

Réponses aux questions :

Exercice 1

- a. La valeur d'indice  $n$  dans  $T$  sera la valeur la plus grande après le premier passage. L'algorithme de tri à bulles est prouvé car on a un variant de boucle ( $T$ , qui est la liste que l'on tri au fur et à mesure) et un invariant de boucle ( $tri==1$ , la condition qui doit être respecté tant que la liste n'est pas triée), donc le programme se termine et fait bien ce que l'on attend de lui.

Algorithme :

$Trie \leftarrow 1$

Compteur  $\leftarrow 0$

Tant que  $Trie=1$

$Trie \leftarrow 0$

Pour  $i$  allant de 0 à longueur tableau - 1 (0 à longueur tableau - Compteur)

Compteur  $\leftarrow$  Compteur + 1

Si  $T[i] > T[i+1]$

$Trie \leftarrow 1$

Echanger  $T[i]$  et  $T[i+1]$

Fin Si

Fin Pour

Fin Tant que

(le vert est pour l'exercice 3)

- b. Le nombre de comparaison d'élément de tableau est  $(n-1)^2$  pour une liste de longueur  $n$ , le nombre maximum d'échange est  $n(n-1)/2$ , le nombre minimum d'échange est lorsqu'à chaque comparaison on ne fait aucun échange c'est-à-dire qu'il est 0. Par exemple :  
Dans une liste de longueur 5, le tableau qui représente le maximum d'échange est :  
[5,4,3,2,1]  
Le nombre d'échange étant 10  
Celui qui représente le minimum d'échange est :  
[1,2,3,4,5]  
Le nombre d'échange étant 0

Exercice 3

- a. La propriété de la portion de tableau dont les indices sont inférieurs à cet indice particulier est que ce sont des valeurs non classées

Amélioration en vert dans l'exercice 1

- b. Oui on peut généraliser

Algorithme :

Trie  $\leftarrow$  1

Start  $\leftarrow$  0

Fin  $\leftarrow$  longueur tableau - 1

Tant que Trie=1

    Trie  $\leftarrow$  0

    Pour i allant de Start à Fin

        Si  $T[i] > T[i+1]$

            Trie  $\leftarrow$  1

            Echanger  $T[i]$  et  $T[i+1]$

        Fin Si

    Fin Pour

    Start  $\leftarrow$  Start + 1

    Renverser le tableau

    Pour i allant de Start à Fin

        Si  $T[i] < T[i+1]$

            Trie  $\leftarrow$  1

            Echanger  $T[i]$  et  $T[i+1]$

        Fin Si

    Fin Pour

    Fin  $\leftarrow$  Fin-1

    Renverser le tableau

Fin Tant que

## DM tri bulle

### Programmes

#### Programme exercice 1

```
def bulle(T):  
    """  
    Fonction avec une liste comme parametre et renvoi une liste triée (le tri à bulle)  
    """  
    trie=1  
    comparaisons=0      # Compteur de comparaisons  
    echanges=0          # Compteur d'echanges  
    while trie==1:      # La condition est tant qu'on fait au moins un echange dans la boucle  
        trie=0          # Si la variable reste à 0 apres la boucle, la liste est triee est on peut sortir de la boucle tant que  
        for i in range(len(T)-1):  
            comparaisons+=1  
            if T[i]>T[i+1]:  
                echanges+=1  
                trie=1  
                T[i],T[i+1]=T[i+1],T[i]  
    return T,echanges,comparaisons
```

#### Programme exercice 3

```
def shaker(T):  
    """  
    Fonction avec une liste comme parametre et renvoi une liste triée (le tri shaker)  
    """  
    trie=1  
    start,fin=0,len(T)-1  
    echange=0  
    while trie==1:  
        trie=0  
        for i in range(start,fin):  
            if T[i]>T[i+1]:  
                echange+=1  
                trie=1  
                T[i],T[i+1]=T[i+1],T[i]  
        start+=1  
        T=list(reversed(T))  
        for i in range(start,fin):  
            if T[i]<T[i+1]:  
                echange+=1  
                trie=1  
                T[i],T[i+1]=T[i+1],T[i]  
        fin-=1  
        T=list(reversed(T))  
    return T,echange
```

#### Programme liste valeurs au hasard (pour les résultats experimentaux)

```
n=int(input())  
a=[random.randint(0,n) for i in range (n)]
```

Les résultats expérimentaux :

Programme Exercice 1 : (nb d'échanges)

Valeurs	1 000	2 000	4000
Test 1	248 502	989 974	4 019 540
Test 2	248 593	1 002 410	4 001 840
Test 3	253 121	1 001 652	3 987 914
Moyenne	250 072	998 012	4 003 031

Programme Exercice 3 : (nb d'échange)

Valeurs	1000	2000	4000
Test 1	246 937	996 016	3 989 519
Test 2	248 712	995 853	3 989 398
Test 3	251 967	1 010 871	4 000 905
Moyenne	249 205	1 000 913	3 993 274

Les deux programmes ont donc des complexités moyennes plus ou moins identique en nombre d'échanges qui est à  $(n^2-n)/4$  .