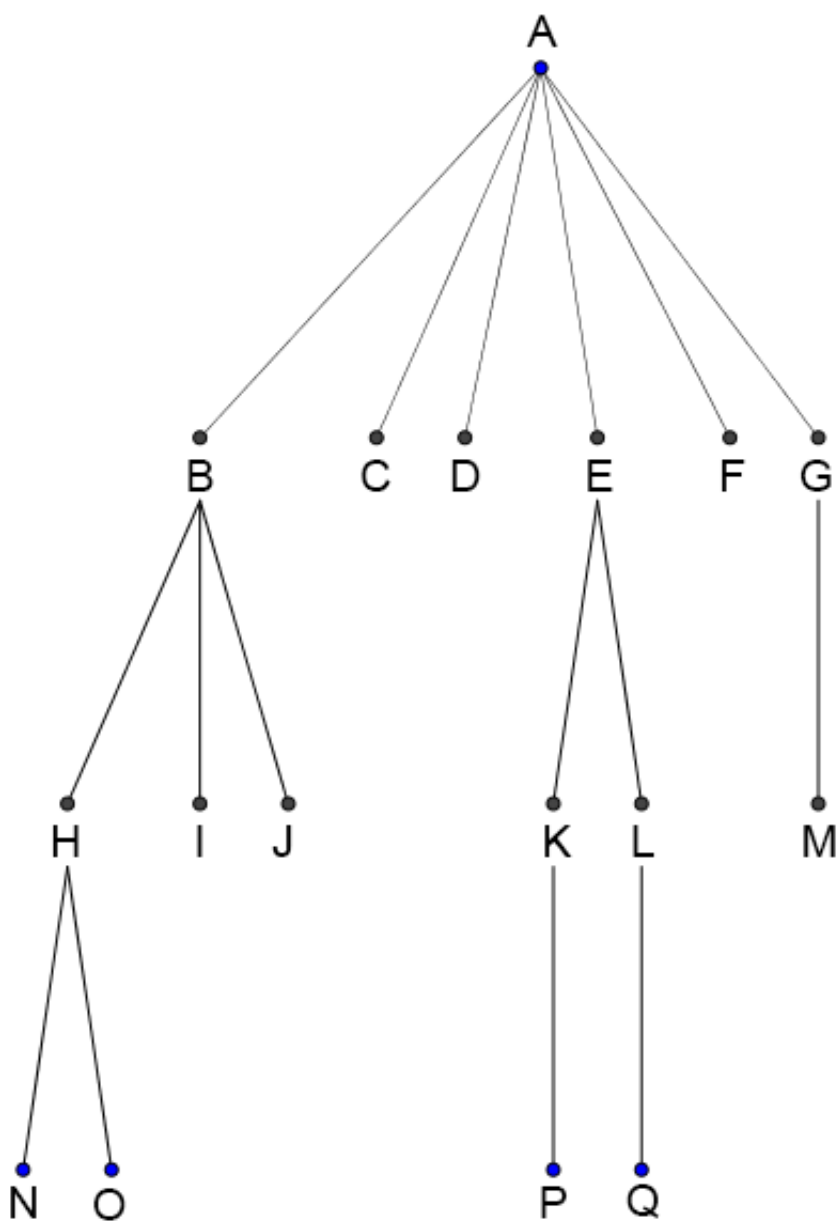


# TD 15 - Les arbres

## I - un pb sur les arbres (pas binaires)

---

Voici la représentation d'un arbre.



1. Compléter le tableau.

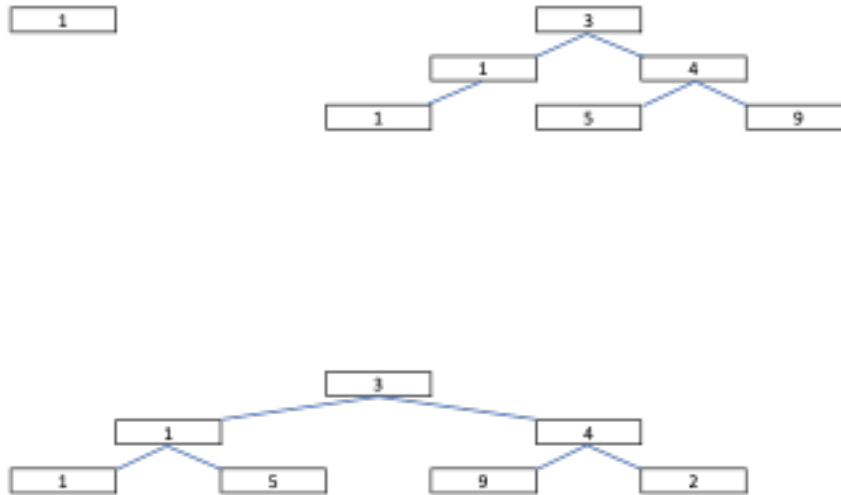
Noeud	interne? V ou F	Feuille? V ou F	Racine? V ou F	profondeur	arité
A	V	F	V	0	6
B	V	F	F	1	3
C	F	V	F	1	0
D	F	V	F	1	0
E	V	F	F	1	2
F	F	V	F	1	0
G	V	F	F	1	1
H	V	F	F	2	2
I	F	V	F	2	0
J	F	V	F	2	0
K	V	F	F	2	1
L	V	F	F	2	1
M	F	V	F	2	0
N	F	V	F	3	0
O	F	V	F	3	0
P	F	V	F	3	0
Q	F	V	F	3	0

2. Calculer:

1. La taille de l'arbre: 17
2. La hauteur de l'arbre: 3
3. L'arité de l'arbre: 6

## II - un cas particulier : les arbres binaires

1. Voici les arbres :



2. 1 feuille. 1 feuille.

3.  $2^3 = 8$ .  $2^4 = 16$ .  $2^h$ .

4.  $2^2 < 13 \leq 2^3$ . Donc la hauteur minimale est 3.

$2^3 < 13 \leq 2^4$ . Donc la hauteur minimale est 4.

Soit  $n$  tel que  $2^{n-1} < h \leq 2^n$ .  $n$  est la hauteur minimale d'un arbre ayant  $h$  feuilles.

5. Qui dit taille minimale dit hauteur minimale.

Donc pour 5 feuilles, la hauteur est 3. Il faut donc que les sous-arbres gauches et droites soient le plus complet possible: il aura 9 noeuds.

Pour 13 feuilles, on fait de même. Il y aura 25 noeuds.

Pour  $h$  feuilles, on fait de même. Soit  $n$  tel que  $2^{n-1} < h \leq 2^n$ .  $n$  est la hauteur minimale de l'arbre.

Le sous-arbre gauche sera complet soit  $2^n - 1$  noeuds soit  $2^n$  noeuds en comptant la racine. Il reste donc  $h - 2^{n-1}$  feuilles dans le sous-arbre de droite. Et on réitère le procédé.

On peut remarquer que si  $h = 1 \times 2^{n-1} + \delta_{n-2} \times 2^{n-2} + \dots + \delta_2 \times 2^2 + \delta_1 \times 2^1 + \delta_0 \times 2^0$  avec les  $\delta_i = 0$  ou 1

alors la taille minimale de l'arbre est:

$$1 \times 2^n + \delta_{n-2} \times 2^{n-1} + \dots + \delta_2 \times 2^3 + \delta_1 \times 2^2 + \delta_0 \times 2^1 - 1$$

6. Voir fichier `TD12_correction.py`.

7. Voir fichier `TD12_correction.py`.

8. Voir fichier `TD12_correction.py`.

9. Voir fichier `TD12_correction.py`.

10. Voir fichier `TD12_correction.py` .
11. Voir fichier `TD12_correction.py` .

### III - Un autre cas particulier: les arbres binaires de recherche

---

1. 3.
2. La plus petite étiquette sur un arbre binaire de recherche se trouve sur le noeud le plus à gauche.  
La plus grande étiquette se trouve sur le noeud le plus à droite.  
L'étiquette médiane: ça dépend. Dans le cas d'un arbre binaire de recherche complet, il s'agit de la racine.
3. Voici l'arbre binaire de recherche complet avec les nombres de 1 à 15:



4. Le parcours infixe.
5. Voir fichier `TD12_correction.py` .
6. Voir fichier `TD12_correction.py` .
7. Voir fichier `TD12_correction.py` .