COST OF GAS

Description:

The cost of gas is insane! Can you believe node traversal is so expensive? I wish I had some sort of map or matrix describing the cheapest cost to any node from any node...
NodeA -> NodeC 32324
NodeB -> NodeA 26786
NodeC -> NodeB 77458 NodeC -> NodeD 19905 NodeC -> NodeG 19455
NodeD -> NodeA 64678 NodeD -> NodeE 57878
NodeE -> NodeF 29999 NodeE -> NodeA 82356
NodeF -> NodeC 77777 NodeF -> NodeA 33333 NodeF -> NodeD 88888 NodeF -> NodeG 88888
NodeG -> NodeA 1
Example submission: A -> B 1 B -> A 1
Resulting matrix: A B A 0 1 B 1 0
THIS IS WHAT YOUR FLAG SHOULD LOOK LIKE: STOUTCTF{0110}

- **No spaces**
- **One line**
- **No letters**

Given a list of node-to-node travel costs and asked to create a matrix that shows the cheapest cost between each pair of nodes. The goal is to process the provided distances and find the minimum cost for each pair. For example, if traveling from NodeA to NodeB costs 1 and from NodeB to NodeA costs 1, the matrix will show these values. Creating a script must be done to calculates the shortest travel costs between nodes using the Floyd-Warshall algorithm. First, it initializes a distance matrix with "infinity" to represent unconnected paths, then sets the diagonal to 0, as the cost to travel from a node to itself is zero. Then populates the matrix with the provided edge costs between nodes. After that, the algorithm iterates through each pair of nodes, updating the distance matrix to reflect the shortest paths by considering each intermediate node. Finally, converts the matrix into a continuous string of integers representing the minimum costs between all nodes and prints the result.

Script

```
import numpy as np

edges = {
    'A': {'C': 32324},
    'B': {'A': 26786},
    'C': {'B': 77458, 'D': 19905, 'G': 19455},
    'D': {'A': 64678, 'E': 57878},
    'E': {'A': 82356, 'F': 29999},
    'F': {'C': 77777, 'A': 33333, 'D': 88888, 'G': 88888},
    'G': {'A': 1}
}

nodes = list(edges.keys())

n = len(nodes)
distance_matrix = np.full((n, n), float('inf'))  # Start with "infinity"

for i in range(n):
    distance_matrix[i][i] = 0

for src in edges:
    for dest in edges[src]:
```

```
        distance_matrix[nodes.index(src)][nodes.index(dest)] = edges[src][dest]

for k in range(n):
    for i in range(n):
        for j in range(n):
            if      distance_matrix[i][j]      >      distance_matrix[i][k]      +
distance_matrix[k][j]:
                distance_matrix[i][j]       =       distance_matrix[i][k]       +
distance_matrix[k][j]

output = ""
for i in range(n):
    for j in range(n):
        output += str(int(distance_matrix[i][j]))

result = f"STOUTCTF{{{output}}}"
print(result)
```

```
┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Cost_of_gas]
└─$ python try12.py
STOUTCTF{0109782323245222911010714010651779267860591107901513689316689278565194567745801990577783107782194556467817446097002057878878771164576333217311495656115561029999115111333331431156565785562143440085112110978332325522301101081401070}
```

Flag	STOUTCTF{0109782323245222911010714010651779267860591107901513689316689278565194567745801990577783107782194556467817446097002057878878771164576333217311495656115561029999115111333331431156565785562143440085112110978332325522301101081401070}