MyMap:

associate():

Since MyMap is a binary searching tree, to associate N elements into MyMap has average time complexity O(Nlog2N).

find():

Same average time complexity as associate(), if there are N elements in MyMap, the average time complexity is O(log2N)

--------------------------------------------------------------------------------------------------------

AttractionMaper:

init():

There is one for-loop nested in another for-loop, if there are N StreetSegments in MapLoader and total A Attractions, the outer for-loop will loop N times, and the time complexity to associate A elements is O(AlogA). Since A is the total times of operations to do, and we don’t need to insert any StreetSegments, the average time complexity of init() is O(N+AlogA)

GetGeoCoord ():

This function is using MyMap.find() to achieve the goal, If there are total N attractions in MyMap, the average time complexity is equal to searching in binary search tree, which is O(logN)

--------------------------------------------------------------------------------------------------------

SegmentMapper:

init():

If there are total N StreetSegment and total A attractions within those StreetSegments, there will be total N+A elements need to insert to MyMap, there for it has average time complexity O(log(N+A)) find the correct position to insert each element, and every StreetSegment and Attraction need to be inserted, there for it will run (N+A) times. Therefore the average time complexity for init() is O((N+A)\*log(N+A))

getSegments():

This function is using MyMap.find() to achieve the goal. If there are total N StreetSegment and total A attractions within those StreetSegments, there will be N+A elements in MyMap, therefore the average time complexity for getSegments() is O(log(N+A))

--------------------------------------------------------------------------------------------------------

Navigator:

Navigate():

If there is total N StreetSegment and A attractions within, the AttractionMapper will contains A elements and the SegmentMapper will contains N+A elements in Navigate(). On each coordinate, it need O(log(N+A)) time to find the correct node and return vector of StreetSegments that relate to this coordinate. There are about total N+1+A coordinates need to search in worse case. Therefore the average time complexity for Navigate() is O((N+A)\*log(N+A)).