

Préparation à l'épreuve pratique NSI

15

Trier rapide d'une table, algorithme de quicksort

Version classique

Nom :

Note : / 20

Classe :

1 Spécification du programme à réaliser

1.1 Description générale

On considère un tableau $T []$ de n éléments numérotés de 1 à n . L'objectif consiste à trier la table.

<u>Entrées</u>	<u>Sorties</u>
Le tableau $T [1..n]$ de n éléments.	La table triée

1.2 Pseudo code de l'algorithme

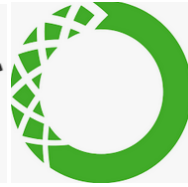
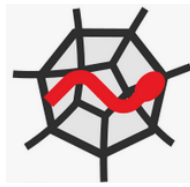
```

fonction tri_rapide(T, p, r)
si p < r alors
    q = partition(T, p, r)
    tri_rapide(T, p, q-1)
    tri_rapide(T, q+1, r)
fin si
retourner T
    
```

```

partition(T, p, r)
x = T[ r ]
i = p - 1
pour j = p jusqu'à r - 1 faire
    si T[ j ] <= x faire
        i = i + 1
        permuter T[ i ] avec T[ j ]
    fin si
fin pour
permuter T[ i + 1 ] avec T[ r ]
retourner i + 1
    
```

Appel de la fonction pour trier la table T : $\text{tri_rapide}(T, 1, n)$



La table T contient n éléments numérotés de 1 à n

2 Codage et mise en œuvre

2.1 Script de l'exercice

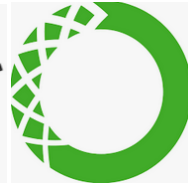
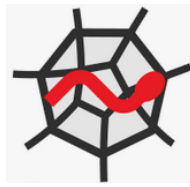


`INSI-PROG-015-Tri-rapide.py`

2.2 Fonction à compléter

Votre fonction à réaliser

```
def tri_rapide(liste, p, r):  
    """  
    Tri rapide de la liste  
    """  
  
    return liste  
  
def partition(liste,p,r):  
    """  
    Partitionnement de la liste en place  
    A la fin de l'algorithme la liste est  
    réordonnée de la manière suivante :  
    [éléments plus petits que liste[r],liste[r],éléments plus grands que liste[r]]  
    """  
  
    return i + 1
```



2.3 Résultats attendus

```
>>> (executing lines 1 to 111 of "NSI-PROG-015-Tri-rapide-corrige.py")
```

```
Liste de départ : [56, 10, -2, 52, -8, 41, -13]
```

```
Liste triée      : [-13, -8, -2, 10, 41, 52, 56]
```

```
Liste de départ : [250, -10, 8, 52, 84]
```

```
Liste triée      : [-10, 8, 52, 84, 250]
```

```
Liste de départ : [-37, 34, -32, 50, 71, 41, -64, 40, -100, -48, -42, -36]
```

```
Liste triée      : [-100, -64, -48, -42, -37, -36, -32, 34, 40, 41, 50, 71]
```

```
Liste de départ : []
```

```
Liste triée      : []
```

```
Liste de départ : [4]
```

```
Liste triée      : [4]
```

