Algorithmique TD5

Magistère Informatique 1ère Année

15 Octobre 2009

Exercice 1 *Tables de Hachage*

Sous l'hypothèse de hachage uniforme montrer que si on tire n clés, la probabilité qu'il y ait au moins une collision est :

$$1 - \frac{m!}{(m-n)!} \cdot \frac{1}{m^n} \approx 1 - e^{-\frac{n^2}{2m}}$$

Exercice 2 *Réhachage*

On peut résoudre le problème de collisions par réhachage. Soit N la taille d'une table de hachage, et soit m le nombre d'élements affectés dans la table. Si m dépasse N, on double la taille de la table de hachage. Il devient alors nécessaire de redéfinir la fonction de hachage et de réaffecter les entrées vers leurs nouvelles positions.

Monter que le coût du réachage peut être négligé en effectuant une analyse de coût amorti. On pourra supposer que le choix de le nouvelle fonction de hachage et le réhachage d'une entrée se font en temps constant.

Exercice 3 *Tris par tas*

1. Transformer un tableau en tas binaire.
2. Transformer un tas binaire en tableau trié.
3. Etudier la complexité de ces opérations.
4. Monter que la construction du tas peut se faire en $O(n)$.

Exercice 4 *Tas 2-3-4* Un tas $2-3-4$ est un arbre dont les clés sont uniquement enregistrées dans les feuilles. Chaque feuille x possède une unique clé $c(x)$ et un pointeur vers son père $p(x)$. Chaque nœud interne x possède :

- Un nombre de fils $n(x)$ qui est égal à $2, 3$ ou 4.
- Un pointeur vers chavun de ses fils.
- Un pointeur vers son père.
- Une valeur $s(x)$ qui est égales à la plus petite clé contenu dans le sous-arbre enraciné en x.

La racine contient également une valeur $h(x)$ qui contient la hauteur du tas. Toutes les feuilles sont à la même profondeur. Pour toutes les questions on demande d'écrire la procédure et de donner sa complexité.

1. Écrire une fonction $MINIMUM(t)$ qui retourne un pointeur vers la feuille de clé minimale du tas t.
2. Écrire une fonction $DIMINUE(t, x, k)$ qui diminue la valeur de la clé de la feuille x de t à k (on suppose que t ne contient pas la clé k).

3. Écrire une fonction $INSERTION(t, k)$ qui insère la clé k dans t (on suppose que t ne contient pas la clé k).
4. Écrire une fonction $SUPPRIME(t, x)$ qui supprime une feuille x donnée de t.
5. Écrire une fonction $UNION(t_1, t_2)$ qui unit les tas t_1 et t_2 en les détruisant.

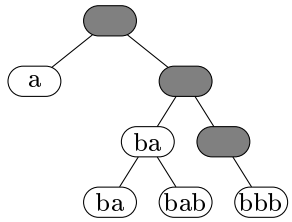
Exercice 5 *Arbres Binaires de Recherche Balisés* Dans un ABR balisé, les clés ne se trouvent qu'aux feuilles et les nœuds internes ont tous exactement 2 fils et contiennent des balises qui permettent d'orienter la recherche. Si x est un nœud interne, alors :

$$c(x.g) < x.balise < c(x.d)$$

1. Montrer que $n(a) = 2 \times f(a) - 1$ avec a un ABR balisé $n(a)$ son nombre de nœuds et $f(a)$ son nombre de feuilles.
2. Écrire les procédures d'insertion et de suppression dans un ABR balisé.
3. Écrire une procédure pour transformer en temps linéaire un ABR balisé en un ABR classique ayant la même structure.
4. Écrire la procédure inverse.

Exercice 6 Etant donné un alphabet Σ ordonné, on définit l'ordre lexicographique sur Σ^* comme suit : Soit $w = a_0a_1...a_p$ et $w' = b_0b_1...b_q$, $w < w'$ si et seulement si w est un prefixe de w' ou $\exists j$ tel que $\forall i < j$ $a_i = b_i$ et $a_j < b_j$.

La structure d'arbre de base peut contenir des mots écrits sur un alphabet à deux lettres. Par exemple l'arbre ci-dessous contient les mots *bab*, *ba*, *baa*, *bbb* et *a*. Soit $\mathcal{L}$ un langage fini sur un alphabet à deux lettres dont la somme des longueurs des mots vaut n. Montrer qu'on peut utiliser la structure d'arbre de base pour trier le langage lexicographiquement en temps $O(n)$.



Exercice 7 Une table de hachage de taille m est utilisée pour contenir n objets, avec $n = \frac{m}{2}$. L'adressage ouvert est utilisé pour résoudre les problèmes de collision.

1. En supposant le hachage uniforme, montrer que pour $i = 1, 2, ..n$, la probabilité que la $i$ème insertion nécessite strictement plus de k sondages est au plus de 2^{-k}.
2. Montrer que pour $i = 1, 2, ..., n$, la probabilité que la $i$ème insertion nécessite plus de $2logn$ sondages est au plus de $\frac{1}{n^2}$.
3. On utilise la variable aléatoire X_i pour représenter le nombre de sondages requis pour la $i$ème insertion. On utilise la variable aléatoire $X = max_{1 \leq i \leq n} X_i$ pour représenter le nombre maximum de sondages que nécessite n'importe laquelle des n insertions.
   Montrer que $P(X > 2logn) = \frac{1}{n}$.
4. Montrer que la longueur moyenne $E[X]$ de la plus longue séquence de sondages est en $O(log(n))$.