Université d’Ottawa
Faculté de génie

École de science d’informatique et de génie électrique

uOttawa
L’Université canadienne
Canada’s university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering and Computer Science

Taxi!
Comprehensive assignement (32%)
CSI2120 Programming Paradigms

Winter 2022

Ce projet est à réaliser de façon individuelle

Part1 1 - due February 7 at 23:00

Part 2 - due February 28 at 23:00

Part 3 and 4 - due 8 April at 23:00

Late submission is accepted for two day with penalty (1 % minus each late hour.)
The assignment is individual work. NO Group work allowed.

Problem description:

In this comprehensive assignment, you will implement a data clustering algorithm named DBSCAN - Density-Based Spatial Clustering of Applications with Noise. Given a large set of data points in a space of arbitrary dimension and given a distance metric, the algorithm can discover clusters of different shapes and sizes, marking as outliers isolated points in low-density regions (whose nearest neighbors are too far away).

The DBSCAN algorithm uses two parameters:

- **minPts:** The minimum number of points (a threshold) in the neighborhood of a point for this one to be considered to belong to a dense region.
- **eps (ε):** A distance measure that is used to identify the points in the neighborhood of any point.

DBSCAN Algo:

```
DBSCAN(DB, distFunc, eps, minPts) {
    C := 0                                                  /* Cluster counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
        Neighbors N := RangeQuery(DB, distFunc, P, eps)     /* Find neighbors */
        if |N| < minPts then {                              /* Density check */
            label(P) := Noise                               /* Label as Noise */
            continue
        }
        C := C + 1                                          /* next cluster label */
        label(P) := C                                       /* Label initial point */
        SeedSet S := N \ {P}                                /* Neighbors to expand */
        for each point Q in S {                             /* Process every seed point Q */
            if label(Q) = Noise then label(Q) := C          /* Change Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously processed */
            label(Q) := C                                   /* Label neighbor */
            Neighbors N := RangeQuery(DB, distFunc, Q, eps) /* Find neighbors */
            if |N| ≥ minPts then {                          /* Density check (if Q is a core point) */
                S := S ∪ N                                  /* Add new neighbors to seed set */
            }
        }
    }
}

RangeQuery(DB, distFunc, Q, eps) {
    Neighbors N := empty list
    for each point P in database DB {                      /* Scan all points in the database */
        if distFunc(Q, P) ≤ eps then {                     /* Compute distance and check epsilon */
            N := N ∪ {P}                                   /* Add to result */
        }
    }
    return N
}

/* Reference: https://en.wikipedia.org/wiki/DBSCAN */
```

Problem to solve

You are managing a taxi fleet in NYC and you would like to identify the best waiting areas for your vehicles. To solve this problem, you have at your disposal a large dataset of taxi trip records; more specifically you will use the 2009 records available here.

Each record of this dataset includes the GPS coordinate of the starting point and end point for the corresponding trip (note that for the most recent years, locations are specified by zone number which is not useful for us). Since we want to identify the best waiting area, we are interested by the starting points. The dataset is contained in a simple comma-separated value file (csv), each line corresponding to a trip record. The different columns correspond to the attributes of each trip and are named as follows (if needed, more details can be found in the web site):

```
vendor_name,Trip_Pickup_DateTime,Trip_Dropoff_DateTime,Passenger_Count,Trip_Distance,
Start_Lon,Start_Lat,Rate_Code,store_and_forward,End_Lon,End_Lat,Payment_Type,
Fare_Amt,surcharge,mta_tax,Tip_Amt,Tolls_Amt,Total_Amt
```

We therefore ask you to use the DB-SCAN algorithm in order to cluster the starting point locations of the trip records in the NYC 2009 taxi dataset. The centers of the largest clusters will become the waiting area for your taxi fleet. Note that in order to accurately apply the DB-SCAN algorithm, the distance between two GPS locations should be measured in meters. However, as a simplification and because the GPS locations are all located in a relatively small area, you can simply use the Euclidean distance between the GPS coordinates.

Programming

You have to write programs under different paradigms that solves different versions of this problem. You will receive additional instructions for each language.

Each program will be marked as follows:

Program produces the correct value	[1.5 points]
Program produces the correct set of items	[1.5 points]
Adherence to programming paradigm	[3 points]
Quality of programming (structures, organisation, etc)	[1 point]
Quality of documentation (comments and documents)	[1 point]

All your files must include a header showing student name and number. These files must be submitted in a zip file.