[LU2IN006] RAPPORT

MINI PROJET

18 FEVRIER

Créé par : El Boukili Ali 21210507
El Haddad Tarek 21215054

SORBONNE UNIVERSITÉ

1- Reformulation courte du sujet

Dans ce mini projet, notre attention se porte sur la gestion d'une bibliothèque, définie comme une collection de livres. Chaque livre est identifié par son titre, l'auteur associé et un numéro d'enregistrement unique.
Le but de ce mini-projet est de se familiariser avec la comparaison des structures de données. Nous mettrons en œuvre une bibliothèque en utilisant deux approches différentes :
| Une implémentation à l'aide d'une liste simplement chaînée de structures (Partie 1).
| Une table de hachage de structures (Partie 2).
| Analyse de performance des deux méthodes (Partie 3).

2- La description des structures manipulées, et la description globale de notre code

Liste chaînée

- *Livre :*
  - **num** : un entier représentant le numéro d'enregistrement du livre.
  - **titre** : un pointeur vers une chaîne de caractères représentant le titre
  - **auteur** : un pointeur vers une chaîne de caractères représentant le nom de l'auteur
  - **suiv** : un pointeur vers une structure Livre, représentant le livre suivant dans la liste chaînée.

- *Biblio :*
  - **L** : un pointeur vers une structure Livre, représentant le premier élément de la bibliothèque. Cette structure est utilisée comme tête fictive pour faciliter la manipulation de la liste chaînée des livres.

Table de Hachage

- *LivreH :*
  - **clef** : un entier représentant la clé du livre dans la table de hachage.
  - **num** : un entier représentant le numéro d'enregistrement du livre.
  - **titre** : un pointeur vers une chaîne de caractères représentant le titre du livre.
  - **auteur** : un pointeur vers une chaîne de caractères représentant le nom de l'auteur du livre.
  - **suivant** : un pointeur vers une structure LivreH, représentant le livre suivant dans la même case de la table de hachage.
- *BiblioH :*

- **nE** : un entier représentant le nombre d'éléments actuellement présents dans la table de hachage.
- **m** : un entier représentant la taille de la table de hachage.
- **T** : un double pointeur vers des structures LivreH, représentant la table de hachage.

- Les fichiers entreeSortieLC.c et entreeSortieH.c contiennent les fonctions permettant de manipuler les fichiers txt.
- Les fichiers entreeSortieLC.h et entreeSortieH.h contiennent les signatures des fonctions permettant de manipuler les fichiers txt.

- Les fichiers BiblioLC.c et BiblioH.c contiennent les fonctions permettant de manipuler les structures respectives à chacune ainsi que des fonctions d'affichage, recherche, suppression, fusion de bibliothèque et recherche de plusieurs exemplaires de tous les ouvrages.
- Les fichiers BiblioLC.h et BiblioH.h contiennent les signatures des fonctions dans les fichiers BiblioLC.c et BiblioH.c

3- Réponses des questions :

1-

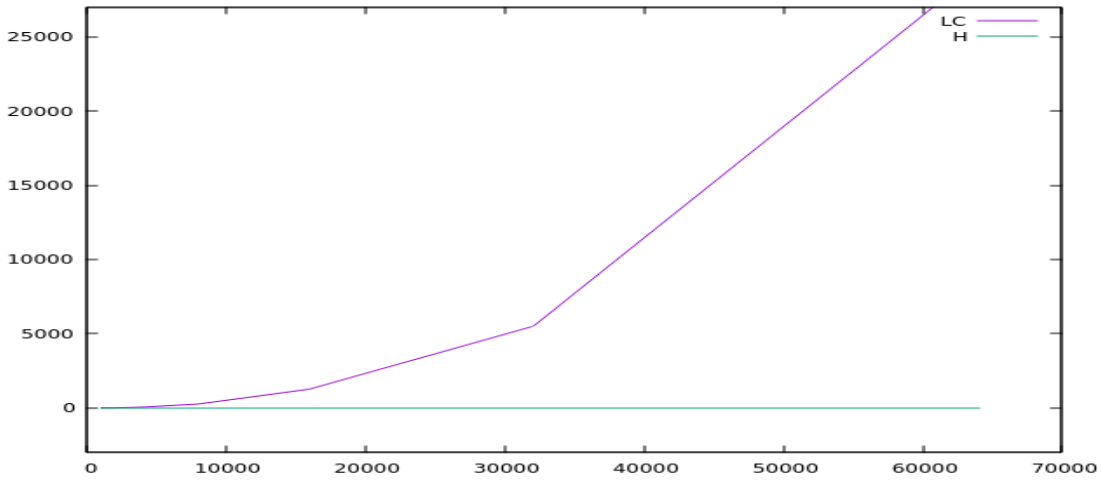
- La structure LC est plus optimisé pour la recherche des livres avec numéro car l'ordre des livres dépends de son numéro d'enregistrement donc plus on s'approche du nombre de livres, plus LC sera optimisé pour la recherche puisque la liste est inversée avec inserer_en_tete ()
- La structure H est plus optimisée pour la recherche des livres par auteur car la clef de hachage dépend de l'auteur du livre, donc la fonction trouvera le livre directement à partir de l'indice(clef) de la table de hachage
- Pour recherche titre on ne saurait déterminer l'efficacité d'une structure par rapport à une autre car le titre n'est pas un critère de triage dans les deux structures

2-

On remarque que la taille de la table de hachage affecte directement la vitesse de recherche, dans un premier temps si la taille est trop petite on se rapprocherai de l'efficacité d'une liste chainée alors que si on augmente la taille plus on aura de clefs de hachage pour chaque auteur unique

Et il faudra parcourir tout le tableau pour trouver un doublon, du coup il faut trouver une taille entre les deux pour avoir la version la plus optimisé

4-



Pour la fonction de recherche d'ouvrage exemplaires la complexité-temps pire des LC étant de $O(n^2)$, nous pouvons observer une courbe exponentielle. Concernant la table de hachage, nous observons une courbe linéaire. On sait que les doublons sont forcément dans la même liste de même indice. Ducoup il faudra appliquer la fonction recherche_ouvrage() de LC m fois mais comme chaque liste sera de petite taille on aura une complexité plus petite que O(n2)

4- La description des jeux d'essais

Les fichiers LC et H se trouvent dans les dossiers respectives LC et H chacun avec un Makefile et un menu dans le main, et pour le fichier compStruct est fait pour les résultats obtenu dans l'exercice 3

LC : ./main GdeBiblio.txt (nb_lignes) fichier.txt (nb_lignes)

H : ./main GdeBiblio.txt (nb_lignes) fichier.txt (nb_lignes)

compStruct : ./compStruct GdeBiblio.txt (nb_lignes)

Vous pouvez aussi utiliser les tests donner dans les fichiers jeux test de chaque dossier

Pour visualiser les courbes, lancer le main de compStruct puis utilisez gnuplot avec le fichier resultats.txt
gnuplot
$ plot "resultats.txt" using 1:2 with lines title "LC"
$ replot "resultats.txt" using 1:3 with lines title "H"