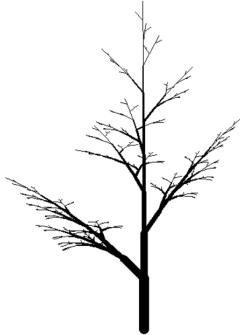
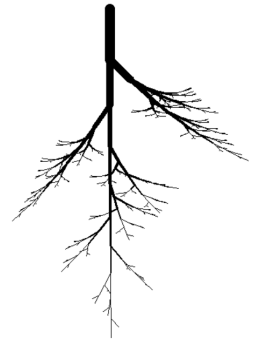


# Arbres 1/3 : Structures hiérarchiques



Nous avons déjà rencontré à plusieurs reprises la notion d'arbre:

- LE DOM(Document Object Model) d'une page html est décrit à l'aide d'un arbre.
- Le système de fichiers de Linux est décrit à l'aide d'un arbre.



Dans cette feuille, garantie sans ordinateur, nous allons découvrir un peu plus cette structure de données abstraites pour :

- -Utiliser un vocabulaire adapté.
- -Parcourir des arbres de différentes manières.

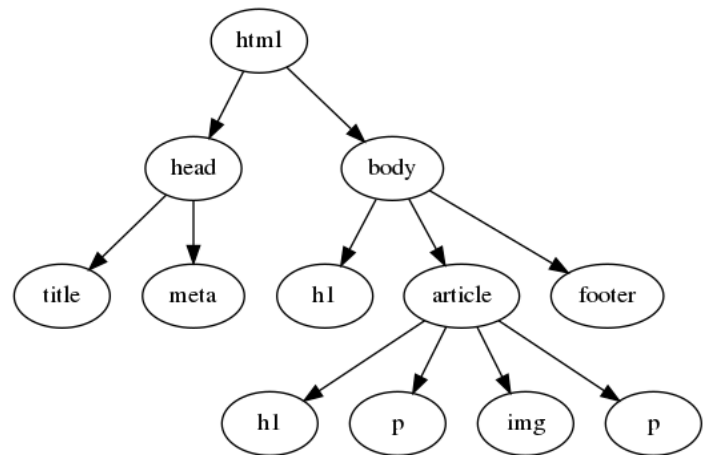
## 1. Un peu de vocabulaire

### Quelques Définitions

- Un *arbre* est une structure de données composée d'une étiquette et d'un nombre quelconque d'enfants, qui sont eux-mêmes des arbres.
- Un *noeud* est une position dans un arbre. À chaque noeud correspond un sous-arbre.
- Une *feuille* est un noeud dans un arbre ou un arbre qui n'a pas d'enfant.
- La *racine* d'un arbre est le noeud qui correspond à l'arbre tout entier. C'est l'unique noeud qui n'a pas de parent.
- La *taille* d'un arbre est le nombre de ses noeuds
- La *profondeur* d'un noeud dans un arbre est la longueur du chemin qui le relie à la racine
- La *hauteur* d'un arbre est la profondeur de son noeud le plus profond (ce noeud est nécessairement une feuille)

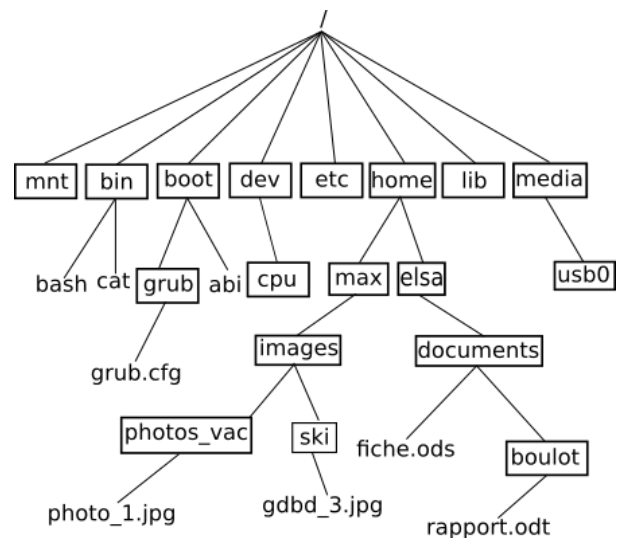
### Exercice 1 : Une page html

1. Combien de feuilles possède cet arbre ?
1. Quelle est l'étiquette de cet arbre ?
1. Quelle est sa taille ?
1. Quelle est sa hauteur ?
1. Quelle est la profondeur du noeud dont l'étiquette est article?
1. Combien d'enfants possède cet arbre ?
1. Quel est le noeud qui possède le plus d'enfants ?



### Exercice 2 : Un système de fichiers

1. Combien de feuilles possède cet arbre?
1. Quelle est l'étiquette de cet arbre ?
1. Quelle est sa taille ?
1. Quelle est sa hauteur ?
1. Quelle est la profondeur du noeud dont l'étiquette est image?
1. Combien d'enfants possède cet arbre ?
1. Quel est le noeud qui possède le plus d'enfants ?



## 2. Parcourir un arbre

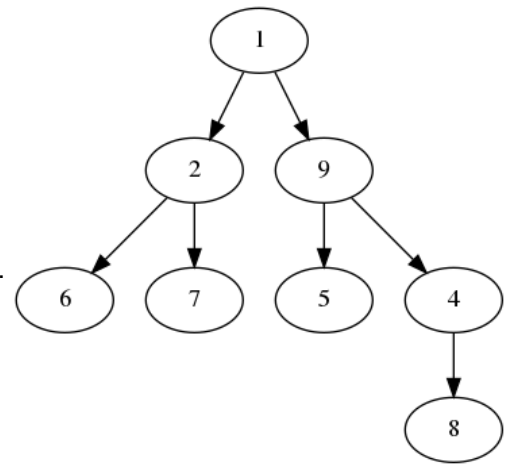
- Parcourir un arbre c'est partir d'un noeud et visiter tous les noeuds de l'arbre une seule fois.
- Ce concept de parcours est très important en algorithmique.
- Les parcours permettent notamment de générer une autre représentation de l'arbre(par exemple une liste) ou d'effectuer une recherche dans une structure arborescente.
- Il existe différentes façons de parcourir un arbre.

## Parcours en largeur (BFS : Breadth First Search)

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

$1 > 2 > 9 > 6 > 7 > 5 > 4 > 8$

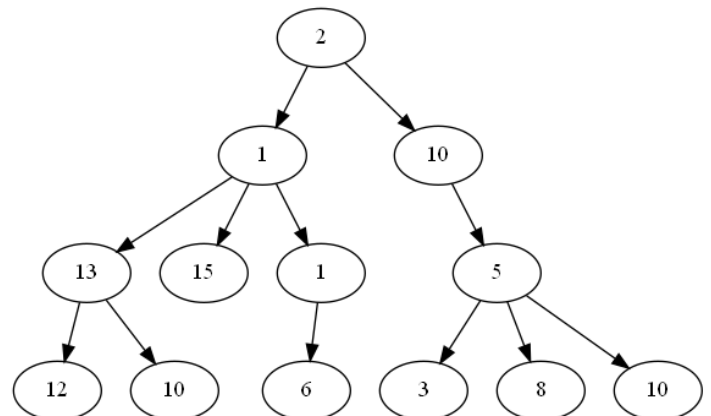
- Plus précisément, on commence par explorer un noeud source, puis ses enfants, puis les enfants non explorés des enfants, etc.



### Exercice 3 :

Ecrire le parcours en largeur de l'arbre ci-contre

Réponse :



## Parcours en profondeur (DFS : Depth First Search)

### Préfixe

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

$1 > 2 > 6 > 7 > 9 > 5 > 4 > 8$

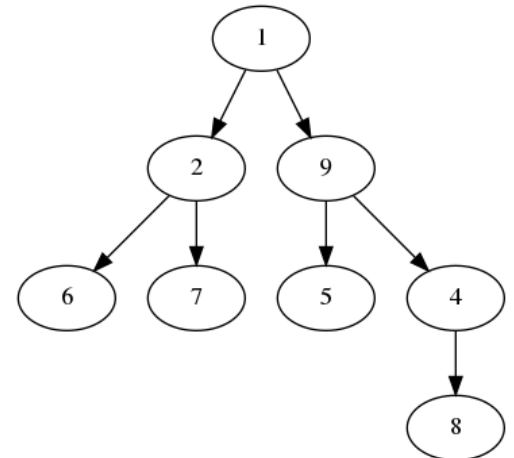
- Ce type de parcours se prête à la récursivité:

*parcours\_profondeur\_pre fixe*(noeud) :

*traitement(etiquette)*

*pour chaque enfant de noeud :*

*parcours\_profondeur\_pre fixe*(enfant)



### Postfixe

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

$6 > 7 > 2 > 5 > 8 > 4 > 9 > 1$

- De façon récursive:

*parcours\_profondeur\_post fixe*(noeud) :

*pour chaque enfant de noeud :*

*parcours\_profondeur\_pre fixe*(enfant)

*traitement(etiquette)*

### Exercice 4 :

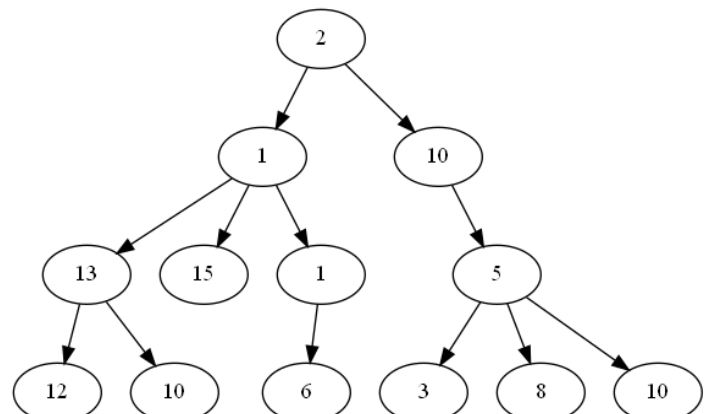
Ecrire le parcours de l'arbre ci-contre :

- En profondeur préfixe

Réponse :

- En profondeur postfixe

Réponse :



### Exercice 5 : Synthèse

1. Quelle est l'étiquette de cet arbre ?

1. Quelle est sa taille ?

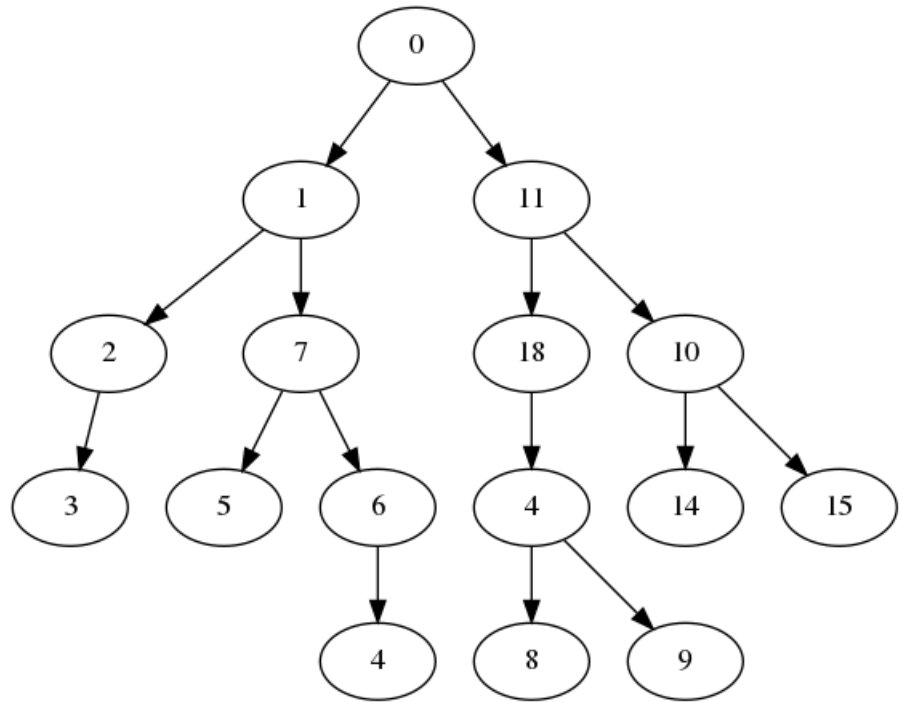
1. Quelle est sa hauteur ?

1. Quelle est la profondeur du noeud dont l'étiquette est 10 ?

1. Combien d'enfants possède cet arbre ?

1. Quel est l'étiquette la plus grande ?

1. Dans quel ordre les étiquettes sont-elles rencontrées dans un parcours en largeur ?



1. Dans quel ordre les étiquettes sont-elles traitées dans un parcours en profondeur postfixe ?

1. Dans quel ordre les étiquettes sont-elles traitées dans un parcours en profondeur préfixe ?