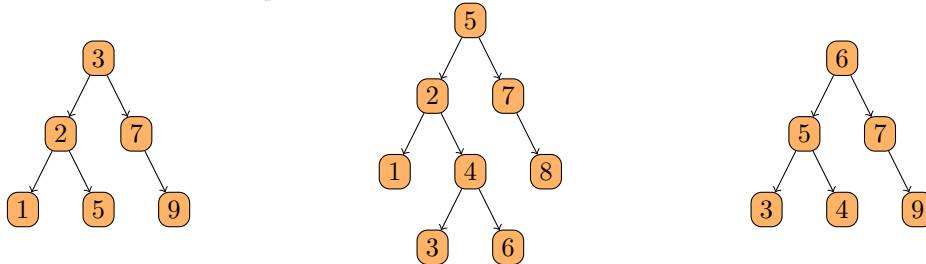


Exercice 1.

Parmi les arbres suivants, lesquels sont des arbres binaires de recherche ?

**Exercice 2.**

Donner tous les arbres binaires de recherche formés de trois noeuds et contenant les entiers 1, 2 et 3.

Exercice 3. a. Dans un arbre binaire de recherche, où se trouve le plus petit élément ? en déduire une fonction `minimum(arb)` qui retourne le plus petit élément de l'ABR `arb` et `None` si cet arbre est vide.

b. Ecrire une fonction `maximum(arb)` qui retourne le plus grand élément de l'ABR `arb` et `None` si cet arbre est vide.

Exercice 4.

Ecrire une variante de la fonction `ajout(arb, v)` qui n'ajoute pas `v` à `arb` si `v` est déjà présent dans `arb`.

Exercice 5.

Ecrire une fonction `est_ABR(arb)` qui retourne `True` si `arb` est un arbre binaire de recherche et `False` sinon.

Exercice 6. a. Ecrire une fonction `remplir(arb, tbl)` qui ajoute tous les éléments de l'ABR `arb` dans le tableau `tbl` dans l'ordre infixe. Utiliser la `tbl.append(x)` pour ajouter les éléments au tableau.

b. Ajouter alors une méthode `lister(self)` à la classe `ABR` qui renvoie un nouveau tableau contenant tous les éléments de l'ABR `self` dans l'ordre croissant.

c. Ecrire finalement une fonction `trier(tbl)` qui reçoit en argument un tableau d'entiers et renvoie un tableau trié contenant les mêmes éléments. Quelle est la complexité de ce tri ?