

TD 10-11 : Projet Rennes2 Express

1 Préambule

Pour ce projet, vous devrez travailler par groupes de 2.

Le projet sera réalisé essentiellement lors des deux dernières séances de TD, et votre présence lors de ces séances de TD est obligatoire.

La date limite de rendu est indiquée sur CURSUS dans l'espace de dépôt.

2 Rennes2 Express



Lors de l'émission télévisée Rennes2 Express, 8 équipes s'affrontent pour tenter de remporter un grand prix ! Les concurrents s'élancent de Marseille le lundi 13 novembre, et ils ont 4 jours pour rallier l'Université Rennes 2, le plus rapidement possible.

Chaque journée de jeu se déroule de la manière suivante :

- Une étape le matin, lors de laquelle les concurrents doivent rallier le plus rapidement possible la ville suivante.
- Une épreuve le midi. Les équipes s'affrontent, et l'équipe gagnante obtient un bonus "Ticket de train" !
- Une étape l'après-midi, lors de laquelle les concurrents doivent rallier la ville suivante.

Tous les déplacements doivent se faire à pied ou en voiture. Les équipes ayant gagné un bonus "Ticket de train" ont le droit d'emprunter une fois le train sur une étape de leur choix lors de la suite de l'aventure.

Au début et à la fin de chaque étape, une borne est disposée sur le parcours, et les candidats doivent badger pour valider leur temps de parcours de l'étape.

Lors de l'édition 2023, des bruits courent... Certaines équipes auraient triché, en empruntant un TGV alors qu'elles n'y étaient pas autorisées !

La règle est simple : on considère qu'une équipe qui parvient à se déplacer plus rapidement que le temps de trajet d'une voiture a emprunté le train.

La société de production de Rennes2 Express nous fournit les informations suivantes :

- les informations sur l'édition 2023 de l'émission avec les étapes, les épreuves et leur gagnants ;
- la liste des bornes de début et de fin de parcours, avec leur position GPS ;
- la liste de tous les enregistrements des équipes sur les bornes, avec l'horaire précis de validation.

Votre rôle sera de déterminer quelles sont la ou les équipes qui ont triché !

3 Travail attendu

Pour valider votre projet, vous devrez fournir un script Python produisant les résultats ci-dessous. Différents niveaux de rendus sont proposés, et seront valorisés dans la note finale.

Niveau 1 Afficher la ou les équipes qui ont triché, c'est à dire qui ont pris un train alors qu'elles n'avaient pas le droit.

Niveau 2 Afficher le nom de l'équipe gagnante au classement général, c'est à dire l'équipe qui a mis le moins de temps cumulé sur l'ensemble des étapes.

Niveau 2 bis Afficher le classement complet des équipes (qui n'ont pas triché), au classement au temps.

Niveau 3 Afficher le nom de l'équipe gagnante en nombre de points (3 points pour le premier de l'étape, 2 points pour le second, un point pour le troisième).

Niveau 3 bis Afficher le classement complet des équipes (qui n'ont pas triché), au classement aux points.

Il est également attendu de fournir un code clair et lisible, qui limite les traitements inutiles et les requêtes redondantes.

4 Implémentation en Python

Pour mener à bien votre mission, vous pourrez utiliser (outre votre intelligence) :

- le package csv pour l'analyse des deux fichiers csv de la production (disponibles sur Cursus) ;
- le package requests pour interroger la base des équipes http://my-json-server.typicode.com/alemaitr/python_opendata_12/rennes2express ;
- le package graphh pour calculer des temps de trajet théoriques en voiture entre deux points. Attention de ne pas utiliser tous vos crédits graphHopper trop vite !
- le package datetime pour la gestion des dates et des durées ;
- les structures de données appropriées : listes, dictionnaires, tuples, ensembles.

Nb : il n'est pas recommandé d'utiliser les Dataframes Pandas ici, car il vous manquerait quelques notions non vues en TD.

5 Rendu de devoir

Votre rendu se fera sous la forme d'un fichier Python. Ce fichier devra être nommé td10-11.py et contenir les noms et numéros étudiant de tous les membres du groupe commentés, en en-tête du fichier, comme dans l'exemple suivant :

```
# 22000002 Paul Machin
# 22000227 Yolène Truc
```

```
import ...
```

Les fichiers déposés sur CURSUS ne devront surtout pas contenir vos clés d'API. Celles-ci devront être lues par votre programme dans un fichier credentials.json (que vous ne fournirez pas pour ne pas divulguer votre clé d'API) au format :

```
{
    "GraphHopper": "..."
```