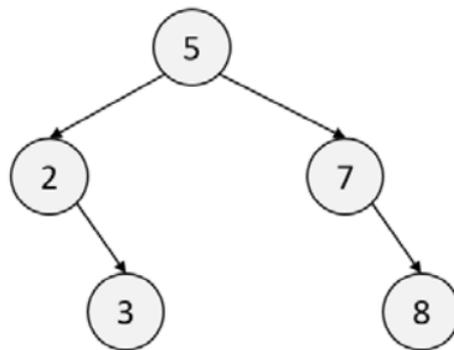


Exercice : Arbre binaire de recherche (ABR)

Partie A : Étude d'un exemple

Considérons l'arbre binaire de recherche ci-dessous.



- 1) Indiquer quelle valeur a le nœud racine et quels sont les fils de ce nœud .
- 2) Indiquer quels sont les nœuds de la branche qui se termine par la feuille qui a pour valeur 3.
- 3) Dessiner l'arbre obtenu après l'ajout de la valeur 6.

Partie B : Implémentation en Python

Voici un extrait d'une implémentation en Python d'une classe modélisant un arbre binaire de recherche.

```
class ABR:
    """Implémentation d'un arbre binaire de recherche (ABR)"""
    def __init__(self,valeur=None):
        self.valeur = valeur
        self.fg = None
        self.fd = None

    def estVide(self):
        return self.valeur == None

    def insererElement(self,e):
        if self.estVide():
            self.valeur = e
        else:
            if e < self.valeur:
                if self.fg:
                    self.fg.insererElement(e)
                else:
                    self.fg = ABR(e)
            if self.valeur > e:
                if self.fd:
                    self.fd.insererElement(e)
                else:
                    self.fd = ABR(e)
```

- 1) Expliquer le rôle de la fonction `__init__`.
- 2) Dans cette implémentation, expliquer ce qui se passe si on ajoute un élément déjà présent dans l'arbre.

- 3) Recopier et compléter les lignes de code surlignées ci-dessous permettant de créer l'arbre de la **partie A**.

```
arbre = ABR(.....)
arbre.insererElement(2)
arbre.insererElement(.....)
arbre.insererElement(7)
arbre.insererElement(.....)
```

- 4) Donner le nom du parcours qui permet de visiter les valeurs d'un arbre binaire de recherche dans l'ordre croissant.
- 5) Écrire la méthode `appartient` de la classe `ABR` qui prend en argument une valeur `e` et renvoie un booléen qui affirme si cette valeur `e` est ou n'est pas dans l'arbre binaire de recherche.