

Partie 1

## Partie 1 : Etude de la fonction interclassement

### 1.1 Etudier le script de cette fonction

1. Quel est le type de chaque paramètre de la fonction ?
2. Quel est le type de chacune des variables internes ?
3. Tester cette fonction à partir de 2 listes passées en argument : [1,4,3,7] et [2,5,8,6]. Que retourne t-elle ?
4. Tester maintenant cette fonction à partir de [1,3,4,7] et [2,5,6,8]. Conclure.
5. Ecrire le prototype de cette fonction à partir des réponses précédentes. (params, variables, returns, exemples)

```
1 def interclassement(L1,L2):
2     lN = []
3     n1, n2 = len(L1),len(L2)
4     i1, i2 = 0,0
5     while i1<n1 and i2<n2:
6         if L1[i1] <= L2[i2]:
7             lN.append(L1[i1])
8             i1 += 1
9         else:
10            lN.append(L2[i2])
11            i2 += 1
12    return lN + L1[i1:] + L2[i2:]
```

Partie 2

## Partie 2 : Algorithmes de tri

### 2.1 On donne la description de 3 algorithmes

#### 2.1.1 Tri par selection

Sur un tableau de n éléments (numérotés de 0 à n-1), le principe du tri par sélection est le suivant :

- rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'indice 0 ;
- rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'indice 1 ;
- continuer de cette façon jusqu'à ce que le tableau soit entièrement trié (jusqu'au rang n-2).

#### 2.1.2 Tri insertion

Trier, c'est déplacer des éléments, et y insérer l'élément rangé, depuis le debut déjà trié de la liste, jusqu'à la fin :

- Hyp : l'élément non rangé est le j. Tous les autres éléments sont rangés jusqu'à j.
- Il faut d'abord conserver sa valeur à l'aide d'une variable temp
- On décale tous les éléments i, depuis le rang j jusqu'à l'élément dont la valeur est inférieure à celle de j (et donc de temp), en redescendant.

### 2.1.3 tri fusion

L'algorithme est naturellement décrit de façon récursive.

- Si le tableau n'a qu'un élément, il est déjà trié.
- Sinon, séparer le tableau en deux parties à peu près égales.
- Trier récursivement le sous-tableau de gauche avec ce même algorithme du tri
- Trier récursivement le sous-tableau de droite avec ce même algorithme du tri
- Fusionner les deux tableaux triés en un seul tableau trié.

## 2.2 Questions

1. A l'aide des scripts python suivants : Quelle fonction correspond à :

- tri par insertion : ...
- tri par selection : ...
- tri fusion : ...

2. La complexité de l'algorithme du tri fusion est  $O(N \times \log(N))$ . Pour les tris simples et par selection, c'est  $O(N^2)$ . Lequel de ces tris est le plus efficace ? Pourquoi ?

```

1 def tri1(L):
2     for j in range(len(L)):
3         temp = L[j]
4         i = j
5         while i>0 and L[i-1]>temp:
6             L[i]=L[i-1]
7             i-=1
8         L[i]=temp
9     return L
10
11 def select(T,debut) :
12     indiceDuMin=debut
13     for k in range(debut+1,len(T)) :
14         if T[k]< T[indiceDuMin] :
15             indiceDuMin=k
16     if indiceDuMin !=debut :
17         T[debut],T[indiceDuMin]=T[indiceDuMin],T[debut]
18
19
20 def tri2(T):
21     for j in range(0,len(T)-1) :
22         select(T,j)
23     return T
24
25 def tri3(L):
26     if len(L) <=1:
27         return L
28     m = len(L)//2
29     gauche = tri3(L[:m])
30     droite = tri3(L[m:])
31     return interclassement(gauche,droite)

```