

```
import random as rd
import matplotlib.pyplot as plt
import numpy as np
```

N= 40

Génération aléatoire de N couples de coordonnées cartésiennes
pour N villes dans un carré de côté unitaire

Calcul de la matrice symétrique NxN des distances intervalles

```
print(matricedistances)
```

```
def trajet(seqvoyage,matricedistances):
```

Définition d'une fonction trajet d'arguments une séquence de villes et la
matrice des distances qui retournant la longueur du trajet fermé

```
return longueur
```

```
def modifseq(seqvoyage):
```

Définition d'une fonction qui retournera une
nouvelle séquence de voyage (nouvseqvoyage)
à partir d'une séquence précédente (seqvoyage)
par substitution et renversement de deux
couples de villes successives

```
return nouvseqvoyage
```

```
essais=50000
```

```
T=0.2
```

```
T0=T
```

```
listelongueurs=[]
```

```
temperatures=[T0]
```

```
coeff=0.9999
```

```
seqvoyageinit=list(range(N))
```

```
seqvoyage=seqvoyageinit
```

```
for e in range(1,essais):
```

Procédure de recuit simulé itérée sur 5000 essais utilisant les deux
fonctions précédentes et la variable température décroissante
(On notera :
newseq la nouvelle séquence
newlong la nouvelle longueur
long l'ancienne longueur de la séquence seqvoyage
ecart l'écart)

liste des températures
et décroissance itérative

```
plt.figure(1)
plt.plot(listelongueurs)
plt.twinx()
plt.ylabel("Température", color="r")
plt.plot(temperatures, "r-", linewidth=2)
plt.figure(2)
plt.subplot(1,2,1)
x=[absc[seqvoyageinit[i]] for i in range(N)]+[absc[seqvoyage[0]]]
y=[ordo[seqvoyageinit[i]] for i in range(N)]+[ordo[seqvoyage[0]]]
plt.plot(x,y,"r.")
plt.plot(x,y,"k-")
plt.subplot(1,2,2)
x=[absc[seqvoyage[i]] for i in range(N)]+[absc[seqvoyage[0]]]
y=[ordo[seqvoyage[i]] for i in range(N)]+[ordo[seqvoyage[0]]]
plt.plot(x,y,"r.")
plt.plot(x,y,"g-")
plt.show()
```