Write the python to implement Travelling Salesman Problem

AIM

To implement a Python program to solve the **Travelling Salesman Problem (TSP)** using **Brute-Force / Permutations** method to find the shortest possible route visiting all cities exactly once and returning to the start.

ALGORITHM

1. Represent the cities and distances as a **distance matrix**.
2. Fix a **starting city**.
3. Generate all possible **permutations of the remaining cities**.
4. For each permutation:
    a. Calculate the **total travel cost** including return to the start city.
    b. Keep track of the permutation with the **minimum cost**.
5. Return the **path** and **minimum cost**.

8 PUZZLE AI.py - C:/Users/gayathri/Downloads/8 PUZZLE AI.py (3.8.2)

File Edit Format Run Options Window Help

```
from itertools import permutations
def tsp(graph, start=0):
    n = len(graph)
    nodes = list(range(n))
    nodes.remove(start)
    min_path = None
    min_cost = float('inf')
    for perm in permutations(nodes):
        cost = 0
        k = start
        for j in perm:
            cost += graph[k][j]
            k = j
        cost += graph[k][start]
        if cost < min_cost:
            min_cost = cost
            min_path = (start,) + perm + (start,)
    return min_path, min_cost
graph = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]
path, cost = tsp(graph)
print("Path:", path)
print("Cost:", cost)
```

```
==============================
Path: (0, 1, 3, 2, 0)
Cost: 80
>>>
```

RESULT

The program successfully computed the shortest route for the given distance matrix.