
Algorithmique Avancée Examen Réparti 2 (1h30)

Seuls documents autorisés : les photocopiés de cours et la copie personnelle. Le barème est indicatif.

Exercice 1 : Trie hybride [2 points]

Le trie hybride est une structure de données permettant de stocker et de rechercher efficacement des mots. Il s'agit d'une structure arborescente dont les nœuds internes sont d'arité 3. La structure a été vue dans le chapitre 3 du cours.

Question 1 Construire le trie hybride résultant de l'insertion de la liste des mots suivants : chien, bol, tri, ballon, chat, thon, bal, salon, rat, achat.

Exercice 2 : Hachage et tri [8 points]

On a un tableau d contenant des numéros d'étudiant à 7 chiffres que l'on souhaite trier dans l'ordre croissant. On utilise pour cela des tables de hachage avec chaînage : lorsqu'il y a collision, on insère *en queue de liste*. On a accès à une fonction `chiffre` prenant en entrée un numéro d'étudiant et un chiffre i et renvoyant le i -ème chiffre du numéro en partant de la droite.

Par exemple `chiffre(1234567, 1) = 7` et `chiffre(1234567, 3) = 5`.

La fonction `chiffre` sera utilisée pour définir diverses fonctions de hachage.

Voilà l'algorithme de tri :

Soit d un tableau contenant les numéros d'étudiant.

Pour i allant de 1 à 7 faire

 Soit t_i une table de hachage de taille 10 dont la fonction de hachage est
 $h_i(k) = \text{le } i\text{-ème chiffre de } k$

 Pour chaque numéro k de d faire

 Ajouter k à t_i

 Fin Pour

Vider d

 Ranger tous les numéros de t_i dans le tableau d

 en parcourant les cases de 0 à 9 et les listes dans l'ordre

Fin Pour

Afficher le contenu de d

Question 1 On va tester l'algorithme sur le tableau d composé des numéros suivants : 3200231, 3200045, 3600529, 3600009, 3300067, 3333333, 3600051, 3600021. Pour chaque étape i , dessiner la table t_i et le tableau d .

Question 2 Prouver la correction de l'algorithme.

Question 3 Après avoir indiqué la mesure de complexité adéquate, calculer la complexité en temps de l'algorithme.

Question 4 Si l'insertion dans les listes se faisait en tête plutôt qu'en queue de liste, l'algorithme renverrait-il une liste triée dans l'ordre croissant ou décroissant ? Prouver le résultat via une preuve de correction ou un contre-exemple.

Question 5 Comment modifier l'algorithme pour trier les numéros dans l'ordre décroissant ?

Exercice 3 : Extraction du maximum d'un arbre 2-3-4 [14 points]

On considère le modèle des arbres 2-3-4 avec éclatements à la descente lors des insertions.

Question 1 Construire l'arbre 2-3-4, noté Ex_1 , obtenu en insérant les clés dans l'ordre suivant : 5, 7, 13, 1, 18, 17, 9, 4, 3, 12, 15. On indiquera toutes les étapes intermédiaires.

Question 2 Donner le pseudo-code d'un algorithme construisant la liste triée, dans l'ordre croissant, des clés d'un arbre 2-3-4. On utilisera obligatoirement les primitives présentées dans le cours.

Question 3 Dans quel nœud d'un arbre 2-3-4 se trouve la clé de valeur maximale ? Justifier la réponse à l'aide d'une induction.

Par la suite, on souhaite extraire la clé de valeur maximale de l'arbre 2-3-4, tout en conservant un arbre de type 2-3-4.

Question 4 Dans quel(s) cas, l'arbre obtenu après une suppression directe de la clé de valeur maximale n'est plus un arbre 2-3-4 ?

Afin de procéder à la suppression de la clé de valeur maximale, on va suivre l'approche suivante : intuitivement, en descendant dans l'arbre à la recherche de la clé, il faut potentiellement effectuer l'inverse des éclatements. On progresse dans l'arbre, depuis la racine vers le nœud contenant la clé recherchée en vérifiant en chaque nœud :

1. Si la clé de valeur maximale est dans le nœud actuel, et qu'il y a au moins 2 clés dans le nœud, alors on supprime simplement la clé.
2. Si le nœud visité ν ne contient pas la clé de valeur maximale, alors on détermine l'enfant E dans lequel continuer la recherche de la clé et
 - a. si E ne contient qu'une seule clé et ν a un autre enfant E' , voisin direct de E , contenant au moins 2 clés, alors on modifie ν , E et E' , afin que E contienne 2 clés et E' une seule clé.
 - b. si E ne contient qu'une clé, et les éventuels autres enfants de ν , voisins directs de E ne contiennent qu'une seule clé aussi, alors on procède à l'inverse d'un éclatement sur ν , E et E' afin que E contienne 3 clés. Ensuite il faut reprendre l'algorithme sur le nœud E .

Question 5 Quel est l'arbre 2-3-4 résultant de l'extraction de 18 dans l'arbre Ex_1 ? Donner les étapes intermédiaires.

Question 6 Construire l'arbre 2-3-4, noté Ex_2 , obtenu en insérant les clés suivantes : 5, 7, 13, 1, 18, 17, 9, 4, 3, 12. Ne pas indiquer les étapes intermédiaires ; on remarque qu'il s'agit de Ex_1 avec une clé en moins.

Puis supprimer sa clé de valeur maximale (en indiquant les étapes intermédiaires).

Question 7 Présenter à travers des schémas les différents cas de figure à traiter pour extraire la clé de valeur maximale d'un arbre 2-3-4 tout en conservant un arbre de type 2-3-4.

Question 8 Donner le pseudo-code de l'extraction de la clé de valeur maximale d'un arbre 2-3-4, renvoyant un couple contenant la clé de valeur maximale et l'arbre 2-3-4 issu de la suppression.

Question 9 Quelle est la mesure de complexité adaptée pour cet algorithme ? Quelle la notion de complexité adéquate cet algorithme (analyse en moyenne, au pire ou amortie) ? Donner des éléments de justifications.

Question 10 Quelle est la complexité de l'algorithme par rapport à la question précédente ?