

Dans ce TD, “trié” signifie “trié par ordre croissant”.

*** Les exercices marqués d’une étoile sont à faire à la maison.**

Exercice 1. *Tri sélection.*

Exécutez à la main le tri par sélection vu en cours sur le tableau suivant :

0	3	1	2
---	---	---	---

Combien de comparaisons avez-vous dû faire ?

Exercice 2. *Tri sur tri.*

Dans cet exercice on utilise

- la fonction `triSelection` du cours, qui trie les tableaux dont les éléments sont comparés par l’opérateur `<`, et
- la fonction `inf` du TD 1, qui prend en argument deux mots et qui retourne -1 si le premier mot est avant le second dans l’ordre lexicographique, 0 si les deux mots sont les mêmes, et +1 sinon.

Écrire une fonction qui prend en argument un tableau de tableaux de caractères `T`, et qui :

- trie par sélection chacun des éléments de `T`, pour l’ordre alphabétique (pour comparer deux caractères on utilise l’opérateur `<`).
- trie ensuite `T` par sélection, pour l’ordre lexicographique.

Par exemple le tableau

c	a	r	a	r	b	r	e	c	e	c	i
---	---	---	---	---	---	---	---	---	---	---	---

est transformé en

a	b	e	r	r	a	c	r	c	c	e	i
---	---	---	---	---	---	---	---	---	---	---	---

Exercice 3. *Tri bourrin.*

Algorithm 1 Tri bourrin

Entrée : tableau `T`

```

1: fonction TRIBOURRIN( $T$ )
2:    $n \leftarrow$  longueur de  $T$ 
3:   pour  $i \leftarrow 0$  à  $n - 2$  faire
4:     pour  $j \leftarrow i + 1$  à  $n - 1$  faire
5:       si  $T[i] > T[j]$  alors
6:         échanger  $T[i]$  et  $T[j]$ 
```

1. Cet algorithme vous semble-t-il correct pour trier un tableau de taille n ? À quel algorithme du cours pourriez vous le comparer ?
2. Que dire de son nombre de comparaisons ? son nombre d’affectations ?

Algorithm 2 Tri du drapeau hollandais

Entrée : tableau T contenant uniquement les valeurs 0, 1 et 2

```
1: fonction TRIDRAPEAU( $T$ )
2:    $p \leftarrow 0$ 
3:    $m \leftarrow 0$ 
4:    $g \leftarrow \text{longueur}(T) - 1$ 
5:   tant que  $m \leq g$  faire
6:     switch  $T[m]$  faire
7:       case 0
8:         échanger  $T[m]$  et  $T[p]$ 
9:          $m \leftarrow m + 1$ 
10:         $p \leftarrow p + 1$ 
11:       case 1
12:          $m \leftarrow m + 1$ 
13:       case 2
14:         échanger  $T[m]$  et  $T[g]$ 
15:          $g \leftarrow g - 1$ 
```

Exercice 4. *Problème du drapeau hollandais.*

L'algorithme ci-dessus permet de trier un tableau contenant uniquement trois valeurs différentes. Ici, on suppose que ces valeurs soient les entiers 0, 1 et 2.

- 1*. Implementez TRIDRAPEAU en Java.
2. Soit $T = \begin{bmatrix} 1 & 1 & 0 & 2 & 0 & 2 \end{bmatrix}$; évaluez l'appel TRIDRAPEAU T à la main.
3. Qu'est-ce qui garantit que tout appel de TRIDRAPEAU termine ?
4. Montrer que pendant l'exécution de TRIDRAPEAU, les propriétés suivantes sont satisfaites :
 - (a) $p \leq m$
 - (b) les éléments d'indice inférieur à p sont des 0.
 - (c) les éléments d'indice supérieur à g sont des 2.
 - (d) les éléments d'indice de p à $m - 1$ sont des 1.
5. Conclure qu'à la fin de l'exécution de l'algorithme, le tableau est trié.
6. Combien d'échanges sont effectués pendant l'exécution de TRIDRAPEAU, dans le pire des cas, la taille de T étant n ? Et dans le meilleur de cas ?
7. Même question que ci-dessus, pour les comparaisons.

Exercice 5. *Drapeau polonais**.

On veut adapter l'algorithme du tri drapeau au cas simple où le tableau à trier ne contient que deux valeurs différentes (par exemple 0 et 1). Ecrivez la fonction `triDrapeauBicolore`, en vous inspirant du code de `triDrapeau`.