Reema Kshetramade

SID: 205328402

Project 4 Report

1. StreetMap
   1. load() – has a time complexity of O(N) with N being the number of lines in the file. This is because each line in the file is read and processed once.
   2. getSegmentsThatStartWith() – has a time complexity of O(S) with S being the number of street segments associated with that coordinate. This is because accessing the segments that are stored in the hash\_map for that particular coordinate takes O(1) time because no traversing through the loop is required. Then it takes O(S) time to copy the street segments stored in the map over to the parameter vector. This simplifies to a complexity of O(S).
2. PointToPointRouter
   1. generatePointToPointRoute() – O(N\*SlogS) with N being the number of coordinates in the StreetMap data and S being the number of segments that start at a particular coordinate. The N part of the complexity is because in the worst case it will have to process all the coordinates in the StreetMap data to find the route. The SlogS comes because for the S segments that start at the current coordinate being visited, they are sorted using quick sort (based on their distance from the end point) which, in the average case, has a complexity of O(SlogS).
3. DeliveryOptimizer
   1. optimizeDeliveryOrder() – O(NlogN) with N being the number of deliveries passed in. This is because the deliveries are ordered using quick sort (based on their distance from the delivery location or the depot) which, in the average case, has a time complexity of (NlogN). There is also a factor of 2N because to calculate the old and new crow distances, there are two for loops that run N times each; however, because this is of a lower degree, it can be dropped.