

Définition d'un arbre

Un arbre est constitué de **noeuds** organisés de manière **hiérarchique**. C'est un **graphe** :

- ...
- ...

Les arbres peuvent servir à trier, à représenter une généalogie, à prendre une décision.

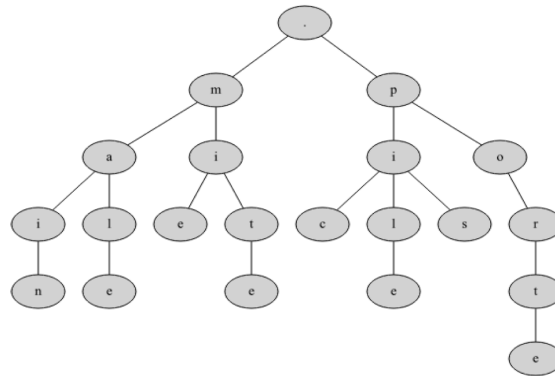


FIGURE 1 – un exemple d'arbre lexicographique

Retrouver le mot P I L E ; \
et rajouter le mot P O R T A I L.

1.1 Caractéristiques d'un arbre

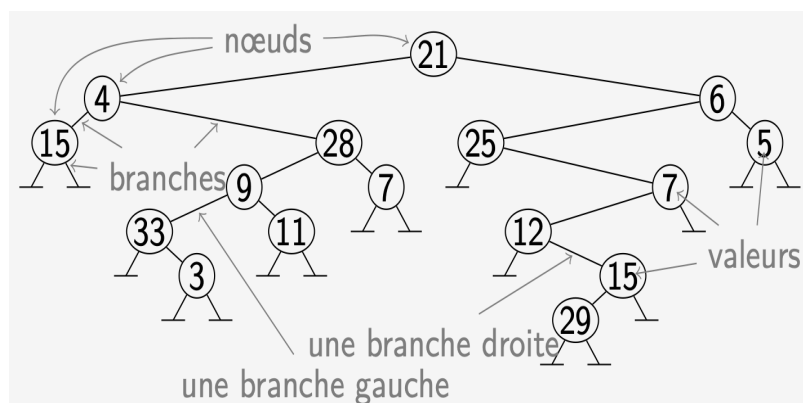


FIGURE 2 – exemple d'arbre binaire et vocabulaire

- Dans un arbre, chaque **valeur** est stockée dans un nœud. On appelle parfois cette *valeur* une ...
- Les **noeuds** sont connectés par des **arêtes**, ou **branches** qui représentent une relation de type ...
- le seul nœud qui n'a pas de prédécesseur est la ...
- Les nœuds qui n'ont pas de fils sont appelés des ...
- Le **niveau d'un nœud** est la distance qui le sépare de la ...

- La **hauteur** d'un arbre, ou sa **profondeur** est égale au *niveau* (à la profondeur) du noeud le plus ...
- la hauteur correspond aussi au **plus grand nombre de** depuis la jusqu'à l'une de ses
- Le **degré** d'un arbre est égal au
- Un arbre de **degré** égal à 1 est ...

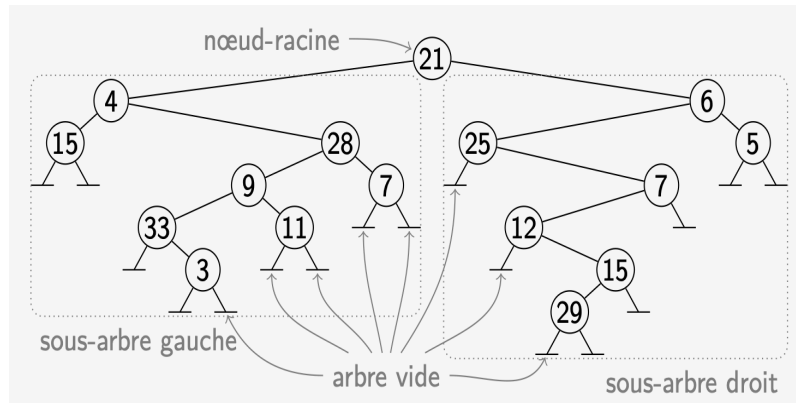


FIGURE 3 – définition récursive des arbres binaires

- La **hauteur** d'un arbre est ...
- La **taille** d'un arbre est son nombre de ...

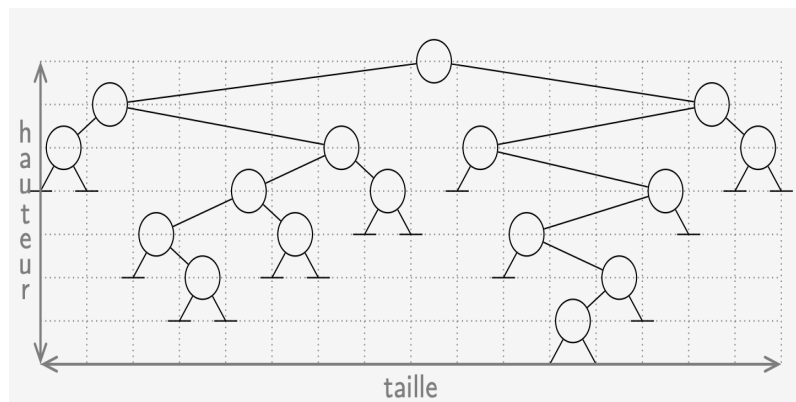


FIGURE 4 – dimensions d'un arbre

1.2 Arbres binaires

- **Arbre binaire** : arbre de degré égal à 2. Chaque noeud a au plus : le **fil** et le **fil**
- **Arbre binaire équilibré** : pour chaque noeud interne, les sous-arbres gauche et droit ont une même (ou qui diffère d'une unité).
- **Arbre binaire complet** : tous les niveaux de l'arbre sont remplis.

Partie 2

Implémentations

2.1 Liste

Un noeud peut être représenté par une liste [clé, fils gauche, fils droit].

Et comme les *fils gauche* ou *fils droit* sont des noeuds, on y mettra une nouvelle liste imbriquée [clé, fils gauche, fils droit].

Pour les feuilles, la liste s'écrit : [clé, None, None].

Le petit arbre suivant peut être représenté par ['r', ['a', None, None], ['b', None, None]]

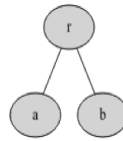


FIGURE 5 – petit arbre de taille 3

2.2 Classe

La classe suivante va permettre d'implémenter les arbres binaires :

```

1 class ArbreBinaire:
2     def __init__(self, valeur):
3         self.valeur = valeur
4         self.fils_gauche = None
5         self.fils_droit = None

```

Exemple :

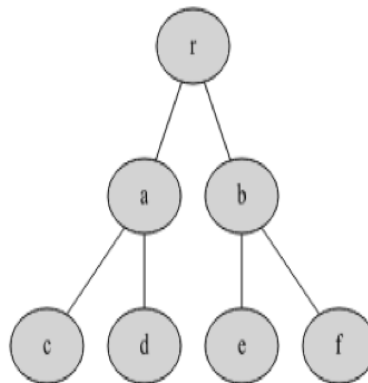


FIGURE 6 – exemple d'arbre de hauteur 2

```

1 racine = ArbreBinaire('r')
2 noeud1 = ArbreBinaire('a')
3 noeud2 = ArbreBinaire('b')
4 noeud3 = ArbreBinaire('c')
5 ...
6 racine.fils_gauche = noeud1
7 racine.fils_droit = noeud2
8 noeud1.fils_gauche = noeud3
9 ...

```

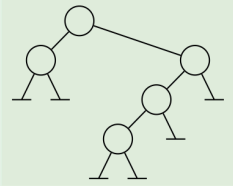
Parcours

Un parcours d'arbre définit dans quel ordre on parcourt ses noeuds. L'algorithme opère alors un **traitement** sur chacune des clés de l'arbre, dans un ordre choisi.

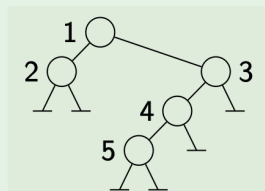
Definitions : Un **parcours** est un algorithme qui appelle une fonction, ou un méthode sur tous les noeuds d'un arbre.

L'**ordre** sur les noeuds dans lequel la méthode est appelée doit être fixé. Il y a plusieurs choix, qui diffèrent par la seule position de l'instruction `Afficher clef [r]`, que l'on aurait pu remplacer par `Traitement clef [r]`.

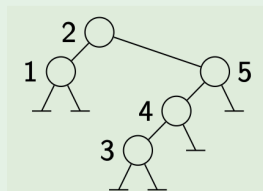
Pour l'arbre :



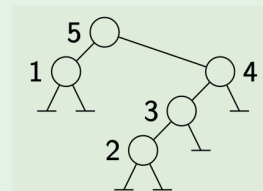
les ordres de traitement des nœuds sont, selon les parcours :



préfixe



infixe



postfixe

FIGURE 7 – exemples de parcours

3.1 Parcours ...

```

1  ParcoursPostfixe ( Arbre binaire T de racine r )
2    ParcoursPostfixe(Arbre de racine fils_gauche[r])
3    ParcoursPostfixe(Arbre de racine fils_droit[r])
4    Afficher clef [ r ]

```

3.2 Parcours ...

```

1  ParcoursPrefixe ( Arbre binaire T de racine r )
2    Afficher clef [ r ]
3    `ParcoursPrefixe(Arbre de racine fils_gauche[r])
4    `ParcoursPrefixe(Arbre de racine fils_droit[r])
5

```

3.3 Parcours ...

```

1  ParcoursInfixe ( Arbre binaire T de racine r )
2    ParcoursInfixe(Arbre de racine fils_gauche[r])
3    Afficher clef [ r ]
4    ParcoursInfixe(Arbre de racine fils_droit[r])

```