Yaseen Haffejee 1827555

Discrete Optimisation Assignment 2

In [1]:

```
import numpy as np
```

Question

Consider the following distance matrix for the 7 city symmetric Travelling Salesman Problem (STSP):

$$M = \begin{bmatrix} 0 & 1.5 & 3 & 13 & 3.5 & 4.5 & 1.5 \\ & 0 & 1.5 & 1.3 & 13 & 13 & 2.3 \\ & & 0 & 1.5 & 3 & 13 & 3 \\ & & & 0 & 1.5 & 13 & 20 \\ & & & & 0 & 1.5 & 3.3 \\ & & & & & 0 & 1.5 \\ & & & & & & 0 \end{bmatrix}$$

where $m_{ij} = m_{ji}$.
Minimize the above problem using the 2-Opt heuristic.
Use $x = (2, 7, 1, 4, 6, 5, 3)$ as your starting solution.
Count the number of improving solutions during the course of your two optimal procedure and list your improving solutions $(x_i, f(x_i))$.
Report also the final solution (route) x^* and the corresponding optimal distance $f(x^*)$.

Solution

In [2]:

```
M = np.array([
    [0,1.5,3,13,3.5,4.5,1.5],
    [0,0,1.5,1.3,13,13,2.3],
    [0,0,0,1.5,3,13,3],
    [0,0,0,0,1.5,13,20],
    [0,0,0,0,0,1.5,3.3],
    [0,0,0,0,0,0,1.5],
    [0,0,0,0,0,0,0]
])
M = M + M.T
```

In [3]:

```
def calculate_distance(x,num_cities,M):
    cost = 0
    for i in range(num_cities-1):
        city1 = x[i]
        city2 = x[i+1]
        cost += M[city1-1][city2-1]
    cost += M[x[num_cities-1]-1][x[0]-1]
    return cost
```

In [4]:

```
def Two_Opt(x_initial,Distance_Matrix):
    print("-----------------------------------------START-------------------------------
    print("-------------------------------------------------------------------------------"
    print(f"The initial Tour is: {x_initial}")
    print(f"The inital Tour Distance is: {calculate_distance(x_initial,len(x_initial),M)}")
    print("-------------------------------------------------------------------------------\
    number_of_cities = len(x_initial)
    Tour = x_initial.copy()
    Best_distance = calculate_distance(x_initial,number_of_cities,Distance_Matrix)
    number_of_improvements = 0
    for i in range(1,number_of_cities-1):

        for j in range(i+1,number_of_cities):
            new_Tour = Tour.copy()
            city_1 = Tour[i]
            city_2 = Tour[j]
            new_Tour[i] = city_2
            new_Tour[j] = city_1

            new_tour_distance = calculate_distance(new_Tour,number_of_cities,Distance_Matri

            if(new_tour_distance < Best_distance):
                number_of_improvements += 1
                Tour = new_Tour.copy()
                Best_distance = new_tour_distance
                print("-------------------------------------------------------------------
                print(f"Improved Solution Found")
                print(f"The improved tour is: {Tour}")
                print(f"The improved cost is: {Best_distance}")
                print("-------------------------------------------------------------------

                i = 1
    print("-----------------------------------------END---------------------------------
    return Tour,Best_distance,number_of_improvements
```

In [5]:

```
x_initial = [2,7,1,4,6,5,3]
Tour,Best_distance,number_of_improvements  = Two_Opt(x_initial,M)
print("------------------------------------------------------------------------------")
print(f"After a total of {number_of_improvements} improvements.")
print(f"The optimal Tour is: {Tour}")
print(f"The distance for the optimal tour is: {Best_distance}")
print("------------------------------------------------------------------------------")
```

```
-----------------------------------------START----------------------------
---------------

------------------------------------------------------------------------
---
The initial Tour is: [2, 7, 1, 4, 6, 5, 3]
The inital Tour Distance is: 35.8
------------------------------------------------------------------------
---

------------------------------------------------------------------------
---
Improved Solution Found
The improved tour is: [2, 4, 1, 7, 6, 5, 3]
The improved cost is: 23.3
------------------------------------------------------------------------
---

------------------------------------------------------------------------
---
Improved Solution Found
The improved tour is: [2, 3, 1, 7, 6, 5, 4]
The improved cost is: 11.8
------------------------------------------------------------------------
---

-----------------------------------------END----------------------------
-------------

------------------------------------------------------------------------
---
After a total of 2 improvements.
The optimal Tour is: [2, 3, 1, 7, 6, 5, 4]
The distance for the optimal tour is: 11.8
------------------------------------------------------------------------
---
```