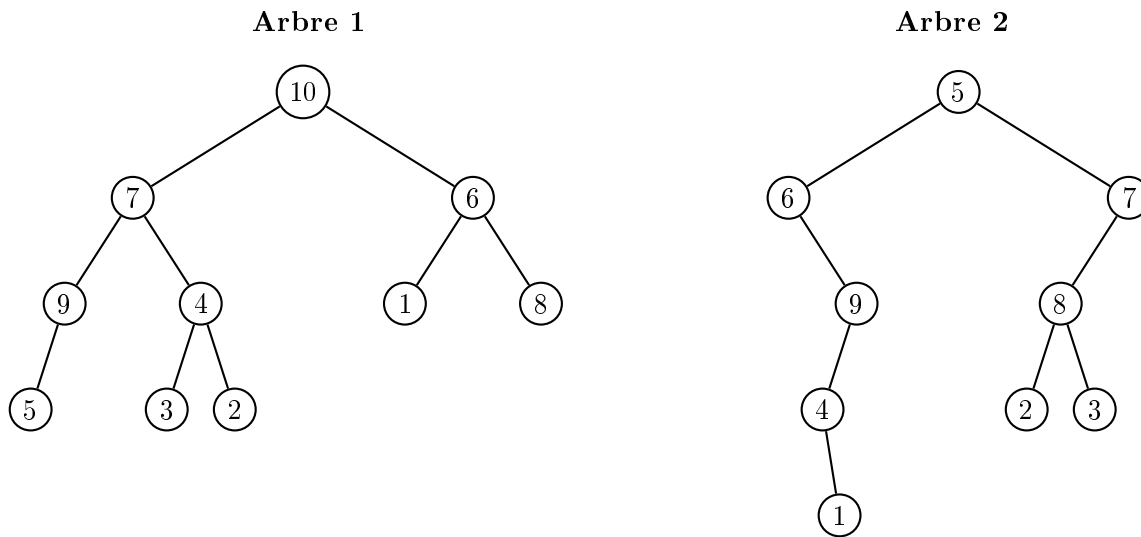


# Exercice : Arbres binaires

---

## Exercice 2



Les arbres 1 et 2 ci-dessus sont des arbres binaires. On suppose que la racine est de profondeur 0.

- 1) Donner la taille et la hauteur des arbres binaires 1 et 2.
- 2) Donner la valeur des feuilles de chaque arbre.
- 3) On étiquette les arbres. En binaire, le noeud racine est 1, son fils gauche 10, son fils droit 11, etc. Quelle est l'étiquette du noeud de valeur 2 dans chaque arbre ?
- 4) Donner une construction récursive de chaque arbre.

## Exercice 3

Nous allons utiliser un module contenant les classes `Noeud` et `Arbre` pour nos arbres binaires. Vous trouverez les fichiers sur l'ENT.

- Pour Jupyter Notebook : `arbre_binaire_jupyter.py`
- Pour les IDLE python : `arbre_binaire.py`

Après avoir récupéré ces fichiers, vous pourrez directement travailler dedans ou faire une copie de travail.

### Interface

L'interface de notre arbre binaire est en programmation orientée objet.

- Un objet `Arbre` a pour attribut **racine** qui contient un objet **Noeud**.
- Un objet `Noeud` possède 3 attributs : `valeur`, `gauche` et `droit`. A la création d'un `Noeud`, l'attribut `valeur` est requis. Les attributs `gauche` et `droit`, s'ils ne sont pas précisés, valent `None`.
- La méthode **`est_vide`** de la classe `Arbre` teste si un arbre est vide. Cette méthode ne s'applique pas à la classe `Noeud`.
- Une méthode **`affiche`** de la classe `Arbre` permet d'afficher l'arbre binaire.

- 1) Créer un arbre binaire composé d'une seule racine.
- 2) Créer un arbre binaire complet de hauteur 2.
- 3) Créer un arbre binaire de hauteur 3, bien tassé, de taille 5.
- 4) Créer les arbres binaires de l'exercice 1.

## Exercice 4

Pour cet exercice, vous pouvez utiliser l'interface en POO de l'exercice précédent ou votre propre interface.

- 1) On donne l'algorithme d'une fonction pour déterminer la taille d'un arbre binaire  $a$  :

Si  $a$  est vide :  
renvoyer 0 (la taille est égale à 0, pas de noeuds)  
Sinon :  
renvoyer  $1 + \text{taille de l'arbre gauche} + \text{taille de l'arbre droit}$

- a) Quel procédé est utilisé par cette fonction pour calculer la taille d'un arbre binaire ?  
b) Programmer cette fonction en python et déterminer la taille des arbres saisis à l'exercice précédent.
- 2) Écrire un algorithme qui permet de calculer la hauteur d'un arbre binaire puis l'exécuter dans votre programme python.
- 3) Créer une fonction **niveau** en Python qui prend un arbre et une profondeur  $p$  en paramètres et qui renvoie la liste des noeuds (valeurs) situés au niveau de profondeur  $p$ .