

2022 sujet 27

Exercice 1

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

```
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])
```

Exercice 2

```
def tri_iteratif(tab):
    for k in range(len(tab) - 1, 0, -1):
        imax = k - 1 # On ignore le dernier pour le max
        for i in range(0, k - 1):
            if tab[i] > tab[imax] :
                imax = i
        if tab[imax] > tab[k] : # On compare ici au dernier
            tab[k], tab[imax] = tab[imax], tab[k]
    return tab

# Cela fonctionne aussi en prenant le dernier pour le max
# mais il y a une itération supplémentaire dans la boucle
def tri_iteratif2(tab):
    for k in range(len(tab) - 1, 0, -1):
        imax = k
        for i in range(0, k):
            if tab[i] > tab[imax] :
                imax = i
        # On pourrait alors remplacer le if par imax != k:
        if tab[imax] > tab[k] :
            tab[k], tab[imax] = tab[imax], tab[k]
    return tab
```