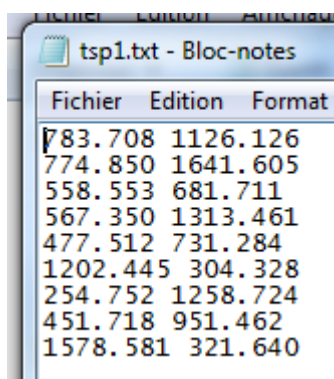


# Complexité et Récursivité

TP1 : 15 Juin 2021

## Problème du voyageur de commerce (TSP).

Il s'agit d'expérimenter le comportement d'un algorithme de recherche exhaustive, si naïf soit-il, sur de petites instances de TSP (Travel Salesman Problem) euclidiens. Vous devez écrire une **fonction** récursive d'énumération qui donnera en sortie le circuit hamiltonien le plus court (et sa longueur) qu'elle trouvera en 10 secondes, 60 secondes et 600 secondes. Une instance est définie par un fichier texte de N lignes (pour N villes) décrivant leurs coordonnées planaires (x,y). Par exemple ; le fichier **tsp1.txt** contient les 9 lignes de texte suivantes :



```
tsp1.txt - Bloc-notes
Fichier  Edition  Format
783.708 1126.126
774.850 1641.605
558.553 681.711
567.350 1313.461
477.512 731.284
1202.445 304.328
254.752 1258.724
451.718 951.462
1578.581 321.640
```

Vous disposez de **10 instances (tsp1.txt à tsp10.txt)** pour tester votre programme. Pour chacune des instances vous devez indiquer les longueurs minimales des tours en fonction du temps cpu. Pratiquement vous devez remplir le tableau suivant :

Instance	Nb villes	Cpu	Longueur	Cpu	Longueur	Cpu	Longueur
tsp1.txt	9	10		60		600	
...		...		...		600	
tsp10.txt		10		60		600	

**Longueur** =

$$\sum_{i=1}^{n-1} \sqrt{(x_{\pi_i} - x_{\pi_{i+1}})^2 + (y_{\pi_i} - y_{\pi_{i+1}})^2} + \sqrt{(x_{\pi_n} - x_{\pi_1})^2 + (y_{\pi_n} - y_{\pi_1})^2}$$

où  $\pi$  représente la « meilleure » permutation des n villes d'une instance donnée.

**Cpu** = temps d'exécution en secondes.

**Vous pouvez travailler par équipe de deux ou trois. Vous devez également me transmettre le fichier source de votre code (C/C++ ou Python) commençant par les lignes de commentaires indiquant les noms et prénoms des membres de l'équipe.**