# ***Practical No.2***

---

Name : Ved Satish Agrawal

Sec : B

Roll no. : 61

Sem : V

Branch : CSE

Subject : DAA

---

**AIM**: Write Program to demonstrate working of Prim's Algorithm.

---

**Problem Statement :**

**A telecommunications organization has offices spanned across multiple locations around the globe.**

**It has to use leased phone lines for connecting all these offices with each other. The cost (in units) of connecting each pair of offices is different and is shown in figure.**

**Calculate the cost of connecting each pair of offices. The organization, thus, wants to use minimum cost for connecting all its offices. This requires that all the offices should be connected using a minimum number of leased lines so as to reduce the effective cost.**

---

## Source Code:

```
from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="ap1")

No_of_city = int(input("Enter no. of city to enter:"))
Citys = []
Loc = []
for i in range(No_of_city):
    city = input()
    Citys.append(city)

for i in range(No_of_city):
    location = geolocator.geocode(Citys[i])
    loc = (location.latitude, location.longitude)
    Loc.append(loc)


from geopy.distance import geodesic
G = []

for i in range(No_of_city):
    lst = []
    for j in range(No_of_city):
        nm1=geodesic(Loc[i], Loc[j]).miles
        lst.append(nm1)
    G.append(lst)

for i in G:
    print(i)
INF = 9999999
V = No_of_city
selected = []

for i in range(V):
    selected.append(0)

no_edge = 0
min_cost = 0
selected[0] = True
print("Edge : Weight\n")

while (no_edge < V - 1):
    minimum = INF
    x = 0
    y = 0
    for i in range(V):
        if selected[i]:
            for j in range(V):
                if ((not selected[j]) and G[i][j]):
                    if minimum > G[i][j]:
                        minimum = G[i][j]
```

```
                    x = i
                    y = j
    print(str(Citys[x]) + "-" + str(Citys[y]) + ":" + str(G[x][y]))
    min_cost += G[x][y]
    selected[y] = True
    no_edge += 1
print("Minimum Cost:",min_cost)
```

## **Output** :-

**Enter no. of city to enter:5**

**Nagpur**

**Mumbai**

**Pune**

**Nashik**

**Thane**

**[0.0, 427.50202366902516, 385.2763086761525, 351.63117601750594, 419.1135964192502]**

**[427.50202366902516, 0.0, 74.550293671268, 87.48224082988973, 9.979747789152853]**

**[385.2763086761525, 74.550293671268, 0.0, 102.55989455322387, 74.1264796790887]**

**[351.63117601750594, 87.48224082988973, 102.55989455322387, 0.0, 77.5569066629451]**

**[419.1135964192502, 9.979747789152853, 74.1264796790887, 77.5569066629451, 0.0]**

**Edge : Weight**

**Nagpur-Nashik:351.63117601750594**

**Nashik-Thane:77.5569066629451**

**Thane-Mumbai:9.979747789152853**

**Thane-Pune:74.1264796790887**

**Minimum Cost: 513.2943101486926**