

EXERCICE 3 (6 points)

Cet exercice porte sur les arbres binaires, les files et la programmation orientée objet.

Cet exercice comporte deux parties indépendantes.

PARTIE 1

Une entreprise stocke les identifiants de ses clients dans un arbre binaire de recherche.

On rappelle qu'un arbre binaire est composé de nœuds, chacun des nœuds possédant éventuellement un sous-arbre gauche et éventuellement un sous-arbre droit.

La taille d'un arbre est le nombre de nœuds qu'il contient. Sa hauteur est le nombre de nœuds du plus long chemin qui joint le nœud racine à l'une des feuilles. On convient que la hauteur d'un arbre ne contenant qu'un nœud vaut 1 et celle de l'arbre vide vaut 0.

Dans cet arbre binaire de recherche, chaque nœud contient une valeur, ici une chaîne de caractères, qui est, avec l'ordre lexicographique (celui du dictionnaire) :

- strictement supérieure à toutes les valeurs des nœuds du sous-arbre gauche ;
- strictement inférieure à toutes les valeurs des nœuds du sous-arbre droit.

Ainsi les valeurs de cet arbre sont toutes distinctes.

On considère l'arbre binaire de recherche suivant :

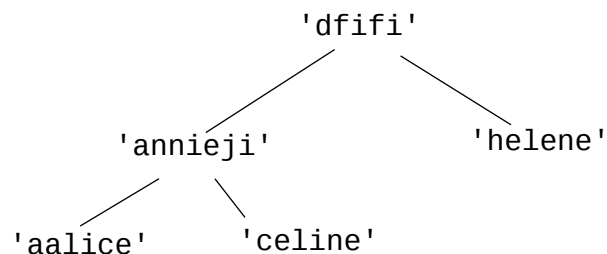


Figure 1. Arbre binaire de recherche.

1. Donner la taille et la hauteur de l'arbre binaire de recherche de la figure 1.
2. Recopier cet arbre après l'ajout des identifiants suivants : 'davidbg' et 'papicoeur' dans cet ordre.
3. On décide de parcourir cet arbre pour obtenir la liste des identifiants dans l'ordre lexicographique.

Recopier la lettre correspondant au parcours à utiliser parmi les propositions suivantes :

- A** - Parcours en largeur d'abord
- B** - Parcours en profondeur dans l'ordre préfixe
- C** - Parcours en profondeur dans l'ordre infixé
- D** - Parcours en profondeur dans l'ordre suffixe (ou postfixé)