

# Projet Intégrateur CSI 2520

## Hiver 2022

#300201661

### Statut du code

- Compile
- Prends beaucoup de temps à finir (16 min avec 8 consommateurs)
- Paramètres actuels :  $N = 4$  ; Nombre de consommateurs : 8

```
Number of points: 232050
SW:(40.700000 , -74.000000)
NE:(40.800000 , -73.930000)

Partition 30000000 : [ 0, 17]
Partition 31000000 : [ 68, 226]
Partition 20000000 : [ 144, 272]
Partition 21000000 : [ 148, 288]
Partition 10000000 : [ 413, 540]
Partition 32000000 : [ 944, 1192]
Partition 3000000 : [1828, 2015]
Partition 0 : [6063, 6514]
Partition 33000000 : [1884, 2187]
Partition 11000000 : [13530, 14022]

Execution time: 16m5.1821504s of 206102 points
Number of CPUs: 16
```

- Observations : les DBScans avec le moins de clusters sont ceux à être achevés en premier

### Références :

DBScan : Basé sur le pseudo-code donné dans l'énoncé du projet

Producteur-Consommateur : Basé sur le code donné sur Brightspace

## Méthodes ajoutées :

- **distance**(x,y LabelledGPSCoord) (float64): retourne la distance euclidienne entre x et y
- **rangeQuery**(point LabelledGPSCoord,coords []LabelledGPSCoord,eps float64 )( []LabelledGPSCoord) : retourne un slice contenant les LabelledGPSCoord voisins de point (la distance entre eux et point <= eps)
- **consomme**(jobs chan int, done \*sync.WaitGroup, grid [4][4][]LabelledGPSCoord) : Partie consommatrice du programme. Cette fonction assure aussi l'exécution du DBScan sur la partition qu'elle « consomme »