In [6]:

```python
#Ayush Sharma 209303312
#Python Code to implement hill climbing algorithm to solve travelling salesman prob

import random


def randomSolution(tsp):
    cities = list(range(len(tsp)))
    solution = []
    for i in range(len(tsp)):
        randomCity = cities[random.randint(0, len(cities) - 1)]
        solution.append(randomCity)
        cities.remove(randomCity)
    return solution


def distance(tsp, solution):
    l = 0
    for i in range(len(solution)):
        l += tsp[solution[i - 1]][solution[i]]
    return l


def getNeighbours(solution):
    neighbours = []
    for i in range(len(solution)):
        for j in range(i + 1, len(solution)):
            neighbour = [*solution]
            neighbour[i] = solution[j]
            neighbour[j] = solution[i]
        neighbours.append(neighbour)
    return neighbours


def getBestNeighbour(tsp, neighbours):
    bestRouteLength = distance(tsp, neighbours[0])
    bestNeighbour = neighbours[0]
    for neighbour in neighbours:
        currentRouteLength = distance(tsp, neighbour)
        if currentRouteLength < bestRouteLength:
            bestRouteLength = currentRouteLength
            bestNeighbour = neighbour
    return bestNeighbour, bestRouteLength

def hillClimbing(tsp):
    currentSolution = randomSolution(tsp)
    currentRouteLength = distance(tsp, currentSolution)
    neighbours = getNeighbours(currentSolution)
    bestNeighbour, bestNeighbourRouteLength = getBestNeighbour(tsp, neighbours)
    while bestNeighbourRouteLength < currentRouteLength:
        currentSolution = bestNeighbour
        currentRouteLength = bestNeighbourRouteLength
        neighbours = getNeighbours(currentSolution)
        bestNeighbour, bestNeighbourRouteLength = getBestNeighbour(tsp, neighbours)
```

```
    return currentSolution, currentRouteLength

tsp = [
    [0, 400, 500, 300],
    [400, 0, 300, 500],
    [500, 300, 0, 400],
    [300, 500, 400, 0]
]
print(hillClimbing(tsp))
```

```
([3, 0, 1, 2], 1400)
```

In [ ]: