Programmation de robots - TD 8

DESS TEXTE

Exercice 1 : étude la classe `URI : :URL` (suite)

Dans cet exercice, nous allons voir comment extraire des informations d'une URL.

a) Ecrire un programme qui, étant donné deux URL A et B, détermine si A et B sont hébergées par le même serveur.

b) Ecrire un programme qui, étant donné deux URL A et B, détermine si les documents pointés par A sont dans un sous-dossier de B.
(Exemple :
http ://www.liberation.fr/quotidien/semaine/lundi/art1.html est dans un sous-dossier de http ://www.liberation.fr/quotidien/semaine mais pas de http ://www.liberation.fr/quotidien/archives)

Exercice 2 : étude des classes `LWP : :UserAgent` et `HTTP : :Request`

Dans cet exercice, nous allons apprendre à programmer un robot prenant en charge l'exécution d'une requête web et la récupération de la réponse.

a) Ecrire un programme qui exécute la requête http ://www.google.fr/search ?q=bora-bora et stocke les liens retournés par google dans une liste qui sera affichée à l'écran.

b) Ecrire un programme qui analyse le contenu de la page située à l'url passée en argument, en extrait les liens :
- pour chaque lien,le programme déterminera s'il est local à la base de l'url ou non, et stockera ce lien en fonction du type (prévenir l'apparition de doublons).

c) Ecrire un programme qui analyse la page située à l'url passée en argument, en extrait les liens :
- pour chaque lien, le programme déterminera s'il est local à la base ou non, et si oui le stockera dans une table (après vérification qu'il ne s'agit pas d'un doublon), et analysera récursivement les pages correspondant aux liens ainsi récupérés.

Conseil : Au fur et à mesure que de nouvelles pages locales sont repérées par l'anlyseur HTML, elles sont ajoutées à une liste, chaque page visitée est sortie de la liste. On se sert des fonctions définies à la question précédente afin de vérifier la valeur de la base, et de vérifier si la page est locale.
On considère les adresses comme des attributs dans une table de hachage. La valeur associée comptabilise le nombre de fois que l'adresse a été détectée par le parseur. Elle sera consultée pour déterminer si cette adresse est "nouvelle" et doit être visitée, ou pas.