

**Projet : Arbre couvrant de poids minimum**

L'objectif de ce projet est de développer un code optimisé pour calculer *le réseau le plus court*<sup>1</sup> reliant des villes françaises, ainsi que la taille de ce réseau. Pour ce faire, on utilisera l'algorithme de Prim sur un graphe initial dont les nœuds sont les villes, les arêtes sont des connexions entre les villes, et le poids associé à chaque arête est la distance entre les villes adjacentes. Initialement, on considère que toutes les villes sont reliées (*i.e.* le graphe initial est complet).

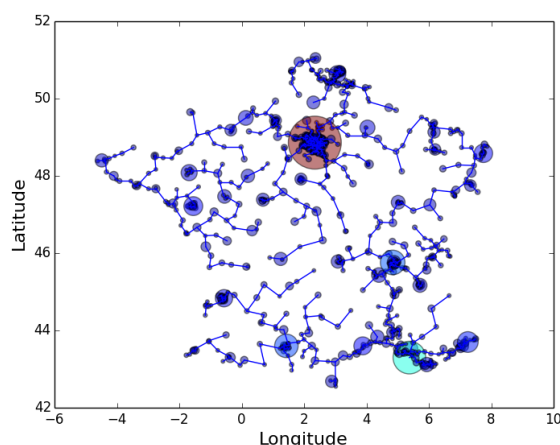


Figure 1: Exemple de réseau le plus court. La taille de chaque bulle correspond à la population. Cette image a été générée avec le script `visualisation.py`.

Pour démarrer ce projet, on fournit :

- Une base de données<sup>2</sup> avec les villes françaises de métropole et leurs caractéristiques. (`listeVilles.csv`)
- Une fonction C++ pour lire les latitudes/longitudes (en degrés) des villes françaises (de  $N$  habitants ou plus) à partir la base de donnée, et pour écrire ces coordonnées dans un autre fichier pour la visualisation. (`lectureVilles.h` et `lectureVilles.cpp`)
- Un script Python pour générer des images avec la position des villes et le graphe obtenu. (`visualisation.py`)
- Un fichier C++ de départ. (`main.cpp`)

Pour comprendre le fonctionnement des fichiers fournis, testez :

```
>> g++ main.cpp lectureVilles.cpp
>> ./a.out
Population minimale ? 250000
...
>> python visualisation.py
```

<sup>1</sup>Plus rigoureusement : un des réseaux les plus courts.

<sup>2</sup>Ce fichier est une version adaptée/modifiée d'un fichier téléchargé à l'adresse: <http://sql.sh/736-base-donnees-villes-francaises>

La distance entre deux villes ( $a$  et  $b$ ) est calculée avec la formule:

$$R \cdot \text{acos}\left(\sin(\text{lat}_a) \sin(\text{lat}_b) + \cos(\text{lon}_a - \text{lon}_b) \cos(\text{lat}_a) \cos(\text{lat}_b)\right)$$

avec  $R \approx 6378\text{km}$ .

Dans ce projet, on considère deux variantes :

1. Le réseau le plus court reliant les villes de  $N$  habitants ou plus (*en utilisant l'algorithme de Prim sur toutes les villes de  $N$  habitants ou plus*).
2. Le réseau obtenu en combinant (1) le réseau national reliant la plus grande ville de chaque département aux plus grandes villes des autres départements (*en utilisant l'algorithme de Prim sur ces grandes villes uniquement*) et (2) les réseaux départementaux reliant les villes de  $N$  habitants ou plus de chaque département (*en utilisant l'algorithme de Prim département par département*).

Pour chacune des variantes, vous devez :

1. Implémenter les algorithmes et calculer la taille du réseau (*il est obtenu additionnant le poids des arêtes du graphe final*). Valider les algorithmes avec un petit nombre de villes.
2. Optimiser au maximum vos implémentations (pour un grand nombre de villes). En particulier, analysez la possibilité (ou non) de vectoriser et/ou de paralléliser les boucles de votre implémentation. Utilisez les rapports de compilation pour alimenter votre analyse.
3. Étudiez la pertinence des optimisations principales en estimant le gain de rapidité avec chacune de ces optimisations (pour un grand nombre de villes). Ensuite, étudiez les performances de votre implémentation en fonction du nombre de villes.

Le code final doit :

1. Contenir la commande de compilation, en commentaire, au dessus du fichier principal (`main.cpp`);
2. Demander la variante et le population minimale  $N$  des villes à considérer;
3. Écrire le fichier `resuGraphe.dat` avec la liste des arêtes, et `resuVilles.dat` avec la population et les coordonnées des villes;
4. Afficher le temps d'exécution (*les parties lecture/écriture de fichiers sont ignorées*).

Votre rapport doit couvrir toutes les parties du projet, et en particulier traiter des aspects 'performance' du projet. Les codes finaux (*nettoyés !*) et le rapport doivent être envoyés par e-mail ([adrien.loseille@inria.fr](mailto:adrien.loseille@inria.fr) et [axel.modave@ensta-paristech.fr](mailto:axel.modave@ensta-paristech.fr)) au plus tard le **lundi 30 avril (midi)**. Une soutenance aura lieu le **mercredi 2 mai**.