

## Entraînement : Arbres structure récursive

Pour tous les exercices, la grille d'évaluation est la suivante.

### Structure de données tas.

A (20)	Exemple tas / tableau correct
D (8)	Structure comprise mais erreurs (d'inattention ?)
E (1)	Exemples faux

### Algorithme.

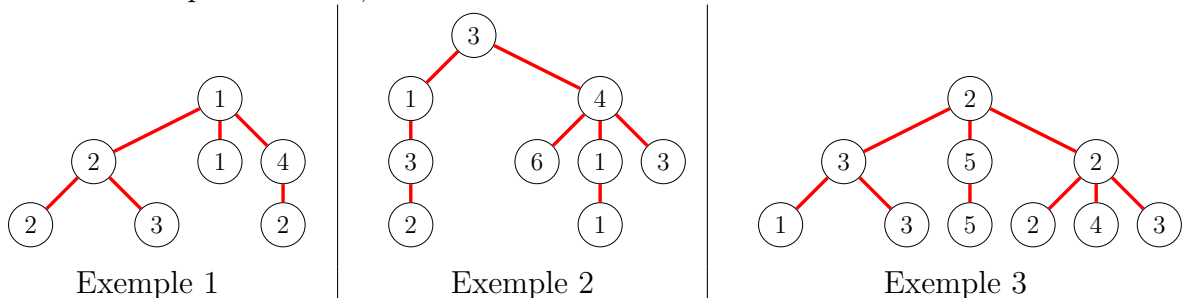
A (20)	L'algorithme répond correctement au problème posé, il est écrit de façon claire et la complexité est correcte.
B (16)	L'algorithme contient quelques erreurs mais reste globalement juste et la complexité est correcte.
C (11)	L'algorithme fonctionne globalement mais complexité non optimale ou complexité calculée fausse.
D (8)	L'algorithme ne fonctionne pas.
E (1)	Algorithme quasi inexistant ou ne répondant pas du tout au problème posé.

### Exercice 1 (Arbres – 7 pts).

On définit la structure suivante pour représenter les arbres :

```
Structure Arbre :
    valeur, un entier
    nbFils, un entier
    Fils, un tableau de taille nbFils contenant des Arbres
```

Et les trois exemples suivants,



(1) Donner les valeurs de **mystere1** et **mystere2** sur les 3 exemples d'arbres donnés.

```
Fonction mystere1(Arbre arbre) :
    s = 1
    Pour chaque fils f de arbre :
        s = s + mystere1(f)
    retourner s

Fonction mystere2(Arbre arbre) :
    m = 0
    Pour chaque fils f de arbre :
        h = mystere2(f)
        Si h > m :
            m = h
```

<pre>    retourner m + 1 Fin fonction</pre>
---------------------------------------------

- (2) Que calculent `mystere1` et `mystere2` ?
- (3) Donner une fonction qui calcule le nombre de *feuilles* de l'arbre (nombre de nœuds qui n'ont pas de fils). La fonction doit retourner 4 sur l'exemple 1, 4 sur l'exemple 2 et 6 sur l'exemple 3.
- (4) On dit qu'un arbre est équilibré si toutes les feuilles sont à la même profondeur. Dans les arbres donnés en exemple, seul l'exemple 3 est équilibré (toutes ses feuilles sont à la profondeur 3, pour les exemples 1 et 2, il y a des feuilles aux profondeurs 2 et 3). Proposer un algorithme qui teste si un arbre donné est équilibré. (Note : il y a plusieurs solutions possibles, plus l'algorithme est optimal en terme de complexité, mieux c'est).