Shreyas Bharadwaj

UID: 705 342 716

Project 4: GooberEats Report

Asymptotic Time Complexity of the Following Functions:

* StreetMap::load()
  + This function, on a general level, makes a single pass through the mapdata.txt file and inserts StreetSegments into the ExpandableHashMap data structure. Supposing that the ExpandableHashMap is implemented like a hash table, insertions and lookups are all done in constant time; that is, Since it inserts such segments, namely those that are in the mapdata.txt, the total runtime of load() is
* StreetMap::getSegmentsThatStartWith()
  + This function is essentially nothing but a lookup for a hashtable of KeyType GeoCoord and ValueType vector<StreetSegment>. Therefore, as long as the load of the hashtable is properly maintained, the runtime in the average case is for a hashtable of say, key-value pairs.
* PointToPointRouter::generatePointToPointRoute()
  + This function is a queue-based breadth first search of the graph of coordinates stored from mapdata.txt, in order to find a list of StreetSegments from the first GeoCoord to the second GeoCoord. Therefore, given a graph of nodes, the breadth first search will have to traverse at most nodes to find the path from the given starting node to the end node. The runtime is hence
* DeliveryOptimizer::optimizeDeliveryOrder()
  + The optimizeDeliveryOrder()method picks the shortest distance from each delivery point to the next, starting and ending with the depot. To do so, it goes through the vector of DeliveryOrder in a nested for loop, and therefore runs intime given DeliveryOrders.