

TP4 : utilisation des tables de hachage

Le but du TP est de mesurer l'efficacité des tables de hachage en adressage ouvert et en adressage chaîné. Pour réaliser cela, nous mettons à votre disposition un ensemble d'unités ainsi qu'un programme principal :

Unité	Ce qu'elle réalise
U_Hachage	Définition d'une clé, définition de la fonction de hachage primaire, définition de la fonction de hachage secondaire et définition de la capacité de la table
U_TableHachageOuvert	Définition d'une table de hachage en adressage ouvert
U_TableHachageChainage	Définition d'une table de hachage avec résolution des collisions par chaînage
U_FichierTexte	Permet la lecture mot à mot du contenu d'un fichier texte
testHachage.pas	Code pour écrire les tests du TP

Les deux unités implantant les tables de hachage disposent des fonctions suivantes :

```
// insere le couple <k,v> dans la table
// exception TablePleine levee si aucune alveole libre ne peut etre trouvee
procedure inserer (k : CLE; v : VALEUR);

// recherche la cle k dans la table et retourne la valeur associee
// exception CleAbsente levee si la cle n'est pas presente
function rechercher (k : CLE) : VALEUR;

// vide la table de son contenu
procedure raz;
```

et d'autres fonctions permettant des comptages.

On met également à disposition plusieurs fichiers pour les tests : un fichier texte1.txt qui est un texte en anglais, un fichier texte2.txt qui comprend des mots de longueur 10 dont l'occurrence de chaque lettre est équiprobable (donc ce ne sont pas des vrais mots), un fichier texte3.txt construit comme le précédent mais pour des mots de longueur 20, un fichier texte4.txt construit comme les précédent mais dont la longueur des mots est variable. Les trois derniers fichiers contiennent 2500 mots.

Dans le TP, nous allons observer :

- l'influence de la fonction de hachage primaire sur le nombre de collisions
- l'influence de la fonction de hachage secondaire sur le nombre de collisions
- l'influence de la méthode de résolution des collisions (ouvert ou chaînage) sur la recherche
- l'influence de la capacité de la table en chaînage

Exercice 1 : Evaluation de la qualité de la fonction de hachage primaire

Dans cet exercice on fixera la capacité de la table à 2500.

Q 1.1 Ecrire une première version de la fonction h où la valeur retournée correspond à la position du premier caractère de la clé dans la table ASCII. Tracer la courbe du nombre moyen de collisions en fonction du taux de remplissage pour chaque fichier texte (nombre de mots insérés variant de 100 à 2500 par pas de 100).

Q 1.2 Ecrire une autre version de la fonction h où la valeur retournée correspond à la somme des positions des caractères de la clé dans la table ASCII. Tracer la courbe du nombre moyen de collisions en fonction du taux de remplissage pour chaque fichier texte (nombre de mots insérés variant de 100 à 2500 par pas de 100).

Q 1.3 Implanter enfin la fonction vue en cours :

$$h(k) = \sum_{i=1}^{\ell} \text{ord}(k[i]) * 128^{\ell}$$

Tracer la courbe du nombre moyen de collisions en fonction du taux de remplissage pour chaque fichier texte (nombre de mots insérés variant de 100 à 2500 par pas de 100).

Q 1.4 Pour chaque texte, tracer sur un même graphique les courbes avec les différentes fonctions de hachage.

Q 1.5 Donner une explication quant aux variations observées.

Exercice 2 : Evaluation de la qualité de la fonction de hachage secondaire

Dans cet exercice, on fixera la capacité de la table à 2500. On choisira comme fonction de hachage primaire celle qui a fourni les meilleurs résultats à l'exercice précédent.

Q 2.1 Modifier le code de la fonction `hprime` pour utiliser un hachage quadratique à la place d'un hachage linéaire. Tester plusieurs couples de valeurs pour `a` et `b` et observer le comportement suivant le fichier test utilisé. Donner un couple de valeurs dont le comportement est mauvais et un couple de valeurs dont le comportement est bon.

On propose d'utiliser maintenant une fonction de hachage primaire et une fonction de hachage secondaire en vue d'utiliser un hachage double.

Q 2.2 Modifier le code de la fonction `hprime` pour utiliser un hachage double et proposer une fonction de hachage primaire et une fonction de hachage secondaire qui, utilisées dans le cadre du hachage double, fournissent des meilleurs résultats que le hachage quadratique ou linéaire.

Q 2.3 Pour chaque texte, tracer sur un même graphique les courbes du nombre moyen de collisions en fonction du taux de remplissage avec les différents hachages : linéaire, quadratique et double (nombre de mots insérés variant de 100 à 2500 par pas de 100).

Q 2.4 Donner vos conclusions quant à l'utilisation de ces fonctions.

Exercice 3 : Comparaison avec la résolution des collisions par chaînage

La capacité de la table est fixée à 2500.

On veut maintenant comparer les résultats obtenus précédemment à ceux qu'on peut obtenir avec une table de hachage où les collisions sont résolues par chaînage.

Pour comparer les performances des tables de hachage, nous allons compter le nombre de comparaisons d'éléments dans le cas d'une recherche fructueuse. Il suffit alors de rechercher tous les mots de la table et de faire la moyenne du nombre de comparaisons.

Q 3.1 Obtenez le nombre moyen de comparaisons dans le cas d'une recherche fructueuse avec la table de hachage en adressage ouvert pour tous les textes lorsque tous les mots sont insérés (on aura choisit les fonctions de hachage donnant les meilleurs résultats dans l'exercice précédent).

Q 3.2 Obtenez le nombre moyen de comparaisons dans le cas d'une recherche fructueuse avec la table de hachage avec résolution des collisions par chaînage pour tous les textes lorsque tous les mots sont insérés.

Q 3.3 Que constatez-vous ? Expliquez pourquoi.

Exercice 4 : Variation de la capacité de la table en adressage chaîné

Q 4.1 Dresser pour chaque texte des courbes montrant la longueur moyenne des listes, la longueur médiane, la longueur minimale et la longueur maximale en fonction du nombre de mots insérés (variant de 100 à 2500 par pas de 100) lorsque la table a une capacité de 2500.

Q 4.2 Dresser des courbes pour chaque texte montrant la longueur moyenne des listes, la longueur médiane, la longueur minimale et la longueur maximale en fonction de la capacité de la table (variant de 100 à 2500 par pas de 100).

Q 4.3 Dresser des courbes pour chaque texte montrant le nombre moyen de comparaisons dans la cas d'une recherche fructueuse en fonction de la capacité de la table (variant de 100 à 2500 par pas de 100).

Q 4.4 Commenter les résultats.