Université Grenoble Alpe
U.F.R. Sciences de l'Homme et de la Société
Master MIASHS - DCISS

T.D. Algorithmique n° 17

Arbres binaires

Exercice 1 : Profondeur

Écrire la fonction **profondeur** qui calcule la profondeur d'un arbre binaire, c'est-à-dire la longueur du plus long chemin allant de la racine vers une feuille.

fonction **profondeur**(a : Arbre) → entier ≥ 0
// profondeur(a) renvoie la longueur du plus long chemin allant de la racine a à une feuille

Exercice 2 : Plus court chemin vers une feuille

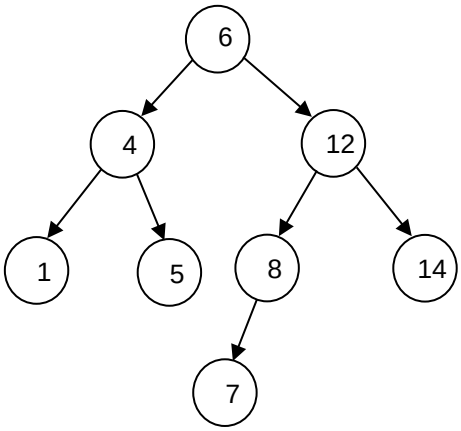
Écrire une fonction **profmin** qui calcule la longueur du plus court chemin allant de la racine jusqu'à une feuille.

fonction **profmin**(a : Arbre) → entier ≥ 0
// profondeur(a) renvoie la longueur du plus court chemin allant de la racine a jusqu'à une feuille

Exercice 3 : Création d'un arbre binaire de recherche

Ecrire un algorithme qui à partir séquence d'entiers représentée dans un fichier, construit un arbre binaire dont les nœuds sont les éléments de la séquence, et qui vérifie les propriétés suivantes :
- tout nœud situé dans un sous-arbre gauche a une valeur inférieure ou égale à celle de la racine
- tout nœud situé dans un sous-arbre droit a une valeur plus grande que celle de la racine.

Par exemple, à partir de la séquence **6, 12, 4, 8, 14, 5, 7, 1**
On obtient l'arbre binaire suivant :



Pour ajouter un élément à l'arbre, écrire l'action **insérerABR** qui insère un entier dans un arbre binaire de recherche :

action **insérerABR**(consulté e : entier ; modifié a : Arbre)
// Effet : insère l'entier n dans l'arbre binaire de recherche a