

Correction

NSI - 2021 Métropole Jour 1 (--)

Exercice 4 - Algorithme du tri fusion

1. a. L'ordre de grandeur du coût, en nombre de comparaisons, de l'algorithme de tri fusion pour une liste de longueur n est $n \cdot \log(n)$. Sa complexité est linéarithmique $O(n \cdot \log(n))$

1. b. L'algorithme du tri à bulles (ou tri par propagation) ou le tri par sélection ou le tri par insertion qui ont un ordre de grandeur du coût, en nombre de comparaisons de n^2 . Leur complexité est quadratique $O(n^2)$. Ils sont moins efficaces que le tri fusion qui utilise la méthode « diviser pour régner ».

2. [7, 4, 2, 1, 8, 5, 6, 3]
[7, 4, 2, 1]
[7, 4]
[2, 1]
[8, 5, 6, 3]
[8, 5]
[6, 3]

3.

Solution 1

```
def moitie_droite(L) :  
    tab = []  
    for i in range(len(L)//2, len(L)) :  
        tab.append(L[i])  
    return tab
```

Solution 2

```
def moitie_droite(L) :  
    return L[n//2:]
```

4.

```
def fusion(L1, L2):  
    L = []  
    n1 = len(L1)  
    n2 = len(L2)  
    i1 = 0  
    i2 = 0  
    while i1 < n1 or i2 < n2 :  
        if i1 >= n1:  
            L.append(L2[i2])  
            i2 = i2 + 1  
        elif i2 >= n2:  
            L.append(L1[i1])  
            i1 = i1 + 1  
        else:  
            e1 = L1[i1]  
            e2 = L2[i2]  
            if e1 > e2 :  
                L.append(e2)  
                i2 = i2 + 1  
            else :  
                L.append(e1)  
                i1 = i1 + 1  
    return L
```