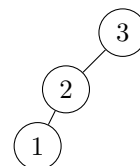
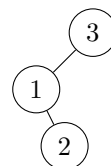
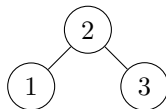
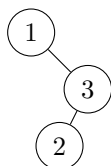
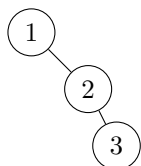

ABR - CORRIGÉ

Exercice 0

Exercice 1 Le plus petit élément d'un ABR est l'élément situé « le plus à gauche de l'arbre ». Pour le trouver, il suffit donc de parcourir tous les sous-arbres gauche jusqu'à trouver un arbre vide.

```

1 def minimum(a):
2     if a:
3         is
4     None: return elif a.gauche is None:
5         return a.valeur
6     else:
7         return minimum(a.gauche)
  
```

Exercice 2

```

1 def inserer_unique(x, a):
2     if a is None:
3         return Noeud(None, x, None)
4     elif x < a.valeur:
5         return Noeud(inserer(x, a.gauche), a.valeur, a.droit)
6     elif x > a.valeur:
7         return Noeud(a.gauche, a.valeur, inserer(x, a.droit))
8     else:
9         return a
  
```

Exercice 3

```

1 def occurrence(x, a):
2     if a is None:
3         return 0
4     elif x < a.valeur:
5         return occurrence(x, a.gauche)
6     elif x > a.valeur:
7         return occurrence(x, a.droit)
8     else:
9         return 1 + occurrence(x, a.gauche) + occurrence(x, a.droit)
  
```

Exercice 4 On définit d'abord une fonction récursive `ajouter(t, a)` qui remplit un tableau `t` avec les valeurs de l'arbre `a` rencontrées en le parcourant dans un ordre infixe. Une deuxième fonction `lister(a)` définit puis remplit un tableau en appelant la fonction `ajouter`, avant de la renvoyer.

```
1 def ajouter(t, a):
2     if a is None:
3         return
4     else:
5         ajouter(t, a.gauche)
6         t.append(a.valeur)
7         ajouter(t, a.droit)

8 def lister(a):
9     t = []
10    ajouter(t, a)
11    return t
```