

Devoir Programmation impérative - Diviser pour régner

MPSI Option Informatique

1 Maximum local

Dans cet exercice, on suppose pouvoir comparer les éléments d'un tableau avec $<$.

1. Écrire une fonction `maximum` renvoyant le maximum d'un tableau (array).

On veut maintenant déterminer un maximum local dans un tableau, c'est à dire un élément qui soit supérieur à l'élément suivant (s'il existe) et l'élément précédent (s'il existe). Par exemple, `[12; 3; 5; 1; 6; 4; 1]` possède des maximums locaux aux indices 2 et 4 (correspondant aux éléments 5 et 6).

2. Écrire une fonction `max_local` renvoyant le premier indice d'un maximum local d'un tableau, en le parcourant de gauche à droite. Quelle est sa complexité?

On veut maintenant s'inspirer de la recherche par dichotomie pour écrire une fonction plus efficace. Pour cela, on regarde le milieu `t.(m)` de `t`:

3. Montrer que si `t.(m+1) > t.(m)` alors on peut chercher un maximum local dans la partie droite de `t` (c'est à dire parmi les éléments d'indices supérieurs ou égaux à `m+1`).
4. En déduire une fonction `max_local2` renvoyant le maximum local d'un tableau en utilisant la remarque précédente. Quelle est sa complexité? Comparer avec la question 2.

2 Fusion de k listes

1. Écrire une fonction `fusion` ayant deux listes triées en argument et renvoyant une liste triée contenant tous les éléments des deux listes.
Par exemple, `fusion [1; 4; 8] [2; 3; 6; 9]` doit renvoyer `[1; 2; 3; 4; 6; 8; 9]`.
2. En utilisant la fonction précédente, écrire une fonction `kfusion` telle que, si `ll` est une liste de listes triées, `kfusion ll` renvoie une liste triée contenant tous les éléments des listes de `ll`.
Par exemple, `kfusion [[1; 4; 8]; [2; 3; 6; 9]; [0; 11]]` doit renvoyer `[0;1; 2; 3; 4; 6; 8; 9; 11]`.
De plus, la complexité doit être $O(n \log(k))$, où k est le nombre de listes de `ll` et n le nombre total d'éléments (c'est à dire la taille de la liste renvoyée par `kfusion ll`).