Instructions to run the code:

Compile c++ code in GNU G++17 7.3.0 or similar

Input must be named “input.def” and it will be generated an “out.txt” text file, you can change this part of the code.

Algorithm:

(1) Read the input and keep coordinates, name and out/in drivers in a vector

(2) Generate the cluster (k-mean optimized) $O(n)$.:

(2.1) We get 16 random centers for the 16 clusters.

(2.2) We connect every point with its nearest cluster center in $O(n)$ (Voronoi Diagrams).

(2.3) We get the mean of every set of points with an amortized quick search in $O(n)$.

(2.4) Repete (2.2) and (2.3) until clusters get stable (30 times).

(3) For every cluster, we will find an optimal path with Square Root Decomposition $O(nlog^2(n))$:

(3.1) Sort point by x/y (two iterations, keep the best).

(3.2) Sort point as in MO’s algorithm but in 2D in $O(nlogn)$ (blocks by y, sort by x).

(3.3) Notice that blocks are not Square Root decomposition, to get the size of the blocks, we multiply sqrt(n) by x, where we do golden ratio ternary search to get optimal x in optimal time (do not have time to implement, but the solution is quite improved).

(3.4) We connect every point in the order of the vector.

(4) Connect drivers and clusters in $O(1)$:

(4.1) Sort points and divers by y and connect them in order.

(5) Give the output.

The total time complexity of the algorithm is $O(n)\,amortized$ and the space complexity is $O(n)$