

Tina Borundia

Roll No : 16

Practical 2

```
In [1]: pip install geopy
```

Requirement already satisfied: geopy in c:\users\hp\anaconda3\lib\site-packages (2.4.1)
Requirement already satisfied: geographiclib<3,>=1.52 in c:\users\hp\anaconda3\lib\site-packages (from geopy) (2.0)
Note: you may need to restart the kernel to use updated packages.

```
In [1]: from geopy.geocoders import Nominatim  
from geopy.distance import geodesic as GD
```

```
In [2]: '''A. Consider the following for deciding connections in same state in India:  
i. Find the latitude and longitude of cities in same state. Consider 4 to 6 cities.  
ii. Calculate the cost of connecting each pair of offices by computing the distance  
between different pair of different cities (as considered in part A) and construct a  
fully connected graph.  
iii. Compute a minimum spanning tree using either Prims or Kruskals Method to find the  
cost of connecting offices in different cities.'''
```

```
Out[2]: 'A. Consider the following for deciding connections in same state in India:\n  
i. Find the latitude and longitude of cities in same state. Consider 4 to 6 cities.\n  
ii. Calculate the cost of connecting each pair of offices by computing the distance\nbetween different pair of different cities (as considered in part A) and construct a\nfully connected graph.\n  
iii. Compute a minimum spanning tree using either Prims or Kruskals Method to find the\ncost of connecting offices in different cities.'
```

```
In [3]: geolocator=Nominatim(user_agent="PrimsAlgo")
```

```
In [4]: print("Enter number of cities ")
n=int(input())
cities=[]
print('Enter city')
for i in range(n):
    city=input()
    cities.append(city)
```

```
Enter number of cities
5
Enter city
Nagpur
Pune
Amravati
Nashik
Yavatmal
```

```
In [5]: longitude=[]
latitude=[]
matrix=[]
for i in range(n):
    location=geolocator.geocode(cities[i])
    longitude.append(location.longitude)
    latitude.append(location.latitude)
for j in range(n):
    matrix.append((latitude[j],longitude[j]))
```

```
In [6]: print(longitude)
```

```
[79.0820556, 73.8544541, 77.64429617998744, 73.7902364, 78.35]
```

```
In [7]: print(latitude)
```

```
[21.1498134, 18.521428, 21.15454115, 20.0112475, 20.15]
```

```
In [8]: print(matrix)
```

```
[(21.1498134, 79.0820556), (18.521428, 73.8544541), (21.15454115, 77.64429617998744), (20.0112475, 73.7902364), (20.15, 78.35)]
```

```
In [9]: pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site-packages (1.24.3)
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
In [10]: import numpy as np
out=[]
for i in range(n):
    dist=[]
    for j in range(n):
        d=(GD(matrix[i],matrix[j])).miles)
        dist.append(d)
    out.append(dist)
arr=np.array(out)
res=arr.reshape(n,n)
print(res)
```

```
[[ 0.          385.27630868  92.79109382 351.63117602  83.53308215]
 [385.27630868  0.          306.00975166 102.55989455 314.15629697]
 [ 92.79109382 306.00975166  0.          261.75754236  82.84746109]
 [351.63117602 102.55989455 261.75754236  0.          296.49028854]
 [ 83.53308215 314.15629697  82.84746109 296.49028854  0.          ]]
```

```
In [11]: INF=9999
min_dist=[]
V=n
selected=np.zeros(n)
no_edge=0
selected[0]=True
add=0
while(no_edge< V-1):
    minimum=INF
    x=0
    y=0
    for m in range(V):
        if selected[m]:
            for n in range(V):
                if((not selected[n]) and out[m][n]):
                    if minimum > out[m][n]:
                        minimum=out[m][n]
                        min_dist.append(minimum)
                        x=m
                        y=n
    print(str(x)+"-"+str(y)+":"+str(out[x][y]))
    add=add+out[x][y]
    selected[y]=True
    no_edge+=1
print("Minimum Dintance is : ",add,"miles")
```

```
0-4:83.53308214561304
4-2:82.8474610857538
2-3:261.7575423593927
3-1:102.55989455322408
Minimum Dintance is :  530.6979801439836 miles
```

In [12]: `pip install pandas`

Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-packages (2.0.3)
 Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\appdata\roaming\python\python311\site-packages (from pandas) (2.8.2)
 Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2023.3.post1)
 Requirement already satisfied: tzdata>=2022.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (2023.3)
 Requirement already satisfied: numpy>=1.21.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas) (1.24.3)
 Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\roaming\python\python311\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
 Note: you may need to restart the kernel to use updated packages.

In [3]: `'''B. Repeat the above for cities in different states.'''`

Out[3]: 'B. Repeat the above for cities in different states.'

In [19]: `print("Enter number of cities of different states ")
 n=int(input())
 cities=[]
 print('Enter city')
 for i in range(n):
 city=input()
 cities.append(city)`

Enter number of cities of different states
 5
 Enter city
 Bhopal
 Ranchi
 Kolkata
 Mumbai
 Patna

In [20]: `longitude=[]
 latitude=[]
 matrix=[]
 for i in range(n):
 location=geolocator.geocode(cities[i])
 longitude.append(location.longitude)
 latitude.append(location.latitude)
 for j in range(n):
 matrix.append((latitude[j],longitude[j]))`

In [21]: `print(longitude)`

[77.401989, 85.3250387, 88.3638953, 72.878176, 85.1235252]

In [22]: `print(latitude)`

[23.2584857, 23.3700501, 22.5726459, 19.0785451, 25.6093239]

In [23]: `print(matrix)`

```
[(23.2584857, 77.401989), (23.3700501, 85.3250387), (22.5726459, 88.3638953),
(19.0785451, 72.878176), (25.6093239, 85.1235252)]
```

In [24]:

```
longitude=[]
latitude=[]
matrix=[]
for i in range(n):
    location=geolocator.geocode(cities[i])
    longitude.append(location.longitude)
    latitude.append(location.latitude)
for j in range(n):
    matrix.append((latitude[j],longitude[j]))
```

In [25]:

```
INF=9999
min_dist=[]
V=n
selected=np.zeros(n)
no_edge=0
selected[0]=True
add=0
while(no_edge< V-1):
    minimum=INF
    x=0
    y=0
    for m in range(V):
        if selected[m]:
            for n in range(V):
                if((not selected[n]) and out[m][n]):
                    if minimum > out[m][n]:
                        minimum=out[m][n]
                        min_dist.append(minimum)
                        x=m
                        y=n
    print(str(x)+"-"+str(y)+":"+str(out[x][y]))
    add=add+out[x][y]
    selected[y]=True
    no_edge+=1
print("Minimum Dittance is : ",add,"miles")
```

0-4:83.53308214561304

4-2:82.8474610857538

2-3:261.7575423593927

3-1:102.55989455322408

Minimum Dittance is : 530.6979801439836 miles