Program 103.Travelling salesman problem

Program:

```
def tsp(graph):
    # Number of cities
    n = len(graph)

    # Initialize dp table with inf
    dp = [[float('inf')] * n for _ in range(1 << n)]

    # Start from city 0
    dp[1][0] = 0

    # Fill dp table
    for mask in range(1 << n):
        for u in range(n):
            if (mask & (1 << u)) != 0:  # If u is in the subset represented by mask
                for v in range(n):
                    if (mask & (1 << v)) == 0:  # If v is not in the subset represented by mask
                        new_mask = mask | (1 << v)
                        dp[new_mask][v] = min(dp[new_mask][v], dp[mask][u] + graph[u][v])

    # Final result
    res = float('inf')
    for u in range(1, n):
        res = min(res, dp[(1 << n) - 1][u] + graph[u][0])

    return res

# Example usage
graph = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]

print(tsp(graph))  # Output: 80
```

Output:

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA\DAA COADS.PYTHON\program 103.py"
80

Process finished with exit code 0
```

Time complexity:
O(n^2.2^2)