

100.Travelling Salesman Problem

AIM: To solve the Travelling salesman problem by greedy technique

PROGRAM:

```
import numpy as np
from itertools import permutations

def tsp_brute_force(distances):
    num_cities = len(distances)
    all_cities = set(range(num_cities))
    min_distance = float('inf')
    best_path = None

    for path in permutations(range(1, num_cities)):
        current_distance = distances[0][path[0]]
        for i in range(num_cities - 2):
            current_distance += distances[path[i]][path[i + 1]]
        current_distance += distances[path[-1]][0]

        if current_distance < min_distance:
            min_distance = current_distance
            best_path = (0,) + path

    return min_distance, best_path

distances = np.array([[0, 10, 15, 20],
                      [10, 0, 35, 25],
                      [15, 35, 0, 30],
                      [20, 25, 30, 0]])

min_dist, best_route = tsp_brute_force(distances)
print(f"Minimum Distance: {min_dist}")
print(f"Best Route: {best_route}")
```

```
Minimum Distance: 80  
Best Route: (0, 1, 3, 2)
```

OUTPUT:

TIME COMPLEXITY: $O(n^2 \cdot 2^n)$