

TP Informatique 27

Exercice 1

On rappelle le principe de l'algorithme du *tri rapide* sur une liste de nombres.

On choisit un élément de la liste (par exemple le premier) qu'on appelle pivot.

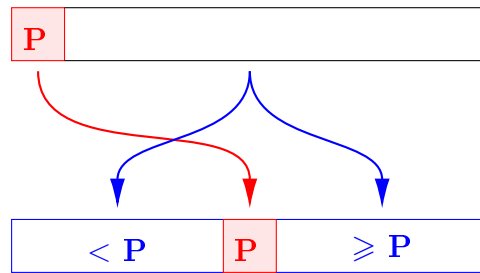


FIGURE 1 – Placement vis-à-vis du pivot

On place les éléments vis-à-vis du pivot puis on effectue récursivement le tri de la liste des éléments à droite du pivot puis à gauche du pivot (pour chaque sous-liste, choix du pivot puis placement à droite et à gauche, etc. ...)

Écrire une fonction `tri_rapide(T)` qui soit une implémentation de l'algorithme du tri rapide sur la liste de nombres `T`. On n'impose pas que cette implémentation soit *en place*.

Exercice 2

Dans cet exercice, on se propose d'effectuer une implémentation en place de l'algorithme du tri rapide.

1. Écrire une fonction `echange(T, i, j)` d'arguments `T` une liste de nombres, `i` et `j` des entiers qui réalise l'échange des éléments `T[i]` et `T[j]` dans la variable `T`. La fonction modifie directement la liste `T`.
2. Écrire une fonction `partition(T, g, d)` d'arguments `T` une liste de nombres, `g` et `d` des entiers qui choisit un élément de `T[g:d]` comme *pivot* (par exemple `T[g]`), modifie directement `T` en plaçant les éléments vis-à-vis du pivot : ceux inférieurs au pivot à gauche du pivot, ceux supérieurs au pivot à droite du pivot et renvoie la position finale du pivot.
3. Écrire une fonction `tri_rapide_rec(T, g, d)` qui réalise récursivement le tri rapide de la sous liste `T[g:d]` en modifiant directement la liste `T`.
4. Écrire une fonction `tri_rapide(T)` qui effectue le tri rapide en place de la liste `T`.

Exercice 3

Le *tri par bulles* consiste à faire remonter, par échanges successifs, les éléments les plus grands dans le haut de la liste comme des bulles d'air dans l'eau. Ainsi, pour une liste T, on parcourt la liste en comparant deux éléments consécutifs et en procédant éventuellement à un échange de positions pour que le plus grand élément remonte. Après le premier parcours, le plus grand élément est en fin de liste puis on recommence ce procédé sur la liste privée du dernier élément, etc. ...

3	7	2	6	5	1	4
3	2	6	5	1	4	7
2	3	5	1	4	6	7
2	3	1	4	5	6	7
2	1	3	4	5	6	7
1	2	3	4	5	6	7
1	2	3	4	5	6	7

FIGURE 2 – Réalisation du tri par bulles

Écrire une fonction `tri_bulles(T)` d'argument T une liste non vide de nombres et qui modifie directement cette liste en la triant selon l'algorithme du tri par bulles puis tester la fonction.