
TD n° 2 - Un peu de tri

Exercice 1.*Échauffement*

1. Écrire une fonction `echange(tab, i, j)` qui permute les valeurs placées dans les cases i et j du tableau `tab`.
2. Écrire une fonction `verifie(tab)` qui vérifie si un tableau est déjà trié.
3. Écrire une fonction `miroir(tab)` qui renvoie l'inverse du tableau `tab` (les premiers éléments à la fin).

Exercice 2.*Tri bulle*

On veut trier un tableau d'entiers.

Le principe du *tri bulle* est de parcourir le tableau du début à la fin en déplaçant la plus grande valeur rencontrée au fur et à mesure que l'on avance. Les premières étapes du tri bulle sont illustrées sur l'exemple suivant (les cases entre crochets sont les cases qui sont comparées, elles sont permutées quand la plus grande valeur est à gauche) :

```
[2 4]3 5 0 1
2 [4 3]5 0 1
2 3 [4 5]0 1
2 3 4 [5 0]1
2 3 4 0 [5 1]
2 3 4 0 1 5
```

Lorsque l'on arrive au bout du tableau, on recommence depuis le début...

1. Réalisez à la main le deuxième parcours du tableau.
2. Combien de parcours du tableau doit-on effectuer pour être sûrs que le tableau est trié ? (prouvez que le tableau est bien trié à la fin)
3. Écrivez la fonction `tri_bulle(tab)` qui trie un tableau par la méthode du tri bulle.
4. Sur un tableau de n cases, combien faut-il effectuer de comparaisons pour trier le tableau ?

Exercice 3.*Tri rapide*

On s'intéresse maintenant à une méthode de tri plus efficace.

Le principe de l'algorithme *quick sort* consiste à choisir une valeur v (appelée *pivot*) dans le tableau `tab` puis à placer les éléments restants dans deux nouveaux tableaux, l'un contenant les valeurs plus petites que le pivot, l'autre les valeurs plus grandes. On trie alors récursivement les deux nouveaux tableaux obtenus (en appliquant la même méthode) et l'on recompose le tableau complet trié (en mettant les éléments plus petits que le pivot, suivis du pivot, suivi des éléments plus grands).

1. Exécuter à la main le fonctionnement de l'algorithme sur le tableau `[3, 0, 11, 2, 9, 2, 8]` en prenant à chaque fois la première case du tableau considéré comme pivot.
2. Écrire une fonction `pivot(tab)` qui renvoie un triplet `(petits, p, grands)` où p est la première valeur du tableau, utilisée comme pivot, et `petits` et `grands` sont deux tableaux contenant les éléments respectivement plus petits et plus grands que le pivot.
3. Écrire une fonction récursive `quick_sort(tab)` qui trie un tableau en appliquant l'algorithme *quick sort*.