

```

# TP6.py

0001| #Exercice n°1
0002| from math import sqrt
0003|
0004| def distance(p1,p2):
0005|     x1,y1=p1
0006|     x2,y2=p2
0007|     return sqrt((x1-x2)**2+(y1-y2)**2)
0008|
0009| def distances(pts,dep):
0010|     n=len(pts)
0011|     tab=[(n+1)*[0] for i in range(n+1)]
0012|     for i in range(n):
0013|         for j in range(i):
0014|             tab[i][j]=distance(pts[i],pts[j])
0015|             tab[j][i]=tab[i][j]
0016|             tab[n][i]=distance(dep,pts[i])
0017|             tab[i][n]=tab[n][i]
0018|     return tab
0019|
0020| def longueur(chemin,dist):
0021|     d=0
0022|     id_pt=len(dist)-1
0023|     for point in chemin:
0024|         d+=dist[id_pt][point]
0025|         id_pt=point
0026|     return d
0027|
0028| def indice(position, dist,dispo):
0029|     n=len(dist)-1
0030|     mini=3*dim # supérieur à la diagonale du carré
0031|     for i in range(n):
0032|         if dispo[i]:
0033|             d=dist[position][i]
0034|             if d<mini:
0035|                 mini=d
0036|                 ind=i
0037|     return ind
0038|
0039| def plus_court(dist):
0040|     n=len(dist)-1
0041|     chemin=[]
0042|     dispo=[True]
0043|     position=n
0044|     while len(chemin)<n:
0045|         position=indice(position,dist,dispo)
0046|         chemin.append(position)
0047|         dispo[position]=False
0048|     return chemin
0049|
0050| from random import randint
0051|
0052| nbpoints,dim=10,20
0053|
0054| def points(n,c):
0055|     liste=[]
0056|     while len(liste)<n:
0057|         x=randint(-c,c)
0058|         y=randint(-c,c)
0059|         if [x,y] not in liste:
0060|             liste.append([x,y])
0061|     return liste
0062|
0063| depart= [randint(-dim,dim),randint(-dim,dim)]
0064|
0065| import matplotlib.pyplot as plt
0066|
0067| pts=points(nbpoints,dim)
0068| tableau=distances(pts,depart)
0069| ch=plus_court(tableau)
0070| print(longueur(ch,tableau))
0071|
0072| x=[u[0] for u in pts]
0073| y=[u[1] for u in pts]
0074| plt.plot(x,y,"x")
0075| plt.plot(depart[0],depart[1],"*")
0076| xliste=[depart[0]]+[pts[k][0] for k in ch]
0077| yliste=[depart[1]]+[pts[k][1] for k in ch]
0078| plt.plot(xliste,yliste) # affichage du chemin
0079| plt.show()
0080|
0081|
0082| # Exercice n°2
0083| # Version itérative
0084| def glouton(distances,dmax):
0085|     n=len(distances)
0086|     d=dmax
0087|     stations=[]
0088|     i=0
0089|     while i!=n:
0090|         while i<n and distances[i]<=d:
0091|             d=d-distances[i]
0092|             i=i+1
0093|         stations.append(i-1)
0094|         d=dmax
0095|     return (stations)
0096|
0097| from random import randint
0098|
0099| trajet=2300
0100| tab=[randint(25,100)] # distance première station
0101| while sum(tab)<trajet:
0102|     tab.append(randint(25,100)) # distances interstations
0103| tab[len(tab)-1]=trajet-(sum(tab)-tab[len(tab)-1]) # dernière distance
0104| res=600
0105|
0106| #print(glouton(tab,res))
0107|
0108| # Version récursive
0109| def choix(distances,dmax,i,stations,d):
0110|     n=len(distances)
0111|     if i!=n:
0112|         if i<n and distances[i]<=d:
0113|             choix(distances,dmax,i+1,stations,d-distances[i])
0114|         else:
0115|             stations.append(i-1)
0116|             choix(distances,dmax,i,stations,dmax)
0117|
0118| def stats(distances,dmax):
0119|     stations=[]

```

```
120|     choix(distances,dmax,0,stations,dmax)
121|     return stations
122|
123| print(stats(tab,res))
124|
125|
126|
```