122. NP-Complete and NP-Hard Problem
Code:

```
from itertools import permutations

def traveling_salesman(graph):
    n = len(graph)
    min_path = float('inf')
    for perm in permutations(range(1, n)):
        current_pathweight = 0
        k = 0
        for j in perm:
            current_pathweight += graph[k][j]
            k = j
        current_pathweight += graph[k][0]
        min_path = min(min_path, current_pathweight)
    return min_path
graph = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]
print(f"The minimum path weight is: {traveling_salesman(graph)}")
```

output:

Time complexity:

F(n)=o(n*m)