

TD2 : implémentation des arbres binaires en python sous forme de liste de listes

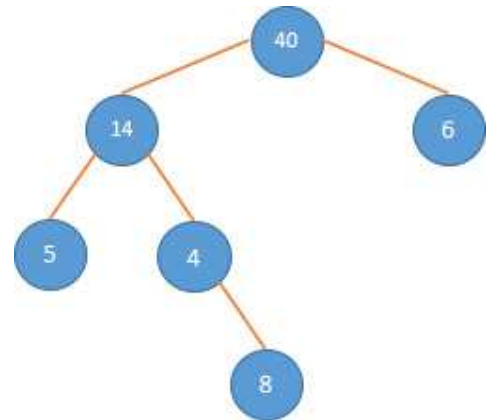
Un arbre binaire peut être implémenté :

- par une liste vide [] s'il s'agit d'un arbre vide
- sinon par une liste [e, g, d] de longueur 3, dans laquelle :
 - o e est l'étiquette de la racine de l'arbre
 - o g est le sous-arbre gauche de l'arbre. Ce sous-arbre est lui-même un arbre.
 - o d est le sous-arbre droit de l'arbre. Ce sous-arbre est lui-même un arbre.

Exercice 1 : Représentations d'arbres binaires :

- a- Dessinez chacun des arbres ci-dessous :
- 1- [5,[3,[],[]],[8,[],[]]]
 - 2- [4,[7,[],[]],[9,[2,[],[]],[3,[],[]]]]
 - 3- [6,[5,[],[4,[],[]]],[0,[1,[],[]],[2,[],[]]]]

- b- Donner la représentation sous forme de liste de listes de l'arbre ci-contre :



Exercice 2 : Programmation fonctionnelle

Préliminaires :

Tester, dans Thonny, les instructions suivantes :

```
>>> max(5,4)
>>> max(5,9)
>>> max(5,5)
```

Déclarer la variable `peuplier` à laquelle on affectera l'arbre de l'exercice 1 question b.

Quelle est la taille de `peuplier` ?

Quelle est la hauteur de `peuplier` ?

Combien de feuille possède `peuplier` ?

Fonctions :

- 1- Ecrire le prédicat (retournant True ou False) `est_vide(noeud)` permettant de reconnaître si l'arbre passé en paramètre est vide.
- 2- Ecrire le prédicat (retournant True ou False) `est_une_feuille(noeud)` permettant de reconnaître si le noeud passé en paramètre est une feuille.
- 3- Ecrire le prédicat (retournant True ou False) `n_a_qu_un_fils(noeud)` permettant de reconnaître si le noeud passé en paramètre a exactement un fils (gauche ou droit).
- 4- Ecrire la fonction `taille(arbre)` retournant la taille de l'arbre passé en paramètre.
- 5- Ecrire la fonction `hauteur(arbre)` retournant la hauteur de l'arbre passé en paramètre. On conviendra conventionnellement que la hauteur de l'arbre vide est -1.

- 6- Ecrire la fonction `nb_feuilles(arbre)` retournant le nombre de feuilles de l'arbre passé en paramètre.
- 7- Ecrire la fonction `nb_noeuds_internes(arbre)` retournant le nombre de nœuds internes de l'arbre passé en paramètre.

Un arbre binaire **filiforme** (ou **dégénéré**) est un arbre dans lequel tous les nœuds internes n'ont qu'un seul fils. Un arbre vide n'est pas *filiforme*.

- 8- Combien de feuilles possède un arbre binaire filiforme de hauteur h ?
- 9- Ecrire le prédicat (retournant True ou False) `est_filiforme(arbre)` permettant de reconnaître si l'arbre passé en paramètre est *filiforme*.

Un arbre binaire **localement complet** (ou **strict**) est un arbre dont tous les nœuds internes possèdent exactement 2 fils (autrement dit, *un arbre binaire complet est un arbre dont chaque nœud possède 0 ou 2 fils*).

- 10- Ecrire le prédicat (retournant True ou False) `est_localement_complet(arbre)` permettant de reconnaître si l'arbre passé en paramètre est *localement complet*.