TD 9 : Applications à l'informatique

1 Hachage

Dans les exercices suivants, on prendra pour notations :

K : le nombre de clés possibles, et on supposera que l'ensemble des clés est précisement $[\![0; K-1]\!]$,

m : la taille de la table de hachage,

n : le nombre de clés que l'on veut stocker dans la table.

Exercice 1. Pour gérer les collisions, on utilise la méthode de chaînage extérieur : chaque case de la table de hachage renvoie vers une liste chaînée contenant les éléments entrant en collision sur cette case. On suppose que la fonction de hachage est uniforme, c'est à dire que pour toute case d'indice i, si on prend k une clé choisie aléatoirement (uniforme), $\mathbb{P}[h(k) = i] = \frac{1}{m}$

1. Montrer que le nombre de collisions dans une case suit une loi de Poisson. Quel est son paramètre ?
2. Calculer la complexité moyenne d'une recherche négative (élément non contenu dans la table).
3. Calculer la complexité moyenne d'une recherche positive (élément contenu dans la table).

Exercice 2. Soit p un nombre premier tel que $K \leq p$. Soient $a, b \in [\![0; p-1]\!], a \neq 0$. On définit la fonction de hachage suivante :

$$h_{a,b}(k) = ((ak + b)\ mod\ p)\ mod\ m$$

1. Soit $r = (ak + b)\ mod\ p$ et $s = (al + b) \mod p$. Montrer que $k \neq l \Rightarrow r \neq s$.
2. Montrer que pour chaque $(r, s) \in [\![0; p-1]\!]^2$ telle que $r \neq s$, il existe une unique paire $(a, b) \in [\![0; p-1]\!]^2$ telle que $a \neq b$.
3. Si on choisit k et l aléatoirement, quelle est la probabilité d'avoir une collision entre $h_{a,b}(k)$ et $h_{a,b}(l)$?

On dit que l'ensemble des fonctions $h_{a,b}$ est une classe universelle de fonctions de hachage.

Exercice 3. Soit $\mathcal{H}$ une classe universelle de fonctions de hachage. On tirera aléatoirement des fonctions dans $\mathcal{H}$. On utilise une table de hachage principale T de fonction $h \in \mathcal{H}$. Pour gérer les collisions de T, on va utiliser la méthode suivante : chaque case d'indice i de T renvoie vers une table de hachage secondaire T_i de taille m_i (avec une fonction $h_i \in \mathcal{H}$ spécifique).

1. Montrer que si $m = n^2$, $\mathbb{P}[$ il exite une collision $] < \frac{1}{2}$.
2. Si h est fixé et $m = n$, que suffit t'il de faire pour éviter toute collision ?
3. Avec la méthode de la question 2, montrer que la structure utilisée est de taille $2n$ en moyenne, i.e. $\mathbb{E}[S] < 2n$ avec $S = \sum_{i=1}^{m} m_i$ la taille totale des différentes tables secondaire T_i.
4. Montrer que $\mathbb{P}[S > 4n] < \frac{1}{2}$.

5. En déduire une méthode générale pour stocker n clés tout en garantissant un temps de recherche constant et un espace mémoire $S = 4n$.

6. Quel défaut majeur voyez vous à cette méthode ?

2 Algorithmes probabilistes

Exercice 4. Soit $n \in \mathbb{N}$ supposé très grand. Soit $a_1, a_2, ..., a_n$ des réels. On veut trouver un a_i plus grand qu'au moins la moitié d'entre eux.

1. Proposer un algorithme déterministe simple. Quelle est sa complexité ?

2. Si on prend k éléments aléatoire et que l'on choisit le plus grand d'entre eux, minorer la probabilité de succès.

Cet algorithme est de type Monte Carlo : la correction n'est pas garantie mais la complexité est bonne.

Exercice 5. On reprend l'algorithme du tri rapide vu en cours d'algorithmique. On cherche à trier un tableau de valeurs $a_1 < a_2 < ... < a_n$ dont l'ordre est quelconque. On précise qu'ici, la complexité correspond au nombre de comparaisons faites.

1. L'algorithme initial est déterministe : on choisit un pivot fixe, disons le dernier élément du tableau. Quelle la complexité en pire cas ?

2. On va maintenant choisir le pivot aléatoirement (uniforme). Pour $i \neq j$, combien de fois a_i et a_j sont comparés ?

3. Majorer la probabilité que a_i et a_j soient comparés ?

4. Quelle est l'espérance de la complexité ? Indication : $\sum_{k=2}^{n} \frac{1}{k} \leq \log(n)$

5. Déduire la complexité moyenne du tri rapide déterministe.

Cet algorithme est de type Las Vegas : la correction est garantie mais la complexité est aléatoire.