

## EXERCICE 2 (3 points)

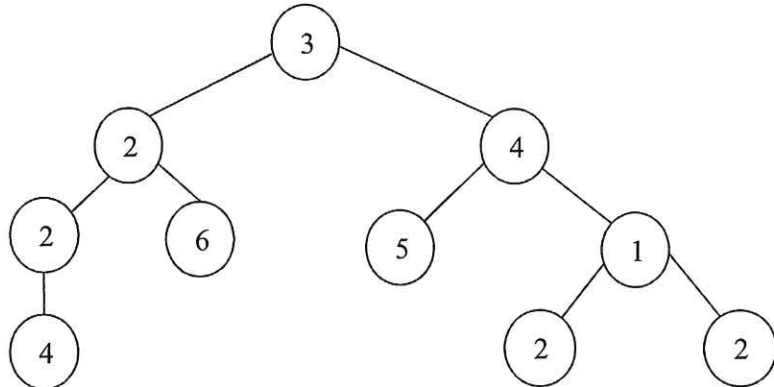
Tous les arbres de cet exercice auront des nœuds avec au plus deux enfants.

On considère des arbres dont les étiquettes sont des nombres entiers positifs.

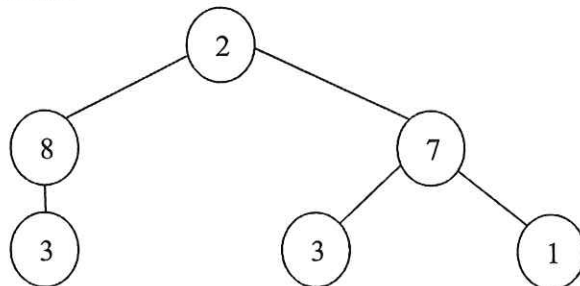
On appelle poids d'un tel arbre la somme de toutes ses étiquettes.

On dira qu'un arbre est un arbre mobile si pour chaque nœud, tous ses sous-arbres sont des arbres mobiles de même poids.

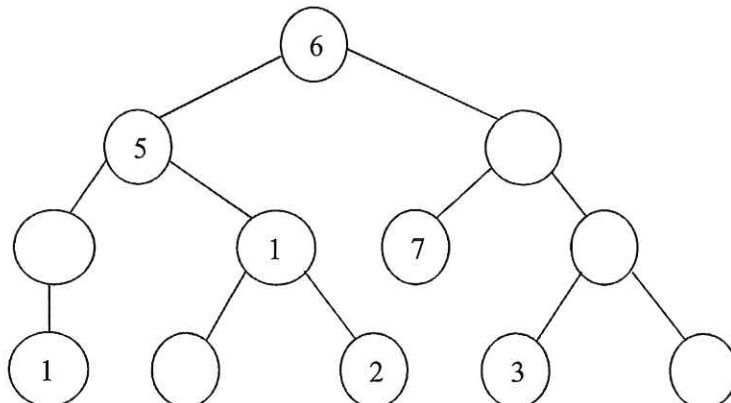
Voici un exemple d'arbre mobile.



Voici un exemple d'arbre non-mobile.



1. Recopier et compléter l'arbre ci-dessous de sorte qu'il soit un arbre mobile.



2. Chaque nœud d'un arbre sera représenté en Python par une liste `[e, lst_sa]` où :
- `e` est l'étiquette du nœud ;
  - `lst_sa` est la liste des sous-arbres de ce nœud.

L'arbre vide sera représenté par une liste vide.

Un arbre est ainsi représenté par une liste correspondant à son nœud racine.

On considère le script Python ci-dessous.

```
n2 = [2, []]
n5 = [5, []]
n8 = [8, []]
n1 = [1, [n2, n5]]
a = [4, [n1, n8]]
```

- a. Dessiner sur votre copie l'arbre représenté par la variable `a`.
- b. Cet arbre n'est pas un arbre mobile, mais on peut le transformer en arbre mobile en ajoutant exactement deux nœuds. Dessiner l'arbre mobile ainsi obtenu et écrire des instructions à ajouter au script précédent pour que la variable `a` représente cet arbre mobile.
3. Le poids d'un arbre peut se calculer de la manière suivante : on parcourt les nœuds de l'arbre en largeur à l'aide d'une file en maintenant à jour une variable `p` égale à la somme des étiquettes des nœuds parcourus.

On utilisera l'interface des files suivante :

<code>creer_file()</code>	Renvoie une file vide
<code>est_vide(f)</code>	Renvoie <code>True</code> si la file <code>f</code> est vide, <code>False</code> sinon
<code>enfiler(f, e)</code>	Ajoute l'élément <code>e</code> dans la file <code>f</code>
<code>defiler(f)</code>	Supprime et renvoie l'élément en tête de la file <code>f</code>

La fonction `poids(arbre)` a pour paramètre un arbre `arbre` et renvoie son poids. Recopier et compléter les lignes 7, 8 et 11 du code ci-dessous.

```
1. def poids(arbre):
2.     if arbre == [] :
3.         return 0
4.     p = 0
5.     file = creer_file()
6.     enfiler(file, arbre)
7.     while ... :
8.         arbre = ...
9.         for sous_arbre in arbre[1] :
10.            enfiler(file, sous_arbre)
11.            p = ...
12.     return p
```

4. Pour déterminer récursivement si un arbre est un arbre mobile, on doit :
- vérifier que tous ses sous-arbres ont le même poids ;
  - vérifier récursivement que tous ses sous-arbres sont des arbres mobiles.

Écrire une fonction récursive `est_mobile(arbre)` qui a pour paramètre un arbre `arbre` et renvoie `True` si cet arbre est un arbre mobile, et `False` sinon.

Pour cela, on pourra utiliser la fonction `poids`.

On rappelle que l'arbre vide est un arbre mobile.