

## Titre : Voyageur de commerce

### Sujet :

Le [problème du voyageur de commerce](#) possède plusieurs variantes. Nous nous intéressons ici au problème symétrique : étant donné un graphe complet de  $n$  nœuds, dont la distance entre deux nœuds  $i$  et  $j$  est la même qu'entre les nœuds  $j$  et  $i$ , quel est le plus court chemin qui passe par tous les nœuds exactement 1 fois.

C'est un problème qui est beaucoup étudié, et plusieurs instances du problème sont disponibles. Ces instances sont représentées sous un format textuel : TSPLIB.

Voici deux exemples de fichiers TSP :

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/bayg29.tsp>

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/att48.tsp>

Le premier contient une matrice d'adjacence, le second n'indique pas la distance entre les nœuds, il faut donc la calculer à partir des coordonnées des nœuds.

Dans ces deux fichiers, le nombre de nœuds du graphe est représenté par la `DIMENSION` de l'instance. La section `EDGE_WEIGHT_SECTION` donne la distance entre chaque nœud sous la forme :

```
distance_1_2    distance_1_3    ...    distance_1_DIMENSION
distance_2_3    distance_2_4    ...    distance_2_DIMENSION
...
distance_DIMENSION-1_DIMENSION
```

Les sections `DISPLAY_DATA_SECTION` et `NODE_COORD_SECTION` donnent les coordonnées des nœuds sous la forme :

```
numéro_du_noeud  x      y
```

La solution à une instance est donc une séquence des nœuds du graphe, par exemple, une solution de la première instance :

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/bayg29.opt.tour>

### Travail demandé :

Développer un programme qui puisse lire des fichiers d'instances TSPLIB, et trouver une solution. Votre programme devra sauvegarder la solution dans un fichier en indiquant la séquence trouvée. Vous afficherez graphiquement le résultat de votre recherche.

Votre programme devra essayer pour chaque instance de trouver la meilleure solution en compétition avec les autres binômes.

**Langages :** python, C++