

FLAT ASSIGNMENT

Implementation of traveling salesman algorithm

program to implement traveling salesman

problem using naive approach.

from sys import maxsize

from itertools import permutations

V = 4

implementation of traveling Salesman Problem

def travellingSalesmanProblem(graph, s):

store all vertex apart from source vertex

vertex = []

for i in range(V):

if i != s:

vertex.append(i)

store minimum weight Hamiltonian Cycle

min_path = maxsize

next_permutation=permutations(vertex)

for i in next_permutation:

store current Path weight(cost)

current_pathweight = 0

compute current path weight

k = s

for j in i:

current_pathweight += graph[k][j]

k = j

current_pathweight += graph[k][s]

```

        # update minimum
        min_path = min(min_path, current_pathweight)
        return min_path

# Driver Code
if __name__ == "__main__":
    # matrix representation of graph
    n=int(input("Enter no of nodes: "))
    for i in range(n):
        l=list(map(int,input().split()))
        graph.append(l)
    s=0
    print("tne minimum length is :",travellingSalesmanProblem(graph,
s))

```

output:

```

Enter no of nodes: 4
0 10 5 20
10 0 35 25
15 35 0 30
20 25 30 0
tne minimum length is : 70

```

Submitted by:

P.Mohan

CSE-DS-A

21071A6741