

TD 10-11 : Projet Terra Aventura

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 Terra Aventura



La région Nouvelle Aquitaine propose une chasse aux trésors touristique, nommée Terra Aventura : <https://www.terra-aventura.fr/>. Ainsi, sur tout le territoire de Nouvelle Aquitaine, environ 600 trésors ont été dissimulés. Chaque *trésor* (aussi appelé *cache*) peut être trouvé en collectant des indices, lors de la découverte d'un lieu. Les aventures sont triées par thématique, et associées à un personnage. Par exemple, le personnage de Zabeth correspond à des aventures de type "Histoire".

3 Problème posé

Deux amis explorateurs, Marco et Polo, souhaitent partir en vacances dans la région Nouvelle Aquitaine, avec pour objectif de dénicher le maximum de trésors possibles. Ils cherchent donc un hébergement qui leur permette d'avoir accès à de nombreux trésors, sachant qu'ils ne souhaitent pas faire plus de 60 minutes de voiture, autour de leur hébergement.

3.1 Sélection des trésors

Terra Aventura propose plus de 600 trésors. Mais Marco et Polo en ont déjà trouvé un certain nombre ! Ils veulent donc se focaliser sur ceux qu'ils n'ont pas encore trouvé. Par ailleurs, Marco et Polo sont férus d'histoire, ils se focaliseront donc sur les trésors de type "Zabeth".

Deux fichiers sont à votre disposition :

- `tresors.json` est un export du site web Terra Aventura. Il contient la liste de tous les trésors disponibles dans la région Nouvelle Aquitaine.
- `trouvailles.csv` contient la liste des trésors déjà trouvés par Marco et Polo (identifiant du trésor et date de découverte).

3.2 Choix de l'hôtel

Pour le choix de l'hébergement, vous aurez à votre disposition l'API suivante, qui rassemble plus de 50000 lieux d'hébergements touristiques en France et dans le monde. <https://data.opendatasoft.com/explore/dataset/osm-hosting-fr%40babel/>

Marco et Polo ont récemment gagné un bon de réduction dans la chaîne d'hôtel IBIS. Ils souhaitent donc loger dans un hôtel IBIS. *Indice : on pourra filtrer la requête à l'API en précisant la valeur "Ibis" pour le champ "Operator"*.

Puisque les Terra Aventura sont situées en Nouvelle-Aquitaine, on s'assurera de ne conserver que les hôtels situés dans un des départements de la région Nouvelle-Aquitaine. On pourra réutiliser le fichier `dico_dpt.py` vu en TD de "SDD en Python" pour obtenir la liste des numéros de départements de Nouvelle Aquitaine.

Enfin, Marco et Polo ont bien noté que leur bon de réduction n'est pas valable dans les hôtels "Ibis Styles", il faudra donc exclure ces hôtels des potentiels résultats.

3.3 Résumé

Le problème posé est donc :

- Étant définie une liste de trésors de type `Zabeth`, non encore trouvées par Marco et Polo
- Étant donné la liste des hôtels Ibis en région Nouvelle Aquitaine (Ibis Styles exclus)
- Trouver parmi ces hôtels lequel a le plus de trésors dans son entourage à moins de 60 minutes de voiture.

Pour le calcul de temps de trajet, vous pourrez utiliser le package `graphh`. Attention de ne pas utiliser tous vos crédits `graphHopper` trop vite !

4 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 0 Produire la liste des trésors candidats telle que décrite dans la section ??, ainsi que la liste des hotels telle que décrite dans la section ??.

Niveau 1 Afficher le nom de l'hôtel IBIS qui permet d'avoir le plus de trésors à moins de 60 minutes en voiture.

Niveau 2 Produire un fichier csv contenant les trésors `Zabeth` placés autour de l'hôtel retenu.

Niveau bonus Produire un affichage cartographique présentant le point de l'hôtel, des trésors à proximité etc...

Il est également attendu de fournir un code clair et lisible, qui limite les traitements inutiles et les requêtes redondantes. Il faudra également utiliser les structures de données appropriées : listes, dictionnaires, tuples, ensembles.

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": "..."  
}
```