BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°44

DURÉE DE L'ÉPREUVE : 1 heure

Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4
Dès que le sujet vous est remis, assurez-vous qu'il est complet.

Le candidat doit traiter les 2 exercices.

EXERCICE 1 (10 points)

Écrire une fonction `enumere` qui prend en paramètre un tableau `tab` (type `list`) et renvoie un dictionnaire `d` dont les clés sont les éléments de `tab` avec pour valeur associée la liste des indices de l'élément dans le tableau `tab`.

Exemple :

```
>>> enumere([])
{}
>>> enumere([1, 2, 3])
{1: [0], 2: [1], 3: [2]}
>>> enumere([1, 1, 2, 3, 2, 1])
{1: [0, 1, 5], 2: [2, 4], 3: [3]}
```

EXERCICE 2 (10 points)

Un arbre binaire est soit vide, représenté en Python par la valeur None, soit un nœud, contenant une étiquette et deux sous-arbres gauche et droit et représenté par une instance de la classe Noeud donnée ci-dessous.

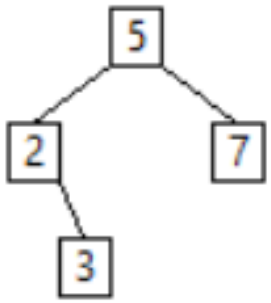
```
class Noeud:
    """Classe représentant un noeud d'un arbre binaire"""
    def __init__(self, etiquette, gauche, droit):
        """Crée un noeud de valeur etiquette avec
        gauche et droit comme fils."""
        self.etiquette = etiquette
        self.gauche = gauche
        self.droit = droit

def parcours(arbre, liste):
    """parcours récursivement l'arbre en ajoutant les étiquettes
    de ses noeuds à la liste passée en argument en ordre infixe."""
    if arbre != None:
        parcours(arbre.gauche, liste)
        liste.append(arbre.etiquette)
        parcours(arbre.droit, liste)
    return liste
```

La fonction récursive `parcours` renvoie la liste des étiquettes des nœuds de l'arbre implémenté par l'instance arbre dans l'ordre du parcours en profondeur infixe à partir d'une liste vide passée en argument.

Compléter le code de la fonction `insere`, présenté page suivante, qui prend en argument un arbre binaire de recherche `arbre` représenté ainsi et une étiquette `cle`, non présente dans l'arbre, et qui :

- renvoie une nouvelle feuille d'étiquette `cle` s'il est vide ;
- renvoie l'arbre après l'avoir modifié en insérant `cle` sinon ;
- garantit que l'arbre ainsi complété soit encore un arbre binaire de recherche.

Tester ensuite ce code en utilisant la fonction parcours et en insérant successivement des nœuds d'étiquette 1, 4, 6 et 8 dans l'arbre binaire de recherche représenté ci- dessous :



```
def insere(arbre, cle):
    """insere la cle dans l'arbre binaire de recherche
    représenté par arbre.
    Retourne l'arbre modifié."""
    if arbre == None:
        return Noeud(cle, None, None) # creation d'une feuille
    else:
        if ...:
            arbre.gauche = insere(arbre.gauche, cle)
        else:
            arbre.droit = ...
        return arbre
```