

# SDD Arbres Exercices Partie I Corrigé

Par convention, la racine a une hauteur de zéro dans les exercices suivants.

## Exercice 1 : Vrai / Faux, vocabulaire des arbres

**VRAI.** Taille et hauteur sont égales en cas d'arbre dégénéré (ou filiforme).

**FAUX.** Seulement dans un parcours préfixe.

**VRAI** car c'est un parcours par niveau.

**VRAI.**

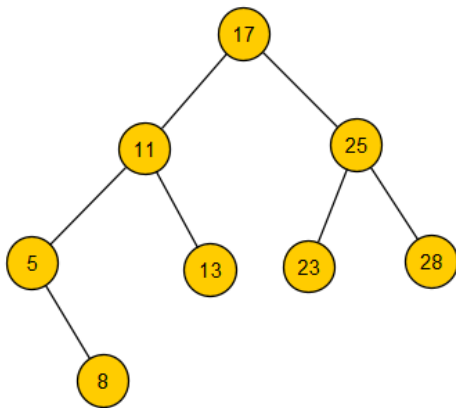
**FAUX.** Le coût de recherche est de  $\log_2(n)$  dans le cas d'un ABR équilibré ce qui est le cas (en général). Il est linéaire dans le cas d'un arbre dégénéré.

**VRAI** (seulement s'il s'agit d'un ABR).

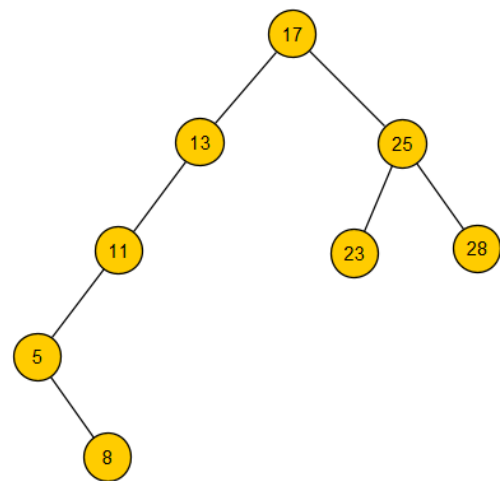
**VRAI.**

## Exercice 2 : Dessins d'ABR

1<sup>er</sup> cas :



2<sup>ème</sup> cas :



## Exercice 3 : Recherche dans un ABR

Le deuxième cas n'est pas possible car toutes les valeurs qui suivent 35 doivent être supérieures à 23 ce qui n'est pas le cas de 22.

## Exercice 4 : Ajout de méthodes dans un ABR

1/ La plus petite valeur d'un ABR est donnée par celle de la feuille la plus à gauche.

2/ La plus grande valeur d'un ABR est donnée par celle de la feuille la plus à droite. Il suffit de remplacer « gauche » par « droite » dans le programme ci-contre.

On suppose qu'une classe a été définie en amont !

```
# Version itérative
def min_it(self) :
    s = self
    # On cherche la feuille la plus à gauche possible
    while s.gauche :
        s = s.gauche

    return s.valeur

# Version récursive
def min_rec(self) :
    # Cas d'arrêt
    if not self.gauche :
        return self.valeur
    # Cas général
    else :
        return self.gauche.min_rec()
```

**Exercice 5 : Parcours d'arbre binaire**

1/A-B-D-H-G-C-F-J-E-I  
2/A-B-H-J-G-D-C-E-I-F  
3/J-H-B-G-A-E-C-I-D-F  
4/J-H-G-B-E-I-C-F-D-A

