ISM MPSI/PCSI, Informatique
Année 2023/2024

TP Informatique 29

Exercice 1

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.

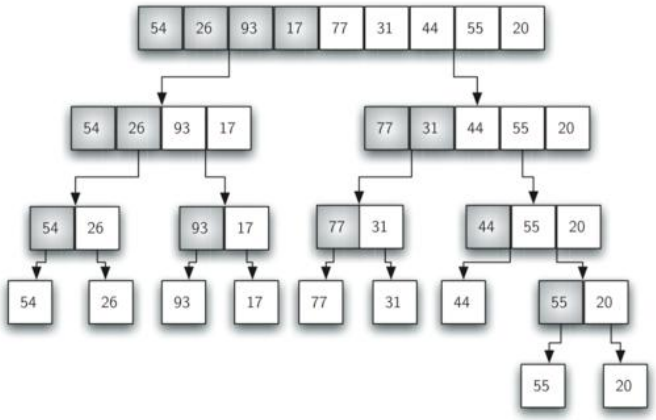

FIGURE 1 – Diviser pour régner - Division en sous-listes

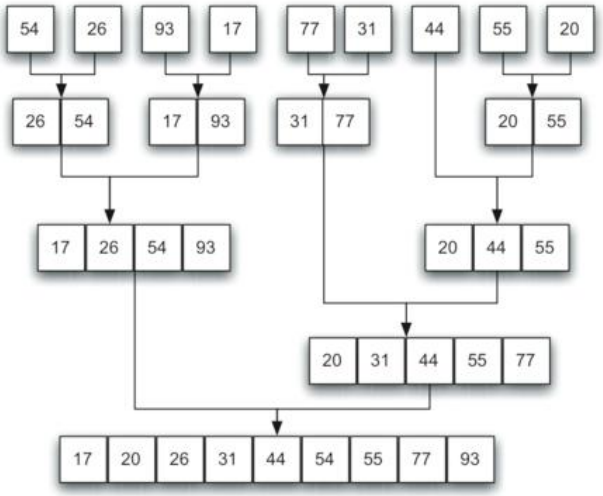

FIGURE 2 – Fusion des listes ordonnées

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.

Exercice 2

Le *tri par comptage* est un tri qui s'applique à une liste d'entiers naturels selon le principe suivant :

— on détermine le plus grand élément de la liste ;
— on construit la liste du nombre d'occurrences dans la liste de tous les entiers de 0 au plus grand ;
— on construit alors une nouvelle liste par répétition du nombre d'occurrences des entiers de 0 au plus grand.

Par exemple, avec la liste `L=[1,3,1,5]` :

— on détermine `max(L)=5` ;
— on construit la liste du nombre d'occurrences `[0,2,0,1,0,1]`, à savoir `0` apparaît zéro fois dans `L`, `1` apparaît deux fois dans `L`, `2` apparaît zéros fois dans `L`, etc. ;
— en parcourant la liste du nombre d'occurences, on construit la liste

```
[0]*0+[1]*2+[2]*0+[3]*1+[4]*0+[5]*1
```

1. Écrire une fonction `tri_comptage(L)` d'argument `L` une liste d'entiers qui renvoie la liste triée par le tri par comptage.
2. Déterminer une configuration simple où le tri par comptage est inefficace.