TD numéro 3
Table de hachage

Programmation impérative avancée, ENSIIE

Semestre 2, 2015–16

Exercice 1 : Fonctions de hachage

On rappelle que la méthode de la division permet d'obtenir une fonction de hachage de l'ensemble des clefs vers $[0, \ldots, m-1]$:

$$h_{div}(k) = k - \left(m \left\lfloor \frac{k}{m} \right\rfloor\right) \qquad \text{(reste de la division euclidienne de } k \text{ par } m\text{)}$$

1. On suppose que les clefs sont des entiers répartis uniformément entre 0 et l, avec $l \gg m$ (autrement dit, la probabilité d'avoir une clef k pour $0 \leq k \leq l$ est de $\frac{1}{l+1}$). Montrer que la fonction h_{div} vérifie l'hypothèse d'uniformité simple.
2. Quel inconvénient possède la méthode si les clefs sont des entiers et $m = 2^p$?
3. On suppose que les clefs sont des chaînes de caractères. On peut associer à la chaîne "$c_1c_2\ldots c_n$" l'entier $c_1 + 256c_2 + \cdots + 256^{n-1}c_n$ (en assimilant caractère et entier ASCII), sur lequel on appellera la fonction de hachage h_{div}.
   Si $m = 255$, que se passe-t-il quand on permute les lettres d'une clef ?

On en déduit que la méthode de la division n'est satisfaisante que pour un entier m premier et éloigné d'une puissance de 2.

4. Si on souhaite stocker 2000 chaînes de caractères avec en moyenne trois collisions. Proposer une taille de table.

On considère maintenant la méthode de la multiplication

$$h_{mult}(k) = \lfloor (kA - \lfloor kA \rfloor) \times m \rfloor$$

où A est une constante strictement comprise entre 0 et 1.

5. Montrer que l'entier retourné est bien entre 0 et $m-1$.
6. Que se passe-t-il si A est rationnel ?

Exercice 2 : Table de hachage

On considère pour l'instant des tables ouvertes.

1. On suppose qu'on a une fonction de hachage simplement uniforme. Montrer que le nombre moyen d'associations par case dans une table de hachage de taille m avec n clefs insérées est de $\frac{n}{m}$.
2. En déduire que la complexité en moyenne de la fonction de recherche est en $O(1 + \frac{n}{m})$.
3. On considère une table avec redimensionnement dynamique : dès que le nombre de clefs insérées est supérieur à la taille de la table, on crée une nouvelle table de taille double et on y met toutes les associations de la table précédente. On fait n insertions dans une table de taille initiale 2^0. Montrer que le coût en temps est de l'ordre de n malgré les redimensionnements. En déduire qu'en moyenne la complexité de l'insertion est en $O(1)$ pour les tables de hachage avec redimensionnement dynamique.