

```

#Name: Kalyani Kulkarni
#Roll no : 3338
#Problem Statement: Identify and Implement heuristic and search strategy for
# Travelling Salesperson Problem
import sys
def nearest_neighbour(curr, unvisited, dist_matrix):
    """Returns the nearest neighbour"""
    nearest = sys.maxsize
    neighbour= None
    for city in unvisited:
        if dist_matrix[curr][city] < nearest:
            nearest=dist_matrix[curr][city]
            neighbour=city
    return neighbour, nearest

def tsp_nn(dist_matrix,i):
    """Solves the tsp problem using nearest neighbour algorithm"""
    n=len(dist_matrix)
    tour=[0]*(n+1) #Initialize the tour
    unvisited=set(range(1,n)) #set of unvisited cities
    curr_city=i #Starting city

    for i in range(1,n):
        next_city, dist= nearest_neighbour(curr_city, unvisited, dist_matrix)
        tour[i]=next_city
        curr_city=next_city
        unvisited.remove(next_city)
        #Return to the starting city
    tour[n-4]=0

    #calculate the total cost of the tour
    cost=sum(dist_matrix[tour[i]][tour[i+1]] for i in range(n-1))
    cost+=dist_matrix[tour[n-1]][tour[0]]
    return tour, cost

#Example
dist_matrix=[]
r=int(input("Enter the size of matrix: "))

for i in range(r):
    r=list(map(int,input().split()))
    dist_matrix.append(r)
print("\n The distance matrix is : \n ")
for i in dist_matrix:
    print(*i)

final_cost=[]
for i in range(0,len(dist_matrix)):
    tour,cost=tsp_nn(dist_matrix,i)
    print("Tour", tour)
    print("Total Cost", cost)
    final_cost.append(cost)
    cost=0
print("Minimum cost: ",min(final_cost))

```

```

Enter the size of matrix: 4
0 5 15 4
5 0 35 25
15 35 0 30
4 25 30 0

```

The distance matrix is :

```

0 5 15 4
5 0 35 25
15 35 0 30
4 25 30 0
Tour [0, 3, 1, 2, 0]
Total Cost 79
Tour [0, 1, 3, 2, 0]
Total Cost 75
Tour [0, 2, 3, 1, 0]
Total Cost 75
Tour [0, 3, 1, 2, 0]
Total Cost 79
Minimum cost: 75

```

[Colab paid products](#) - [Cancel contracts here](#)

