Name: GANESH K

USN: 2VX23UE015 ¶

## **Expt-04 B : Travelling Salesperson Problem**

```
In [1]:
           1 from itertools import permutations
In [10]:
           1
             n = 4
           2
           3 | graph = [[0, 30, 4, 6],
                      [30, 0, 10, 5],
           5
                       [4, 10, 0, 20],
           6
                       [6, 5, 20, 0]]
           7
           8 | start_city = 0
           9
          10 def TSP(graph, start_city):
          11
                  dist = []
          12
          13
                  for cur_city in range(n):
                      if cur city != start city:
          14
          15
                          dist.append(cur_city)
          16
          17
                  min_cost = float('inf')
          18
          19
                  next_Perm = permutations(dist)
          20
                  for i in next_Perm:
          21
          22
                     cur_cost = 0
          23
                      k = start_city
          24
                      for j in i:
          25
          26
                          cur_cost = cur_cost + graph[k][j]
          27
                          k = j
          28
          29
                      cur_cost = cur_cost + graph[k][start_city]
          30
                      min_cost = min(min_cost, cur_cost)
          31
          32
          33
                  return min_cost
          34
          35
             min_TourCost = TSP(graph, start_city)
          36
             print("Minimum Tour Cost of TSP:", min_TourCost)
          37
```

Minimum Tour Cost of TSP: 25

```
In [9]:
          1 n = int(input("Enter number of cities: "))
          3 print("Enter the cost matrix (enter each row separated by space):")
             graph = []
          5
             for i in range(n):
          6
                 row = input().split()
          7
                 graph.append([int(x) for x in row])
          8
          9
                 start_city = 0
         10
         11 def TSP(graph, start_city):
         12
                 dist = []
         13
                 for cur_city in range(n):
         14
         15
                     if cur_city != start_city:
                         dist.append(cur_city)
         16
         17
                 min_cost = float('inf')
         18
         19
         20
                 next_Perm = permutations(dist)
         21
         22
                 for i in next_Perm:
         23
                     cur_cost = 0
         24
                     k = start_city
         25
                     for j in i:
         26
         27
                         cur_cost = cur_cost + graph[k][j]
         28
                         k = j
         29
         30
                     cur_cost = cur_cost + graph[k][start_city]
         31
         32
                     min_cost = min(min_cost, cur_cost)
         33
         34
                 return min_cost
         35
         36 min_TourCost = TSP(graph, start_city)
         37
         38 print("Minimum Tour Cost of TSP:", min_TourCost)
```

```
Enter number of cities: 4
Enter the cost matrix (enter each row separated by space): 0 30 4 6
30 0 10 5
4 10 0 20
6 5 20 0
Minimum Tour Cost of TSP: 25
```