Corrigé sujet 27 - Année : 2022

Sujet 27 - 20222

1. Exercice 1

Script Python

```python
def taille(arbre,lettre):
    if arbre[lettre]==['','']:
        return 1
    elif arbre[lettre][0]=='':
        return 1+taille(arbre,arbre[lettre][1])
    elif arbre[lettre][1]=='':
        return 1+taille(arbre,arbre[lettre][0])
    else:
        return 1+taille(arbre,arbre[lettre][0])+taille(arbre,arbre[lettre][1])
```

Commentaires

1. La correction suit les indications du sujet en traitant les 4 cas. On peut faire autrement (et plus simplement).
2. Ce sujet est sans doute difficile car il utilise une représentation des arbres binaires inhabituelle, en plus de mélanger diverses notions du programme (récursivité, arbre, dictionnaire, listes)

2. Exercice 2

Script Python

```python
def tri_iteratif(tab):
    for k in range( len(tab)-1 , 0, -1): #(1)
        imax = 0
        for i in range(0 , k ): #(2)
            if tab[i] > tab[imax] :
                imax = i
        if tab[imax] > tab[k] :
            tab[k] , tab[imax] = tab[imax] , tab[k]
    return tab
```

1. L'indice du dernier élément d'un tableau `tab` est `len(tab)-1`. On parcourt ici dans l'ordre inverse (revoir l'instruction `range` si nécessaire)
2. Cette portion du programme est une recherche classique de maximum.

3. Ici on échange le maximum trouvé avec l'élément d'indice `k`

Attention

1. En dépit du nom `tri_itératif`, il fallait reconnaître ici l'algorithme du tri par sélection.
2. Les listes étant mutables, `tab` est modifiée par la fonction (tri en place), alors que le `return` finale peut laisser penser qu'on veut récupérer un "nouveau tableau".
3. Le test ligne 7 peut paraître surprenant mais comme on a cherché le maximum entre les indices `0` et `k-1`, on doit vérifier qu'il ne se trouve pas à l'indice `k` (dans ce cas l'échange n'est pas nécessaire). On aurait pu chercher entre `0` et `k` et éviter ce test.