Plus court ou plus rapide ? (MEDIUM/HARD)

Auteur(s) :

Hokanosekai

Catégorie :

Prog

Description :

Tous les jours, je me rends à la fac en bus. Mais aujourd'hui, le bus est en panne. Je dois donc trouver un autre moyen de transport. Je me rends compte que je peux prendre le métro. Mais je ne sais pas quel est le chemin le plus court pour aller à la fac. Pouvez-vous m'aider ?

```
nc 161.35.21.37 40002
```

Exemple de graphe créé par le serveur :

```
{
  "graph": {
    "nodes": ["A", "B", "C"],
    "edges": [
      {"source": "A", "target": "B", "weight": 5},
      {"source": "B", "target": "C", "weight": 3},
      {"source": "A", "target": "C", "weight": 10}
    ]
  },
  "source": "A",
  "destination": "C"
}
```

Puis on retourne sous le format suivant :

```
A B 5
B C 3
A C 10
-----
A > C
```

Exemple de réponse attendue :

```
{
  "path": ["A", "B", "C"],
```

```
  "weight": 8
}
```

Flag : UHOCTF{Fake_flag}

Solution :

On comprend grâce au titre du challenge qu'il va falloir utiliser un algorithme pour trouver le plus court chemin entre deux points. On peut utiliser l'algorithme de Dijkstra pour cela.

Nous allons donc commencer par parser le résultat du serveur pour créer un graphe.

Ensuite nous devons implémenter l'algorithme de Dijkstra pour trouver la solution.

```python
def dijkstra(graph, source, destination):
    distances = {node: float('inf') for node in graph["nodes"]}
    distances[source] = 0

    heap = [(0, source)]
    while heap:
        current_distance, current_node = heapq.heappop(heap)

        if current_distance > distances[current_node]:
            continue

        if current_node == destination:
            break

        for edge in graph["edges"]:
            if edge["source"] == current_node:
                neighbor = edge["target"]
                distance = current_distance + edge["weight"]
                if distance < distances[neighbor]:
                    distances[neighbor] = distance
                    heapq.heappush(heap, (distance, neighbor))

    return distances[destination]
```

On peut ensuite envoyer la réponse au serveur.

Dans le cas de ce challenge le serveur nous envoie aux total 10 graphes à la suite. Il faut donc automatiser la résolution du challenge.

Flag : UHOCTF{Th4nk's_K4r311}

Hosting

```
sudo docker build -t dijkstra .
sudo docker network create -d bridge dijkstra
```

The command to start the challenge is:

```
sudo docker run -p 40002:40002 --detach --name dijkstra --network dijkstra
dijkstra:latest
```

The command to stop the challenge (since CTRL+C won't work) is:

```
sudo docker stop dijkstra
```