



# L3 LFTI - Exercices d'algorithmique : permutations

Karën Fort

21 février 2017



**Important** : vous devez écrire ce qui vous est demandé en Java et rendre un fichier nommé TD2\_NomPrenom.java contenant les méthodes demandées dans les exercices ainsi qu'une méthode *main* qui permettra d'exécuter votre code sur un exemple. Vous pouvez utiliser Eclipse (<https://www.eclipse.org/>) ou NetBeans (<https://netbeans.org/>) ou n'importe quel logiciel de votre préférence, mais la responsable de TD ne l'installera et ne le configurera pas pour vous.

**Attention** : tout plagiat (notamment, sur Internet) ou copie directe sur le voisin ou la voisine sera sanctionné par un 0. Les communications entre vous sont autorisées, mais le code doit être le vôtre.

## 1 Algorithme de Narayana

### 1.1 Finir d'implémenter le voyageur de commerce

Rappel concernant le problème du voyageur de commerce :

« étant donné une liste de villes, et des distances entre toutes les paires de villes, déterminez un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ. » [Wikipédia, consultée le 22 février 2016]

Étant donné le code qui vous est fourni (voir sur le site <http://www.schplaf.org/kf/Enseignement.php>) :

1. ajoutez le calcul de la distance parcourue pendant le voyage total
2. ajoutez le calcul du minimum de la distance (vous pouvez pour cela vous inspirer du TD précédent)
3. affichez la liste des villes parcourues (en une distance minimale), dans le bon ordre

### 1.2 Ajouter 2 villes

Ajouter Grenoble et Tours à la liste des villes. Les distances sont prises sur Google maps (distance à pied).

Adaptez le code si nécessaire.

Combien de temps cela prend-il en plus ?

## 2 Bonus : algorithme de Heap

Implémentez l'algorithme de Heap pour le voyageur de commerce tel que proposé.