

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

Exercice 1 - Algorithmes de tri

Partie A : Manipulation de listes

1. Le code Python affiche 8, étant le nombre d'éléments dans la liste puis il affiche la liste [8, 7, 18, 16, 12, 9, 17, 3].
2. Pour afficher les éléments de la liste de l'indice 2 à 4 de la liste notes, on peut utiliser les codes :

Solution 1

```
print(notes[2:5])
```

Solution 2

```
for i in range(2, 5):  
    print(notes[i])
```

Partie B : Tri par insertion

1.

```
while i >= 0 and liste[i] > element_a_inserer :  
    liste[i+1] = liste[i]
```

2. Après une itération de la boucle for, la liste est [7, 8, 18, 14, 12, 9, 17, 3].
3. Après trois itérations de la boucles for, la liste est [7, 8, 14, 18, 12, 9, 17, 3].

Partie C : Tri Fusion

1. C'est un algorithme récursif car il s'appelle lui-même à la troisième étape.
2. Étape 1 : Comparer les cartes visibles (la première de chaque tas).
Étape 2 : Placer la carte avec la valeur la plus petite dans la main.
Étape 3 : Recommencer jusqu'à l'épuisement des tas.
3.

```
tri_fusion(liste, i_debut, i_partage)  
tri_fusion(liste, i_partage+1, i_fin)  
fusionner(liste, i_debut, i_partage, i_fin)
```
4. La première ligne du code de la question 3 sert à importer depuis le module math la fonction floor. La fonction `floor()` renvoie la partie entière d'un nombre.

Partie D : Comparaison du tri par insertion et du tri fusion

1. C'est le tri fusion qui a été utilisé car à chaque étape on fusionne deux tas triés.
2. Le tri fusion a une complexité en $O(n \log_2 n)$ et le tri par insertion à une complexité en $O(n^2)$.
3. Le tri fusion divise la liste à trier en deux à chaque itération sur le principe « diviser pour régner ». Le tri par insertion utilise deux boucles imbriquées parcourant la liste à trier.