

# Arbre binaire équilibré

## Arbre binaire équilibré

On appelle arbre binaire **équilibré** un arbre binaire tel que :

- la différence entre la hauteur de son sous-arbre à gauche et la hauteur de son sous-arbre à droite vaut -1, 0 ou 1,
- son sous-arbre à gauche est équilibré,
- son sous-arbre à droite est équilibré.

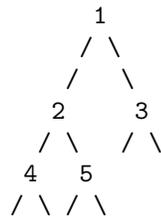
On précise, de plus, que **l'arbre binaire vide est équilibré**.

On convient que la hauteur d'un arbre binaire est le nombre de nœuds rencontrés lors du plus long parcours en profondeur issu de la racine. Ainsi l'arbre binaire vide a une hauteur de 0.

**Dans cet exercice**, on représente les arbres binaires ainsi :

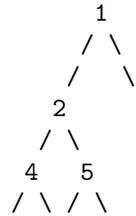
- l'arbre binaire vide est représenté par **None**,
- un arbre binaire non-vide est représenté par le tuple (<sous-arbre à gauche>, <valeur de la racine>, <sous-arbre à droite>).

Ainsi l'arbre binaire



est représenté par  $(((\text{None}, 4, \text{None}), 2, (\text{None}, 5, \text{None})), 1, (\text{None}, 3, \text{None}))$ . Il est équilibré.

D'autre part,



est représenté par `((None, 4, None), 2, (None, 5, None)), 1, None`. Il n'est pas équilibré.

## Objectif

On demande d'écrire deux fonctions :

- la fonction `hauteur` prend en paramètre un tel arbre et renvoie sa hauteur,
- la fonction `est_equilibre` qui prend en paramètre un tel arbre et renvoie `True` s'il est équilibré, `False` dans le cas contraire.

## Exemples

```
>>> arbre_1 = (None, 0, (None, 1, None))
>>> hauteur(arbre_1)
2
>>> est_equilibre(arbre_1)
True

>>> arbre_2 = ((None, 1, None), 0, (None, 2, None))
>>> hauteur(arbre_2)
2
>>> est_equilibre(arbre_2)
True

>>> arbre_3 = ((None, 1, None), 0, (None, 2, (None, 3, None)))
>>> hauteur(arbre_3)
3
>>> est_equilibre(arbre_3)
True

>>> arbre_4 = (((None, 4, None), 2, (None, 5, None)), 1, None)
>>> hauteur(arbre_4)
3
>>> est_equilibre(arbre_4)
False
```