

CH7 LES ARBRES

I. GENERALITES SUR LES ARBRES

Un **arbre** est une structure de données constituée de **nœuds**, qui peuvent avoir des **enfants** (qui sont eux-mêmes des nœuds). Cette structure est hiérarchisée et le sommet de l'arbre est appelé **racine**.

Un **nœud** est une position dans un arbre. À chaque nœud correspond un sous-arbre. Chaque nœud est défini par son **étiquette**.

Une **feuille** est un nœud qui n'a pas d'enfant.

Le sommet d'un arbre est appelé la **racine**. C'est le seul nœud qui n'a pas de parents.

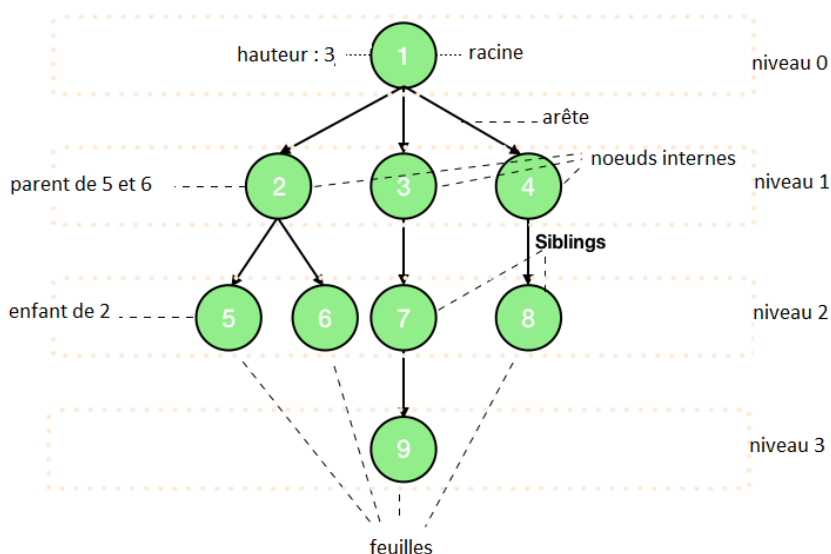
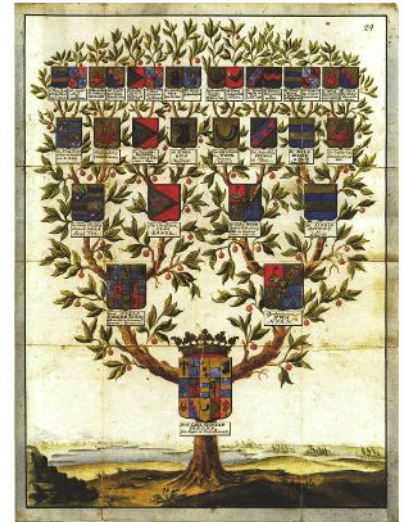
Un **nœud interne** est un nœud autre que la racine et qui n'est pas une feuille.

Une **branche** est une suite finie de nœuds consécutifs de la racine vers une feuille.

La **taille** d'un arbre est le nombre de ses nœuds.

La **profondeur** d'un nœud est la longueur du chemin qui le relie à la racine.

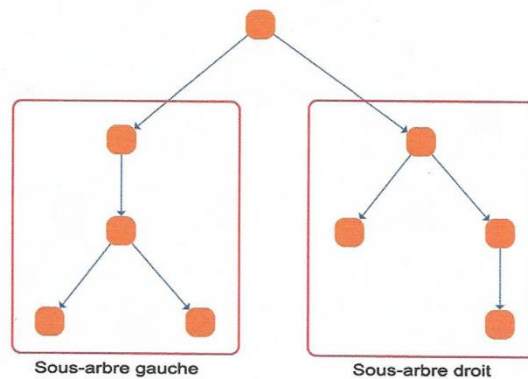
La **hauteur** d'un arbre est la profondeur de son nœud le plus profond (c'est donc nécessairement une feuille).



II. ARBRES BINAIRES

Un **arbre binaire** correspond à un des cas suivants :

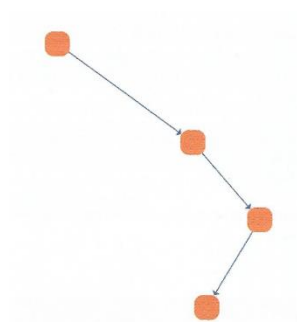
- Un arbre vide (il ne contient aucun nœud).
- Un arbre non vide, dont la structure est la suivante :
 - Un des nœuds est la racine de l'arbre
 - Les nœuds restants sont séparés en deux sous-ensembles, qui forment récursivement deux sous-arbres (binaires) appelés respectivement sous-arbre gauche et sous-arbre droit.
 - La racine est reliée à la racine de ces deux sous-arbres gauche et droit.



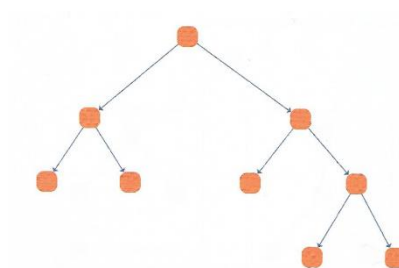
Il existe des cas particuliers d'arbres binaires :

- **Arbre binaire dégénéré (ou filiforme)** : c'est un arbre dont les nœuds possèdent au maximum un enfant.
- **Arbre localement complet** : c'est un arbre binaire dont chacun des nœuds possède soit deux, soit aucun enfant.
- **Arbre complet** : c'est arbre localement complet et dont toutes les feuilles sont au niveau hiérarchique le plus bas
-

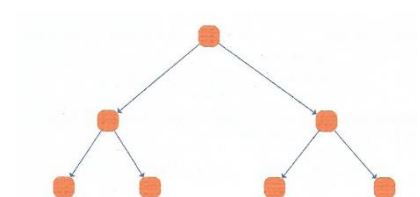
Arbre dégénéré



Arbre localement complet



Arbre complet



Les définitions données sur les arbres (racine, enfant, feuille, taille, hauteur ...) sont bien sur valables pour les arbres binaires. Les différents types de parcours sont également les mêmes.

III. PARCOURIR UN ARBRE

Parcourir un arbre c'est partir d'un nœud et visiter tous les nœuds de l'arbre une seule fois. Ce concept de parcours est très important en algorithmique.

Les parcours permettent notamment de générer une autre représentation de l'arbre (par exemple une liste) ou d'effectuer une recherche dans une structure arborescente.

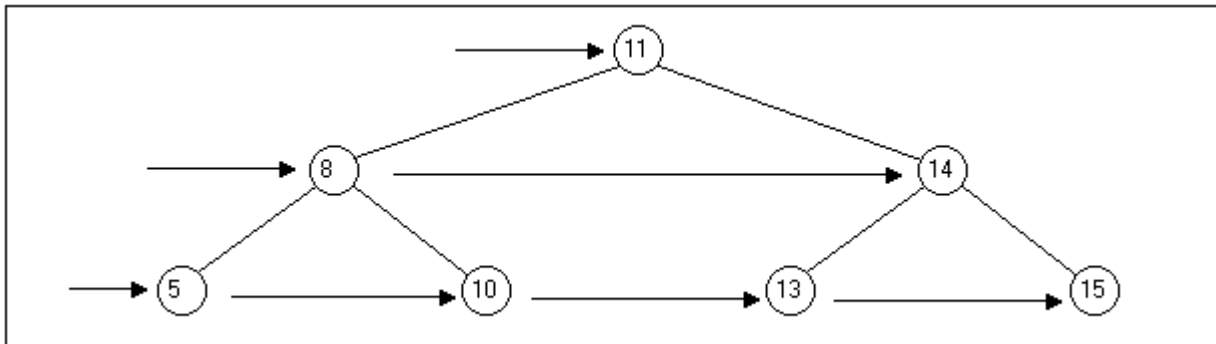
Il existe différentes façons de parcourir un arbre.

Parcours en largeur (BFS : Breadth First Search)

Dans un parcours en largeur, les nœuds sont parcourus par profondeur croissante. Avec l'exemple donné ci-dessous, les étiquettes seront traitées dans l'ordre suivant :

11 > 8 > 14 > 5 > 10 > 13 > 15

Plus précisément, on commence par explorer un nœud source, puis ses enfants, puis les enfants non explorés des enfants, etc.

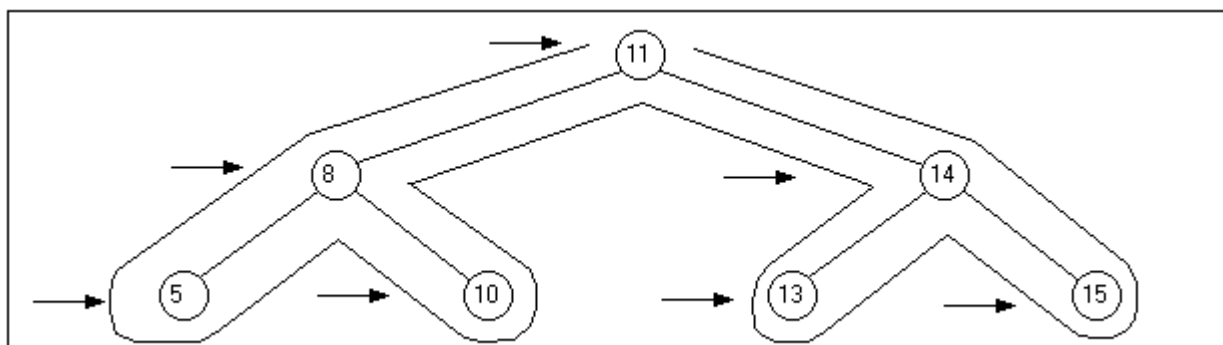


Parcours en profondeur (DFS : Depth First Search)

Préfixe

- Un parcours en profondeur préfixe est un parcours où l'on effectue le traitement de chaque nœud avant d'explorer le sous-arbre correspondant. Avec l'exemple donné ci-contre, les étiquettes seront visitées dans l'ordre suivant :

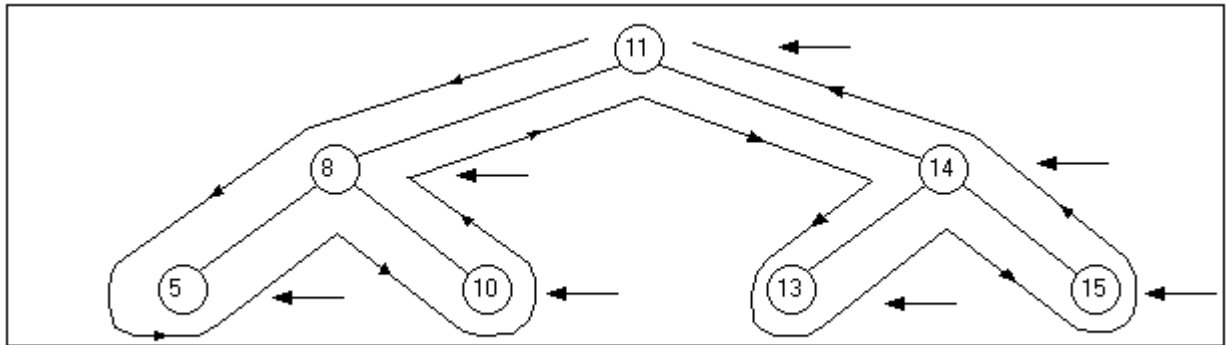
11 > 8 > 5 > 10 > 14 > 13 > 15



Postfixe

- Un parcours en profondeur postfixe est un parcours où l'on effectue le traitement de chaque nœud après avoir exploré le sous-arbre correspondant. Avec le même exemple donné ci-dessous, les étiquettes seront visitées dans l'ordre suivant :

5 > 10 > 8 > 13 > 15 > 14 > 11



Infixe

- Un parcours en profondeur infixe est un parcours où l'on effectue le traitement du sous-arbre gauche, puis la racine puis le traitement du sous-arbre droit.

Avec l'exemple précédent on obtient les étiquettes dans l'ordre suivant :

5 > 8 > 10 > 11 > 13 > 14 > 15

On peut implémenter facilement les parcours en profondeur de manière récursive :

ParcoursPréfixe(arbre A):

Afficher la racine de A

Pour tous les enfants e de la racine :

ParcoursPréfixe(Sous-Arbre issu de e)

ParcoursPostfixe(arbre A):

Pour tous les enfants e de la racine :

ParcoursPostfixe(Sous-Arbre issu e)

Afficher la racine de A

ParcoursInfixe(arbre A):

Pour tous les enfants de la racine

ParcoursInfixe(Sous-arbre gauche issu de e)

Afficher la racine de A

ParcoursInfixe(sous-arbre droit issu de e)

IV. ARBRE BINAIRE DE RECHERCHE

Un **arbre binaire de recherche (ABR)** est un arbre binaire dont les nœuds contiennent des valeurs qui peuvent être comparées entre elles, et tel que, pour tout nœud de l'arbre, toutes les valeurs situées dans le sous-arbre gauche (resp. droit) sont plus petites (resp. plus grandes) que la valeur située dans le nœud.

On emploie également le terme **clé** à la place de valeur.

Un **ABR** étant un arbre binaire, le vocabulaire des arbres binaires est valable.

Le parcours en profondeur infixe d'un ABR parcourt les nœuds de l'arbre dans l'ordre croissant de leurs valeurs.