

Project #3 (Map Routing), ECE368

Name: Abhishek Srikanth

Login: srikanth@purdue.edu

Overall Approach to the problem and general summary of solution

1. Read the map file first and populate an array of 'Vertice' points with the information provided in the map file.
2. The adjacency list for each Vertice is populated using edge information.
3. For indexing/locating the Vertices in the Vertice array, an alternate integer array is used to store location of each ID value.
4. The query file is then read and each query processed.
5. The source element is located in $O(1)$ time using the addressing integer array and is placed at the top of the Vertice array.
6. The Vertice array is used to maintain both the min-heap and the visited list.
 - a. The upper portion of the array is the min-heap and it shrinks every iteration.
 - b. The lower portion is the visited list and it grows in size per iteration.
7. At the start of each iteration, the first element of min-heap is swapped with the last element in the min-heap and then the new first element is downward heapified.
8. After this, each element in adjacency list of above swapped element is updated and upward heapified if need, provided not present in the completed list. (Check completed list using binary search)
9. Loop repeats until either min-heap is empty or destination element has been popped from the min-heap.
10. The path is calculated by following predecessors from destination.
11. Vertice array is then reset (distance and predecessor reset).
12. New query is processed until no more queries remain.

Assumptions made

The map file will have vertex data with ID number in continuous ascending order starting from the index of 0.

NOTE: To add `-lm` flag so that math.h square root function might work.

Help Received

None

Problems encountered

1. Getting a valid data structure that would provide good time complexity for execution even as it does not take too much memory.
2. In my program an infinity was represented using a -1 and this caused lot of problems making a downward heapify function since each '-1' situation had to be individually checked and acted upon accordingly.

Comments/Feedback

The exercise was enjoyable and challenging although not hard. It was not too time consuming either.