# StreetMap

***load()***: If the *mapdata* file contains N lines of data, the *load()* function in StreetMap has average time complexity O(N).

***getSegmentsThatStartWith()***: If the streetmap holds a total of G geocoordinates, and each geocoordinate is associated with S segments on average, the *getSegmentsThatStartWith()* function has average time complexity O(S) (assuming the coordinates are well-distributed across the hash map).

# PointToPointRouter

***generatePointToPointRoute()***: The function is built on the A\* search algorithm. It uses the ***map*** data structure (*#include <map>*) for the following three variables:

1. ***parents***: a map from *GeoCoord*’s to *GeoCoord*’s where each *GeoCoord* key is associated with a *GeoCoord* value adjacent to it through which runs the most optimal path to that key. Each value is therefore the best “parent” of each key.
2. ***openList***: a map from *GeoCoord*’s to doubles which maps each GeoCoord the algorithm has yet to push into the closed list to a double representing its *g* value, i.e. the distance along the currently-most-optimal path to it.
3. ***closedList***: a map from *GeoCoord*’s to doubles which maps each *GeoCoord* (that the algorithm has found the most optimal path to) to its optimal *g* value, i.e. a double representing the distance along that optimal path.

Furthermore, auxiliary functions such as *findCurrentCoord()* also use the map data structure to identify the *GeoCoord* with minimal *f* value i.e. the distance along the optimal path to a *GeoCoord* plus the Euclidean distance between the *GeoCoord* and the end *GeoCoord*.

# DeliveryOptimizer

***optimizeDeliveryOrder()***: The function is built on the Simulated Annealing optimisation algorithm. The algorithm only uses the *vector* container to represent delivery plans. Shuffling of the delivery plan vectors is implemented via the Fisher-Yates shuffling algorithm.