

Partiel d'Algorithmique de L3 (AL5) éléments de corrections

 corrigé partiel et non officiel,
à lire d'un œil critique 

Exercice 1 :

Question 1 (facile) :

Méthode

La méthode est déjà détaillée au début du sujet avec un exemple :

- *profondeur d'un nœud* = sa distance par rapport à la racine
- *profondeur moyenne des nœuds* =
[somme des profondeurs des nœuds] / [nombre de nœuds]
(c'est la définition classique de la moyenne, aucun piège)

Correction

$$\text{Profondeur moyenne} = (0+1+1+2+2+3+3)/7 = 12/7$$

Question 2 :

Méthode

Nombre de nœuds dans un arbre de racine $a =$

$$\begin{aligned} & 1 \text{ (la racine } a \text{ elle même)} \\ & + \text{ nombre de nœuds dans le fils gauche de } a \\ & + \text{ nombre de nœuds dans le fils droit de } a \end{aligned}$$

On en déduit un algorithme récursif, qu'on écrira en pseudo-code, en utilisant une syntaxe proche de celle des corrigés de TD, avec les fonctions suivantes :

- $\text{clef}(a) =$ clef de a (si ce n'est pas une feuille)
- $G(a) =$ fils gauche de a
- $D(a) =$ fils droit de a
- $\text{estVide}(a) =$ Vrai si a est une feuille/nœud externe (sans clef et sans fils)
Faux si a est un nœud interne (avec une clef et 2 fils)

Correction

Procédure NbNoeuds(ABR a) :

début

si (estVide(a)) **alors**

retourner 1;

sinon

retourner (1 + NbNoeuds($G(a)$) + NbNoeuds($D(a)$));

fin

Question 3 :

3.1) Écrire l'algorithme

Méthode

On raisonne comme pour la question précédente

Somme des profondeurs des nœuds dans un arbre de racine a qui est à une profondeur $p = p$ (la profondeur de a lui même, donc 0 s'il est la racine de l'arbre)

 + la somme des profondeur des nœuds dans le fils gauche de a

 + la somme des profondeur des nœuds dans le fils droit de a

On en déduit un algorithme qui est le même que le précédent, avec un argument p en plus, en remplaçant les 1 par des p , et en faisant augmenter la profondeur de 1 lorsqu'on passe de a à l'un de ses fils.

Puis on appelle cet algorithme, dans une autre fonction par exemple, avec la racine a comme nœud de départ, et 0 comme argument pour la profondeur p (puisque la racine est à profondeur 0).

Correction

```
Procédure FonctionAux(ABR a, entier p) : //Aux pour Auxiliaire
début
  si (estVide(a)) alors
    retourner p;
  sinon
    retourner (p + FonctionAux(G(a), p+1) + FonctionAux(D(a), p+1));
fin
```

```
Procédure SommeProf(ABR a) :
début
  retourner FonctionAux(a, 0);
fin
```

3.2) Appliquer l'algorithme

Méthode

Pour montrer qu'on applique cet algorithme sur les arbres donnés, on peut par exemple dessiner les appels de fonctions successifs, mais donner simplement le résultat retourné par l'algo était peut-être suffisant.

(sur ces schémas, on notera FA pour FonctionAux, on désignera les nœuds sans numéro par "fils gauche/droit de..." pour les reconnaître)

Correction

- 3ème arbre :

SommeProf(feuille unique)

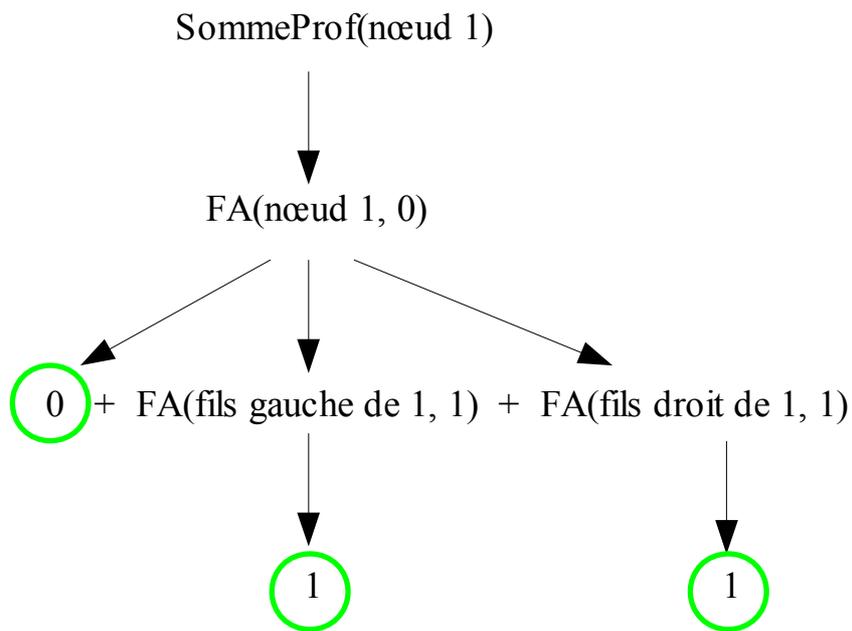


FA(feuille unique, 0)



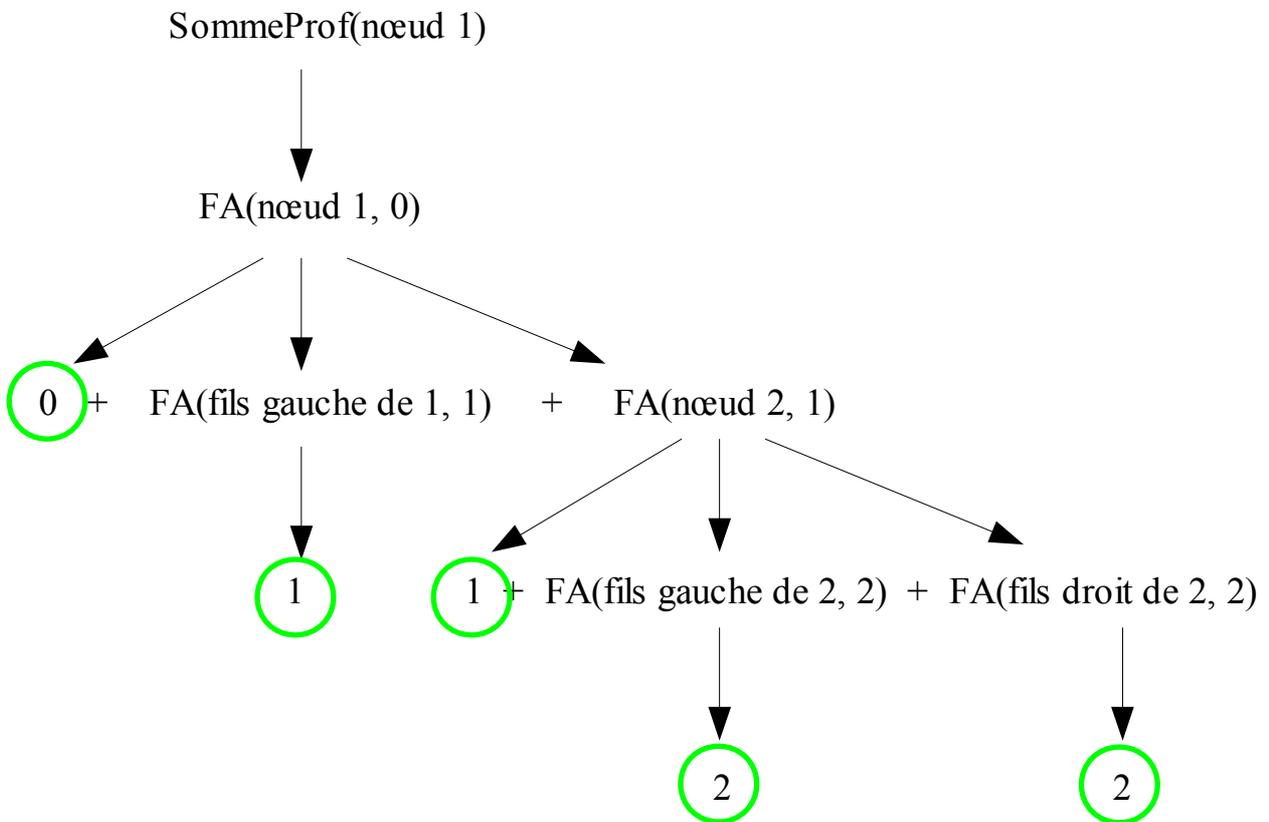
Somme des profondeurs des nœuds = 0

- 2ème arbre :



Somme des profondeurs des nœuds = $0 + 1 + 1 = 2$

- 1er arbre :



Somme des profondeurs des nœuds = $0 + 1 + 1 + 2 + 2 = 6$

Question 4 (facile) :

Méthode

Par définition, la profondeur moyenne est la somme des profondeurs (qu'on sait calculer grâce à la question 3) divisée par le nombre de nœuds (qu'on sait aussi calculer, grâce à la question 2). Il suffit donc d'écrire une fonction qui divise l'un par l'autre les 2 résultats renvoyés par SommeProf et NbNoeuds.

Remarque : Même si les questions 2 et 3 n'étaient pas traitées, on pouvait quand même utiliser les fonctions NbNoeuds et SommeProf pour faire la question 4.

Correction

Procédure ProfMoyenne(ABR a) :

début

retourner (SommeProf(a) / NbNoeuds(a));

fin

Question 5 :

Méthode

Cette question est la même que la question 4 avec une petite variation. Le plus simple est donc de la résoudre comme on a résolu la question 4 : en réécrivant les fonctions NbNoeuds et SommeProf de façon à ce qu'elles ne prennent en compte que les nœuds internes.

Dans la nouvelle fonction NbNoeudsInt, les feuilles ne devront plus renvoyer 1, mais 0 (puisqu'elles ne comptent plus).

De même, dans la nouvelle fonction FonctionAuxInt, les feuilles ne devront plus renvoyer leur profondeur p, mais 0.

Correction

(version "interne" de la question 2)

Procédure NbNoeudsInt(ABR a) :

début

si (estVide(a)) **alors**

retourner 0;

 //il n'y a que cette ligne qui change (0 au lieu de 1)

sinon

retourner (1 + NbNoeudsInt(G(a)) + NbNoeudsInt(D(a)));

fin

