

## Série N°03 – Les arbres

### Exercice 01 : Comparer deux arbres

L'objectif de cet exercice est d'écrire une fonction permettant de comparer deux arbres binaires. On considère que deux arbres sont égaux si leurs structures sont identiques et si les valeurs stockées dans les nœuds sont identiques. On considère que deux arbres sont différents si leurs structures sont différentes ou si les valeurs stockées dans les nœuds sont différentes.

1. Définir la structure de données permettant de représenter un arbre binaire d'entiers.
2. Écrire une fonction récursive permettant de comparer deux arbres binaires.
3. Écrire la version itérative de la fonction précédente.

### Exercice 02 : Arbre symétrique

Un arbre binaire symétrique est un type particulier d'arbre binaire où les sous-arbres gauche et droit de sa racine sont des miroirs l'un de l'autre par rapport. En d'autres termes, si vous imaginez l'arbre comme une structure géométrique, la partie gauche de la racine est la réflexion miroir de la partie droite.

Pour cet exercice, vous devez écrire une fonction qui détermine si un arbre binaire est symétrique ou non. La fonction retourne *1 (true)* si l'arbre est symétrique et *0 (false)* sinon.

### Exercice 03 : Somme d'un chemin

Écrire une fonction qui prend en paramètre un arbre binaire, et un entier, et qui teste s'il existe un chemin à partir de la racine jusqu'à une feuille de l'arbre dont la somme est égale à l'entier passé comme paramètre. La fonction doit retourner vrai si le chemin existe, et faux sinon.

### Exercice 04 : Conversion en arbre binaire de recherche équilibré

Un arbre binaire de recherche équilibré est un arbre binaire de recherche dont les sous-arbres gauche et droit ont une hauteur égale ou différant d'au plus 1 pour chaque nœud. L'objectif de cet exercice est de convertir un tableau trié en un arbre binaire de recherche équilibré. Écrire une fonction "convertAVL" qui prend en paramètre un tableau trié et retourne un arbre binaire de recherche équilibré construit à partir des éléments du tableau.

La fonction ne doit effectuer aucune rotation. Elle doit simplement construire l'arbre à partir du tableau en insérant les éléments dans la bonne position.

### Exercice 05 : Corriger un arbre binaire de recherche

Étant donné un arbre binaire de recherche, les valeurs de deux nœuds ont été échangées par erreur. L'objectif de cet exercice est de trouver et échanger les deux valeurs sans modifier sa structure de l'arbre ni de la recréer. Vous devez écrire une fonction qui prend en entrée la racine de l'arbre et réaliser la correction en échangeant les valeurs des nœuds incorrectement placés.

**Indication :** le parcours infixe d'un arbre binaire de recherche doit retourner les valeurs des nœuds dans un ordre croissant.