

## TP Informatique 03

### Exercice 1

Le *tri par insertion* consiste à itérer le procédé suivant : si les  $k$  premiers éléments sont triés, on vient insérer le  $k + 1$ -ème à sa place avec les  $k$  premiers.

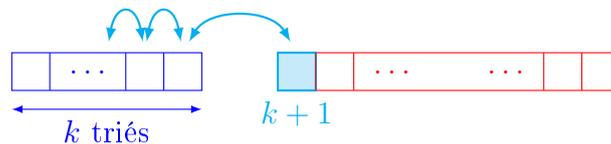


FIGURE 1 – Tri par insertion

Ainsi, pour une liste  $T$ , on examine les deux premiers éléments et on les échange éventuellement pour qu'ils soient ordonnés, puis on considère le troisième élément et on vient, par échanges successifs, l'insérer vis-à-vis des deux premiers déjà triés, etc. ...

1. Écrire une fonction `tri_ins(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 insertion.
2. L'implémentation de ce tri par sélection est-elle en place? Déterminer sa complexité temporelle.
3. Écrire une version améliorée du tri par insertion qui évite les échanges en privilégiant des « remontées » de certaines valeurs dans la liste au cours du processus.

### Exercice 2

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.

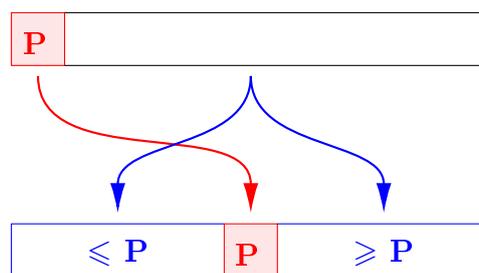


FIGURE 2 – Placement vis-à-vis du pivot

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 3

On rappelle le principe de l'algorithme du *tri fusion*.

Pour trier une liste, on partitionne celle-ci en deux listes de même taille (ou presque selon la parité) que l'on trie récursivement puis que l'on fusionne.

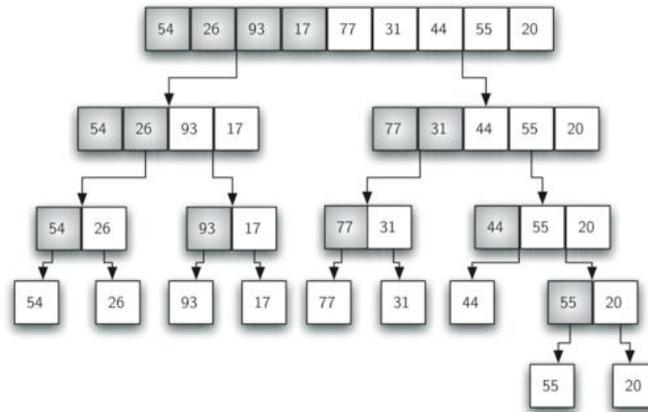


FIGURE 3 – Diviser pour régner - Division en sous-listes

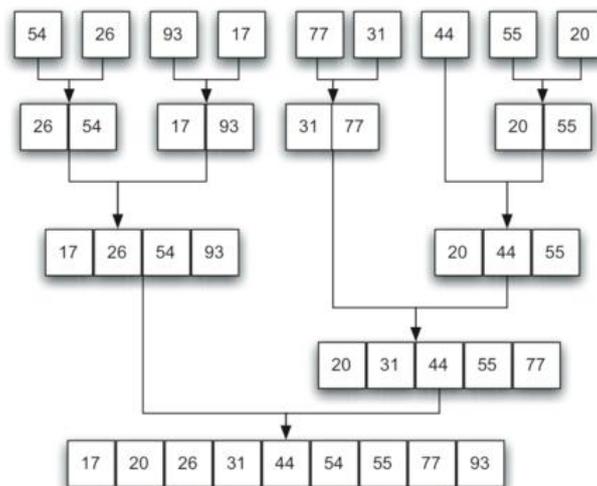


FIGURE 4 – Fusion des listes ordonnées

1. Écrire une fonction `fusion(T,g,m,d)` d'argument `T` une liste, `g`, `m`, `d` des entiers tels que `T[g:m]` et `T[m:d]` soient des sous-listes triées et qui modifie la liste `T` en modifiant la plage d'indices `[g:d]` pour que celle-ci contienne la fusion des deux sous-listes triées `T[g:m]` et `T[m:d]`. On pourra utiliser la construction d'une liste « dos-à-dos ».
2. Écrire une fonction `tri_fusion_rec(T,g,d)` qui réalise récursivement le tri fusion de la sous-liste `T[g:d]` en modifiant directement la liste `T`.
3. Écrire une fonction `tri_fusion(T)` qui effectue le tri fusion de la liste `T`.
4. Expliquer pourquoi il ne s'agit pas d'un tri en place.