Dennis Zang

Report

MyMap:

[associate()]
If a MyMap object currently holds onto N pairs of elements, then associate() will typically take O(log(N)) time. Since associate is a binary search tree, it will search for a particular key value in $\log_2(N)$ iterations utilizing binary search (before inserting a node or replacing a value in constant time). Of course, if the map is in the middle of inserting values (and has yet to reach N), the time complexity will still be proportional to log(N). Another case to consider is if the map is very unbalanced, and in such cases, the search will start to become O(N). Still, the average case can still be considered O(log(N)), since the tree will likely not be so unbalanced.

[find()]
Again, with N pairs of elements, find() will typically take O(log(N)) time. As mentioned above, binary search will take an average of $\log_2(N)$ iterations.

AttractionMapper:

[init()]
Given N StreetSegments and A Attractions, init() will take O(N+Alog(A)) time. The function will iterate through N StreetSegments (from the MapLoader parameter) in O(N) time, and for each Attraction it finds, “associate()” it into a MyMap binary search tree (in O(log(A)) time, A times).

[getGeoCoord()]
The getGeoCoord() function mirrors the basic find function for the template class MyMap, taking O(log(A)) time. For each of the A Attractions in the binary search tree, the function will make $\log_2(A)$ iterations to find the value.

SegmentMapper:

[init()]
Given N StreetSegments and A Attractions, init() will take O((N+A)log(N+A)) time. In the init() function, SeqmentMapper will create a map each GeoCoord to any associated StreetSegment and Attraction. So, it must go through 2N+A iterations to access each GeoCoord (resulting in about N+A GeoCoords [due to some overlaps]) (each StreetSegment by itself has 2 GeoCoords, and will take twice as many steps to process), and then “associate” then into a binary search tree in log(N+A) time. This rounds down to O((N+A)log(N+A)) time (ignoring the constant coefficient).

[getSegments()]
The getSegments() function mirrors the basic find function for the template class MyMap, taking O(log(N+A)) time. For each of the N+A Attractions in the binary search tree, the function will make $\log_2(N+A)$ iterations to find the vector of StreetSegments.

Navigator:

[navigate()]
The navigate() function, given N+A potential geoCoords, each time it travels to a new intersection (or destination), it must iterate through the SegmentMapper object log(N+A) times to get a viable GeoCoordinate, and average O(log(N+A)) time to record all traveled GeoCoords in a MyMap object (used like a set). The number of StreetSegments the function will traverse through is proportional to N+A (N times for storing a new GeoCoord in a priority queue, and A possible destinations to check whenever the function reaches a new StretSegment). So, iterating proportionally to N+A and dealing with binary search trees in log(N+A) time per iteration, the time complexity is O((N+A)log(N+A)).

*for all of the above, N refers to the number of StreetSegments, and A refers to the number of attractions