

## 102. Travelling salesman problem

### Code:

```
def tsp(dist):
    n = len(dist)
    dp = [[float('inf')] * n for _ in range(1 << n)]
    dp[1][0] = 0

    for mask in range(1 << n):
        for u in range(n):
            if mask & (1 << u):
                for v in range(n):
                    if mask & (1 << v) and u != v:
                        dp[mask][u] = min(dp[mask][u], dp[mask ^ (1 << u)][v] + dist[v][u])

    final_mask = (1 << n) - 1
    return min(dp[final_mask][u] + dist[u][0] for u in range(1, n))

dist = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]

print(tsp(dist))
```

### Output:

80

### Time Complexity:

- $T(n) = O(2^n * n^2)$