

Tri fusion

Ce TP se traite en OCaml.

A Rappels

- En OCaml, la paire composée de `x` et `y` se note `x,y`. Réciproquement, si `c` est une paire alors `let x,y = c` permet d'accéder aux coordonnées de la paire (on peut aussi faire cela dans un `let ... in`).
- Dans un filtrage de liste, `[]` est la liste vide, et `t::q` est le motif d'une liste composée d'une tête suivie d'une queue. Mais on peut également filtrer directement cette queue en la remplaçant par des motifs! Ainsi, `t::[]` est un motif valide pour une liste (liste à un seul élément), et `t0::t1::q` aussi (liste à au moins deux éléments).
- Dans utop, taper `#use "file.ml";;` permet de charger tout le fichier `file.ml` dans la console (comme si on l'avait copié/collé).
- Dans utop, vérifiez que vos fonctions ont bien le type auquel vous vous attendez. Sinon, il y a un probable bug...
- Les messages du compilateur sont pertinents.

B Tri fusion

B.1 Préambule

Une fonction `random_list : int -> int -> int` est fournie pour vous aider à tester. Sa documentation est écrite dans le code source.

0. Créez une fonction `is_sorted : 'a list -> bool` qui teste si une liste est triée (elle s'évalue à `true` si la liste est triée), en temps et espace linéaire en la longueur de la liste.
1. Testez! En particulier, `is_sorted [3; 4; 0; 1]` doit valoir `false`.

B.2 Tri fusion

Le **tri fusion** est un algorithme de tri qui procède ainsi pour trier une liste :

- Séparer arbitrairement la liste en deux moitiés à peu près égales (peu importe comment!).
 - Trier récursivement les deux moitiés.
 - Fusionner les deux moitiés triées.
2. Proposer un ou des cas de base pour cet algorithme.
 3. Écrire une fonction `divide : 'a list -> 'a list * 'a list` qui prend en entrée une liste et la divise en une paire de listes ayant (à peu près) le même nombre d'éléments.
 4. Testez!!
 5. Écrire une fonction `merge : 'a list -> 'a list -> 'a list` qui prend en entrée deux listes triées et les fusionne en une seule liste triée.
 6. Testez!!
 7. En déduire une fonction `mergesort : 'a list -> 'a list` qui réalise un tri fusion.
 8. Testez!!
 9. Quelle est la complexité de `mergesort`?