

Chapter 1

Arbres

Programmes:

Contenus	Capacités attendues	Commentaires
Arbres : structures hiérarchiques. Arbres binaires : nœuds, racines, feuilles, sous-arbres gauches, sous-arbres droits.	Identifier des situations nécessitant une structure de données arborescente. Évaluer quelques mesures des arbres binaires (taille, encadrement de la hauteur, etc.).	On fait le lien avec la rubrique « algorithmique ».
Algorithmes sur les arbres binaires et sur les arbres binaires de recherche.	Calculer la taille et la hauteur d'un arbre. Parcourir un arbre de différentes façons (ordres infixe, préfixe ou suffixe ; ordre en largeur d'abord). Rechercher une clé dans un arbre de recherche, insérer une clé.	Une structure de données récursive adaptée est utilisée. L'exemple des arbres permet d'illustrer la programmation par classe. La recherche dans un arbre de recherche équilibré est de coût logarithmique.

Support: Notebook Jupyter *TD_arbre.ipynb*

Donald Knuth, auteur de l'ouvrage de référence sur l'algorithmique, en plusieurs volumes, intitulé *The Art of Computer Programming* (L'art de la programmation informatique), considère les arbres comme la structure la plus fondamentale de toute l'informatique.

1.1 Définition

Un arbre est constitué de nœuds ou sommets, reliés entre eux par des arêtes selon une relations pères-fils. Lorsqu'un sommet est distingué par rapport aux autres, et que l'on impose une orientation, on obtient la notion d'**arborescence**. Ce sommet est la **racine**. Par abus de langage on utilise le mot arbre pour arborescence. On distingue trois types de nœuds :

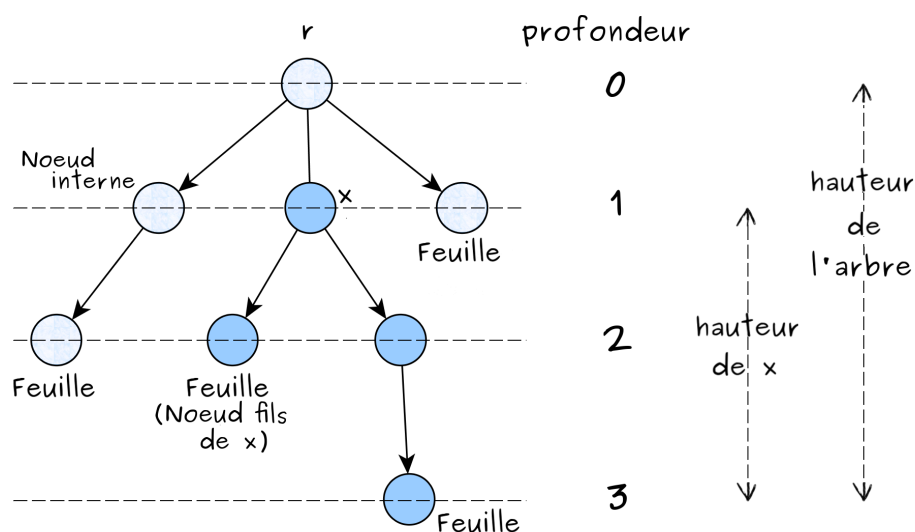
La racine: unique nœud ne possédant pas de parent.

les feuilles (ou nœuds externes): éléments ne possédant pas de fils dans l'arbre ;

les nœuds internes: éléments possédant des fils (sous-branches).

1.2 Mesures

- La profondeur d'un nœud est la distance (nombre d'arêtes) de la racine au nœud.
- La hauteur d'un arbre est la plus grande profondeur d'une feuille de l'arbre.
- La taille d'un arbre est son nombre de nœuds,
- La longueur de cheminement est la somme des profondeurs de chacune des feuilles.
- L'arité (ou le degré) d'un nœud est le nombre de fils du nœud.
- L'arité d'un arbre est l'arité maximale des nœuds de l'arbre.



Dans le cas de figure ci-dessus, l'arité de $\mathcal{A}$ est 3, celle de x vaut 2, sa profondeur vaut 1, sa hauteur vaut 2, et la hauteur de l'arborescence vaut 3. En foncé, un sous-arbre de racine x . Dans la suite, on ne s'intéressera qu'aux arbres d'arité 2: les arbres binaires. Ils sont les plus utilisés en informatique et les plus simples à étudier. En outre il est toujours possible de "binariser" un arbre non binaire, ce qui nous permettra dans ce chapitre de n'étudier que les structures d'arbres binaires.

1.3 Arbres binaires

Un arbre binaire est défini de la façon suivante : ou bien il est vide, ou bien il est constitué d'un nœud racine qui possède deux fils au maximum, un fils gauche et un fils droit, qui sont tous les deux des arbres binaires. Attention, ils ne sont pas échangeables !

Il s'agit donc d'une définition récursive.

Détaillons cette définition:

Définition : Un arbre binaire est un ensemble fini de sommets qui est :

- soit l'arbre vide, noté Δ ou NIL ;
- soit un triplet (e, g, d) , appelée nœud, dans lequel:
 - e est l'élément, ou encore étiquette, de la racine de l'arbre.
 - g est le sous-arbre gauche de l'arbre.
 - d est le sous-arbre de droite.

Lorsque l'arbre est étiqueté, ce qui est le cas notamment pour les arbres binaires de recherche que nous verrons plus loin, un sommet comporte un champ supplémentaire, qui est son contenu ou sa clé.

Exemple L'arbre binaire $\mathcal{A}_1 = (3, (1, \Delta, \Delta), \Delta)$ où Δ désigne l'arbre vide est représenté par:

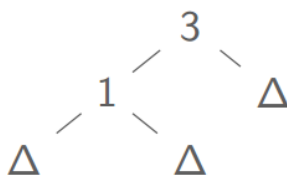


Figure 1.1: Un arbre binaire étiqueté

1. Quelques arbres binaires Dessinez chacun des arbres ci-dessous, donnez sa taille et sa hauteur, le nombre de feuilles, le nombre de nœuds à chaque profondeur.

1. $(1, \Delta, \Delta)$
2. $(3, (1, \Delta, (4, (1, \Delta, (5, \Delta, \Delta))), \Delta)), \Delta)$
3. $(3, (1, (1, \Delta, \Delta), \Delta), (4, (5, \Delta, \Delta), (9, \Delta, \Delta)))$
4. $(3, (1, (1, \Delta, \Delta), (5, \Delta, \Delta)), (4, (9, \Delta, \Delta), (2, \Delta, \Delta)))$

2. Des arbres binaires Combien de feuilles au minimum comporte un arbre binaire de hauteur h ? Au maximum ? Combien de nœuds au minimum comporte un arbre binaire de hauteur h ? Au maximum ?

3. Squelettes d'arbres binaires On appelle squelette ou forme d'arbres binaires tout arbre binaire dans lequel on ne tient pas compte des étiquettes. Combien y a-t-il de squelettes d'arbres binaires de taille 0, de taille 1 ? Dessinez tous les squelettes d'arbres binaires de taille 2, 3, 4 ; donnez-en le nombre.

4. Taille et hauteur Proposez des algorithmes pour calculer:

1. la taille d'un arbre binaire.
2. la hauteur d'un arbre binaire. On conviendra conventionnellement que la hauteur de l'arbre vide Δ est -1.

1.3.1 Caractéristiques des arbres binaires (Hors programme NSI)

Les arbres binaires sont caractérisés par le fait que chaque nœud possède au plus deux fils. D'autres caractéristiques sont définies, qui permettent par exemple d'identifier des arbres binaires pour lesquels le coût de certaines opérations sera minimal, ou de définir des algorithmes spécifiques à ces arbres:

- Arbre filiforme ou dégénéré: arbre dans lequel tous les nœuds internes n'ont qu'un seul fils. (Un arbre filiforme ne possède donc qu'une unique feuille.)
- Arbre parfait: arbre non vide dans lequel tous les niveaux sont remplis à l'exception éventuelle du dernier, et dans ce cas les feuilles du dernier niveau sont alignées à gauche.
- Un arbre équilibré ou AVL: l'arbre binaire A est équilibré si tous les chemins menant de la racine à une feuille ont pour longueur $\text{hauteur}(A)$ ou $\text{hauteur}(A)-1$. Ainsi, l'accès à n'importe lequel des nœuds est en moyenne minimisé.

Exercices: Se repérer dans cette forêt d'arbres

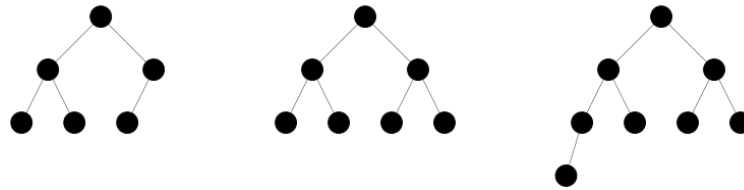


Figure 1.2: Arbres parfaits à 6, 7 et 8 sommets.

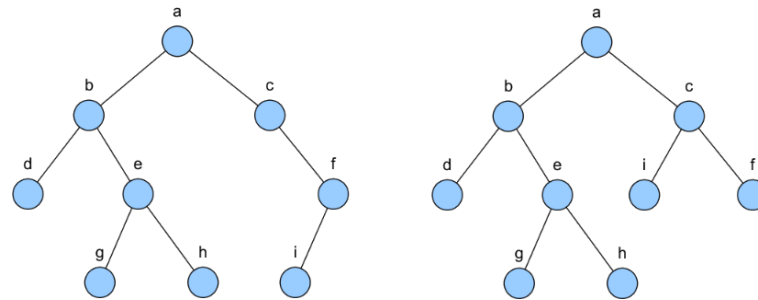


Figure 1.3: Quel est l'arbre équilibré ?

Compter ces arbres

1. Combien de nœuds au maximum comporte un arbre localement complet de hauteur h ? Au minimum ?
2. Combien de nœuds comporte un arbre complet de hauteur h ?
3. Combien de squelettes d'arbres parfaits de hauteur h existe-t-il ?
4. Combien de squelettes d'arbres filiformes de hauteur h existe-t-il ?

Reconnaître ces arbres Proposez des algorithmes pour les prédicats

1. reconnaître un arbre binaire filiforme
2. reconnaître un arbre binaire localement complet
3. reconnaître un arbre binaire complet
4. reconnaître un arbre binaire parfait

Superposer deux arbres Proposez un prédicat pour tester l'égalité du squelette de deux arbres binaires.

1.4 Implémentation objet d'un arbre binaire

On peut définir seulement une classe `Noeud`, une classe `Arbre` ou les deux. En voici une définition avec une seule classe `Arbre`:

```
class Arbre:
    def __init__(self, valeur):
        self.v = valeur
        self.fg = None
        self.fd = None

    def ajout_gauche(self, val):
```

```

        self.fg = Arbre(val)

def ajout_droit(self, val):
    self.fd = Arbre(val)

def affiche(self):
    """permet d'afficher un arbre"""
    if self == None:
        return None
    else:
        return [self.v, Arbre.affiche(self.fg), Arbre.affiche(self.fd)]

def taille(self):
    if self == None:
        return 0
    else:
        return 1 + Arbre.taille(self.fg) + Arbre.taille(self.fd)

def hauteur(self):
    if self == None:
        return 0
    elif self.fg == None and self.fd == None:
        return 0
    else :
        return 1+max(Arbre.hauteur(self.fg), Arbre.hauteur(self.fd))

def get_valeur(self):
    if self == None:
        return None
    else:
        return print(self.v)

```

A faire

1.4.1 Autres algorithmes

Calcul de la taille et de la profondeur. A vérifier sur l'arbre précédent.

1.5 Arbre binaire de recherche (ABR)

1.5.1 Définition

Un arbre binaire de recherche (ou ABR) est une structure de donnée qui permet de représenter un ensemble de valeurs si l'on dispose d'une relation d'ordre sur ces valeurs.

Définition Soit E un ensemble muni d'une relation d'ordre, et soit A un arbre binaire portant des valeurs de E .

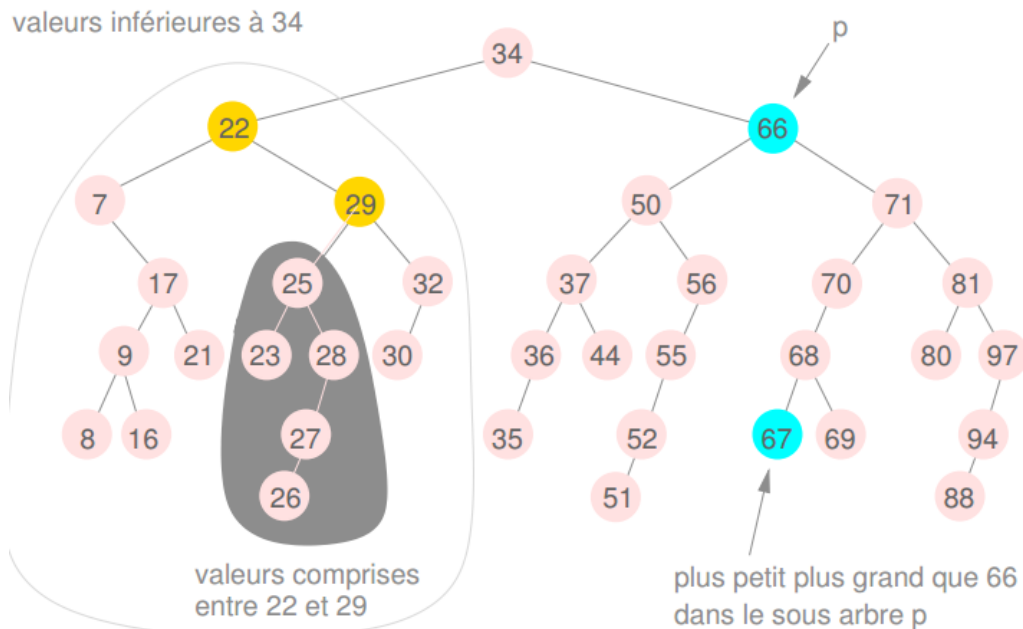
L'arbre $\mathcal{A}$ est un arbre binaire de recherche si pour tout noeud p de $\mathcal{A}$, la valeur de p est strictement plus grande que les valeurs figurant dans son sous-arbre gauche et strictement plus petite que les valeurs figurant dans son sous-arbre droit. Cette définition suppose donc qu'une valeur n'apparaît au plus qu'une seule fois dans un arbre de recherche.

Autrement dit, un arbre binaire de recherche est un arbre binaire tel que :

1. Chaque sommet possède une clé prise dans un ensemble ordonné.
2. Pour tout sommet x :

- Pour tout sommet y du sous-arbre de gauche de x clé $(y) < \text{clé}(x)$.
- Pour tout sommet y du sous-arbre de droite de x , clé $(x) < \text{clé}(y)$.

Exemple:



1.5.2 Primitives

Les opérations caractéristiques sur les arbres binaires de recherche sont:

1. l'insertion,
2. la suppression,
3. la recherche d'une valeur.

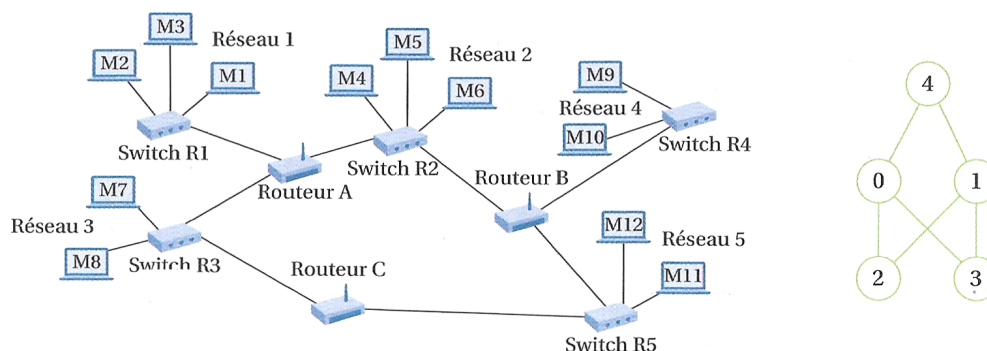
Ces opérations sont peu coûteuses si l'arbre n'est pas trop déséquilibré. En pratique, les valeurs sont des clés permettant d'accéder à des enregistrements.

1.6 Annexe

Une définition plus formelle d'un arbre:

1.6.1 Les arbres sont des graphes particuliers

Un graphe est la donnée d'un ensemble fini V de points ou sommets (vertices en anglais) et d'un ensemble de liens entre ces points.



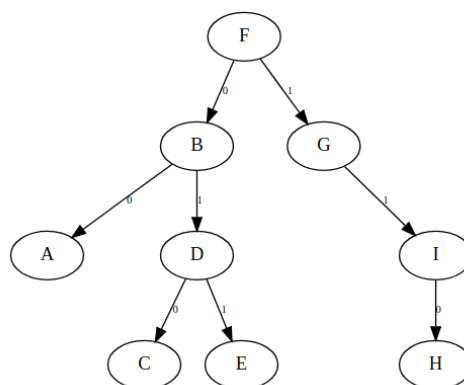
Graphes d'un réseau et graphes dont les sommets sont numérotés.

Exemples: système de fichiers, base de données, sites web, fichiers XML.

Définition Soit $G = (S, A)$ un graphe orienté ou non. Les définitions ci-dessous sont équivalentes:

1. G est un arbre $\Leftrightarrow G$ connexe acyclique.
2. G est un arbre $\Leftrightarrow \forall (x, y) \in S^2, \exists! \mathcal{C}(x, y)$.

où $\mathcal{C}(x, y)$ est une chaîne simple d'extrémités x et y .



Propriétés: Soit G un graphe non orienté d'ordre n . Les propositions suivantes sont équivalentes :

- G est un arbre.
- G est connexe et possède $n-1$ arêtes.
- G est sans cycles simples et possède $n-1$ arêtes.