

NUMERIQUE et SCIENCES INFORMATIQUES

Épreuve de l'enseignement de spécialité

Sujet d'entraînement

Partie Pratique

Classe Terminale de la voie générale

Le candidat doit traiter les 2 exercices

Le sujet comporte 4 pages



Exercice 1 (4 points)

Une entreprise spécialisée dans la réparation d'objets veut valoriser son employé.e ayant réalisé le plus de réparation durant le mois.

L'ensemble des employé.e.s et du nombre de réparations effectuées durant le mois par chacun est stocké sous forme d'un dictionnaire où la clé est le nom de l'employé.e et la valeur associée le nombre de réparations effectuées.

Écrire une fonction `employe_e_du_mois` qui :

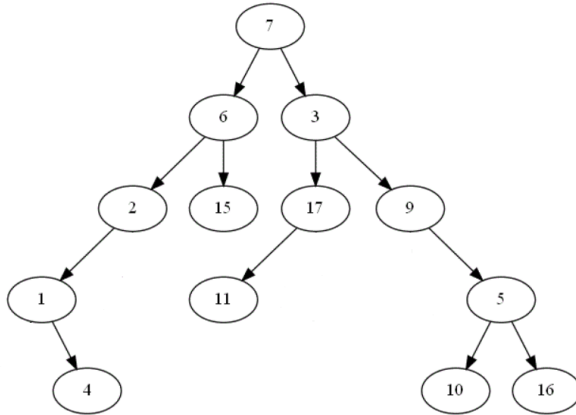
- Prend en paramètre un dictionnaire `employes` non vide dont les clés sont des chaînes de caractères et les valeurs associées sont des entiers ;
- Renvoie un tuple dont :
 - La première valeur est la clé du dictionnaire associée à la valeur maximale ; en cas d'égalité sur plusieurs clés, on choisira la plus petite suivant l'ordre alphabétique
 - La seconde valeur est la valeur maximale présente dans le dictionnaire.

Exemples :

```
>>> employe_e_du_mois({'Bosson': 183, 'Haquetive': 214, 'Valleauriz': 164, 'Tremolo': 62})
('Haquetive', 214)
>>> employe_e_du_mois({'Raipatou': 204, 'Maud-Elle' : 204 , 'Bosson': 199, 'Haquetive': 198,
                       'Valleauriz': 204 , 'Tremolo': 134})
('Maud-Elle', 204)
```

Exercice 2 (4 points)

Dans cet exercice, un arbre binaire de nombres entiers distincts deux à deux et strictement positifs est stocké sous la forme d'un dictionnaire où les clefs sont les nombres entiers des nœuds de l'arbre et les valeurs, pour chaque clef, la liste des nombres entiers des fils gauche et droit du nœud, 0 signifie l'absence d'un tel fils.



est stocké dans le dictionnaire :

```
arbre = {7:[6,3], 6:[2,15], 15:[0,0], 2:[1,0], 1:[0,4], 4:[0,0], 3:[17,9], 17:[11,0],  
9:[0,5], 5:[10,16],10:[0,0],16:[0,0], 11:[0,0]}
```

Le but est d'écrire une fonction récursive `nb_paire` prenant en paramètres un arbre binaire, `arbre` sous la forme d'un dictionnaire, et un nombre entier `racine` qui est la valeur du sommet de l'arbre, et qui renvoie le nombre de nœuds de valeur paire.

Compléter la fonction récursive `nb_pair` et proposer des tests pour vérifier le bon fonctionnement de cette fonction :

```
def nb_pair(arbre:dict,racine:int)->int:
    """
    arbre est un dictionnaire dont les clés sont des entiers et dont les valeurs associées sont une l.
    de deux nombres entiers.
    Fonction qui renvoie le nombre de noeuds ayant une valeur paire.
    """
    # préconditions
    assert ... , "l'objet entré comme premier argument dans la fonction doit être de type dictionnaire"
    assert ... , "l'objet entré comme second argument dans la fonction doit être un nombre entier"
    assert ... , "l'objet entré comme second argument dans la fonction doit être strictement positif"

    # cas d'un noeud sans fils droit ni gauche
    if ... :
        if racine %2 ==0:
            return 1
        return 0

    # cas d'un noeud avec fils droit mais sans fils gauche
    elif ... :
        if racine %2 ==0:
            return 1 + nb_pair(arbre,...)
        return ...

    # cas d'un noeud avec fils gauche mais sans fils droit
    elif ... :
        if racine %2 ==0:
            return 1 + nb_pair(arbre,...)
        return ...

    # cas d'un noeud avec deux fils
    else:
        if racine %2 ==0:
            return ...
        return ...

    # tests de bon fonctionnement à rajouter en dessous :
```