

CHAPITRE 6.2 : LES ARBRES BINAIRES

Introduction : tournoi à élimination directe

Dans ce type de compétition, un joueur est éliminé au premier match qu'il perd. Le vainqueur est celui qui gagne tous ses matches.

PARTIE A : On suppose qu'il y a 16 joueurs

1. Représenter cette situation à l'aide d'un arbre.
2. En combien de tours se déroule ce tournoi ?
3. Combien de matches au total seront joués pour désigner un vainqueur ?

PARTIE B : On suppose qu'il y a N joueurs.

1. Quelle condition peut-on envisager sur N pour que le tournoi soit simple à organiser ?
2. Sous cette condition :
 - a. En combien de tours se déroule ce tournoi ?
 - b. Combien de matches au total seront joués pour désigner un vainqueur ?

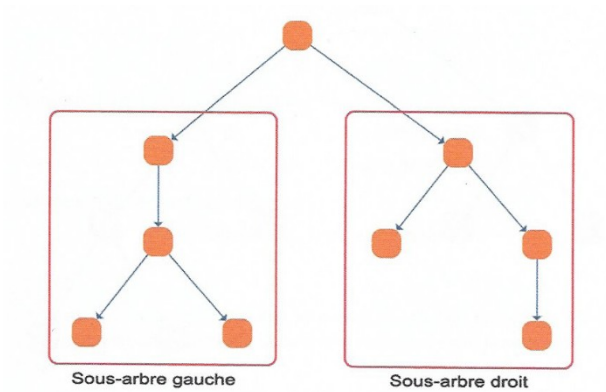
1. DEFINITIONS

La situation précédente est décrite à l'aide d'un **arbre binaire**.

Un **arbre binaire** est un cas particulier de structure arborescente, où chaque nœud a *au maximum* deux enfants.

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

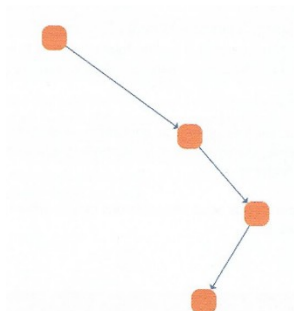
- Un arbre vide (il ne contient aucun nœud).
- Un arbre non vide, dont la structure est la suivante :
 - Un des nœuds est la racine de l'arbre
 - Les nœuds restants sont séparés en deux sous-ensembles, qui forment récursivement deux sous-arbres (binaires) appelés respectivement sous-arbre gauche et sous-arbre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



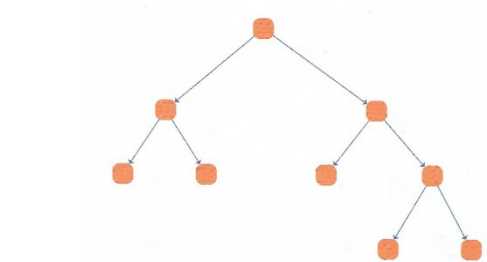
Il existe des cas particuliers d'arbres binaires :

- **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 un arbre localement complet et dont toutes les feuilles sont au niveau hiérarchique le plus bas

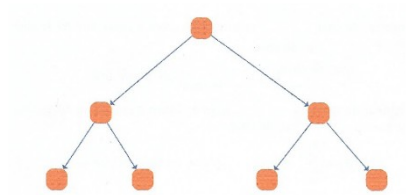
Arbre dégénéré



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.

Exercices d'application : n° 1 à 3

2. LE TYPE ABSTRAIT ARBRE

De manière classique, on représente un arbre en Python par un objet d'une classe, appelée (très souvent ...) Nœud.

```
class Noeud:
    """ Nœud d'un arbre binaire """
    def __init__(self, g, v, d):
        self.gauche = g
        self.valeur = v
        self.droite = d
```

Un objet de cette classe contient trois attributs :

- **gauche** pour le sous-arbre gauche
- **valeur** pour la valeur contenue dans le nœud
- **droite** pour le sous-arbre droit

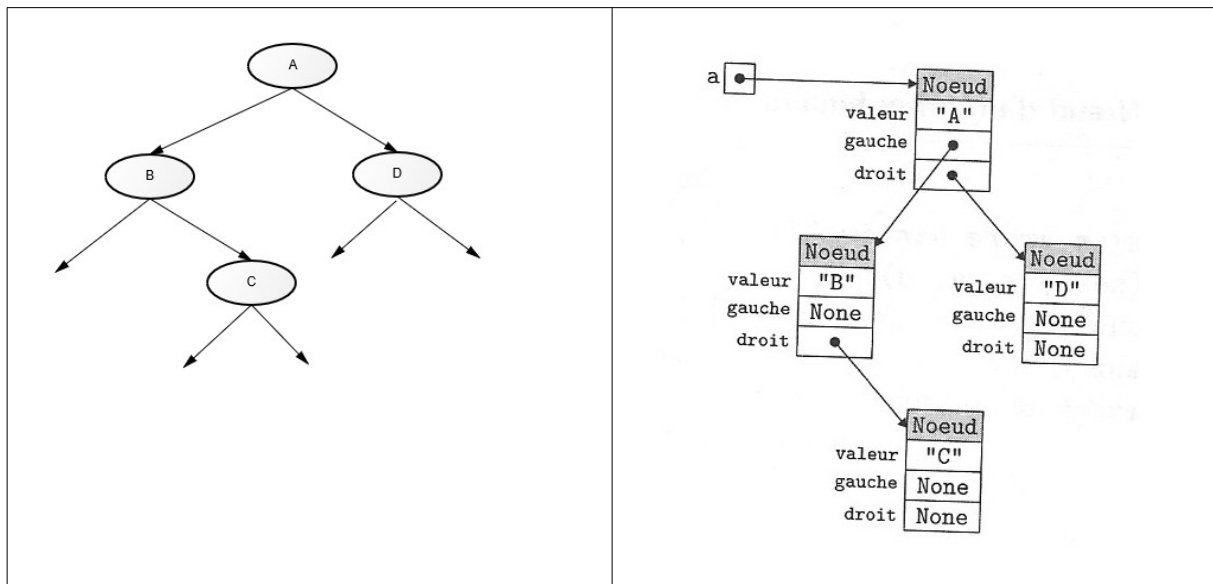
L'arbre vide est représenté par la valeur **None**.

Pour construire un arbre, on applique le constructeur autant de fois qu'il y a de nœuds.

Par exemple l'arbre défini par

```
a = Noeud(Noeud(None, « B », Noeud(None, « C », None)), « A »,
Noeud(None, « D », None))
```

est représenté par l'arbre ci-dessous, le principe de construction étant détaillé à droite :



On peut bien sûr créer des méthodes pour cette classe afin de déterminer sa taille, sa hauteur, ses parcours ...

Application : TP Arbres binaires