Projet de programmation concurrente Java
Recherche d'expressions régulières sur le Web

Travail à rendre. Le travail est à rendre pour le 15 novembre 2019 sous forme électronique sur Madoc sous forme d'archive zip ou tar.gz, qui comprendra :

- le code source **commenté** de vos programmes ;
- un script ant ou bash commenté permettant **de construire et lancer vos programmes** ;
- un court rapport au format **pdf** détaillant les problèmes de concurrence que vous avez réglés.

Problème. Pour pouvoir répondre efficacement aux requêtes des utilisateurs, Google Search maintient un index de toutes les pages du Web, accompagné de la liste des mots qu'elles contiennent. Cet index est établi grâce à des *robots d'indexation* (en anglais *web crawlers*) qui explorent les pages Web en suivant leurs hyperliens.

Le but de ce projet est d'implémenter notre propre robot d'indexation pour rechercher des expressions régulières sur le Web. Par exemple, la commande suivante doit rechercher toutes les pages contenant le mot « Nantes » accessibles depuis la page Wikipédia de la ville.

```
1 $ java WebGrep Nantes https://fr.wikipedia.org/wiki/Nantes
2 https://fr.wikipedia.org/wiki/Nantes
3 https://fr.wikiquote.org/wiki/Nantes
4 https://fr.wiktionary.org/wiki/Nantes
5 https://fr.wikivoyage.org/wiki/Nantes
6 ...
```

Pour obtenir les résultats, on commencera par chercher une occurrence de l'expression dans la page donnée en argument. Si la page contient l'expression, on continuera la recherche en suivant tous les liens de la page, récursivement. On pourra également afficher les lignes contenant les occurences trouvées, ou autres, comme dans grep. Les contraintes suivantes doivent être respectées.

- La recherche doit être parallélisée entre n threads.
- Il ne faut pas d'entrelacement entre les sorties pour deux pages.
- Il ne faut pas explorer deux fois la même URL.

Ressources. On pourra se baser sur les exemples suivants pour les parties du programme liés à la lecture des pages Web :

Lecture d'URL : `https://docs.oracle.com/javase/tutorial/networking/urls/readingURL.html`

Recherche de regex : `https://stackoverflow.com/questions/1670593/java-i-have-a-big-string-of-html-and-need-to-extract-the-href-text`