

Leçon 2 : Les arbres

Université Virtuelle de Côte d'Ivoire



Table des matières

I - 1- Généralité sur les arbres	3
II - Application 1 :	5
III - 2- Conception de la structure Arbre	6
IV - Application 2 :	8
Ressources annexes	9
Solutions des exercices	10

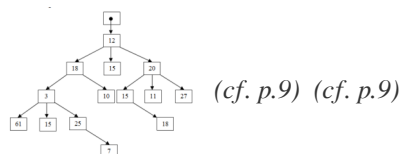
1- Généralité sur les arbres

I

1.1- Arbre

Un *ARBRE*, appelé aussi *ARBRE N-AIRE*, est un ensemble de nœuds permettant de définir une hiérarchie sans cycle.

Dans un *arbre N-aire*, chaque *nœud* possède au maximum *N nœuds suivants*. La représentation d'un arbre en informatique se fait à l'envers : la *racine* se trouve en haut et les *branches* se développent vers le bas.



L'arbre ci-dessus est un arbre 3-aire de 14 nœuds

1.2- Un nœud

Un *NŒUD*, appelé aussi *SOMMET*, contient un *ÉLÉMENT* et indique les *NŒUD SUIVANTS*.

Pour reprendre l'image d'un arbre généalogique, nous parlerons de nœuds fils, nœud père et nœud frère

1.3- La racine

La *RACINE* d'un arbre est le *nœud initial*. Tous les autres nœuds de l'arbre suivent directement ou indirectement la racine.

La racine est le seul nœud sans père.

1.4- Une Feuille

Une *FEUILLE* est un nœud qui n'a pas de suivant.

Si un arbre n'a qu'un nœud, il s'agit de la racine qui est aussi une feuille

1.5- Une Branche

Une *BRANCHE* est un chemin qui rejoint deux nœuds.

1.6- La Hauteur d'un nœud

La *HAUTEUR* d'un nœud est égale au nombre de branches le séparant de la feuille la plus éloignée plus un.

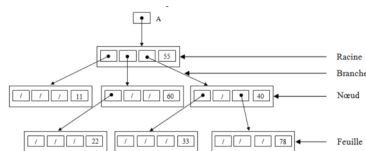
La hauteur d'un arbre vaut alors la hauteur du nœud racine.

1.7- La profondeur d'un nœud

La *PROFONDEUR* d'un nœud est égale au nombre de branches le séparant de la racine plus 1.

La profondeur de la racine vaut 1

1.8- Illustration des définitions



Hauteur de l'arbre est $2 + 1 = 3$

Application 1 :

II

Exercice

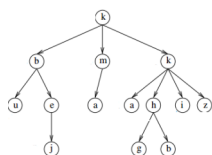
[solution n°1 p.10]

Un arbre est un ensemble

- ☐ de branches reliées entre elles
- ☐ de nœuds permettant de définir une hiérarchie
- ☐ de feuilles permettant de définir une
- ☐ hiérarchie

Exercice

[solution n°2 p.10]



L'arbre ci-dessus est un arbre de nœuds

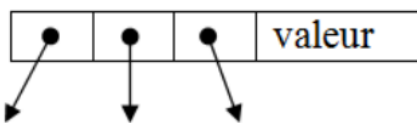
La racine de l'arbre est

La hauteur est

2- Conception de la structure Arbre

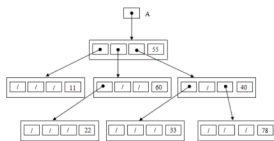


Comme pour une liste chaînée, un *nœud* permet d'accéder aux nœuds fils. Le nœud peut référencer jusqu'à trois nœuds suivants. Ceux-ci pourront être stockés dans trois attributs nommés gauche, milieu et droit ou dans un tableau de trois éléments.



Représentation d'un nœud

Tout comme la liste chaînée a été construite, il est simple d'utiliser la structure nœud pour construire un arbre :



Pour définir les variables utilisées dans l'exemple ci-dessus, il faut :

- Définir le type des éléments de l'arbre :

TYPE Nœud = Structure

gauche : *Arbre*

milieu : *Arbre*

droit : *Arbre*

Info : *Entier*

Fin Structure

- Définir le type du pointeur :

TYPE Arbre = ^Nœud

- Déclarer une variable pointeur :

Var

un_arbre : Arbre

- Allouer une cellule mémoire qui réserve un espace en mémoire et donne à un_arbre la valeur de l'adresse de l'espace mémoire
Allouer(un_arbre)
- Libérer une cellule mémoire qui était occupé par l'élément un_arbre
Desallouer(un_arbre)

- Pour récupérer l'adresse d'une Noeud il faut :
- *variable* \leftarrow @variable
- affecter des valeurs à l'espace mémoire un_arbre :
un_arbre^.Info \leftarrow 78
un_arbre^.gauche \leftarrow Nil
un_arbre^.milieu \leftarrow Nil
un_arbre^.droit \leftarrow Nil

- *Remarque :*

Il est d'ailleurs possible de représenter un arbre avec une écriture standard utilisant des parenthèses et des crochets. Chaque arbre est alors représenté par une parenthèse ouvrante, l'information du sommet, suivie de ses sous arbres suivants entre crochets.

Notons que chaque sous arbre est lui-même un arbre qui utilise la même notation.

L'écriture finale est réflexive.

Exemple Création de l'arbre illustrant le schéma ci-dessus :

(55, [premier sous-arbre]-[deuxième sous-arbre]-[troisième sous-arbre]) //Création de la racine et du premier niveau de l'arbre

Le premier sous-arbre s'écrit : (11, []-[]-[]). //Création du premier sous-arbre simplement car il constitue une feuille

Le deuxième sous-arbre s'écrit : (60,[(22, []-[]-[])-[]-[]). //Création du deuxième sous-arbre avec sa descendance 22

Le troisième sous-arbre s'écrit : (40,[(33, []-[]-[])]-[(78, []-[]-[])]-[]) //Création du troisième sous-arbre avec ses descendances 33 et 78

Ce qui donne finalement :

(55 ,
[(11, []-[]-[])]-
[(60,[(22, []-[]-[])]-[]-[])-
[(40,[(33, []-[]-[])]-[(78, []-[]-[])]-[])]).

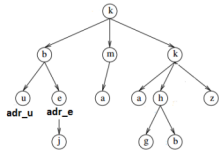
Ce parcours est appelé parcours PRÉFIXE.

Remarque

Dans le cas d'un arbre 3-aire. Chaque nœud possède au plus 3 sous arbres que nous appellerons respectivement gauche, milieu et droit. Pour un arbre N-aire de taille supérieure, il aurait été plus pratique de stocker les sous arbres dans un tableau. Avec cette conception d'arbre, l'utilisateur de la classe doit faire attention pour construire l'arbre désiré.

IV

[solution n°3 p.10]

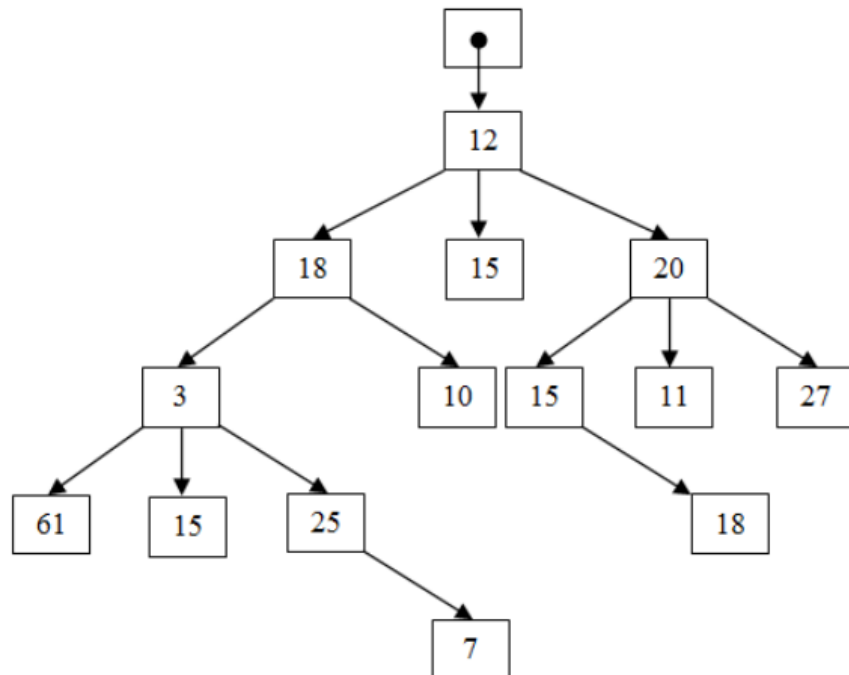


Algorithme CréationNoeud

Fin

Ressources annexes

>



Solutions des exercices



> Solution n°1

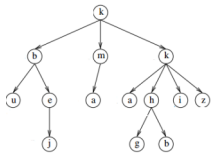
Exercice p. 5

Un arbre est un ensemble

- ☐ de branches reliées entre elles
- ☒ de nœuds permettant de définir une hiérarchie
- ☐ de feuilles permettant de définir une hiérarchie
- ☐ hiérarchie

> Solution n°2

Exercice p. 5



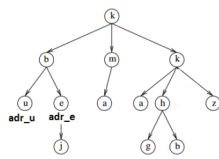
L'arbre ci-dessus est un arbre de 14 nœuds

La racine de l'arbre est k

La hauteur est 3

> **Solution n°3**

Exercice p. 8



Soit le graphique ci-dessus, Complétez l'algorithme suivant pour la création du nœuds b.

Algorithme CréationNœud

TYPE Nœud = Structure

gauche : Arbre

milieu : Arbre

droit : Arbre

val : caractère

Fin Structure

TYPE Arbre = ^Nœud

Var

un_arbre_b : Arbre

un_arbre_u : Arbre

un_arbre_e : Arbre

Début

Allouer(un_arbre_b)

un_arbre_b^.val ← 'b'

un_arbre_b^.gauche ← @un_arbre_e

un_arbre_b^.milieu ← Nil

un_arbre_b^.droit ← @un_arbre_u

Fin