

# 7.1

## Les arbres (ou arborescence)

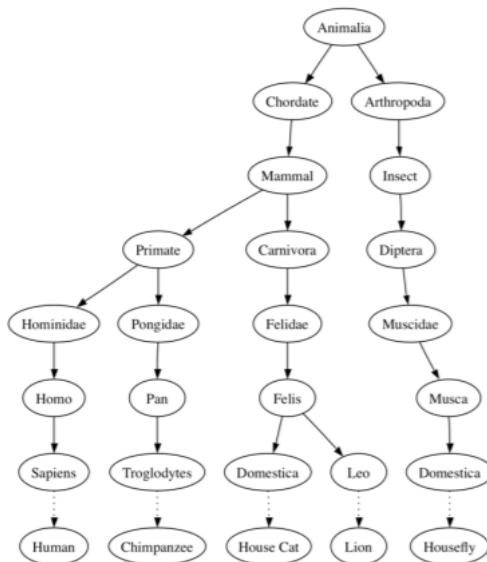
NSI TERMINALE - JB DUTHOIT

### 7.1.1 Introduction

Il n'existe pas qu'une façon linéaire de représenter les données, comme les listes, les tableaux, les dictionnaires, les piles et les files. Nous pouvons également structurer les données de façon hiérarchique.

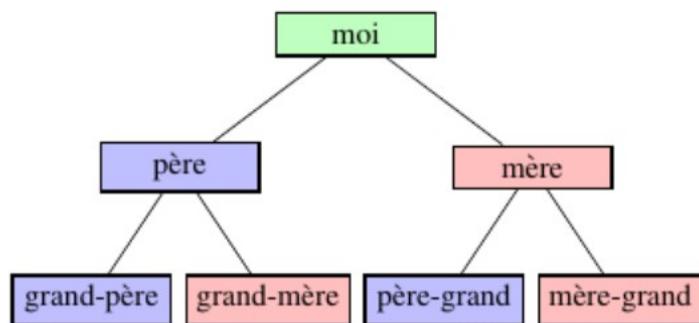
### 7.1.2 Exemples de situations où l'on rencontre des arbres

- En biologie



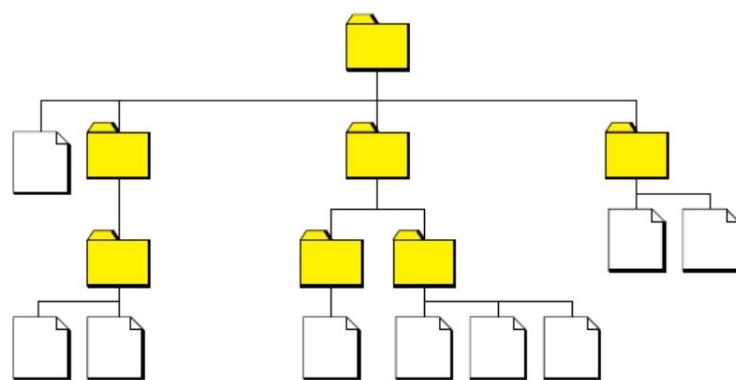
Arbre de classification

- En généalogie



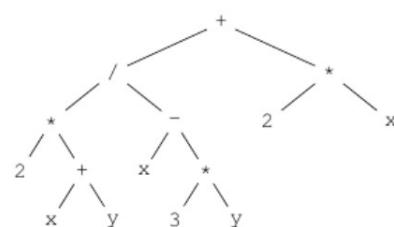
Arbre généalogique

- en informatique



Arbre de classification : arborescence des fichiers

- En mathématiques :



Arbre syntaxique

### 7.1.3 Définition

#### Définition

Un **arbre** est une structure de données constituée de nœuds, qui peuvent avoir des enfants (et qui sont aussi des nœuds)

#### Définition

Le sommet de l'arbre est appelé **racine**.

#### Définition

Un nœud qui ne possède pas d'enfant est appelé une **feuille**.

#### Définition

Les nœuds autres que la racine et les feuilles sont appelés **noeuds internes**.

#### Définition

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

#### Remarque

► Un arbre a donc autant de feuilles que de branches !

## Définition

| l'**arité** d'un arbre est le nombre maximal d'enfants qu'un noeud peut avoir.

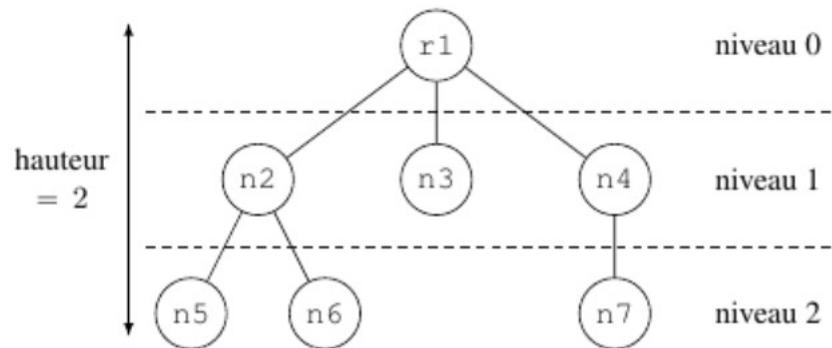
## Définition

| La **taille** d'un arbre est le nombre de noeuds qui le compose.

## Définition

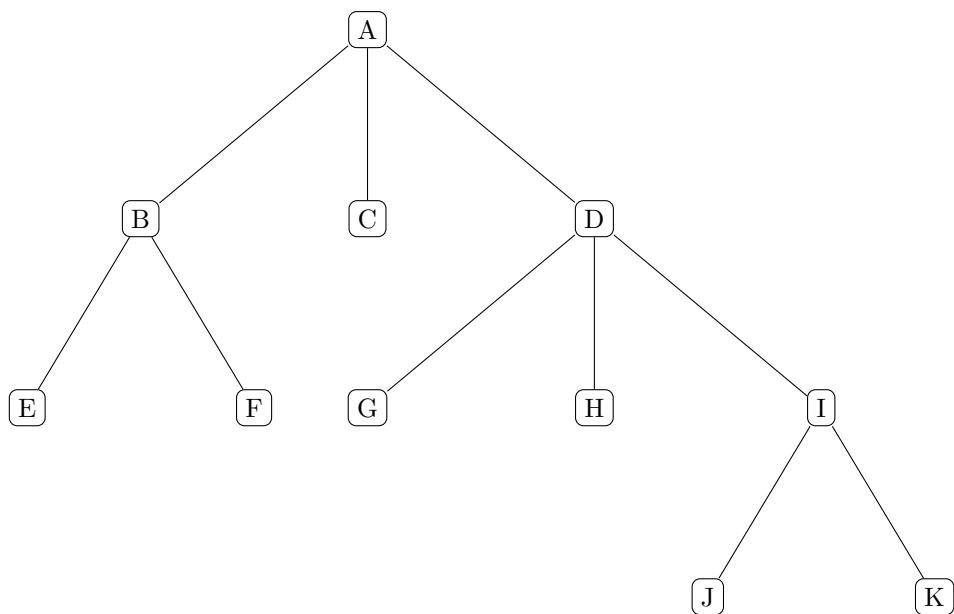
La **hauteur** d'un arbre est la profondeur à laquelle il faut descendre pour trouver la feuille la plus éloignée de la racine.

## Exemple



### Exercice 7.1

On considère l'arbre ci-dessous.



Répondre aux questions suivantes :

- Nombre de noeuds ?
- Nombre de racine ? Laquelle ?
- Nombre de feuilles ? Lesquelles ?
- Nombre de branches ?

- Nombre de noeuds internes ?
- Quel est son arité ?
- Quelle est sa taille ?
- Quelle est sa hauteur ?

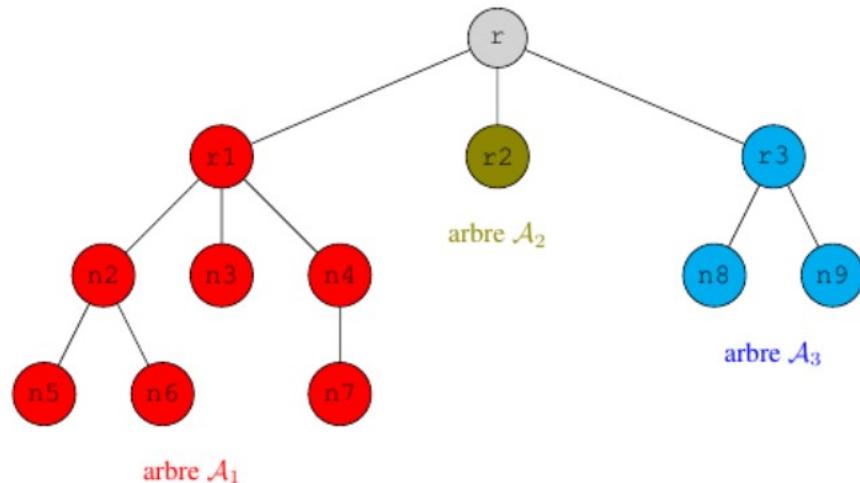
### Remarque

| En informatique, les arbres poussent vers le bas :-)

## Définition Définition récursive d'un arbre ou d'une arborescence

Il est souvent utile de voir un arbre comme structure de données récursive : Un arbre est :

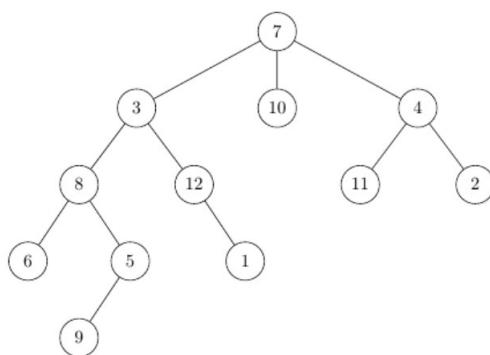
- soit un arbre réduit à un seul noeud unique  $r$
- soit un arbre constitué d'une racine  $r$  et d'une suite de sous-arbre  $A_1, A_2 \dots$



Définition récursive d'un arbre

### Exercice 7.2

Considérons l'arbre ci-contre :



- Déterminer la taille, le nombre de feuilles et le nombre de noeuds internes de A.
- Quels sont les noeuds de niveau 3 ?
- Quelle est la hauteur de l'arbre A ?

### 7.1.4 Utilisation de la POO pour représenter en Python un arbre, appelé aussi arborescence

Pour représenter un arbre (une arborescence) en Python, on peut utiliser la POO, comme pour les listes chaînées.

L'objet de la classe contient deux attributs : un attribut valeur (dans lequel on stocke une valeur quelconque, appelée étiquette) et un attribut fils dans lequel on stocke les fils sous la forme d'un tableau.

#### Exercice 7.3

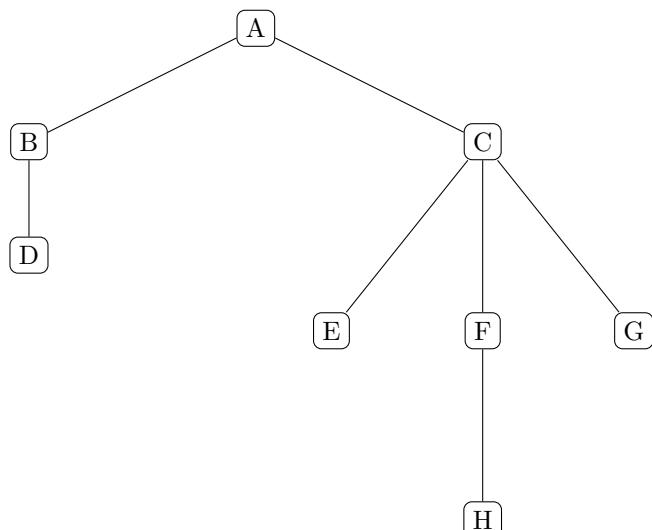
Construire la classe `Noeud`. Pour les feuilles, on mettra `[]` pour le fils \*\*\*

#### Exercice 7.4

Construire l'arbre donné en exemple plus haut.

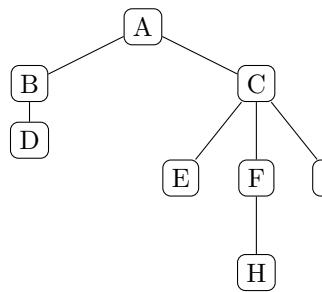
#### Exercice 7.5

Construire l'arbre ci-dessous :



#### Exercice 7.6

Créer une fonction récursive `représente(arbre, p=0)` qui permet un affichage d'un arbre comme ceci :



A  
 - B  
 -- D  
 - C  
 -- E  
 -- F  
 ---- H  
 -- G

Représentation possible d'un arbre. Les trets indiquent la profondeur

\*\*\*

### 7.1.5 Utilisation d'autres structures afin de représenter une arborescence en Python

#### Exercice 7.7

Utiliser la structure de dictionnaire `dict` pour représenter l'arbre donné en introduction.  
\*\*

#### Exercice 7.8

Utiliser la structure `list` pour représenter l'arbre donné en introduction.  
\*\*