

# Algorithmique

## TP : Isotoponymes et villes confondues

Raphaël Ginoulhac

September 21, 2019

On reprend les sorties de l'algorithme :

Histogram of the repetitions of town names :

Number of town names used by 1 towns exactly : 31665  
Number of town names used by 2 towns exactly : 1083  
Number of town names used by 3 towns exactly : 200  
Number of town names used by 4 towns exactly : 87  
Number of town names used by 5 towns exactly : 39  
Number of town names used by 6 towns exactly : 13  
Number of town names used by 7 towns exactly : 10  
Number of town names used by 8 towns exactly : 5  
Number of town names used by 9 towns exactly : 2

Histogram of the repetitions of town coordinates :

Number of town coordinates used by 1 towns exactly : 33030  
Number of town coordinates used by 2 towns exactly : 1043  
Number of town coordinates used by 3 towns exactly : 20  
Number of town coordinates used by 4 towns exactly : 1

Creating the set N of towns whose name is repeated at least once, and C of towns whose coordinates are repeated at least once.

Size of N : 3515

Size of C : 2150

The intersection of N and C has 207 elements.

The number of towns where we can be mistaken by hearing about a town A close to a town B is : 0.

Pour la dernière question, on fait au maximum  $207 \cdot 4 \cdot 9 = 7452$  itérations. La méthode naïve implémentée au dessus fait 4 parmi 210 (4 boucles et 207 éléments dans l'ensemble) = environ 78 Millions d'itérations. Donc le gain de temps est d'un facteur 10.000.

La méthode naïve consiste à imbriquer 4 boucles et à chercher parmi les éléments de N inter C. Une méthode plus efficace consiste à parcourir les éléments de N inter C avec une première boucle, puis faire une deuxième boucle sur les villes v2 de mêmes coordonnées que v1 (4 possibilités max d'après l'histogramme), une troisième boucle sur les villes v3 de même nom que v1 (9 possibilités max).

Dans cette dernière boucle, on calcule le `set_intersection` entre les villes de même nom que `v2` et celles de mêmes coordonnées que `v3`. S'il est non vide, c'est qu'on a trouvé un exemple de quadruplet `v1,v2,v3,v4` répondant aux exigences de la question.