Contrôle 03

Exercice 3

On implémente une structure d'arbre binaire grâce à la classe **Node** du cours, dont voici un extrait :

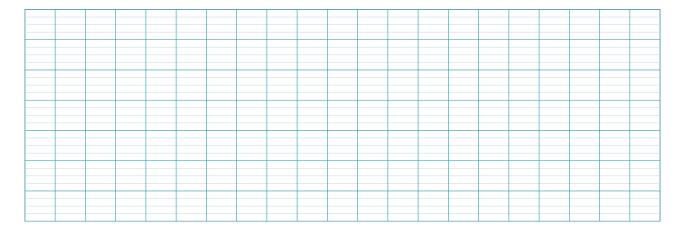
Python

```
class Node:
def __init__(self, v, left=None | int, right=None | int):
    self.value = v
    self.left = left # vaut None ou bien un entier
    self.right = right # vaut None ou bien un entier
```

La notation **None** | int signifie que le paramètre concerné peut être **None** ou une valeur de type int.

On aimerait savoir, étant donnée une instance de la classe **Node** nommée **root**, si l'arbre binaire de racine **root** est un ABR (arbre binaire de recherche).

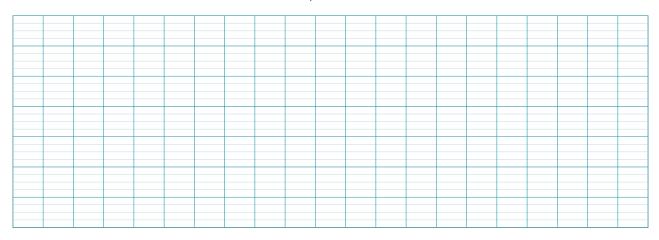
1. Rappeler quel parcours des nœuds d'un ABR permet d'obtenir les valeurs qu'ils contiennent dans l'ordre croissant et expliquer son fonctionnement.



Le parcours dont il est désormais question est celui de la question 1.

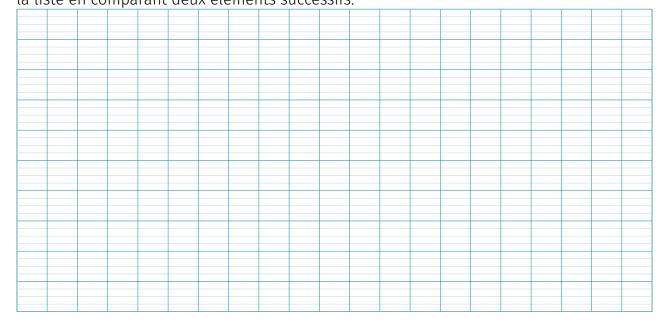
- 2. Écrire la fonction récursive parcours qui
- en entrée prend une instance **n** de la classe **Node**;

- renvoie la liste des valeurs obtenue en parcourant l'arbre dont **n** est la racine.



- 3. Écrire la fonction est_triee qui
- en entrée prend une liste d'entiers;
- renvoie **True** si celle-ci est triée dans l'ordre croissant et **False** sinon.

Il est interdit d'utiliser la méthode **sort** ou la fonction **sorted** : la fonction doit parcourir la liste en comparant deux éléments successifs.



En déduire le code d'une fonction **est_un_abr** qui

- en entrée prend une instance **n** de la classe **Node**;
- renvoie **True** si l'arbre de racine **n** est un ABR et **False** sinon.