

Algorithmique TD 6 : Arbres et arbres binaires de recherche

Exercice 1. Parcours d'arbres

Écrire des algorithmes itératifs effectuant le parcours d'un arbre

a. en profondeur préfixe

b. en largeur.

On pourra utiliser des structures de donnée dynamiques comme les piles, files, etc.

Exercice 2. Parcours infixe, préfixe, postfixe

Pour chaque problème suivant, quel parcours en profondeur vous semble le plus adapté :

a. former un tableau trié contenant les éléments d'un arbre binaire de recherche,

b. copier efficacement un arbre binaire de recherche,

c. organiser une grande consultation nationale (on supposera que la démocratie participative idéale est basé sur une hiérarchie de niveaux de représentativité connectée en arbre).

Exercice 3. Arbres binaires de recherche

a. Écrire les algorithmes de recherche dans un arbre binaire de recherche

1. d'un élément donné,

2. du minimum, du maximum,

3. du successeur d'un élément donné, de son prédécesseur,

4. de l'élément médian.

b. Écrire l'algorithme d'insertion d'un élément dans une feuille d'un arbre binaire de recherche.

c. Proposer un algorithme de tri sur un tableau, qui utilise en interne un arbre binaire de recherche.

d. Écrire un algorithme **Coupe** qui, à partir d'un élément x , découpe un arbre binaire de recherche en deux arbres binaires de recherche contenant des éléments inférieurs (respectivement supérieurs) à x . En déduire un algorithme d'insertion à la racine

e. Écrire un algorithme effectuant la suppression d'un élément d'un arbre binaire de recherche s'il existe. On distinguera les trois cas, où l'élément à supprimer possède 0, 1 ou 2 descendants.

Exercice 4. Tri par tas

a. Ré-écrire l'algorithme de tri par tas quand le tas est représenté par un tableau où pour tout noeud $T[i]$, son fils gauche est l'élément $T[2i + 1]$ et son fils droit l'élément $T[2i + 2]$.

b. Tracer l'exécution de cet algorithme sur le tableau $[5, 8, 6, 3, 7, 10, 2, 1, 12, 4, 11, 9]$

Exercice 5. File à priorité

a. Implémenter une file à priorité à partir d'un tas. On utilisera la représentation en tableau d'un tas, et on supposera disposer d'une fonction `Priorité` qui retourne pour chaque élément un entier positif indiquant son niveau de priorité.