Hill climbing algorithm for Heuristic search

Program:

from sys import maxsize

from itertools import permutations

V = 4

def travellingSalesmanProblem(graph, s):

vertex = [] # Changed variable name to lowercase 'vertex'

for i in range(V): # Fixed capitalization of 'for'

if i != s: # Changed capitalization of 'if'

vertex.append(i)

min\_path = maxsize # Changed variable name to lowercase 'min\_path'

next\_permutation = permutations(vertex) # Changed variable name to lowercase

'next\_permutation'

for i in next\_permutation: # Fixed capitalization of 'for'

current\_pathweight = 0 # Changed variable name to lowercase 'current\_pathweight'

k = s # Changed variable name to lowercase 'k'

for j in i: # Fixed capitalization of 'for'

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

k = j

current\_pathweight += graph[k][s]

min\_path = min(min\_path, current\_pathweight)

return min\_path # Changed capitalization of 'return'

if \_\_name\_\_ == "\_\_main\_\_":

graph = [[0, 10, 15, 20], [10, 0, 35, 25],

[15, 35, 0, 30], [20, 25, 30, 0]]

s = 0

print(travellingSalesmanProblem(graph, s)) # Changed capitalization of 'print'

Output:

80