

Practical No. : 2

Name : Mohan Kadambande

Roll No. : 13212

Aim : Implement A* Algorithm for Traveling Salesman Problem (TSP) search problem.

Code :

```
from itertools import permutations
def tsp(cost):
    # Number of nodes
    numNodes = len(cost)
    nodes = list(range(1, numNodes)) # The remaining nodes (excluding the start node)

    minCost = float('inf')

    # Generate all permutations of the remaining nodes
    for perm in permutations(nodes):
        currCost = 0
        currNode = 0 # Start from node 0

        # Calculate the cost of the current permutation
        for node in perm:
            currCost += cost[currNode][node]
            currNode = node

        # Add the cost to return to the starting node (node 0)
        currCost += cost[currNode][0]

        # Update the minimum cost if the current cost is lower
        minCost = min(minCost, currCost)

    return minCost

if __name__ == "__main__":
    # Get the number of cities (nodes)
    num_cities = int(input("Enter the number of cities: "))

    # Initialize the cost matrix
    cost = []
    print(f"Enter the cost matrix ({num_cities}x{num_cities}):")

    for i in range(num_cities):
        row = list(map(int, input(f"Enter row {i+1} (space-separated): ").split()))
```

```
cost.append(row)

# Call the TSP function to get the minimum cost
result = tsp(cost)
print(f"The total cost of the tour is: {result}")
```

Output:

```
Enter the number of cities: 3
Enter the cost matrix (3x3):
Enter row 1 (space-separated): 10 20 30
Enter row 2 (space-separated): 30 20 10
Enter row 3 (space-separated): 30 20 10
The total cost of the tour is: 60
```