

# Algorithmique

## Activité algorithmes de tri

### Trier des villes

Il vous est demandé aujourd'hui d'analyser par groupes de 4 élèves 8 algorithmes de tri. L'objectif étant de trier des villes en fonction de leur distance par rapport à Grenoble. Cette classification est par exemple très utile lors de la réalisation d'une application de cartographie.

Pour ce faire, un jeu de données contenant les différentes villes et villages de France ainsi que leurs coordonnées GPS est disponible sur le site [data.gouv.fr](https://data.gouv.fr).

Un code minimal permettant de charger le fichier et d'afficher la distance entre les différentes villes est fourni : [GitHub](#)

Ce code minimal contient 3 jeux de données pour tester votre algorithme :

1. Small.csv : 28 premiers échantillons des villes de France
2. Isere.csv : les 537 communes de l'Isère
3. Fr.csv : les 36850 communes françaises

### Un peu de maths

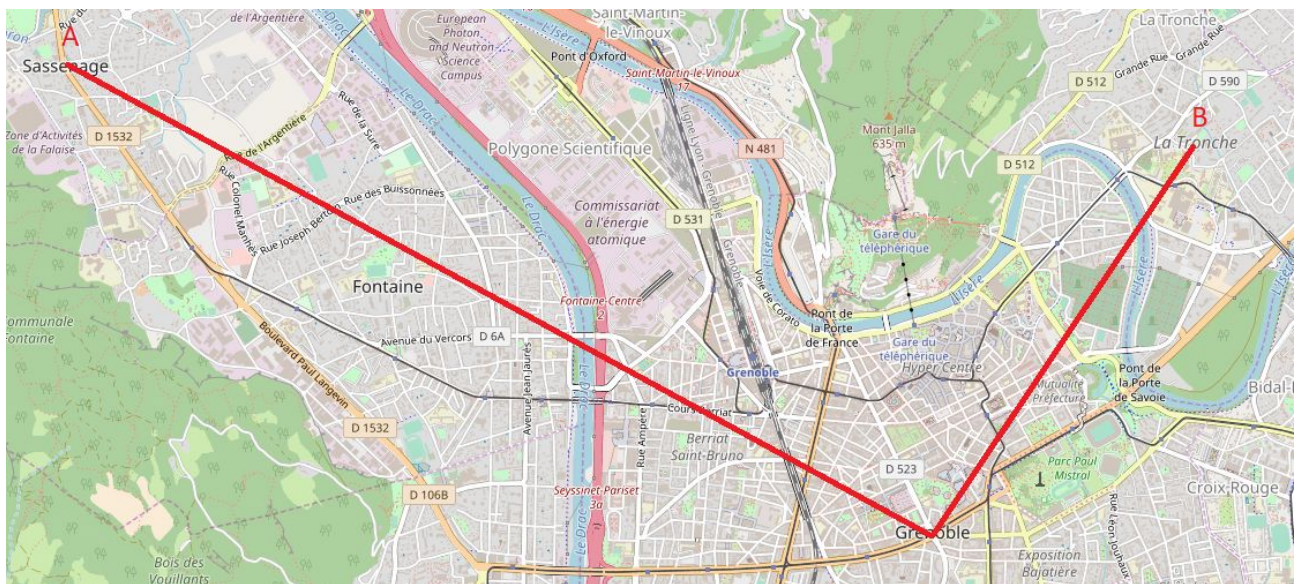
Avant d'aborder la question du tri d'un tableau, il est nécessaire d'aborder la question de la comparaison entre 2 éléments du tableau.

L'élément A est-il plus grand que l'élément B ?

## A > B ?

Dans le contexte d'un tri de distance, la question à se poser est : Ma ville A est elle plus proche de Grenoble que ma ville B ? Ou encore, la distance entre A et Grenoble est elle supérieure à la distance entre B et Grenoble ?

$\text{dist}(A, \text{Grenoble}) > \text{dist}(B, \text{Grenoble})$  ?



Pour répondre à cette question, vous pouvez utiliser la ressource suivante :

<http://www.movable-type.co.uk/scripts/latlong.html>

Ecrire la fonction **distanceFromGrenoble(city)** qui calcule la distance à Grenoble

Ecrire la fonction **isLess(A,B)** qui retourne vrai si i est plus prêt de Grenoble que j.

⚠ La distance ne doit pas forcément être calculée avec une très grande précision

## Permutation

Écrivez la fonction **swap(A,B)** qui permet de permuter 2 éléments d'un tableau.

## Algorithmes de tri

Voici les différents algorithmes de tri à implémenter :

<https://www.toptal.com/developers/sorting-algorithms>

- Tri par insertion
  - [Animation et algorithme](#)
  - [Danse roumaine](#)
- Tri par sélection
  - [Animation et algorithme](#)
  - [Danse gitane](#)
- Tri à bulles
  - [Animation et algorithme](#)
  - [Danse hongroise](#)
- Tri de Shell
  - [Animation et algorithme](#)
  - [Danse hongroise](#)
- Tri par fusion
  - [Animation et algorithme](#)
  - [Danse transylvanienne](#)
- Tri par tas
  - [Animation et algorithme](#)
  - [Danse hongroise](#)
- Tri rapide
  - [Animation et algorithme](#)
  - [Danse hongroise](#)
- Tri rapide (3 partitions)
  - [Animation et algorithme](#)

## Pour aller plus loin

Pour en savoir plus :

- [Un mémoire d'étudiant des arts et métiers](#)

Vous pouvez maintenant tester la distance routière entre 2 villes en utilisant une API :

- [Open Source Routing Machine](#)
- [Google Distance Matrix](#)