ENSEIRB 1ière année, 2003/2004, filière informatique

RATTRAPAGE DE STRUCTURES ARBORESCENTES

Durée 2h, documents autorisés, livres interdits

1. Quel est le temps d'exécution du tri par tas sur un tableau A de longueur n déjà trié en ordre croissant ? Et en ordre décroissant ? Justifiez votre réponse en rappelant les étapes du tri et en calculant leur nombre d'opérations dans ces deux cas.
2. Supposons qu'un noeud x soit inséré dans un arbre rouge et noir par la procédure RN-inserer et aussitôt supprimé avec RN-supprimer. L'arbre rouge et noir résultant est-il le même que l'arbre initial ? Justifier la réponse.
3. On considère des ensembles dynamiques de chaînes de bits $a = a_0 a_1 ... a_p$, où $a_i \in \{0, 1\}$ pour $0 \leq i \leq p$ que l'on souhaite stocker en vue d'implémenter efficacement la recherche et le tri. Pour cela, nous allons utiliser une structure d'arbre à base. Un arbre à base est un arbre binaire tel que tout arc issu d'un noeud vers son enfant gauche est étiqueté par la lettre 0 et tout arc issu d'un noeud vers son enfant droit est étiqueté 1. Une chaîne de bits peut alors être codée par un chemin descendant de la racine vers un noeud de l'arbre ayant une profondeur égale au nombre de lettres de la chaîne et tel que la suite des lettres associée au chemin soit égale à celle de la chaîne.

   Donner un exemple d'un ensemble de chaînes de bits et lui associer un codage au moyen d'un arbre à base. Montrer sur l'exemple qu'il est nécessaire de définir la notion de ***noeud vide*** afin de différencier un chemin codant une chaîne de l'ensemble, d'un chemin ne codant aucune chaîne de l'ensemble.

   Écrire un type abstrait correspondant à l'arbre à base, ainsi qu'un type abstrait correspondant à une chaîne de bits qui soient adaptés aux besoins de la recherche et du tri. Proposer une implémentation pour chacun de ces types. Évaluer la complexité des opérations dans le pire des cas.

   Écrire une fonction de recherche d'une clef $a = a_0 a_1 ... a_p$ dans un arbre à base. Évaluer la complexité de cette opération dans le pire des cas.

   Soit S un ensemble de chaînes de bits distinctes dont la somme des longueurs (c'est-à-dire leur nombre de lettres) est égale à n. Montrer que l'on peut trier lexicographiquement S en $O(n)$.