import sys

class Graph:

    def \_\_init\_\_(self, vertices):

        self.V = vertices

        self.graph = [[0 for column in range(vertices)] for row in range(vertices)]

    def printSolution(self, dist):

        print("Vertex \tDistance from Source")

        for node in range(self.V):

            print(node, "\t", dist[node])

    def minDistance(self, dist, sptSet):

        min\_val = sys.maxsize

        min\_index = 0

        for u in range(self.V):

            if dist[u] < min\_val and not sptSet[u]:

                min\_val = dist[u]

                min\_index = u

        return min\_index

    def dijkstra(self, src):

        dist = [sys.maxsize] \* self.V

        dist[src] = 0

        sptSet = [False] \* self.V

        for \_ in range(self.V):

            x = self.minDistance(dist, sptSet)

            sptSet[x] = True

            for y in range(self.V):

                if (

                    not sptSet[y]

                    and self.graph[x][y] > 0

                    and dist[y] > dist[x] + self.graph[x][y]

                ):

                    dist[y] = dist[x] + self.graph[x][y]

        self.printSolution(dist)

g = Graph(9)

g.graph = [

    [0, 4, 0, 0, 0, 0, 0, 8, 0],

    [4, 0, 8, 0, 0, 0, 0, 11, 0],

    [0, 8, 0, 7, 0, 4, 0, 0, 2],

    [0, 0, 7, 0, 9, 14, 0, 0, 0],

    [0, 0, 0, 9, 0, 10, 0, 0, 0],

    [0, 0, 4, 14, 10, 0, 2, 0, 0],

    [0, 0, 0, 0, 0, 2, 0, 1, 6],

    [8, 11, 0, 0, 0, 0, 1, 0, 7],

    [0, 0, 2, 0, 0, 0, 6, 7, 0],

]

g.dijkstra(0)