

Exercice 5 (4 points)

Cet exercice porte sur les arbres binaires, la programmation orienté objet et la récursivité

Dans un arbre binaire, chaque nœud admet au plus deux enfants, appelés sous-arbre gauche et sous-arbre droit. On considère dans cet exercice des arbres binaires étiquetés avec des nombres entiers.

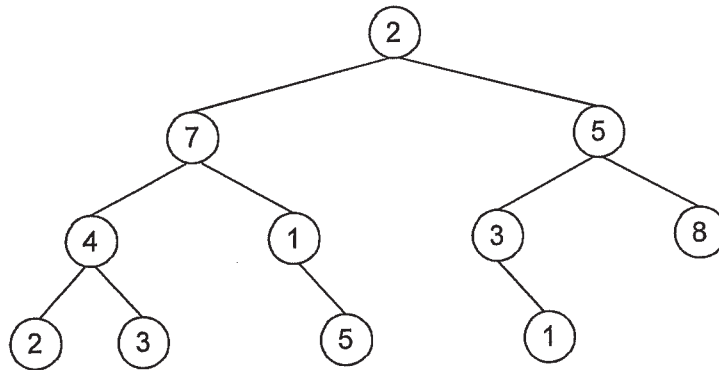
On définit un chemin racine-feuille dans un tel arbre comme une liste ordonnée de nœuds telle que

- le premier nœud est la racine ;
- chaque nœud suivant est enfant du précédent ;
- le dernier nœud est une feuille.

On appellera somme d'un chemin racine-feuille la somme des étiquettes des nœuds du chemin.

Enfin, la plus grande somme racine-feuille d'un arbre est la plus grande somme qu'il est possible d'obtenir en considérant tous les chemins racine-feuille de l'arbre.

1. Déterminer la plus grande somme racine-feuille de l'arbre représenté ci-dessous.



2. La classe Noeud ci-dessous implémente le type abstrait d'arbre binaire.

```
class Noeud:

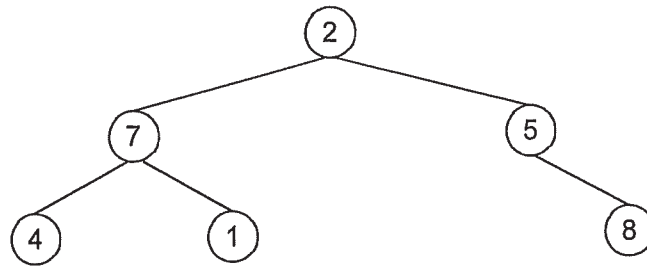
    def __init__(self, v):
        self.etiquette = v
        self.sag = None
        self.sad = None

    def niveau(self):
        if self.sag!=None and self.sad!=None:
            hg = self.sag.niveau()
            hd = self.sad.niveau()
            return 1+max(hg, hd)
        if self.sag!=None:
            return self.sag.niveau()+1
        if self.sad!=None:
            return self.sad.niveau()+1
        return 0

    def modifier_sag(self, nsag) :
        self.sag = nsag

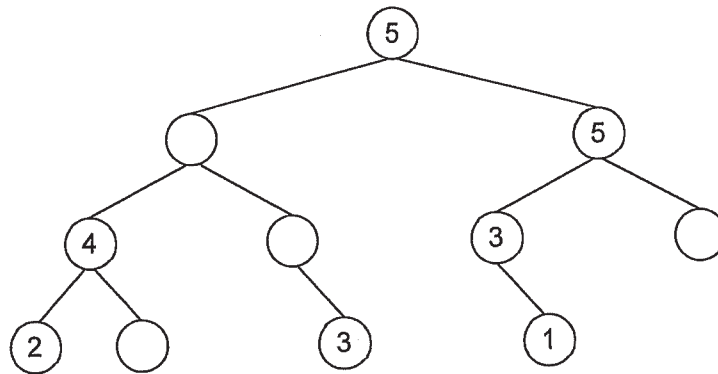
    def modifier_sad(self, nsad) :
        self.sad = nsad
```

- a. Écrire une suite d'instructions utilisant la classe `Noeud` permettant de représenter l'arbre ci-dessous.



- b. Que renvoie l'appel de la méthode `niveau` sur l'arbre ci-dessus ?
3. S'inspirer du code de la méthode `niveau` pour écrire une méthode récursive `pgde_somme` qui renvoie la plus grande somme racine-feuille d'un arbre.
4. On appelle arbre magique un arbre binaire dont toutes les sommes des chemins racine-feuille sont égales.

- a. Recopier et compléter l'arbre ci-dessous pour qu'il soit magique.



- b. Un arbre est magique si ses sous-arbres sont magiques et qu'ils ont de plus la même plus grande somme racine-feuille. Écrire une méthode récursive `est_magique` qui renvoie `True` si l'arbre est magique et `False` sinon.