

Algorithmique et programmation 3

EXERCICES

5-Arbres ordonnés

Exercice 5-1 : premiers algorithmes sur les arbres ordonnés

Pour chaque question, écrire l'algorithme demandé et donner sa complexité en temps (après avoir défini la taille du problème).

- Restituer la plus grande (resp. la plus petite) clé d'un ABR non vide. La plus grande clé est-elle dans une feuille ?
- Ecrire une version itérative de la recherche du max.
- Restituer la clé qui précède (resp. qui suit) une clé C de l'arbre dans l'hypothèse où C n'est pas la plus petite (resp. plus grande) clé dans l'arbre.
- (maison) Même question mais : on sait que C est strictement supérieure au min de l'arbre mais pas forcément présente. La valeur restituée doit être la plus grande des clés de l'arbre qui sont inférieures à C .

Exercice 5-2 : insertion dans un arbre ordonné

- Ecrire en pseudo-code une version récursive fonctionnelle de l'insertion dans un arbre ordonné en veillant à ne pas gâcher de mémoire.
- (maison) Ecrire en pseudo-code une version itérative fonctionnelle.
- (maison) L'insertion est-elle toujours réalisée sous une feuille ?

Exercice 5-3 : suppression dans un arbre ordonné

Ecrire une version procédurale récursive de la suppression dans un arbre ordonné.

Exercice 5-4 : insertion à la racine d'un arbre ordonné

- Analyser puis écrire un sous-programme qui sépare efficacement un arbre ordonné en deux arbres ordonnés : celui des clés inférieures à une certaine valeur de référence (le pivot), celui des clés supérieures au pivot. Le pivot est une clé absente de l'arbre. Votre méthode ne doit pas se contenter d'insérer une à une les clés de l'arbre initial dans l'un ou l'autre des arbres résultats selon qu'elles sont inférieures ou supérieures au pivot.
- En déduire une fonction d'insertion d'une clé à la racine d'un arbre ordonné.