
TD n° 3 - Du tri récursif

Exercice 1.

Tri fusion

La méthode de tri dite *tri fusion* est une méthode récursive permettant de trier rapidement un tableau. L'idée est de séparer le tableau en deux, de trier séparément chacune des deux moitiés (récursivement, à l'aide la même méthode de tri fusion) puis de « fusionner » les deux demi-tableaux triés en un seul grand tableau trié.

1. Exécutez à la main le fonctionnement de l'algorithme sur le tableau $[3, 5, 1, 7, 6, 2, 4, 0]$.
2. Comment coupe-t-on un tableau en deux moitiés de même taille (à 1 près) en *python* ?
3. Écrivez une fonction `fusion(tab1, tab2)` qui prend en argument deux tableaux **que l'on suppose déjà triés** et qui renvoie un tableau contenant les éléments de `tab1` et `tab2` dans l'ordre (on pourra écrire une fonction récursive ici aussi...).
4. En utilisant la fonction `fusion` précédente, écrivez une fonction `tri_fusion(tab)` qui trie récursivement un tableau de la manière suivante :
 - si le tableau est de taille 1 ou 0, on le renvoie directement (il est forcément déjà trié) ;
 - sinon, on coupe le tableau en deux moitiés, on trie chacune des moitiés à l'aide de la fonction `tri_fusion` (d'où le caractère *récursif* de la fonction) et on fusionne ces deux moitiés triées en utilisant la fonction `fusion`.

Exercice 2.

De l'autre côté du miroir

1. Écrivez les fonctions suivantes de manière itérative puis récursive¹ :
 - `append(a, l)` qui ajoute l'élément `a` à la fin de la liste `l` ;
 - `concat(l1, l2)` qui concatène les listes `l1` et `l2` (les éléments de `l1` suivis des éléments de `l2` dans une même liste) ;
 - `reverse(l)` qui renvoie la liste `l` retournée (premiers éléments à la fin).

On se propose d'améliorer la version récursive de `reverse(l)` en définissant une fonction `reverse_concat(l1, l2)` qui renvoie la concaténation de `reverse(l1)` et `l2`.

2. Écrivez la fonction `reverse_concat(l1, l2)` de manière naïve.
3. Ré-écrivez directement la fonction `reverse_concat(l1, l2)` de manière récursive (sans utiliser les fonctions `concat(l1, l2)` et `reverse(l)`).
4. En utilisant la fonction `reverse_concat(l1, l2)`, écrivez la fonction `reverse(l)` (en deux lignes).

1. En itératif, vous ne pouvez utiliser que `longueur(l)`, `element(i, l)`, `insertion(i, a, l)` et `suppression(i, l)`. En récursif, vous ne pouvez utiliser que `vide(l)`, `cons(a, l)`, `tete(l)` et `queue(l)`.