

SÉRIE N° 8 (SUITE): ARBRES BINAIRES-RÉVISION

Exercice 1:

Dans cet exercice, on cherche à stocker un ensemble de contacts téléphoniques dans un arbre binaire de recherche.

Un contact téléphonique est composé d'un nom (représenté par une chaîne de caractères) et d'un numéro de téléphone (représenté par un entier).

Un contact téléphonique sera représenté dans cet exercice par un dictionnaire

Exemple :

D={nom : ' bennani', tel : '212623145623'}

Définition d'un arbre binaire de recherche :

- L'arbre contient deux types d'éléments : des nœuds (chacun contenant un contact et deux sousarbres fils) et des feuilles (qui ne contiennent aucune information).
- Lorsqu'un nœud contient un contact ayant le numéro de téléphone n, tous les contacts ayant un numéro de téléphone strictement inférieur à n se trouvent dans le sous-arbre gauche, et tous les contacts ayant un numéro de téléphone strictement supérieur à n se trouvent dans le sous-arbre droit

- 1) Écrivez une fonction **listeContacts** qui prend en paramètre un arbre et qui retourne la liste des contacts de cet arbre, triée par ordre croissant des numéros de téléphone.
- 2) Écrivez une fonction **rechercheContact** qui prend en paramètre un arbre et un numéro de téléphone, et qui retourne le nom du contact ayant ce numéro de téléphone, ou la chaîne de caractères vide sinon. Votre fonction doit profiter des caractéristiques de l'arbre binaire de recherche.
- 3) Écrivez une fonction **MaxContact** qui prend en paramètre un arbre et qui retourne le contact qui le plus grand numéro de téléphone
- 4) Écrivez une fonction **ajouteContact** qui prend en paramètre un arbre et un contact, et qui ajoute ce contact dans l'arbre. L'ajout doit se faire en conservant les propriétés de l'arbre binaire de recherche.

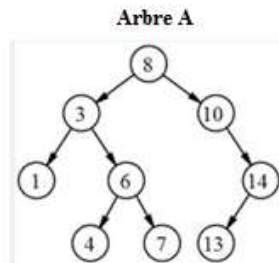


Exercice 2:

1. Ecrire la fonction **Croissant(A)** qui prend en paramètre un arbre binaire de recherche A et qui retourne une liste croissante des valeurs de l'arbre

Rq: Ne pas utiliser un algorithme de tri

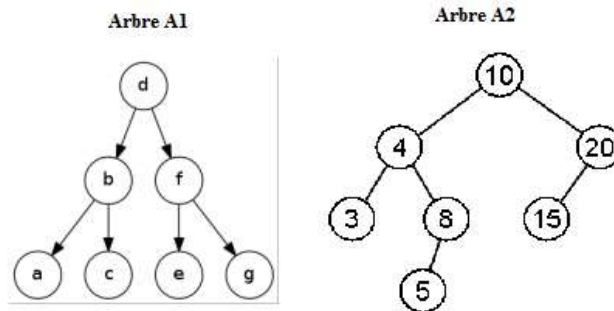
Exemple :



L'appel de la fonction **L= Croissant(A)** retourne la liste suivante : **L=[1,3,4,6,7,8,10,13,14]**

2. Ecrire la fonction **Complet(A)** qui prend en paramètre un arbre binaire A et qui retourne un booléen indiquant si A est un arbre binaire parfait complet ou non

Exemple :



L'appel de la fonction **Complet(A1)** retourne **True** et l'appel de la fonction **Complet(A2)** retourne **False**

Exercice 3:

On considère les expressions arithmétiques sur les entiers n'utilisant que les opérateurs +, -, x, et ÷. Ces expressions peuvent être représentées par des arbres binaires dont les nœuds internes (nœuds non vides qui ne sont pas des feuilles) sont étiquetés par l'un des quatre opérateurs, tandis que les feuilles sont étiquetées par des entiers.

Par exemple, l'arbre représenté sur la figure 1 représente l'expression arithmétique $(3 - 2) \times (7 + 10 \div 2)$

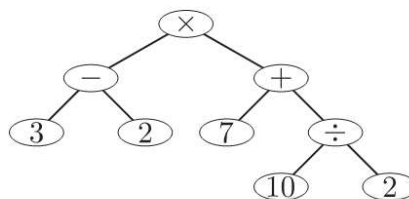


FIGURE 1 – Représentation de l'expression $(3 - 2) \times (7 + 10 \div 2)$ par un arbre binaire.



Remarque :

Les opérateurs sont représentés par la chaîne de caractères qui leur correspond. Par exemple, l'opérateur + est représenté par '+' :

- 1) écrivez la fonction **valide(a)** qui teste si un arbre binaire a ne contenant que des entiers et des opérateurs, représente bien une expression arithmétique valide (chaque nœud étiqueté par un opérateur doit avoir deux fils non vides et les entiers ne doivent apparaître que sur les feuilles de l'arbre).

Indication : On peut vérifier que la condition est vraie pour la racine puis tester récursivement si les sous-arbres droit et gauche sont également valides

- 2) Écrivez récursivement la fonction **eval(a)** qui renvoie la valeur correspondant à l'expression arithmétique représentée par l'arbre a (par exemple, sur l'arbre illustré par la figure 1, la fonction eval(a) doit renvoyer 12). On suppose ici que l'arbre représente une expression valide.

Indication : Si le nœud considéré contient un entier, l'évaluation en ce nœud doit renvoyer la valeur de l'entier, si par contre le nœud contient par exemple un '+', l'évaluation doit renvoyer le résultat de l'évaluation du fils gauche plus le résultat de l'évaluation du fils droit..