25. Travelling Salesman Problem (Brute-force)

from itertools import permutations

def tsp\_brute\_force(graph, start):

vertices = list(graph.keys())

vertices.remove(start)

min\_path = float('inf')

best\_route = []

for perm in permutations(vertices):

current\_cost = 0

k = start

for j in perm:

current\_cost += graph[k][j]

k = j

current\_cost += graph[k][start] # return to start

if current\_cost < min\_path:

min\_path = current\_cost

best\_route = [start] + list(perm) + [start]

return best\_route, min\_path

graph = {

'A': {'A': 0, 'B': 10, 'C': 15, 'D': 20},

'B': {'A': 10, 'B': 0, 'C': 35, 'D': 25},

'C': {'A': 15, 'B': 35, 'C': 0, 'D': 30},

'D': {'A': 20, 'B': 25, 'C': 30, 'D': 0}

}

path, cost = tsp\_brute\_force(graph, 'A')

print("TSP Path:", path)

print("Total Cost:", cost)

OUTPUT:

