Exp-9 travelling salesman problem

Code

from itertools import permutations

def calculate\_cost(graph, path):

cost = 0

for i in range(len(path) - 1):

cost += graph[path[i]][path[i+1]]

cost += graph[path[-1]][path[0]]

return cost

def tsp(graph):

n = len(graph)

cities = list(range(n))

min\_path = None

min\_cost = float('inf')

for perm in permutations(cities[1:]):

current\_path = [0] + list(perm)

current\_cost = calculate\_cost(graph, current\_path)

if current\_cost < min\_cost:

min\_cost = current\_cost

min\_path = current\_path

return min\_path, min\_cost

graph = [

[0, 10, 15, 20],

[10, 0, 35, 25],

[15, 35, 0, 30],

[20, 25, 30, 0]

]

path, cost = tsp(graph)

print("Minimum Cost Path:", path)

print("Minimum Cost:", cost)

output

