SIO algorithmique : Différents types de tri

1. Tri par sélection

Principe: on parcourt la liste pour chercher le plus petit élément, et on le permute avec le premier élément de la liste, et on recommence l'opération avec les éléments suivants de la liste

Exemple :

liste initiale	5	7	1	2	4	3
étape 1	1	7	5	2	4	3
étape 2	1	2	5	7	4	3
étape 3	1	2	3	7	4	5
étape 4	1	2	3	4	7	5
liste triée	1	2	3	4	5	7

Exercice 1:

1. Créer une fonction echange(T,i,j) qui échange dans la liste T les éléments de rang i et j, et qui renvoie la liste T.
2. On considère l'algorithme suivant:

```
fonction tri_selection (T)
Variables:
T=liste des éléments à trier
n=longueur de la liste T
pour i variant de 0 à n-2
    indice_min=i      # on initialise l'indice du minimum
    pour j variant de i+1 à n-1
        si T[j]<T[indice_min]
            indice_min=j
        finsi
    fin pour
    echange(T,i,indice_min)
fin pour
retourner T
```

Dérouler "à la main" cet algorihme pour T=[7,5,1,2,4,3].

3. Ecrire cet algorithme avec Python et le tester avec T=[7,5,1,2,4,3] puis avec T=[3,7,1,2,2,3]

2. Tri par la méthode de la bulle

Principe: on propage par permutations successives le plus grand élément à la fin de la liste, et on recommence l'opération avec les autres éléments de la liste

Exemple :

liste initiale	5	7	1	2	4	3
étape 1	5	1	2	4	3	7
étape 2	1	2	4	3	5	7
étape 3	1	2	3	4	5	7
liste triée	1	2	3	4	5	7

Exercice 2:

1. Reprendre votre fonction echange(T,i,j) de l'exercice 1.

2. Créer une fonction tri_bulle(T) qui trie la liste T en utilisant le tri à bulle.

3. Tester avec T=[7,5,1,2,4,3] puis avec T=[3,7,1,2,2,3]. On fera afficher les étapes intermédiaires pour vérifier qu'on utilise bien le tri à bulle.

3. Tri par comptage

Principe: on crée une liste de compteurs dans lequel on met pour chaque élément, le nombre d'éléments de la liste à trier qui lui sont inférieurs et on utilise la liste de compteurs pour le tri

Le premier élément de la liste triée est celui dont le compteur est nul, on permute aussi les éléments correspondants dans la liste des compteurs

Exemple :

liste initiale	5	7	1	2	4	3
étape 1	[1]	7	[5]	2	4	3
étape 2	1	[2]	5	[7]	4	3
étape 3	1	2	[3]	7	4	[5]
étape 4	1	2	3	4	7	5
liste triée	1	2	3	4	5	7

liste compteurs	4	5	0	1	3	2
étape 1	[0]	5	[4]	1	3	2
étape 2	0	[1]	4	[5]	3	2
étape 3	0	1	[2]	5	3	[4]
étape 5	0	1	2	3	5	4
liste compteurs triée	0	1	2	3	4	5

Exercice 3:

1. Reprendre votre fonction echange(T,i,j) de l'exercice 1.

2; Crere une fonction compteurs(T) qui prend en paramètre la liste à trier et renvoie la liste des compteurs Tc .

2. Créer une fonction tri_comptage(T) qui trie la liste T en utilisant le tri par comptage.

3. Tester avec T=[7,5,1,2,4,3] puis avec T=[3,7,1,2,2,3]. On fera afficher les étapes intermédiaires pour vérifier qu'on utilise bien le tri par comptage.

4. Tri par transposition

Principe: deux éléments consécutifs sont comparés et permutés, si nécessaire, et un retour en arrière permet de vérifier si l'ordre initial a été modifié ou non

Exemple :

liste initiale	5	7	1	2	4	3	
étape 1	5	[1]	[7]	2	4	3	transposition
étape 2	1	[5]	[7]	2	4	3	**retour en arrière**
étape 3	1	5	[2]	[7]	4	3	transposition
étape 4	1	[2]	[5]	7	4	3	**retour en arrière**
étape 5	1	2	5	[4]	[7]	3	transposition
étape 6	1	2	[4]	[5]	7	3	**retour en arrière**
étape 7	1	2	4	5	[3]	[7]	transposition
étape 8	1	2	4	[3]	[5]	7	**retour en arrière**
liste triée	1	2	[3]	[4]	5	7	**retour en arrière**

1. Reprendre votre fonction echange(T,i,j) de l'exercice 1.

2. Créer une fonction tri_transpo(T) qui trie la liste T en utilisant le tri par transposition

3. Tester avec T=[7,5,1,2,4,3] puis avec T=[3,7,1,2,2,3]. On fera afficher les étapes intermédiaires pour vérifier qu'on utilise bien le tri par transposition.