

Les arbres binaires

Chapitre 10: les arbres binaires

77

Motivation

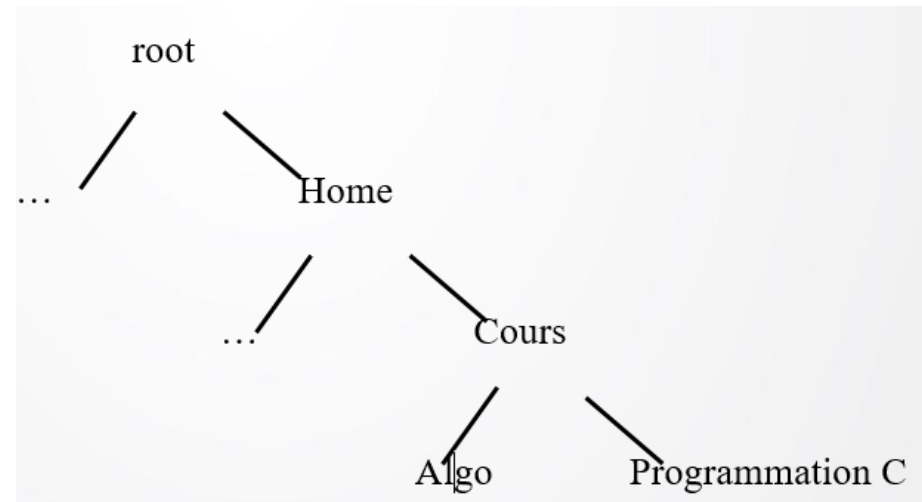
- Les structures (Tableaux, listes) sont des structures linéaires:
 - ▣ Les données sont organisées de manière ordonnée les uns à la suite des autres
 - ▣ Pour chercher un élément, nous sommes obligés de parcourir toute la structure de donnée jusqu'à le trouver.
- ➔ La recherche d'un élément dans un arbre binaire de recherche est beaucoup plus rapide que la recherche dans une structure de donnée linéaire.

Chapitre 10: les arbres binaires

78

Définition

- Un arbre est une structure de données composée d'un ensemble de nœuds.
- Chaque nœud contient les données spécifiques de l'application et des pointeurs vers d'autres nœuds (d'autres sous-arbres).
- Plusieurs traitements en informatique sont de nature arborescente tel que:
 - ▣ La représentation des expressions arithmétiques,.. Etc.
 - ▣ La hiérarchie des répertoires et des fichiers

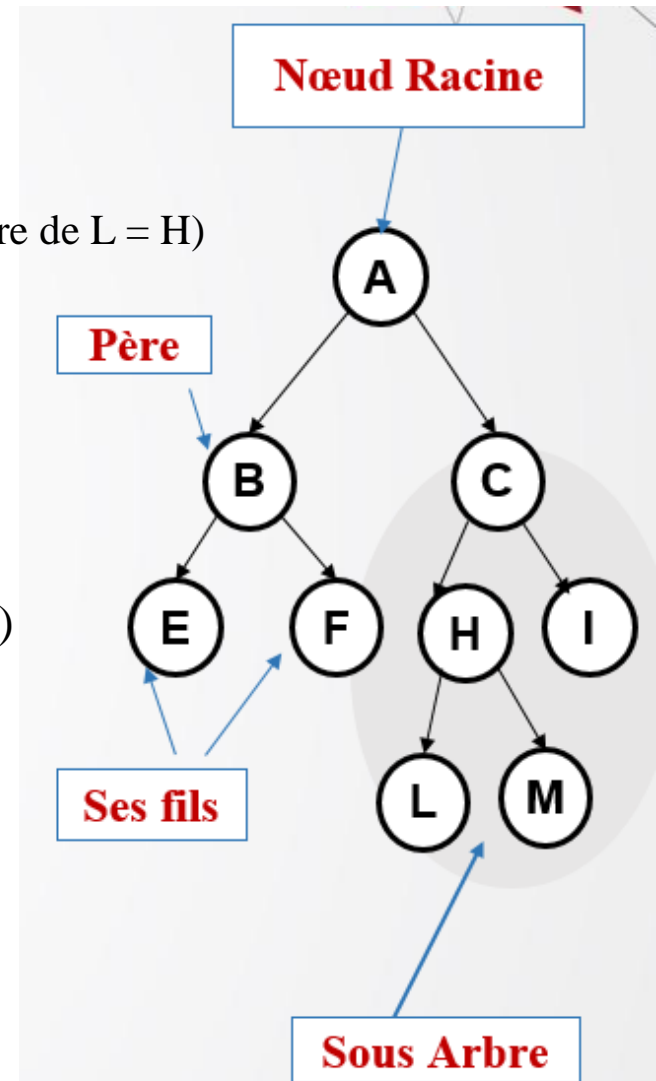


Chapitre 10: les arbres binaires

79

Terminologie

- Le prédécesseur s'il existe s'appelle **père** (père de C = A, père de L = H)
- Le successeur s'il existe s'appelle **fil**
(fils de A = { B,C }, fils de H = { L,M })
- Le nœud qui n'a pas de prédécesseur s'appelle **racine** (A)



Chapitre 10: les arbres binaires

80

Terminologie

- Le nœud qui n'a pas de successeur s'appelle **feuille** (Exemples: E,F,L,M,I)
- Un nœud **descendant** n d'un autre nœud X est tout nœud se trouvant dans le chemin partant du nœud X jusqu'à une feuille (y compris le nœud feuille).

Exemple: Les descendants de $C=\{H,I,L,M\}$, de $B=\{E,F\}$

- Un nœud **ascendant** n d'un autre nœud X est tout nœud se trouvant dans le chemin partant du nœud X jusqu'à la racine(y compris la racine).

Exemple: Les ascendants de $L=\{H,C,A \}$, $E=\{B,A\}$

Chapitre 10: les arbres binaires

81

Terminologie

- Le nœud qui n'a pas de successeur s'appelle **feuille** (Exemples: E,F,L,M,I)
- Un nœud **descendant** n d'un autre nœud X est tout nœud se trouvant dans le chemin partant du nœud X jusqu'à une feuille (y compris le nœud feuille).

Exemple: Les descendants de $C=\{H,I,L,M\}$, de $B=\{E,F\}$

- Un nœud **ascendant** n d'un autre nœud X est tout nœud se trouvant dans le chemin partant du nœud X jusqu'à la racine(y compris la racine).

Exemple: Les ascendants de $L=\{H,C,A \}$, $E=\{B,A\}$

Chapitre 10: les arbres binaires

82

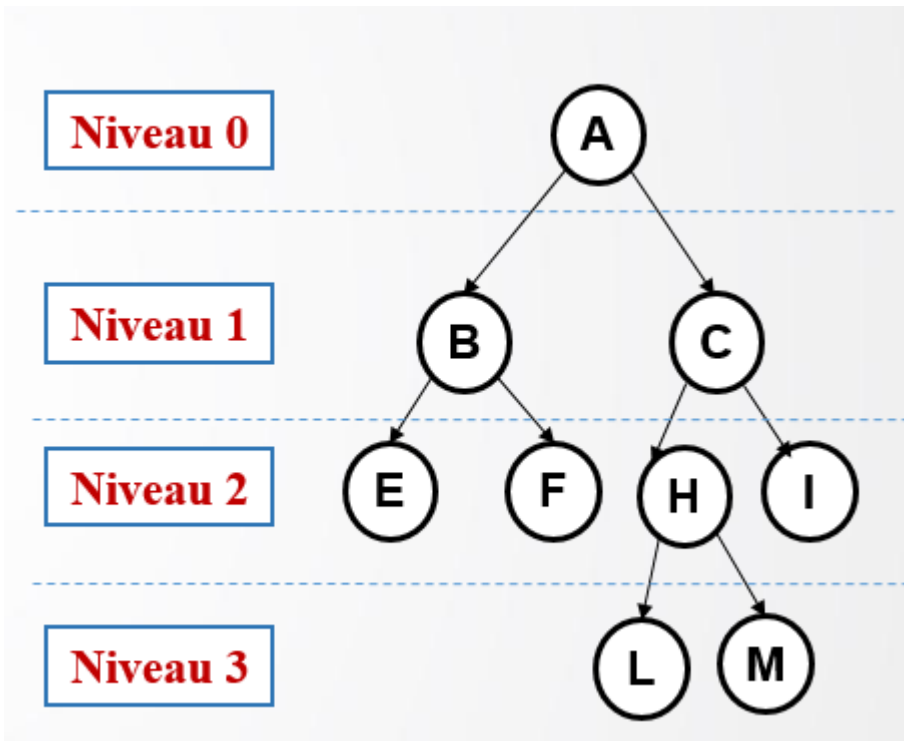
Mesures sur les arbres

➤ Taille d'un arbre

- On appelle taille d'un arbre le nombre total de nœuds de cet arbre.
- Taille de l'arbre suivant = 9
- Un arbre vide est de taille 0.

➤ Niveau d'un nœud

- Le niveau de la racine = 0
- Le niveau de chaque nœud = niveau de son père + 1
- Niveau de $\{E, F, H, I\} = 2$



Chapitre 10: les arbres binaires

83

Mesures sur les arbres

➤ Profondeur (Hauteur) d'un arbre

- C'est le niveau maximum dans cet arbre.

Profondeur de l'arbre suivant = 3

➤ Degré d'un nœud

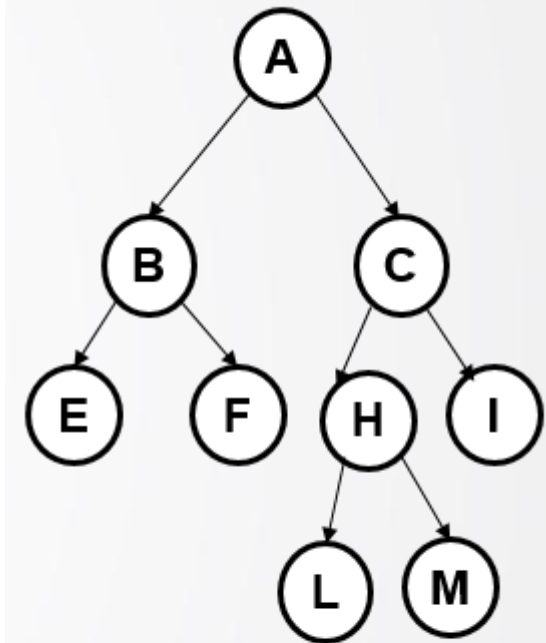
- Le degré d'un nœud est égal au nombre de ses fils.
- Degré de (A = 2, B = 2, C = 2, E = 0, H = 2)

➤ Degré d'un arbre

- C'est le degré maximum de ses nœuds.

Le degré d'un arbre binaire est égal à 2.

Si le degré d'un arbre est égal à N, l'arbre est dit **N-aire**.

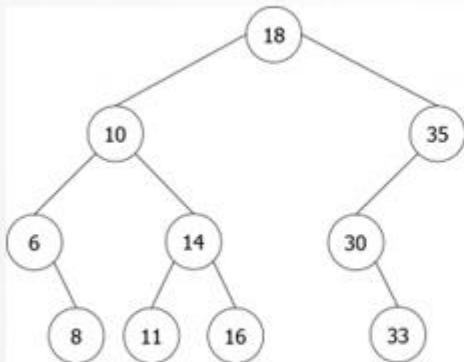


Degré de l'arbre = 2

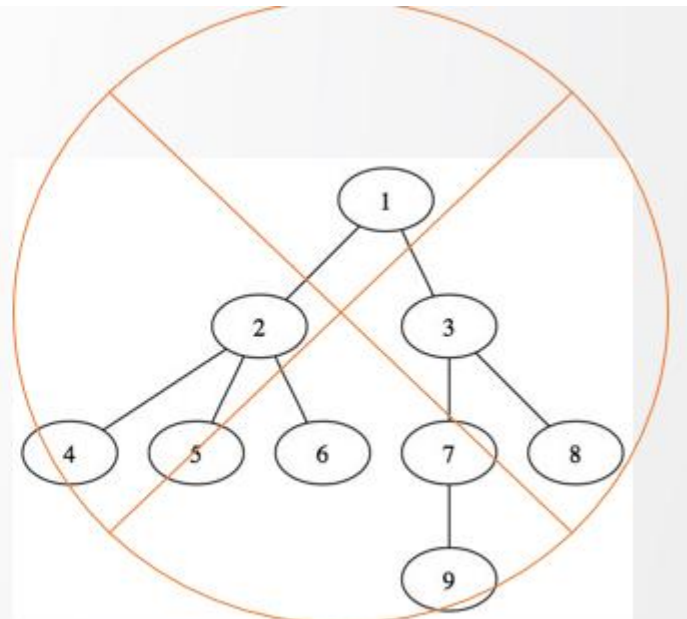
Chapitre 10: les arbres binaires

84

Un arbre **binaire** est un arbre où chaque nœud a un fils gauche, un fils droit ou les deux à la fois.
➔ c'est un arbre où le degré maximum d'un nœud est égal à 2.



Arbre binaire



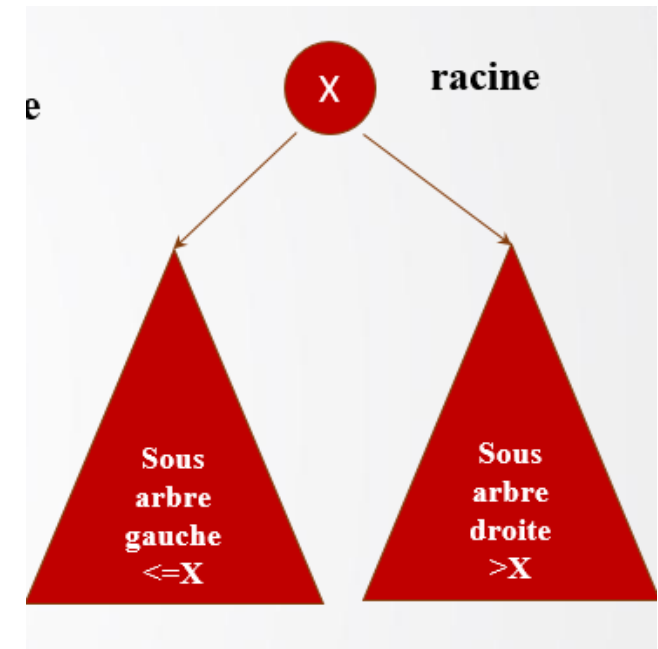
Arbre non binaire

Chapitre 10: les arbres binaires

85

Un arbre binaire A de racine X est dit **arbre binaire de recherche** (ABR) si et seulement si :

- ✓ Toute valeur associée à un nœud de son sous-arbre principal gauche est $\leq X$
- ✓ Toute valeur associée à un nœud de son sous-arbre principal droit est $> X$
- ✓ Tout sous-arbre de A est lui-même un ABR.

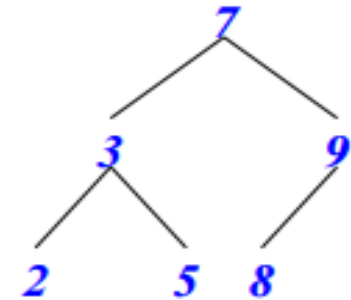
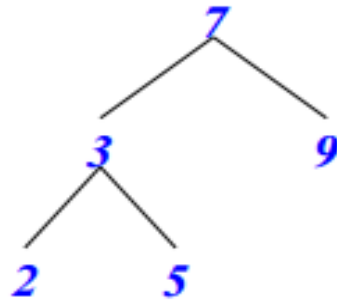


Chapitre 10: les arbres binaires

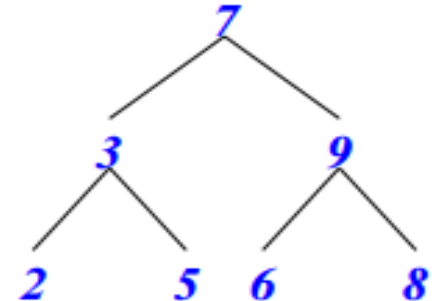
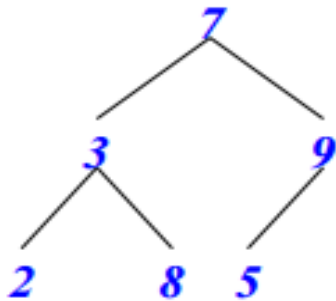
86

□ exemples

Des ABR



Des arbres non ABR



Chapitre 10: les arbres binaires

87

Structure

- 3 types de données sont stockées dans un nœud. :
 - La donnée data
 - Un pointeur de type Nœud vers le sous arbre gauche
 - Un pointeur de type Nœud vers le sous arbre droit
 - Relations entre types: Structure récursive
 - Un arbre binaire est caractérisé par une racine qui est un nœud
 - Les descendants d'un nœud sont des arbres binaires
 - définition récursive de l'arbre en fonction d'elle-même.
- Solution** : les descendants d'un nœud sont des pointeurs vers d'autres nœuds.

Chapitre 10: les arbres binaires

88

Struct Nœud

```
{  
  TYPE data;           // data peut avoir n'importe  
                        // quel type  
  Struct Nœud * FG;    // FG et FD sont deux  
                        // pointeur vers d'autres  
                        // noeuds */  
  Struct Nœud * FD;  
};
```

```
Typedef Struct Nœud * Arbre;
```

Chapitre 10: les arbres binaires

89

Le parcours

- Le parcours d'un arbre consiste à passer par tous ses nœuds pour en effectuer un traitement.
- On distingue deux types de parcours :
 - ✓ Parcours en profondeur
 - ✓ Parcours en largeur

Chapitre 10: les arbres binaires

90

parcours en profondeur

Dans un parcours en profondeur, Commençant par la racine:

1. On descend le plus profondément possible dans l'arbre puis
2. Une fois qu'une feuille est atteinte, on remonte pour explorer les autres branches en commençant par la branche "la plus basse" parmi celles non encore parcourues.

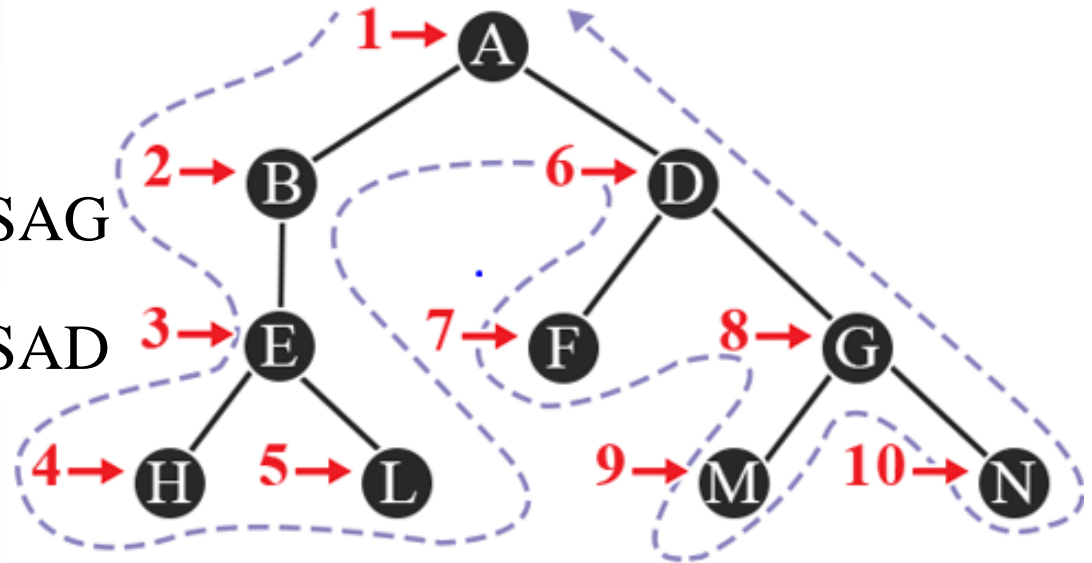
Chapitre 10: les arbres binaires

91

Parcours Préfixé

La racine est traitée en premier

- 1 Traiter la racine
- 2 Parcours préfixé du SAG
- 3 Parcours préfixé du SAD



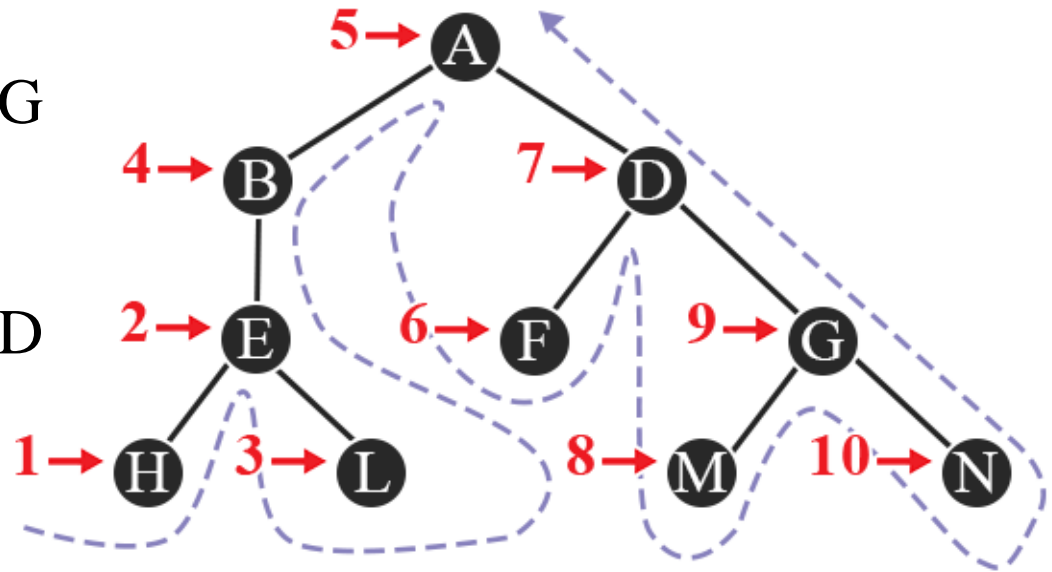
□ Parcours préfixé : A, B, E, H, L, D, F, G, M, N.

Chapitre 10: les arbres binaires

92

La racine est traitée entre les deux appels récuratifs

- 1 Parcours Infixé du SAG
- 2 Traiter la racine
- 3 Parcours Infixé du SAD



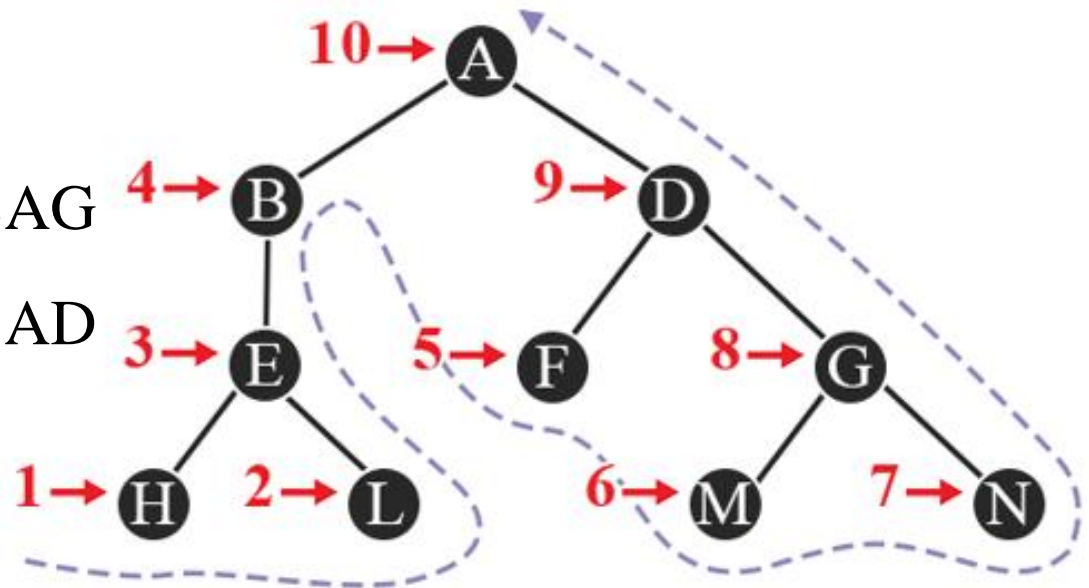
□ Parcours infixé : H, E, L, B, A, F, D, M, G, N.

Chapitre 10: les arbres binaires

93

La racine est traitée après les deux appels récuratifs

- 1 Parcours Postfixé du SAG
- 2 Parcours Postfixé du SAD
- 3 Traiter la racine



□ Parcours postfixé : H, L, E, B, F, M, N, G, D, A.

Chapitre 10: les arbres binaires

94

- Ajout d'un nœud
- Algo INSERT(v) // insère la clé v dans l'arbre **itératif**
 - ▣ $x \leftarrow \text{racine}$
 - ▣ si $x = \text{null}$ alors initialiser avec une racine de clé v et retourner
 - ▣ tant que vrai faire
 - si $v = \text{cle}(x)$ alors retourner
 - si $v < \text{cle}(x)$
 - alors si $\text{gauche}(x) = \text{null}$
 - alors attacher nouvel enfant gauche de x avec clé v et retourner
 - sinon $x \leftarrow \text{gauche}(x)$
 - sinon si $\text{droit}(x) = \text{null}$
 - alors attacher nouvel enfant droit de x avec clé v et retourner
 - sinon $x \leftarrow \text{droit}(x)$