UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

LICENCE Informatique
Info0501 – 2018/2019

Travaux dirigés n° 3

Arbres

Exercice 1 (Arbres binaires)

1°) En spécifiant les attributs de l'arbre et des cellules, dessinez l'arbre binaire dont l'indice de la racine est 6 et qui est représenté par les attributs du tableau ci-contre.

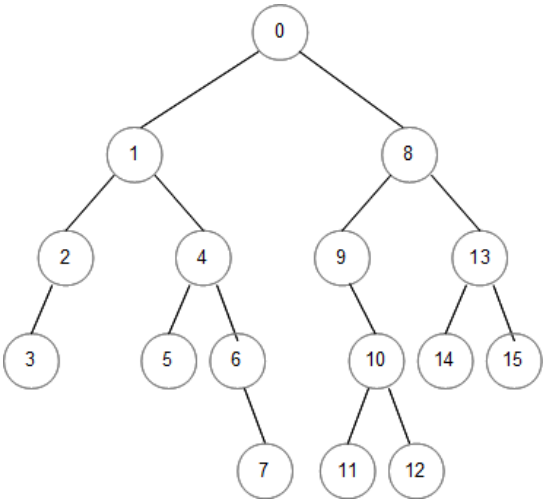
indice	*cle*	*gauche*	*droite*
1	12	7	3
2	15	8	NIL
3	4	10	NIL
4	10	5	9
5	2	NIL	NIL
6	18	1	4
7	7	NIL	NIL
8	14	6	2
9	21	NIL	NIL
10	5	NIL	NIL

2°) Définissez une procédure en temps $O(n)$ qui, étant donné un arbre binaire à n noeuds, affiche la clé de chaque noeud de l'arbre

a) de façon récursive ;

b) de façon non récursive.

3°) Déduisez le temps d'exécution de la recherche d'un noeud de clé donnée dans un arbre binaire.



4°) Soit l'arbre binaire ci-contre, donnez l'ordre d'affichage des noeuds en utilisant un parcours

a) infixe ;

b) préfixe ;

c) postfixe.

5°) Définissez les trois procédures de parcours de la question précédente en donnant une version récursive et une version non récursive.

Exercice 2 (Arbres binaires de recherche)

1°) Quelle propriété un arbre binaire doit-il satisfaire pour être un arbre binaire de recherche ?

2°) En utilisant l'ensemble de clés {2, 5, 5, 6, 7, 8}, dessinez un arbre binaire de recherche

a) de hauteur 2 ;

b) de hauteur 4.

3°) Soit l'arbre binaire de recherche ci-contre, donnez le chemin d'interrogation suivi pour trouver

a) la clé 13 en partant de la racine ;

b) la clé minimale ;

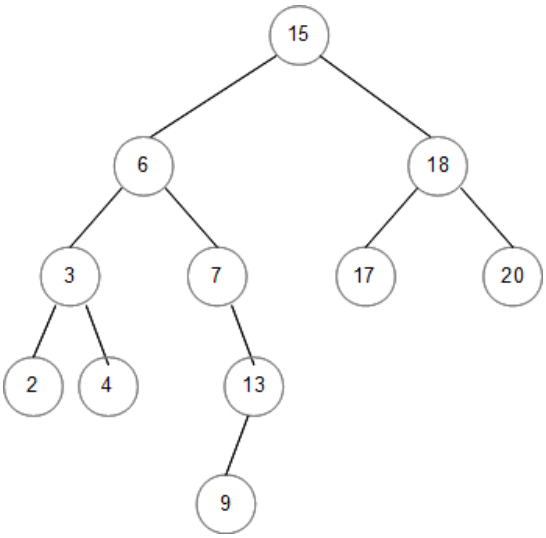
c) la clé maximale ;

d) le successeur du noeud de clé 15 (le successeur d'un noeud est le noeud qui suit ce noeud dans l'ordre déterminé par un parcours infixe de l'arbre. Autrement dit, si toutes les clés sont distinctes, le successeur d'un noeud x est le noeud possédant la plus petite clé supérieure à $x.cle$.) ;

e) le successeur du noeud de clé 13 ;

f) le prédéceseur du noeud de clé 20 (le prédécesseur d'un noeud est le noeud qui précède ce noeud dans l'ordre déterminé par un parcours infixe de l'arbre. Autrement dit, si toutes les clés sont distinctes, le prédécesseur d'un noeud x est le noeud possédant la plus grande clé inférieure à $x.cle$) ;

g) le prédécesseur du noeud de clé 6.



4°) Définissez les procédures suivantes sur un arbre binaire de recherche et donnez leur temps d'exécution :

a) `afficher-arbre-trié` : affiche toutes les clés de l'arbre dans l'ordre trié ;

b) `rechercher-arbre-récursif` : recherche un noeud ayant une clé donnée. Étant donné un pointeur sur la racine de l'arbre et une clé k, retourne un pointeur sur un noeud de clé k s'il en existe un et NIL sinon ;

c) `rechercher-arbre-itératif` : même question que la précédente, mais cette fois sans appel récursif ;

d) `minimum-arbre` : retourne un pointeur sur le noeud dont la clé est un minimum de l'arbre ;

e) `maximum-arbre` : retourne un pointeur sur le noeud dont la clé est un maximum de l'arbre ;

f) `successeur-arbre` : retourne un pointeur sur le successeur d'un noeud ou NIL s'il possède le maximum ;

g) `prédécesseur-arbre` : retourne un pointeur sur le prédécesseur d'un noeud ou NIL s'il possède le minimum.

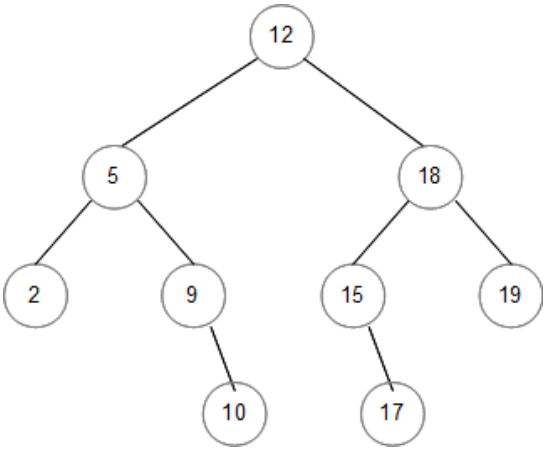
5°) Soit l'arbre binaire de recherche ci-contre, décrivez le fonctionnement

a) de l'insertion d'un noeud de clé 13 ;

b) de la supression du noeud de clé 15 ;

c) de la suppression du noeud de clé 5 ;

d) de la suppression du noeud de clé 12.



6°) Définissez les procédures suivantes sur un arbre binaire de recherche et donnez leur temps d'exécution :

a) `insérer-arbre` : insère un noeud dans l'arbre en conservant la propriété d'arbre binaire de recherche ;

b) `transplanter` : remplace un sous-arbre, considéré comme enfant de son parent, par un autre sous-arbre ;

c) `supprimer-arbre` : supprime un noeud de l'arbre en conservant la propriété d'arbre binaire de recherche.

Références

Cormen T. H., Leiserson C. E., Rivest R. L. et Stein C., "Algorithmique" 3e édition, Dunod, 2010.