

# Rapport de TP : Isotoponymes et villes confondues

Mohammed Amine KHELDOUNI

12 décembre 2016

## Introduction

Dans ce TP, on se propose de faire l'étude des homonymes et isotopes dans une liste des villes de France (ville.txt puis ville2.txt).

On utilise pour cela les dictionnaires en C++, ou tableaux associatifs, ( $map < T1, T2 >$ ) puis des ensembles ( $set < T >$ ) pour trouver les isotoponymes d'une ville, c'est à dire les villes ayant un nom commun avec une autre et des coordonnées égales à celle d'une autre ville.

## Histogramme des occurrences de noms

Dans cette première section, on cherche à compter le nombre d'occurrence de chacun nom de ville dans notre ensemble de villes du fichier villes.txt

Pour cela, on utilise un tableau associative noté *names\_occurrence* qui a chaque *string* nom de ville associe un *int*, son occurrence dans la liste de villes. Il suffit alors de parcourir une seule fois la liste de villes pour faire le comptage.

Enfin, on remplit l'histogramme des noms pour associer chaque occurrence au nombre de villes qui ont cette occurrence (en nom dans un premier temps).

Il ne faut néanmoins pas oublier de créer un ordre sur les villes comme fait dans le fichier *towns.h*. Pour plus tard, on crée également un ordre sur les *Point2D* (cf *operator <* et *operator =* dans *point2d.h*).

```
/home/amine/.CLion2016.2/system/cmake/generated/villes2-c368ef67/c368ef67/Release/isotopo
Reading town file: /home/amine/Documents/2A/Prog Av/TP6/code/villes2/villes2/villes.txt
File read in: 0.072341 s
Number of towns in file: 35180
A random town, using format "name[lat,lon](x,y)": Brassempouy[43.6333,-0.7](401.437,6288.58)
Number of towns with a name occurrence of 1 is 31665
Number of towns with a name occurrence of 2 is 1083
Number of towns with a name occurrence of 3 is 200
Number of towns with a name occurrence of 4 is 87
Number of towns with a name occurrence of 5 is 39
Number of towns with a name occurrence of 6 is 13
Number of towns with a name occurrence of 7 is 10
Number of towns with a name occurrence of 8 is 5
Number of towns with a name occurrence of 9 is 2
```

## Histogramme des occurrences de coordonnées

De même que pour la première section, on crée une *map* cette fois ci liant un *Point2D* à son occurrence. C'est à dire, des coordonnées au nombre de villes ayant les mêmes coordonnées. On remplit de la même manière qu'en section 1, l'histogramme des coordonnées, et on trouve l'histogramme suivant pour le fichier *villes.txt* :

```
#####  
  
Number of towns with a coords' occurrence of 1 is 31665  
Number of towns with a coords' occurrence of 2 is 1083  
Number of towns with a coords' occurrence of 3 is 200  
Number of towns with a coords' occurrence of 4 is 87  
Number of towns with a coords' occurrence of 5 is 39  
Number of towns with a coords' occurrence of 6 is 13  
Number of towns with a coords' occurrence of 7 is 10  
Number of towns with a coords' occurrence of 8 is 5  
Number of towns with a coords' occurrence of 9 is 2  
  
#####
```

## Création des ensembles N et C

Pour créer les ensembles N et C (*set* < *Town* >), il suffit de parcourir la liste des villes une seule fois également.

En effet, si on trouve une occurrence de nom (respectivement de coordonnées) qui est strictement plus grande que 1 dans nos tableaux associatifs (*names\_occurrence* et *coords\_occurrence*), et l'insérer dans l'ensemble respectif.

## Création de l'intersection $N \cap C$

Pour trouver l'intersection des deux ensembles et donc l'ensemble des villes ayant un nom d'occurrence au moins 2 et des coordonnées d'occurrence également strictement plus grande que 1, on suit les étapes suivantes :

- On crée un vecteur de villes de taille égale au minimum des cardinaux de N et C.
- On effectue l'intersection en calculant le pointeur de  $N \cap C$  grâce à la fonction *set\_intersection* de la STL.
- Enfin, on coupe le vecteur pour enlever les villes vides (constructeur vide) jusqu'à avoir le bon nombre d'éléments.

On réduit ainsi le nombre d'éléments de :

$$\frac{Card(N \cap C)}{Card(towns)}$$

La compilation du programme joint au rapport nous donne alors le résultat suivant :

```
#####  
Number of Towns that have same name as other Towns : 3515  
#####  
Number of Towns that have same coordinates as other Towns : 2150  
#####  
Number of Towns that have same name as another Town and same coordinates as another one : 207  
#####
```

Soit un ensemble environ mille fois moins grand que l'ensemble initial.

$$\frac{207}{35181} \approx 5.8 * 10^{-3}$$

## Recherche de villes "ambigues"

Dans cette section, on cherche les villes  $v_1$  telles que :

$$\begin{aligned} \exists v_2, v_3, v_4 \in Towns, \quad coord(v_1) = coord v_2, \quad name(v_1) = name(v_3) \\ coord(v_3) = coord(v_4), \quad name(v_2) = name(v_4) \end{aligned}$$

C'est à dire qu'on cherche les villes dont on peut se tromper en en parlant, car elles possèdent des homonymes (villes de même nom) et des isotopes (villes de même coordonnées).

Pour cela, on parcourt quatre fois la liste des villes pour trouver 4 villes liées par les relations précédentes. Heureusement, on n'a pas à parcourir notre ensemble de départ à 35181, il suffit de chercher nos isotoponymes dans  $N \cap C$ . Aisni :

```
#####  
  
Number of town we can be wrong about we are comparing names and coords is 0  
Found tuples of Towns in N intersected with C in : 0.109083 seconds.  
  
#####  
  
End of program !  
  
Process finished with exit code 0
```

Dans le fichier initial, on ne trouve aucun quadruple de villes qui correspond aux quatre conditions.

En ajoutant quelques villes dans un fichier *villes2.txt* comme suit :

```
35181    Foo 7.1 7.1
35182    Bar 7.1 7.1
35183    Foo 2.55 2.55
35184    Bar 2.55 2.55
35185
```

Le compilateur fournit bien le nombre de villes dites "ambigues" :

```
#####

Number of towns with a coords' occurrence of 1 is 31657
Number of towns with a coords' occurrence of 2 is 1086
Number of towns with a coords' occurrence of 3 is 202
Number of towns with a coords' occurrence of 4 is 87
Number of towns with a coords' occurrence of 5 is 39
Number of towns with a coords' occurrence of 6 is 13
Number of towns with a coords' occurrence of 7 is 10
Number of towns with a coords' occurrence of 8 is 5
Number of towns with a coords' occurrence of 9 is 2

#####

Number of Towns that have same name as other Towns : 3527

#####

Number of Towns that have same coordinates as other Towns : 2154

#####

Number of Towns that have same name as another Town and same coordinates as another one : 211

#####

Number of town we can be wrong about we are comparing names and coords is 4
Found tuples of Towns in N intersected with C in : 0.120643 seconds.

#####

End of program !

Process finished with exit code 0
```

Avec un résultat en temps de

$$\tau \approx 0.12s$$

Comparé à quatre boucles sur l'ensemble total des villes, on a :

$$\frac{Card(N \cap C)^4}{Card(villes)^4} \approx 10^{-10}$$

## Conclusion

La comparaison de notre algorithme avec un algorithme naïf prouve son efficacité. En effet, on a réussi grâce aux fonctions d'intersections implémentées sur C++, à réduire un problème de très grande dimension en un problème très simple en réduisant l'ensemble d'étude et de recherche des isotoponymes.

## Références

- Slides du cours de Renaud MARLET, ENPC - MOPSI (2016-2017)
- <http://stackoverflow.com/questions/7666509/hash-function-for-string>