Dylan Lee

205300889

Project 4 Report

Big-O Time Complexity

* StreetMap
  + load() – If mapdata.txt holds N lines, get will be retrieved one time, resulting in a time complexity of O(N).
  + getSegmentsThatStartWith() – If the ExpandableHashMap holds N associations, and ExpandableHashMap.find() has a time complexity of O(1), getSegmentsThatStartWith() has a time complexity of O(1).
* PointToPointRouter
  + generatePointToPointRoute() – For generatePointToPointRoute(), I implemented the A\* algorithm. For my implementation, I had two three ExpandableHashMaps, two of which mapped GeoCoords to doubles to store the gScores and fScores of GeoCoords, respectively. The other ExpandableHashMap mapped GeoCoords to GeoCoords, to store which GeoCoord (value), each GeoCoord (key) came from. This could then be traced back to find the optimal path. I also implemented a priority\_queue to keep track of the GeoCoord with the lowest fScore, and a set to keep track of what was contained within the priority\_queue, because they cannot be searched.
* DeliveryOptimizer
  + optimizeDeliveryOrder() – If the vector deliveries holds D DeliveryRequests, the time complexity is O(D^2 + D) or just O(D^2). This is because the deliveries array is looped through in two single loops and one nested loop.