**Prosit 1 : Petit Détour**

# Contexte :

Nous avons été missionnés pour optimiser le trajet des agents se chargent de la pose d’éclairage intelligents dans la ville.

# Mot clés :

Recherche opérationnelle : Discipline mathématique et informatique qui vise à résoudre des problèmes de décision ou d'optimisation en utilisant des méthodes algorithmiques et des techniques quantitatives.

Algorithme : Suite finie et non ambiguë d'instructions ou de règles permettant de résoudre un problème ou d'accomplir une tâche spécifique.

ADEME : L'Agence de l'Environnement et de la Maîtrise de l'Énergie est un établissement public français qui agit dans les domaines de l'environnement, de l'énergie et du développement durable.

Arbre couvrant minimal : Dans un graphe, un arbre couvrant minimal est un sous-graphe qui est un arbre, qui inclut tous les sommets du graphe de départ et dont la somme des poids de ses arêtes est minimale.

Graphes et théorie des graphes : Les graphes sont des structures mathématiques utilisées pour représenter des relations entre des objets. La théorie des graphes étudie les propriétés et les applications des graphes.

Complexité – Asymptotique : La complexité asymptotique d'un algorithme mesure comment son temps d'exécution ou son utilisation de la mémoire évolue en fonction de la taille de l'entrée. Elle est souvent exprimée en notation Big O.

Chaîne eulérienne : Une chaîne eulérienne dans un graphe est un chemin qui traverse chaque arête exactement une fois.

Intersection : En mathématiques, l'intersection de deux ensembles est l'ensemble des éléments communs aux deux ensembles.

7 ponts de Königsberg : Problème classique de la théorie des graphes qui consiste à déterminer s'il est possible de traverser tous les sept ponts de la ville de Königsberg (aujourd'hui Kaliningrad) une seule fois et de revenir au point de départ sans passer deux fois sur le même pont.

Occupation CPU

# Problématique :

**Comment optimiser l’itinéraire en ayant un passage unique dans chaque rue ?**

# Contraintes :

- Passage unique

- Partir du point de depart et y revenir.

- Si le passage unique est impossible, donner le nombre de passages minimal.

- Bonne performance de l’algorithme.

# Généralisation :

Conception d’algorithme eulérien.

Les bases de Python

# Livrables :

Modélisation du graphe

Démonstration de l’algorithme et du programme. (Préciser le temps d’exécution du programme)

# Pistes de solution :

Conversion de l’itineraire en matrice adjacente

Chaque intersection serait un Sommet et chaque rue une arete

# Plan d’action :

Explorer les ressources.

Remodéliser le problème sous forme de graphe.

Etudier les différentes manières de modéliser un graphe sur python.

Faire un choix de modélisation.

Faire une condition de passage unique sur le programme

Avoir une progression de l’itineraire sur le programme pour suivre l’ordre de passage sur les aretes.

Un algo pour passage unique, dans le cas contraire : un autre pour l’algo optimisé.

Implémenter l’algorithme (structure de données, analyse de performance).

# Réalisation du plan d’action :

Voici donc le schéma du prosit :

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement

J’ai utilisé le graphe suivant :

Une image contenant capture d’écran, ligne, espace, cercle

Description générée automatiquement

J’ai essayé de représenter la situation avec le schéma suivant mais je n’étais pas satisfait des résultats suivants :

Une image contenant capture d’écran, cercle

Description générée automatiquement

J’ai donc réaliser un script python qui prend en entre la liste d’adjacence du graphe et qui retourne un cycle eulérien valide si il y en a un.

Le code est dispo sur mon [github](https://github.com/HLaplace/CESI_A3_algo_avance).