TP 1
Construire un crawler minimal

Présentation du sujet

Écrivez un crawler single-threaded.

À partir d'une URL d'entrée unique (https://ensai.fr/), le crawler doit télécharger une page, puis attendre au moins cinq secondes avant de télécharger la page suivante.

Votre programme doit trouver d'autres pages à explorer en analysant les balises de liens trouvées dans les documents précédemment explorés.

Votre programme doit se terminer lorsque le crawler arrive à 50 urls trouvées ou si il ne trouve plus de liens à explorer.

Une fois terminé, votre programme écrira dans un fichier **crawled_webpages.txt** toutes les urls trouvées.

Règles:

- Ne pas crawler un site web qui vous l'interdit
- Respecter la politeness

Ce papier peut vous aider:
Pant, Gautam & Srinivasan, Padmini. (2003). Crawling the Web. Web Dynamics. 10.1007/978-3-662-10874-1_7.

En bonus

- Lire le fichier sitemap.xml des sites pour réduire les requêtes aux urls tout en découvrant plus de pages, vous pourrez alors augmenter le threshold sur le nombre de documents maximum à trouver
- Créer une base de données relationnelle pour stocker les pages web trouvées, leur age
- Multi threader votre crawler tout en respectant la politeness et les robots.txt

Ce qui est demandé

Un dossier avec votre crawler écrit en **python**.
Le code devra s'executer dans un fichier **main.py** à la racine du projet.
Vos fonctions publiques devront être testées.

L'explication du code et de son execution devra être décrite dans un fichier **README.md** à la racine du projet. Ce fichier devra aussi comprendre le nom des contributeurs du projet.

Si vous êtes familier avec GitHub, vous pouvez aussi m'envoyer le lien vers votre repository.

Les librairies dont vous aurez besoin

- Pour requêter les urls et lire les robots.txt: https://docs.python.org/fr/3/library/urllib.html
- Pour lire les fichiers html: https://www.crummy.com/software/BeautifulSoup/bs4/doc/