

6. Implementation of TSP using heuristic approach

```
def tsp_nearest_neighbor(distances):
    num_cities = len(distances)
    unvisited_cities = set(range(num_cities))
    current_city = 0 # Start from city 0
    tour = [current_city]
    total_distance = 0

    while unvisited_cities:
        nearest_city = min(unvisited_cities, key=lambda city: distances[current_city][city])
        total_distance += distances[current_city][nearest_city]
        current_city = nearest_city
        unvisited_cities.remove(nearest_city)
        tour.append(current_city)

    total_distance += distances[current_city][0] # Return to the starting city
    tour.append(0)

    return tour, total_distance
```

Example usage:

```
distances = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]
```

```
tour, total_distance = tsp_nearest_neighbor(distances)
print("Nearest Neighbor TSP Solution:")
print("Tour:", tour)
print("Total Distance:", total_distance)
```

OUTPUT

Nearest Neighbor TSP Solution:

Tour: [0, 0, 1, 3, 2, 0]

Total Distance: 80

