

Exercice 2 :

```
def tri_iteratif(tab):
    for k in range(len(tab)-1, 0, -1): #on parcourt tous les éléments sauf le dernier de gauche à droite.
        imax = k-1 #on permute le plus grand élément trouvé avec l'avant dernier.
        for i in range(0, k-1):#on va de 0 à k-1 dans tab.
            if tab[i] > tab[imax] :#si on trouve un élément > à imax :
                imax = i #imax devient i.
        if tab[imax] > tab[k] : #si tab[imax] est > tab [k].|
            tab[k] , tab[imax] = tab[imax] , tab[k] #on permute les deux valeurs.
    return tab
```

EXERCICE 1 :

```
def taille(arbre, lettre):
    if arbre[lettre][0] == '' and arbre[lettre][1] == '':
        return 1
```

On vérifie si l'arbre ne possède pas de branche et possède juste un seul nœud donc juste la racine la taille est donc de 1.

```
elif arbre[lettre][0] == '' and arbre[lettre][1] != '':
    return 1 + taille(arbre, arbre[lettre][1])
```

On vérifie maintenant si l'arbre possède une branche non vide fils droit mais pas de fils gauche, si oui on renvoie 1 qui correspond à la racine + la taille de l'arbre du côté du fils droit.

```
elif arbre[lettre][0] != '' and arbre[lettre][1] == '':
    return 1 + taille(arbre, arbre[lettre][0])
```

A l'inverse : On vérifie maintenant si l'arbre possède une branche non vide fils gauche mais pas de fils droit, si oui on renvoie 1 qui correspond à la racine + la taille de l'arbre du côté du fils gauche.

```
elif arbre[lettre][0] != '' and arbre[lettre][1] != '':
    return 1 + taille(arbre, arbre[lettre][0]) + taille(arbre, arbre[lettre][1])
```

Pour finir on vérifie si l'arbre possède une branche non vide du côté du fils gauche et une branche non vide du côté du fils droit, si oui l'arbre possède les deux branches alors on renvoie 1 (la racine) + la taille de l'arbre du côté du fils gauche + la taille de l'arbre du côté du fils droit.

```
a = {'F':['B','G'], 'B':['A','D'], 'A':['',''], 'D':['C','E'], \
     'C':['',''], 'E':['',''], 'G':['','I'], 'I':['','H'], \
     'H':['','']} 
```

Puis stocké la valeur de a

Résultat :

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