

## TD – 01

## Exercices d'applications

**Savoirs et compétences :**

- Alg – C17 : tris d'un tableau à une dimension de valeurs numériques (tri par insertion, tri rapide, tri fusion).

**Exercice 1 – Représentation graphique**

Soit la liste de nombres suivante : [10, 3, 7, 5, 9, 7, 8, 0, 8].

**Question 1** Pour le tri par insertion, le tri rapide et le tri fusion, représenter graphiquement l'évolution de la liste de nombres.

**Question 2** Donner, pour chacun des cas, la complexité dans le meilleur des cas, le pire des cas et le cas moyen.

**Exercice 2 – Tri à bulles**

Pour trier une liste selon la méthode du tri à bulles, on réalise des balayages successifs : à chaque balayage, on compare les éléments du tableau 2 à 2 et on les réordonne.

**Question 1** En utilisant la liste donnée dans l'exercice 1, donne la séquence d'échanges permettant d'arriver à la liste triée.

**Question 2** Donner un algorithme naïf permettant de trier un algorithme selon la méthode du tri à bulles.

**Question 3** Dans quel cas est-on dans le meilleur des cas ? Quelle alors la complexité de l'algorithme ?

**Question 4** Dans quel cas est-on dans le pire des cas ? Quelle alors la complexité de l'algorithme ?

**Question 5** En remarquant qu'à l'étape  $i$ , les  $i$  derniers éléments sont triés, proposer un nouvel algorithme du tri à bulles.

**Question 6** En remarquant qu'à l'itération  $i$ , il est possible d'arrêter le tri là où la dernière inversion a eu lieu à

l'étape  $i - 1$ , proposer un nouvel algorithme du tri à bulles.

**Exercice 3 – Tri cocktail, tri shaker**

On remarque dans le tri à bulles qu'un balayage permet de faire remonter directement la bulle la plus grosse, cependant, les plus petites bulles ne descendent que d'un indice par balayage. Le tri cocktail propose donc de réaliser, à chaque itération, un balayage dans les deux sens.

**Question 1** En utilisant la liste donnée dans l'exercice 1, donne la séquence d'échanges permettant d'arriver à la liste triée.

**Question 2** Donner l'algorithme naïf permettant de trier une liste selon la méthode du tri cocktail.

**Question 3** Partant des remarques du tri à bulles, proposer une amélioration du tri cocktail.

**Question 4** Donner la complexité de l'algorithme.

**Exercice 4 – Tri par sélection**

Le tri par sélection consiste en rechercher le plus petit élément d'une liste, puis en le positionner en première place. Dans la liste non triée restante, on cherche le plus petit élément, puis on le positionne en première place etc.

**Question 1** En utilisant la liste donnée dans l'exercice 1, donne la séquence de listes permettant d'arriver à la liste triée.

**Question 2** Donner l'algorithme naïf permettant de trier une liste selon la méthode du tri par sélection.

## Tri insertion

```
def tri_bulles_naif(l):
    for i in range(0, len(l)-1):
        for j in range(0, len(l)-1):
            if l[j]>l[j+1]:
                l[j],l[j+1]=l[j+1],l[j]
```

```
def tri_bulles(l):
    for i in range(0, len(l)-1):
        for j in range(0, len(l)-i-1):
            if l[j]>l[j+1]:
                l[j],l[j+1]=l[j+1],l[j]
```

## Tri cocktail

```
fonction tri_cocktail (array liste)
    échangé := vrai
    Répéter tant que échangé = vrai
        échangé := faux

        Répéter pour tout i entre 0 et liste.taille - 2
            si liste[i] > liste[i + 1]
                [[Echanger (liste[i], liste[i+1])
                échangé := vrai
            fin si
        fin Répéter

        Répéter pour tout i (décroissant) entre liste.taille-2 et 0
            si liste[i] > liste[i + 1]
                [[Echanger (liste[i], liste[i+1])
                échangé := vrai
            fin si
        fin Répéter

    fin tant que
fin fonction
```

## Tri par sélection

```
def tri_selection(t) :
    n = len(t)
    for i in range(n) :
        k = i
        for j in range(i+1,n) :
            if t[k] > t[j] :
                k = j
        t[k],t[i] = t[i],t[k]
```

