Analyse fonctionnelle

Sujet: Calculer l'itinéraire optimal via les transports en commun parisiens
Niveaux de complexité :

- Trajet le plus rapide
- Tenir compte des exigences de l'utilisateur (correspondances et chemin obligatoire)
- Compter les travaux actuels dans le calcul de l'itinéraire

Etapes:

- Chargements des données (Python, Ma +A, 4h)
    - Création d'une base de données à partir de 4 fichiers textes : nombre d'arguments, nom de station, latitude, longitude, lignes associées
    - Création d'une base de données de tous les horaires de chaque arrêt, quelque soit le transport
    - Faire une documentation explicative
- Création du graphe (C, A, 6h)
    - A partir des deux bases de données, création de listes doublement chaînées. Chaque nœud a pour structure le nom de la station, ses coordonnées, les lignes associées, une variable changement non initialisée. Les liaisons sont typées selon la méthode de transport et valuées selon le temps de parcours.
    - Relier les deux bases de données via le nom de l'arrêt (ou un code_id)
    - Calculer le temps de parcours une fois les horaires correspondants récupérés (pour une heure demandée ?)
- A*(C, T, 10h)
    - Etablir un lien géographique de proximité à vol d'oiseau entre les diverses stations. Condition d'arrêt : lien de proximité viable déterminé
    - Etablir l'heuristique selon le niveau de complexité exigé
    - Exécuter A*. Sortie : fichier texte comportant les nœuds de l'itinéraire calculé et les horaires de chaque transition
- Interface graphique (Python, Me, 4h)
    - Chargement du logo
    - Récupération des arrêts départ et arrivée : saisie de l'utilisateur
    - Récupération des conditions liées aux niveaux de complexité : saisie de l'utilisateur
    - Gestion des boutons pour valider et quitter
    - Vérification des éléments saisis
    - Transmission des informations utilisateurs (via fichier texte) pour exécuter A*
    - Récupération des données de l'itinéraire optimal calculé (via fichier texte)
    - Affichage du trajet déterminé

- Affichage dynamique (Python, Ma+Me, 6h)
    - Incrustation de l’image du graphe
    - Représenter l'itinéraire sur cette image