Assignment: NP-Completeness and Heuristic Algorithms

Note: You will discuss Question 1 as part of the Group Assignment. (Check this week's Group Assignment on Canvas for details).

1. **NP-Completeness:** Consider the Travelling Salesperson (TSP) problem that was covered in the exploration.
   Problem: Given a graph G with V vertices and E edges, determine if the graph has a TSP solution with a cost of at most k.
   Prove that the above stated problem is NP-Complete.

2. **Implement Heuristic Algorithm:**
   a. Below matrix represents the distance of 5 cities from each other. Represent it in the form of a graph

	A	B	C	D	E
A	0	2	3	20	1
B	2	0	15	2	20
C	3	15	0	20	13
D	20	2	20	0	9
E	1	20	13	9	0

   b. Apply Nearest-neighbor heuristic to this matrix and find the approximate solution for this matrix if it were for TSP problem.
   c. What is the approximation ratio of your approximate solution?
   d. Implement Travelling Salesman Problem using the nearest-neighbor heuristic.

   Input: The input Graph is provided in the form of a 2-D matrix (adjacency matrix). Consider the first node as the starting point.

   Sample input:

```
G = [
    [0, 2, 3, 20, 1],
    [2, 0, 15, 2, 20],
    [3, 15, 0, 20, 13],
    [20, 2, 20, 0, 9],
    [1, 20, 13, 9, 0],
]
```

   Output: A list of indices indicating the path taken. You must return the sequence of nodes, the path taken starting from node 0. In this example, G is 5x5, indicating there are 5 nodes in this graph: 0-4. You will always begin with node 0, and your

path should include every node exactly once, and only go between nodes with a nonzero edge between them. You path will end at the starting node.

Sample output (For above graph G):

[0, 4, 3, 1, 2, 0]

Note: Not all graphs are fully connected: some rows in G may have more than one 0. These indicate absence of an edge.

Name your function **solve_tsp(G)**. Name your file **TSP.py**.