# 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 = 5

def travellingSalesmanProblem(graph, s):

    vertex = []

    for i in range(V):

        if i != s:

            vertex.append(i)

    min\_path = maxsize

    next\_permutation = permutations(vertex)

    for i in next\_permutation:

        current\_pathweight = 0

        k = s

        for j in i:

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

            k = j

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

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

    return min\_path

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

    graph = [

        [0, 10, 15, 20, 25],

        [10, 0, 35, 25, 12],

        [15, 35, 0, 30, 13],

        [20, 25, 30, 0, 13],

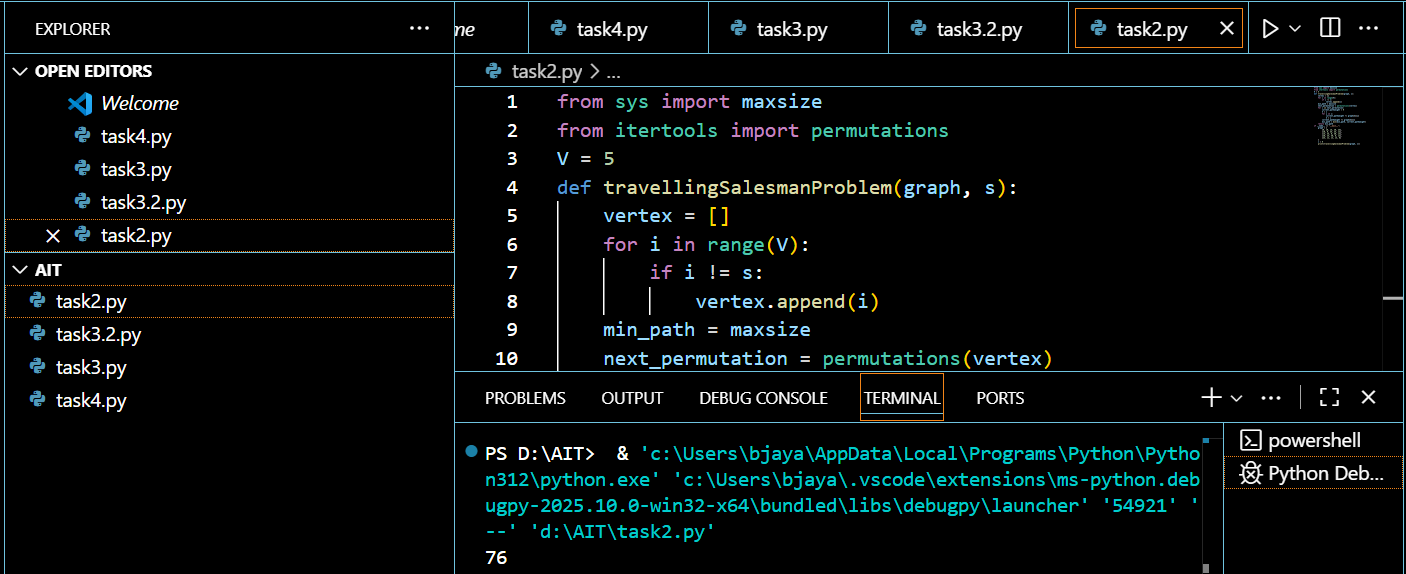
        [25, 12, 13, 13, 0]

    ]

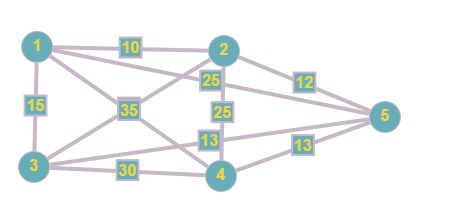
    s = 0

    print(travellingSalesmanProblem(graph, s))

**Output:**

****

**GRAPH:**



**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.