

SDD Arbres Exercices Partie I Corrigé

Exercice 1 : QCM, vocabulaire des arbres (une seule réponse possible)

1/ **Réponse b)** . Les feuilles du sous-arbre droit ont la hauteur maximale.

2/ **Réponse a)**. C'est le nombre total de nœuds.

3/ **Réponse c)**. L'arbre possède un nœud ayant trois fils.

Exercice 2 : QCM, vocabulaire du parcours d'arbres (une seule réponse possible)

1/ **Réponse b)**. Lecture des nœuds de gauche à droite, étage par étage.

2/ **Réponse d)**. Lecture de la racine, puis du sous-arbre gauche (en préfixe), puis du sous-arbre droit (en préfixe).

3/ **Réponse c)**. Lecture du sous-arbre gauche (en infixe), puis de la racine, puis du sous-arbre droit (en infixe).

4/ **Réponse a)**. Lecture du sous-arbre gauche (en postfixe), puis du sous-arbre droit (en postfixe), puis de la racine.

Exercice 3 : QCM, arbre d'expression arithmétique (plusieurs réponses possibles)

Réponse a), c) et e). Les opérateurs mathématiques mentionnés sont des opérateurs binaires, il faut donc deux opérandes soient deux fils exactement pour chaque nœud interne.

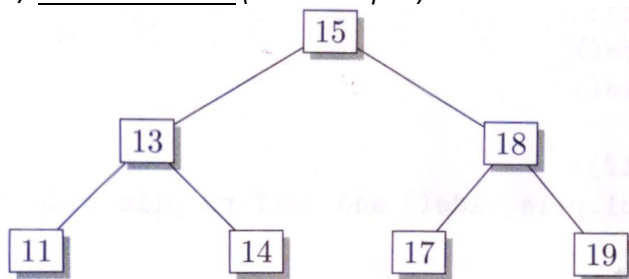
Les nombres sont obligatoirement sur les feuilles, seul un opérateur peut avoir des fils.

Exercice 4 : QCM, ABR (une seule réponse possible)

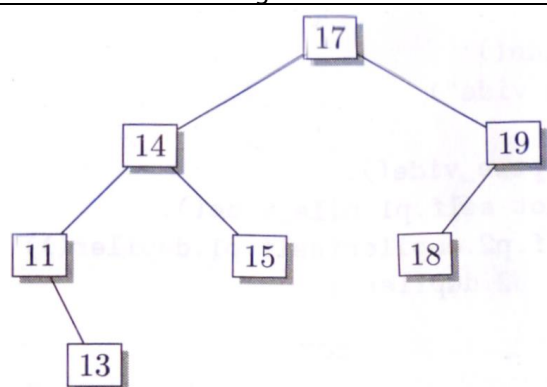
Réponse a)

Exercice 5 : Construire un ABR

1/ Hauteur de deux (arbre complet)

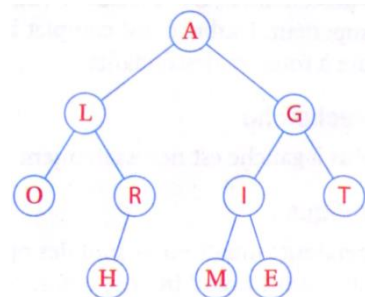


2/ 4 nœuds au sous-arbre gauche et 2 à celui de droite



Exercice 6 : Reconstruire un arbre

Astuce : Le nœud « A » étant obligatoirement la racine (parcours préfixe), on sait que grâce au parcours infixe, les nœuds LORH sont dans le sous-arbre gauche et que les nœuds GITME dans le sous-arbre droit.



Exercice 7 : ABR

1/

a) Il y a 9 feuilles.

b) Il est composé des nœuds 19 et 21.

c) Sa hauteur est de 4 (la racine comptant 1) et sa taille de 9.

d) La valeur `val` doit être strictement supérieur à 15 (fils droit) mais strictement inférieur à 18 (racine), soit 16 ou 17.

2/

- a) On trouve : 12 – 13 – 15 – 16 – 18 – 19 – 21 – 23 – 32.
- b) On trouve : 12 – 13 – 16 – 15 – 21 – 19 – 32 – 23 – 18.

Remarque : le parcours infixe affiche les valeurs d'un ABR dans l'ordre croissant.

Exercice 8 (*) : Ajout d'un élément dans un ABR

2/ Voici le programme complet, avec le parcours infixe affichant les clés dans l'ordre croissant (pour vérification).

```
# Classe créant un arbre binaire de recherche
class Node :
    def __init__(self, val) :
        self.value = val
        self.parent = None # Valeur par défaut
        self.left = None # Valeur par défaut
        self.right = None # Valeur par défaut

    # Surcharge de la fonction `print`
    def __str__(self) :
        return str(self.value)

    # Ajout d'une valeur dans l'ABR
    def add(self, val) :

        # Cas où la valeur se situera dans le sous-arbre gauche
        if val < self.value :
            # Cas d'arrêt
            if self.left is None : # S'il s'agit d'une feuille
                self.left = Node(val)
                self.left.parent = self # Lien vers le parent
            # Cas général
            else :
                self.left.add(val)

        # Cas où la valeur se situera dans le sous-arbre droit
        else :
            # Cas d'arrêt
            if self.right is None : # S'il s'agit d'une feuille
                self.right = Node(val)
                self.right.parent = self # Lien vers le parent
            # Cas général
            else :
                self.right.add(val)
```

```
# Parcours infixe qui doit afficher l'ABR dans l'ordre croissant
# de ses valeurs
def inf_print(self) :
    if self is None :
        return

    if self.left is not None :
        self.left.inf_print()

    print(self, end = " ")

    if self.right is not None :
        self.right.inf_print()
```

```
# Jeu de tests
tree = Node(15)
print(tree) # Attendu : 15
tree.add(10)
tree.add(23)
tree.add(19)
tree.add(25)
tree.add(20)

tree.inf_print() # Attendu : tri dans l'ordre croissant
```

Résultats obtenus

15
10 15 19 20 23 25