Cody Do

105140467

Smallberg 3C

Project 4 Report

**StreetMap**

*load():*

O(NGS+NGR ); If N is the number of lines in mapdata.txt, G is the number of unique GeoCoords per street, S is the average number of StreetSegments per street and R is the average number of reversed StreetSegments per street, then Big O of this program is N\*(G\*(S+R)). In expanded form, this is NGS+NGR.

*getSegmentsThatStartWith():*

O(N); If N is the number of StreetSegments associated with the input GeoCoord, then Big O of this program is simply N.

**PointToPointRouter:**

*generatePointToPointRoute():*

This class utilizes the A\* algorithm. There are two primary data structures that this program uses: a Node (implemented in support.h/.cpp) and a List. A Node has a pointer to another Node. This other node is known as the parent Node as its the parent Node is what was used to create the current Node. For the starting Node, the Parent Node is simply nullptr. Nodes also have a starting GeoCoord and a destination GeoCoord (the GeoCoord that we wish to reach). It also stores a StreetSegment and a boolean variable to that tell whether it’s the starting Node or not (for appropriate actions). Inside the implementation of the function itself, there are two lists. One is called openList and is meant to function as a priority queue (this was done because I couldn’t figure out how to create a comparison between two Node pointers) and one is called closedList.

**DeliveryOptimizer:**

*optimizeDeliveryOrder():*

O(N2); If N is the number of DeliveryRequests in the vector, then the Big O of this program is the outer while-loop (N) times the inner while loop (N). In order words, Big O is N2.