

Mini-Projet (TME3 et 4) , LU2IN006 par Beeverly Gourdet et Bettina Mubiligi

Le projet repose sur la gestion d'une bibliothèque, soit un ensemble de livres identifiés par leurs titre, auteur et numéro d'enregistrement.

Le projet est séparé en 3 parties : la gestion de bibliothèque avec une liste chaînée de struct, une table de hachage, et enfin, la comparaison des temps de calcul avec ces deux structures pour la recherche dans une bibliothèque donnée.

La partie 1 traite la structure en liste de structures, et est composée de 4 fichiers principaux : biblioLC.c, biblioLC.h, entreeSortieLC.h et entreeSortieLC.c, les deux premiers étant dédiés à la définition des structures Livre et Biblio, ainsi que les fonctions de manipulation de ces dernières, et les deux derniers fichiers permettent de créer une bibliothèque à partir d'un fichier texte ou d'enregistrer une bibliothèque sous forme de fichier texte.

La partie 2 traite la structure avec table de hachage, cette fois avec 2 fichiers principaux : biblioH.c et biblioH.h, définissant deux nouvelles structures adaptées aux tables de hachages : BiblioH et LivreH, avec de nouveaux paramètres : le nombre d'éléments contenus dans la table de hachage ainsi que sa taille dans BiblioH, et la clé associée au livre dans LivreH.

Nous avons également implémenté des fonctions main.c, mainh.c et main2.c pour tester les parties 1, 2 et 3 respectivement, ainsi que essaiBib.c et essaiBibH.c.

Partie 3 : comparaison des temps de calculs pour les deux structures, pour une recherche par numéro, titre ou auteur (Q 3.1) ou une recherche des ouvrages avec plusieurs exemplaires (Q 3.3) et enfin discussion sur les complexités avec les courbes obtenues (Q 3.4).

Q 3.1

Dans le cas où le livre recherché est présent dans la bibliothèque et que l'on souhaite effectuer des recherches par son numéro, son titre et son auteur, une table de hachage est plus appropriée car elle offre des temps de recherche en moyenne constants et une meilleure complexité algorithmique que la liste.

En compilant et en exécutant notre fichier test main2.c, on observe que la recherche par paramètres d'un livre avec la table de hachage prends globalement moins de temps qu'avec une liste de structures.

En revanche, si le livre est absent de la bibliothèque, une liste chaînée peut être plus adaptée. Parcourir une liste pour vérifier l'absence d'un élément peut être plus efficace, car dans le pire des cas, cela nécessite de parcourir tous les éléments de la liste, or si une collision se produit avec la table de hachage et que plusieurs livres sont associés à la même case, on doit parcourir chaque livre pour vérifier l'absence du livre recherché, ce qui peut entraîner un temps de recherche plus long.

Q 3.2

Si la table de hachage devient trop remplie (par exemple si le nombre d'éléments dépasse un certain seuil par rapport à la taille de la table), les temps de calcul peuvent commencer à augmenter en raison des collisions plus fréquentes, lorsque deux éléments différents se voient attribuer la même case dans la table de hachage. Il faudrait ajuster la taille de la table en fonction du nombre d'éléments stockés à comparer.

Q 3.4

En comparant les temps de calcul pour la recherche d'ouvrages identiques avec chaque structure, on obtient un graphe (fichier `graphe_comparaison`) qui nous montre l'écart de temps lié à la complexité des deux algorithmes `recherche_identiques` et `recherche_identiquesH` (ici, une complexité temps pire cas en $O(n^2)$ où n est la taille de la bibliothèque).

On observe que la recherche par table de hachage prends globalement plus de temps, cela peut être dû au fait que plus la taille de la bibliothèque augmente (mais pas celle de la table de hachage), plus les collisions sont nombreuses, ce qui nest pas le cas avec la structure de liste.

En effet, avec la table de hachage, la recherche d'ouvrages identiques demande de parcourir la liste chaînées des livres (collisions avec livres occupant la même case de la table) un problème qui n'est pas rencontré avec les listes. Pour améliorer les temps de calcul par hachage, il faudrait systématiquement adapter la taille de la table à celle de la biobibliothèque.