Algorithmique avancée – Examen Réparti 1
UPMC — Master d'Informatique —
Novembre 2012 – durée 2h

Les seuls documents autorisés sont les polys de cours, ainsi que la copie double personnelle.

1 Questions simples et diverses [6 points]

Question 1.

1. Etant donné une file binomiale contenant n éléments, proposer un algorithme qui retourne la liste triée des éléments (utiliser les primitives associées à la structure de file binomiale). Donner la complexité de cet algorithme dans le pire des cas.
2. Etant donné un arbre bicolore contenant n éléments, proposer un algorithme qui retourne la liste triée des éléments. Donner la complexité de cet algorithme dans le pire des cas.
3. Etant donné une table de hachage contenant n éléments, proposer un algorithme qui retourne la liste triée des éléments. Donner la complexité de cet algorithme dans le pire des cas.

Question 2. Quelles sont les valeurs du nombre minimal d'éléments et du nombre maximal d'éléments que peut contenir un arbre 2-3-4 de hauteur h ?

2 Interclassement de tableaux triés[8 points]

Question 1. Expliquer comment faire l'interclassement de deux tableaux triés de tailles n_1 et n_2 en $n_1 + n_2$ comparaisons au pire.

Question 2. Il s'agit maintenant d'interclasser N tableaux triés, de tailles $n_1 \leq n_2 \leq \ldots \leq n_N$, en les interclassant deux à deux jusqu'à obtenir un seul tableau.

1. Traiter l'exemple de 4 tableaux triés, de tailles respectives 3, 4, 5 et 6
   - en interclassant par ordre croissant des tailles
   - en interclassant par ordre décroissant des tailles

   Quel est dans chaque cas le nombre de comparaisons au pire ?
2. Donner un algorithme qui interclasse N tableaux triés en les interclassant deux à deux, par ordre croissant des tailles.

Question 3.

1. Montrer que cet algorithme est optimal en nombre de comparaisons dans le cas le pire (sous l'hypothèse que l'interclassement de deux tableaux triés de tailles n_1 et n_2 nécessite $n_1 + n_2$ comparaisons).
2. Quelle est la complexité de l'algorithme précédent pour interclasser N tableaux triés de tailles respectives $1, 2, 4, \ldots, 2^N$?

Question 4. On considère la structure "suite de tableaux triés", qui est analogue à celle de file binomiale, à la différence que chaque tournoi binomial est remplacé par un tableau trié : les tailles des tableaux sont deux à deux différentes et toutes les tailles sont des puissances de 2.
Par exemple $S_6 =< [1, 5, 8, 10], [2, 6] >$ est une suite de tableaux triés de 6 éléments.

1. Quelle est la suite de tableaux triés obtenue en ajoutant 3 puis 7 à S_6 ? Décrire l'algorithme d'insertion.
2. Quel est le coût de l'insertion d'un nouvel élément dans une suite de tableaux triés contenant n éléments ? En déduire que la construction par adjonctions successives d'une suite de tableaux triés contenant n éléments a un coût amorti en $O(\log n)$.

3 Appartenance à un ensemble [6 points]

Etant donné un ensemble A de n éléments, l'objectif est de stocker les éléments de A avec peu de mémoire, pour ensuite tester l'appartenance à A de façon probabiliste.
La structure est un tableau T de m bits, et on utilise k fonctions de hachage uniformes et indépendantes $h_1, \ldots, h_k$ à image dans $\{0, \cdots, m-1\}$. Pour "placer" un élément x de A dans cette structure, on met la valeur 1 dans chaque $T[h_i(x)]$, pour $i = 1, \ldots, k$.

Fonction `Consruction` (ensemble A, entier m, fonctions de hachage $h_1, \ldots, h_k$)
T = table de m bits, initialisée à 0
PourChaque a_i dans A
  PourChaque fonction de hachage h_j
    $T[h_j(a_i)] = 1$
  FinPour
FinPour
Retourne T

Ensuite, pour rechercher si un élément y appartient à l'ensemble, on calcule tous les $h_i(y)$ et on vérifie que tous les bits correspondant dan T valent 1.

Fonction `Appartient?` (élément y, table T , fonctions de hachage $h_1, \ldots, h_k$)
PourChaque fonction de hachage h_j
  Si $T[h_j(y)] = 0$ **Alors Retourne** NON
FinPour
Retourne OUI

Lorsque la fonction `Appartient?` retourne NON, c'est que l'élément y n'est pas dans A mais il est possible qu'elle retourne OUI pour un élément qui n'appartient pas à l'ensemble (on parle alors de *faux positif*).

Question 1. Construire la table T pour $A = \{9, 11\}$, avec $m = 5$, $k = 2$, et les fonctions de hachage $h_1(x) = x \mod 5$ et $h_2(x) = 2 \times x + 3 \mod 5$.
Appliquer ensuite la fonction `Appartient?` pour $y = 15$ et $y = 16$. Expliquer le résultat.

Question 2. Etant donné les tables T_A et T_B de taille m, associées à 2 ensembles A et B, en utilisant les mêmes fonctions de hachage $h_1, ..., h_k$. Expliquer comment construire la table associée à l'union $A \cup B$.

Question 3. Dans la suite on considère une table T_A associée à un ensemble A.
Montrer que la probabilité pour qu'un bit donné de la table soit égal à 0 est $p_0 = (1 - 1/m)^{kn}$.
En déduire la probabilité p_r que la fonction `Appartient?` donne un faux positif vaut $(1 - \exp(-kn/m))^k$, pour m grand (on utilisera l'approximation $(1 - 1/m)^{kn} \sim \exp(-kn/m)$).

Question 4. Etant donné un ensemble de n éléments, on peut jouer sur la taille de la table et le nombre de fonctions de hachage pour minimiser la probabilité de faux positifs.
Calculer la valeur de k qui minimise la probabilité p_r, pour m grand.
Discuter du choix des valeurs de k et m.