Exercise 7:

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy import stats
import networkx as nx
import random
from src.my_random.sales import*
```

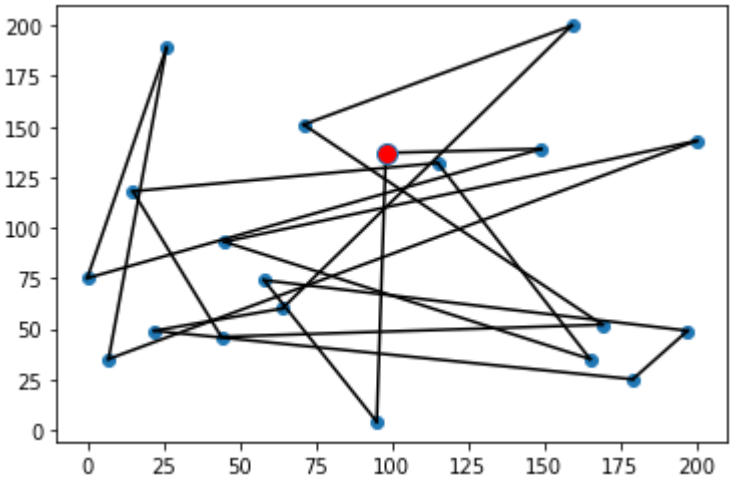
a)

In [ ]:
```
n = 20
stations = gen_stations(n)
route = np.arange(0,n)
random.shuffle(route)
cost = euclDist(stations[:,route[n-1]],stations[:,route[0]])
```

In [ ]:
```
plt.figure()
plt.plot([stations[0,route[0]],stations[0,route[n-1]]],[stations[1,route[0]],s
for i in range(n-1):
    cost += euclDist(stations[:,route[i]],stations[:,route[i+1]])
    plt.plot([stations[0,route[i]],stations[0,route[i+1]]],[stations[1,route[i
plt.scatter(stations[0,:],stations[1,:])
plt.plot(stations[0,0],stations[1,0],marker='o',markerfacecolor='red',markersi

print('Total cost of this route:',cost)
```

```
Total cost of this route: 2410.3705531657756
```



b)

In [ ]:
```
df = pd.read_csv(r'C:/Users/lenovo/Documents/DTU/02443/cost.csv',header=None)
df
```

Out[ ]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	225	110	8	257	22	83	231	277	243	94	30	4	265	274	250	87	83
1	255	0	265	248	103	280	236	91	3	87	274	265	236	8	24	95	247	259
2	87	236	0	95	248	110	25	274	250	271	9	244	83	250	248	280	29	26
3	8	280	83	0	236	28	91	239	280	259	103	23	6	280	244	259	95	87
4	268	87	239	271	0	244	275	9	84	25	244	239	275	83	110	24	274	280
5	21	265	99	29	259	0	99	230	265	271	87	5	22	239	236	250	87	95
6	95	236	28	91	247	93	0	247	259	244	27	91	87	268	275	280	7	8
7	280	83	250	261	4	239	230	0	103	24	239	261	271	95	87	21	274	255
8	247	9	280	274	84	255	259	99	0	87	255	274	280	3	27	83	259	244
9	230	103	268	275	23	244	264	28	83	0	268	275	261	91	95	8	277	261
10	87	239	9	103	261	110	29	255	239	261	0	259	84	239	261	242	24	25
11	30	255	95	30	247	4	87	274	242	255	99	0	24	280	274	259	91	83
12	8	261	83	6	255	29	103	261	247	242	110	29	0	261	244	230	87	84
13	242	8	259	280	99	242	244	99	3	84	280	236	259	0	27	95	274	261
14	274	22	250	236	83	261	247	103	22	91	250	236	261	25	0	103	255	261
15	244	91	261	255	28	236	261	29	103	9	242	261	244	87	110	0	242	236
16	84	236	27	99	230	83	7	259	230	230	22	87	93	250	255	247	0	9
17	91	242	28	87	250	110	6	271	271	255	27	103	84	250	271	244	5	0
18	261	24	250	271	84	255	261	87	28	110	250	248	248	22	3	103	271	248
19	103	271	8	91	255	91	21	271	236	271	7	250	83	247	250	271	22	27

In [ ]:

```
w = np.zeros(n)
for l in range(n):
    w[l] = np.sum(df.values[l,:])

e = list(range(0,n,1))
pointA = random.choices(e,weights=1/w)
print('Route:',route)
indexA = np.where(route==pointA)[0]
print('A:',pointA,', index:',indexA)
```

```
Route: [ 9 11  6  8  4 15  1 19 12 17 14 16  2 10  7  5 18 13  3  0]
A: [0] , index: [19]
```

In [ ]:

```
n = 20
iterations = 1000
permutations = 2
stations = gen_stations(n)
route = np.arange(0,n)
optRoute = route
random.shuffle(route)
k = 0
costMin = np.sum(np.sum(df.values))
    l=1/np.sqrt(1+k)
```

```
    #T=-np.log(1+k)
    k+=1
    for j in range(permutations):
        w = np.zeros(n)
        e = list(range(0,n,1))
        for l in range(n):
            w[l] = np.sum(df.values[l,:])
        pointA = random.choices(e,weights=T/w)
        pointB = random.choices(e,weights=T/df.values[:,pointA])
        indexA = np.where(route==pointA)[0]
        indexB = np.where(route==pointB)[0]
        route[indexA], route[indexB] = route[indexB], route[indexA]

    cost = df.values[route[len(route)-1]][route[0]]
    for i in range(len(route)-1):
        cost += df.values[route[i]][route[i+1]]
    if cost<costMin:
        costMin = cost
        optRoute = route
    else:
        route = optRoute


print('The estimated optimal route follows this sequence:', optRoute,'and enta
plt.figure()
plt.plot([stations[0,route[0]],stations[0,route[n-1]]],[stations[1,route[0]],s
for i in range(n-1):
    plt.plot([stations[0,route[i]],stations[0,route[i+1]]],[stations[1,route[i
plt.scatter(stations[0,:],stations[1,:])
plt.plot(stations[0,0],stations[1,0],marker='o',markerfacecolor='red',markersi
```

```
<ipython-input-66-b04bce06d701>:20: RuntimeWarning: divide by zero encountered
in true_divide
  pointB = random.choices(e,weights=T/df.values[:,pointA])
```

```
The estimated optimal route follows this sequence: [ 0  4  3 12  7 10  8  2 18
11 15 13  6 17  9 14  5 19 16  1] and entails the following cost: 1912
```

Out[ ]:

```
[<matplotlib.lines.Line2D at 0x1bef6a389d0>]
```