

Correction

NSI - 2021 Métropole Jour 1 (21-NSIJ1ME2)

Exercice 5 - Itérative vs Réursive

1. Le couple (1, 3) est une inversion car $8 > 7$.
2. Le couple (2, 3) n'est pas une inversion car $3 < 7$.

Partie A : Méthode itérative

1. a.

Cas n°1 : fonction1([1, 5, 3, 7], 0) renvoie 0
Cas n°2 : fonction1([1, 5, 3, 7], 1) renvoie 1
Cas n°3 : fonction1([1, 5, 2, 6, 4], 1) renvoie 2

1. b. Cette fonction permet de déterminer combien de valeur, située après la $i^{\text{ième}}$ valeur sont inférieures à cette $i^{\text{ième}}$ valeur.

```
2. def nombre_inversions(tab):  
    """ Renvoie le nombre d'inversions. """  
    cpt = 0  
    for i in range(len(tab)):  
        cpt += fonction1(tab, i)  
    return cpt
```

3. Pour une liste de taille n , on voit qu'il faut parcourir presque toute la liste pour chaque élément de la liste, ce qui conduit à une complexité quadratique $O(n^2)$.

Partie B : Méthode itérative

1. Le tri fusion a une complexité linéarithmique $O(n \cdot \log(n))$.

Solution 1	Solution 2
<pre>def moitie_gauche(tab): milieu = len(tab) // 2 if len(tab)%2: milieu += 1 return tab[:milieu]</pre>	<pre>from math import ceil def moitie_gauche(tab): milieu = ceil(len(tab)/2) return tab[:milieu]</pre>

```
3. def nb_inversions_rec(tab, n = 0):  
    """ Renvoie le nombre d'inversions. """  
    if len(tab) <= 1: # Cas trivial  
        return 0  
    else: # Cas récursif  
        gauche = moitie_gauche(tab)  
        droite = moitie_droite(tab)  
        ng = nb_inversions_rec(gauche, n) # Nombre d'inversion dans la moitié gauche  
        nd = nb_inversions_rec(droite, n) # Nombre d'inversion dans la moitié droite  
        n = nb_inv_tab(sorted(gauche), sorted(droite)) # Nb d'inversion pr les 2 tableaux triés  
        return ng + nd + n
```