TD3 : Algorithmes de tri

Corrigé

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 `[10, 3, 7, 5, 9, 7, 8, 0, 8]`, écrire la séquence d'échanges permettant d'arriver à la liste triée.

Première série de comparaison sur la liste faisant monter 10 : [10, 3, 7, 5, 9, 7, 8, 0, 8], [3, 10, 7, 5, 9, 7, 8, 0, 8], [3, 7, 10, 5, 9, 7, 8, 0, 8], [3, 7, 5, 10, 9, 7, 8, 0, 8], [3, 7, 5, 9, 10, 7, 8, 0, 8], [3, 7, 5, 9, 7, 10, 8, 0, 8], [3, 7, 5, 9, 7, 8, 10, 0, 8], [3, 7, 5, 9, 7, 8, 0, 10, 8], [3, 7, 5, 9, 7, 8, 0, 8, 10]
Deuxième série de comparaison à partir des deux premiers éléments : [3, 7, 5, 9, 7, 8, 0, 8, 10], [3, 5, 7, 9, 7, 8, 0, 8, 10], [3, 5, 7, 7, 9, 8, 0, 8, 10], [3, 5, 7, 7, 8, 9, 0, 8, 10], [3, 5, 7, 7, 8, 0, 9, 8, 10], [3, 5, 7, 7, 8, 0, 8, 9, 10]
Troisième série de comparaison à partir des deux premiers éléments : [3, 5, 7, 7, 8, 0, 8, 9, 10], [3, 5, 7, 7, 0, 8, 8, 9, 10]
Quatrième série de comparaison à partir des deux premiers éléments : [3, 5, 7, 7, 0, 8, 8, 9, 10], [3, 5, 7, 0, 7, 8, 8, 9, 10]
Cinquième série de comparaison à partir des deux premiers éléments : [3, 5, 7, 0, 7, 8, 8, 9, 10], [3, 5, 0, 7, 7, 8, 8, 9, 10]
Sixième série de comparaison à partir des deux premiers éléments : [3, 5, 0, 7, 7, 8, 8, 9, 10], [3, 0, 5, 7, 7, 8, 8, 9, 10]
Septième série de comparaison à partir des deux premiers éléments : [3, 0, 5, 7, 7, 8, 8, 9, 10], [0, 3, 5, 7, 7, 8, 8, 9, 10]
Huitième série de comparaison à partir des deux premiers éléments : [0, 3, 5, 7, 7, 8, 8, 9, 10]

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

```
def tri_bulles_naif(l):
    for i in range(len(l)):
        for j in range(len(l)-1): #on ne peut pas comparer le dernier
              lment   avec un suivant
            if l[j]>l[j+1]:
                l[j],l[j+1]=l[j+1],l[j]
```

Question 3. Dans quel cas est-on dans le meilleur des cas ? Quelle alors la complexité de l'algorithme ?
Quand la liste est triée, il n'y a pas d'échange d'élément. On a qu'en même les deux boucles `for` imbriquées.

Question 4. Dans quel cas est-on dans le pire des cas ? Quelle alors la complexité de l'algorithme ?
Quand la liste est triée dans le sens inverse, il y a échange de tous les éléments. On a qu'en même les deux boucles `for` imbriquées.

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

```
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]
```

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.

```
def tri_bulles_optimise(tableau):
    permutation = True
    passage = 0
```

```
    while permutation == True:
        permutation = False
        passage = passage + 1
        for en_cours in range(0, len(tableau) - passage):
            if tableau[en_cours] > tableau[en_cours + 1]:
                permutation = True
                tableau[en_cours], tableau[en_cours + 1] = tableau[en_cours
                    + 1],tableau[en_cours]
```

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 7. En utilisant la liste donnée dans le tri à bulles, donne la séquence d'échanges permettant d'arriver à la liste triée.

Question 8. Donner l'algorithme naïf permettant de trier une liste selon la méthode du tri cocktail.

Question 9. Partant des remarques du tri à bulles, proposer une amélioration du tri cocktail.

Question 10. Donner la complexité de l'algorithme.