

Task :2

Implementation of **Hill climbing algorithm for Heuristic search** approach using following constraints in python.

Aim: To Implement Hill climbing algorithm for Heuristic search approach for travelling salesman problem using python

Algorithm:

Step 1: start

Step 2: define TSP with (graph, s) and assign value for vertex.

Step 3: store all vertex apart from source vertex.

Step 4: store minimum weight hamiltonian cycle and assign permutation (vertex).

Step 5: store current path weight (cost) and compute current path weight.

Step 6: Update minimum and matrix representation of the graph values and print it.

Step 7: stop

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

Result:

Thus the Implementation of Hill climbing algorithm for Heuristic search approach for travelling salesman problem using python was successfully executed and output was verified.