

Structures hiérarchiques : les arbres

En informatique, toutes les structures de données ne sont pas linéaires. Il existe bien d'autres structures. Nous allons étudier ici les structures hiérarchiques, aussi appelés arbres.

Exemples de structures arborescentes

Arborescence de fichiers

Les arbres servent à décrire des arborescences comme par exemple la structure des fichiers sur votre ordinateur : Sur Unix/Linux, le sommet de l'arborescence est désigné par `/`. On trouve en dessous un certain nombre de dossiers principaux comme `bin`, `lib`, `home`, `usr` etc... chacun ayant un rôle prédéfini : `home` sert par exemple à stocker les répertoires personnels de tous les utilisateurs du système. Pour avoir la liste de ce qui se trouve à la racine du système, on tape la commande

```
$ ls -l
total 176
drwxr-xr-x  2 root root 12288 avril 15 19:14 bin
drwxr-xr-x  3 root root  4096 avril 17 09:35 boot
drwxrwxr-x  2 root root  4096 oct. 21 2017 cdrom
drwxr-xr-x 20 root root  4940 avril 17 08:53 dev
drwxr-xr-x 202 root root 12288 avril 16 19:19 etc
drwxr-xr-x  8 root root  4096 juin 14 2019 home
etc...
```

Les dossiers sont indiqués par le caractère `d` en début de ligne. L'arborescence se déroule de manière analogue pour chacun de ces dossiers, par exemple, dans le dossier `/usr` on retrouve une autre arborescence :

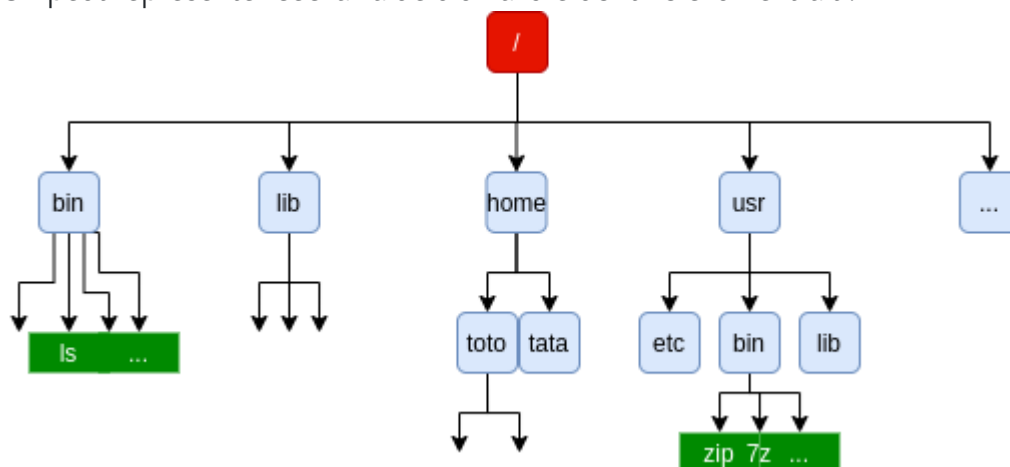
```
$ ls -l /usr
total 248
drwxr-xr-x  2 root root 131072 avril 16 19:18 bin
drwxr-xr-x  3 root root  4096 janv. 20 2018 etc
drwxr-xr-x  2 root root  4096 avril  5 20:22 games
drwxr-xr-x 116 root root 20480 avril  8 19:04 include
drwxr-xr-x 197 root root 20480 avril 16 19:17 lib
etc...
```

et ainsi de suite jusqu'à un moment où on ne trouve plus de dossiers, mais uniquement des fichiers. C'est le cas dans `/usr/bin` par exemple.

```
$ ls -l /usr/bin
total 1424264
-rwxr-xr-x  1 root root      96 avril  7 14:05 2to3-2.7
-rwxr-xr-x  1 root root 10104 avril 23 2016 411toppm
-rwxr-xr-x  1 root root   39 août  9 2019 7z
-rwxr-xr-x  1 root root   40 août  9 2019 7za
-rwxr-xr-x  1 root root   40 août  9 2019 7zr
etc...
```

Nous n'avons plus ici que des fichiers et l'arborescence se termine.

On peut représenter ceci à l'aide d'un arbre dont voici un extrait :



A vous de jouer

Identifiez des exemples concrets dans lesquels la structure d'arbre intervient.

Vocabulaire sur les arbres

Reprenons l'exemple précédent :

- On retrouve la **racine** de notre arborescence en rouge : c'est là où notre arbre commence.
- On identifie les dossiers et sous dossiers en bleu : on les appelle des **noeuds**.
- les arêtes qui relient les noeuds sont appelées aussi **branches** de l'arbre.
- Les noeuds à un même niveau sont dits **frères**. Les dossiers descendent d'un noeud sont les **fil**s de ce noeud.
- On a de même le noeud **père** qui désigne le noeud dont un noeud descend.
- Les fichiers en vert représentent les terminaisons de l'arbre. Lorsqu'un noeud n'a pas de descendants, on parle de **feuilles**.

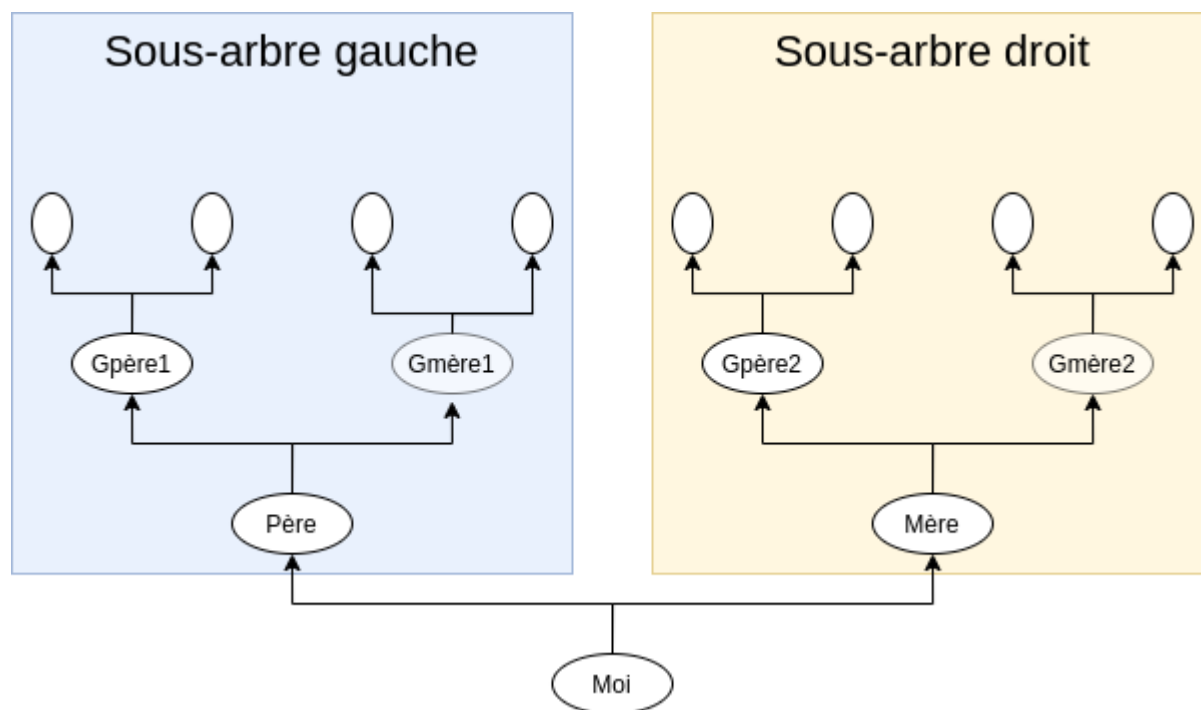
On définit la **profondeur** d'un noeud ou d'une feuille comme étant le nombre de branches traversées pour aller du noeud à la racine de l'arbre.

La **hauteur** de l'arbre est la profondeur maximale des noeuds de l'arbre.

Les arbres binaires

Les **arbres binaires** sont un cas particulier d'arbre dans lequel de chaque noeud **partent au plus deux branches**.

Très clairement la structure d'arborescence de fichiers vue ci dessus n'est pas un arbre binaire. Si on descende l'arbre de nos descendants, ce n'est pas non plus un arbre binaire car une personne peut avoir plus que 2 enfants. Par contre, si on s'intéresse à nos ascendants, nous avons alors un arbre binaire :

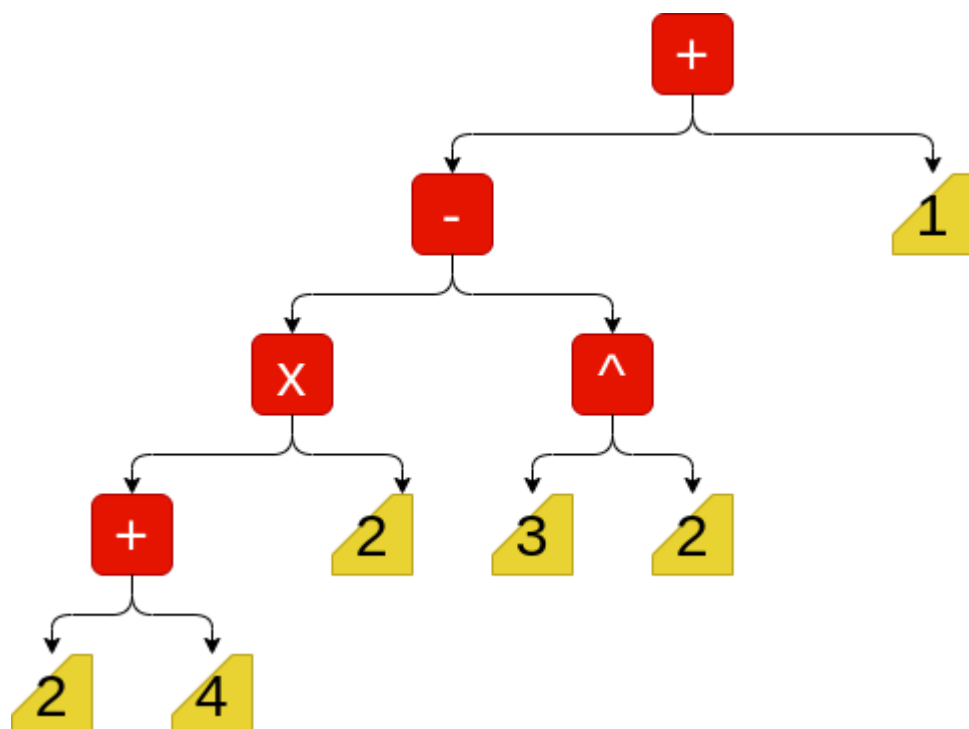


Il est intéressant de remarquer que dans un arbre binaire, les sous-arbres formés en prenant comme racine les fils d'un noeud *qui n'est pas une feuille* sont aussi des arbres binaires. On les nomme sous-arbre gauche et sous-arbre droit du noeud considéré. Sur l'illustration, nous avons représenté les sous arbres gauche et droit de la racine.

Un exemple d'arbre binaire : l'arbre de calcul.

On considère l'expression suivante : $A = (2+4) \times 2 - 32 + 1$

On peut représenter cette expression par un arbre binaire dans lequel les noeuds sont les opérations et les feuilles, les nombres. Cela présente beaucoup d'avantages pour calculer l'expression car une fois l'expression écrite sous forme d'arbre, l'algorithme permettant de l'évaluer est aisé. Nous aurons l'occasion d'y revenir un peu plus tard.

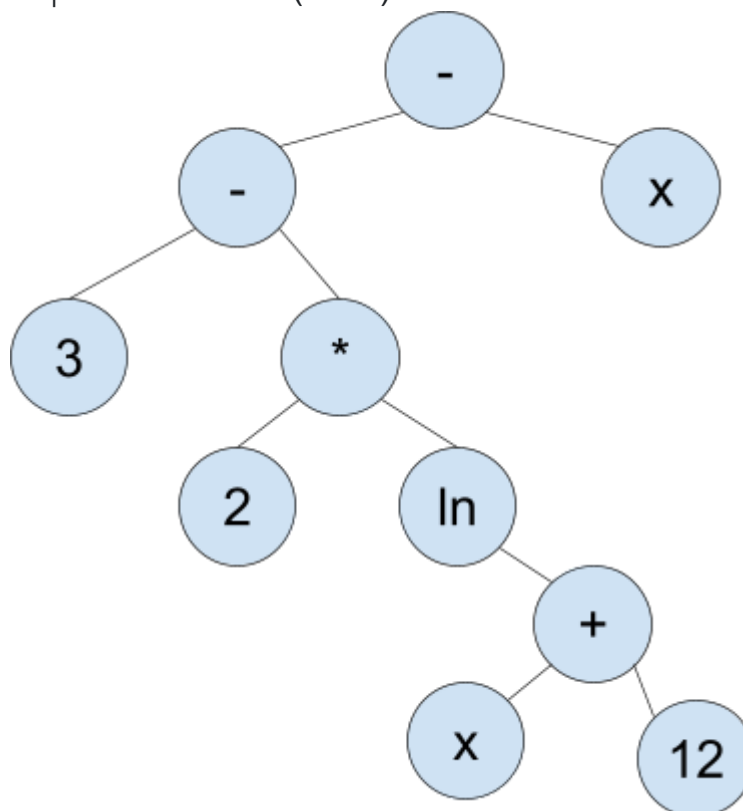


Pour obtenir cet arbre, l'idée est de parcourir notre expression à la recherche de l'opération **de moins forte priorité**. Celle-ci prend alors place dans un noeud. On sépare alors notre expression en deux sous expressions : gauche et droite, permettant chacune de construire un sous arbre gauche et droit en répétant la même méthode de construction. Lorsqu'il n'y a plus d'opération, nous sommes au niveau d'une feuille et la construction du sous-arbre s'arrête.

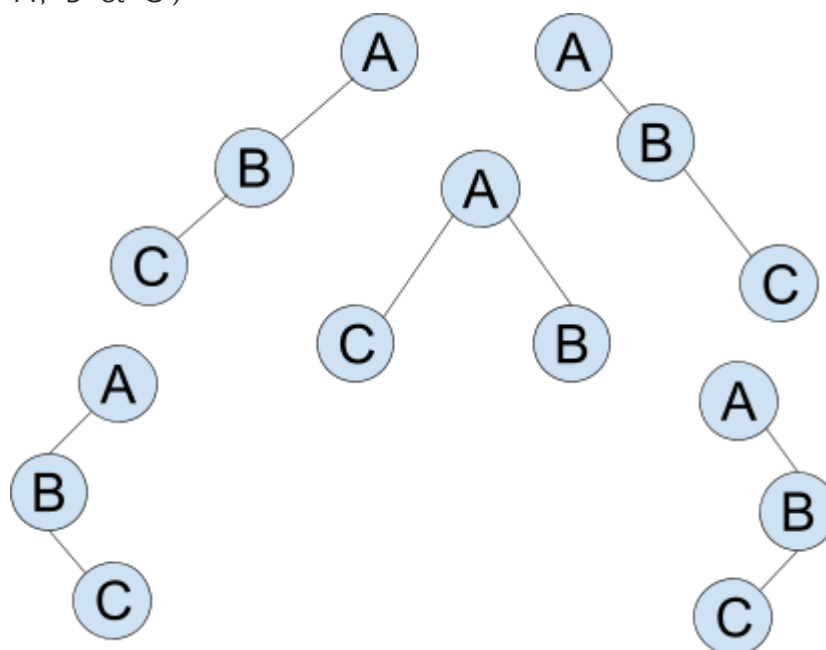
A vous de jouer

1. Quelle est la hauteur de l'arbre ci-dessus ?
L'arbre a une hauteur 4
2. Quelle est la feuille de moindre profondeur ?
C'est la feuille 1
3. Quelles sont les feuilles qui sont à la même hauteur que 3 ?
Ce sont les feuilles 2 et 2

4. Ecrire un arbre binaire correspondant à l'expression $B = 3 - 2 \ln(x + 12) - x$



5. Ecrire tous les arbres binaires à 3 noeuds (on pourra les nommer "A", "B" et "C")



6. Combien y a-t-il d'arbres binaires à 5 noeuds ? on pourra remarquer qu'il y a 14 arbres binaires à 4 noeuds.
[Il y en a 42 types différents](#)

Conclusion

Les arbres sont donc partout, en informatique, en mathématiques mais aussi dans la vie de tous les jours.

Nous les reverrons donc très régulièrement cette année et étudierons un certain nombre d'algorithmes relatifs aux arbres.

Nous allons voir en TP comment implémenter ces arbres en Python. Il y a plusieurs manières de procéder : on peut par exemple utiliser des tableaux, mais on peut aussi définir notre propre type arbre en le construisant sur mesure. Cela est rendu possible grâce à la notion nouvelle cette année de **programmation orientée objet**. Notre TP va permettre d'explorer cette nouvelle voie.