Helen Wang – CS32

405320396

3/14/20

Proj4 Report

StreetMap:

* load():
  + if the file contains N StreetSegments, load() has O(N)
  + load() iterates through each of N lines in the file and calls addStreetSegment to add a segment to the hashmap; addStreetSegment calls find and associate which have O(1) so addStreetSegment has O(1)
* getSegmentsThatStartWith():
  + if StreetMap holds N geo-coordinates and geo-coordinate is associated with S StreetSegments on average, getSegmentsThatStartWith() has O(S)
  + getSegmentsThatStart() calls find, which has O(1), to get the vector of slots and then iterates through each slot to get the StreetSegment at the slot and add it to the segs vector parameter

PointToPointRouter

* generatePointToPointRoute(): I used A\* algorithm for this, so I will describe the data structures I used instead
  + Created struct StreetSegmentNode holding a StreetSegment, its value f, and a pointer to the previous StreetSegment; a nested private class to compare StreetSegmentNodes; a minheap of StreetSegmentNodes using stl priorityqueue where StreetSegmentNodes were compared by their f value
  + Used vector of StreetSegmentNodes to keep track of nodes taken off minheap to deallocate after the route was generated
  + Used vector of StreetSegments to hold StreetSegments whenever a method was called that put StreetSegments in the function call
  + Used ExpandableHashMap mapping geo-coordinates to StreetSegments to hold the final route generated

DeliveryOptimizer

* optimizeDeliveryOrder():
  + if there are N DeliveryRequests, optimizeDeliveryOrder() has O(N^2)