Université de Batna 2
Département Informatique
Module : Algorithmique et structures de données

Série V : Arbres

Exercice 1 : La numérotation en ordre hiérarchique d'un arbre binaire consiste à numéroter en ordre croissant à partir de 1, tous les nœuds possibles d'un arbre binaire, à partir de la racine, niveau par niveau, et de gauche à droite sur chaque niveau. Écrire un sous-programme qui produit la liste des nœuds d'un arbre binaire dans l'ordre hiérarchique.

Exercice 2 : Écrire pour chaque représentation d'un arbre binaire (chaînée ou contiguë)
a/ Une fonction qui calcule la hauteur d'un arbre binaire.
b/ Une fonction qui calcule la longueur de cheminement d'un arbre binaire (on ne supposera pas connue à l'avance la taille de l'arbre).
c/ Une fonction qui indique pour chaque nœud de l'arbre le nombre de ses descendants.

Exercice 3
a/ Écrire une fonction qui teste l'égalité de deux arbres binaires étiquetés.
b/ Un arbre binaire S figure dans un arbre binaire T s'il lui est égal ou s'il est égal à un sous-arbre de T. Écrire une fonction récursive figure(S, T) qui indique si S figure dans T; si c'est le cas, la fonction renvoie l'adresse du sous-arbre qui est égal à S et sinon elle renvoie NULL.

Exercice 4 : On définit récursivement une fonction entière φ sur l'ensemble des arbres binaires. Appelons g(A) et d(A) respectivement le sous-arbre gauche et le sous-arbre droit de A.

$$\varphi(A) = \begin{cases} 0 & \text{si A est l'arbre} - \text{vide} \\ \max(\varphi(g(A)), \varphi(d(A))) & \text{si } \varphi(g(A)) \neq \varphi(d(A)) \\ \varphi\big(d(A)\big) + 1 & \text{si } \varphi(g(A)) = \varphi(d(A)) \end{cases}$$

Écrire une fonction récursive basée sur le parcours suffixe qui calcule φ.

Exercice 5
a/ Donner les différentes formes de représentation linéaire des arbres binaires.
b/ Est-il possible de reconstruire un arbre binaire à partir de sa forme linéaire? Dans le cas contraire, quelles informations supplémentaires faut-il ajouter?
c/ Écrire les sous-programmes de linéarisation.

Exercice 6 : On considère les arbres planaires généraux. Une représentation possible de ces arbres est la suivante : une liste de listes. Chaque nœud de la liste de base représente un nœud de l'arbre. Ensuite, à chaque nœud de l'arbre, on associe la liste de ses fils.
1) Donner une structure de données respectant cette représentation. En utilisant cette structure de données, écrire :
2) La fonction `element_de()` qui teste l'appartenance d'un nœud à un arbre planaire.
3) La fonction `pere()` qui retourne l'adresse du père d'un nœud dans un arbre planaire.
4) La fonction `hauteur_noeud()` qui calcule la hauteur (niveau) d'un nœud dans un arbre planaire.
5) La fonction `hauteur()` qui calcule la hauteur d'un arbre planaire.
6) La fonction `ancetres()` qui trouve tous les ancêtres d'un nœud. Cette fonction retourne un pointeur sur une liste (à créer) des ancêtres.
7) La fonction `descendants()` qui trouve tous les descendants d'un nœud. Cette fonction retourne un pointeur sur une liste (à créer) des descendants.