the following form: java ProgramName map.txt [-show] [-directions startIntersection endIntersection] [-meridianmap]

You can draw a map of the University of Rochester/Monroe county/New York City, find the direction of the shortest path between two arbitrary points, and also find the minimum spanning tree for a meridian of the U of R campus.

The most important factor that will affect the program’s runtime is the way to store edges and vertices. As the map gets more complex, there are a large number of edges and vertices, and thus how to get/set an arbitrary item matters a lot. Initially I use List to store the data, but then it turns out the rumtime is too long (O(n) for each search). That’s when I switch to HashMap because the average runtime for its get() method is O(1) (as the set is unordered). This significantly increase the speed of the program.

**\* Extra credits \***

**1. Simple and elegant mapping design**

**2. Interactive mapping**

**NOTE: I noticed that some pairs of intersections in monroe.txt and nys.txt can’t be connected to each other, so it’s better to try multiple times if the program doesn’t function.**

**The functionality and correctness of the algorithm finding shortest path can be proven correct using pairs in ur.txt**

**For monroe.txt and nys.txt, considering the long runtime due to large data set, here are some sample pairs that are proven viable:**

**for monroe.txt:**

Canvas monroe.txt [-show] [-direction i1400 i7054]

**for nys.txt:**

Canvas nys.txt [-show] [-direction i123456 i124000]