

TD - Le tri fusion

Nom : _____

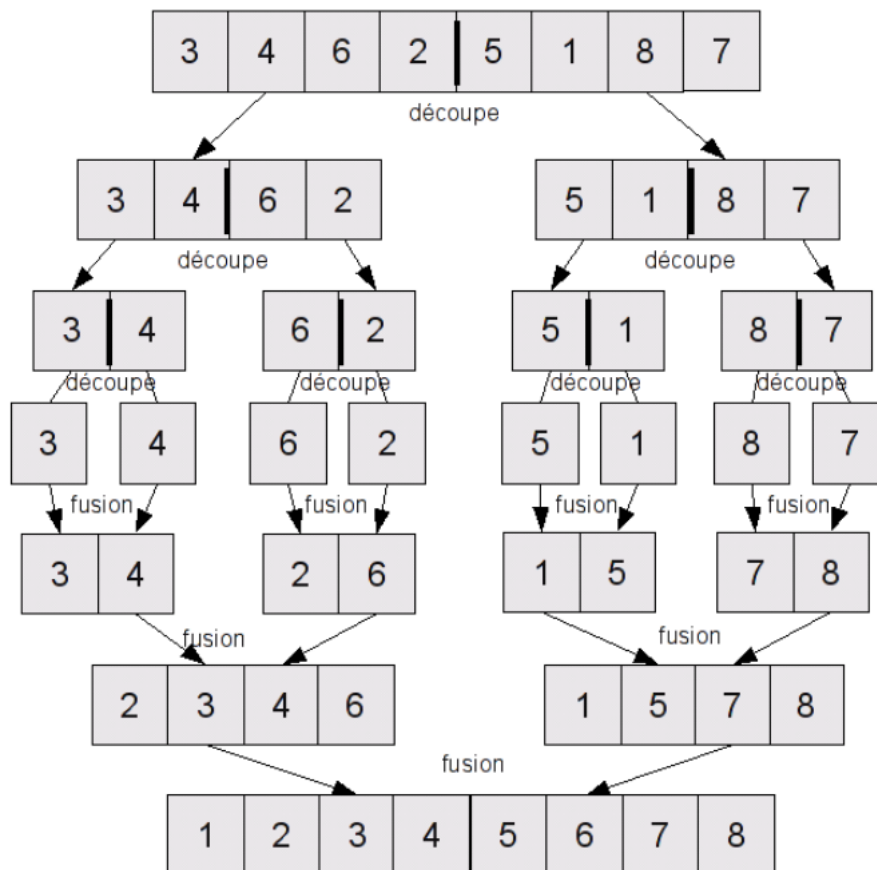
Introduction

Le tri fusion consiste à trier **récurivement** les deux moitiés de la liste, puis à fusionner ces deux sous-listes triées en une seule. La condition d'arrêt à la récursivité sera l'obtention d'une liste à un seul élément, car une telle liste est évidemment déjà triée.

Voici donc les trois étapes (**diviser, régner et combiner**) de cet algorithme :

- Diviser la liste en deux sous-listes de même taille (à un élément près) en la "coupant" par la moitié.
- Trier récursivement chacune de ces deux sous-listes. Arrêter la récursion lorsque les listes n'ont plus qu'un seul élément.
- Fusionner les deux sous-listes triées en une seule.

Ce schéma illustre le processus:



D'une manière générale, la complexité du tri par fusion est de l'ordre de $n \log(n)$ tout en comptant le temps de calcul nécessaire pour réaliser les différents découpages. Néanmoins, malgré cette rapidité d'exécution, cet algorithme n'est que très rarement utilisé. En effet, en raison des découpages répétés, l'algorithme nécessite de l'espace mémoire supplémentaire qui peut être un facteur critique lorsque le jeu de données à trier est conséquent.

Implémentation "Pythonnesque" du tri fusion...

Voici une version très "pythoneuse" du tri fusion, vous trouverez par ailleurs de nombreuses autres versions.

L'algorithme du tri fusion

fonction trifusion(T):

- Si la taille du tableau T est inférieure ou égale à 1:
 - On renvoie le tableau (un seul élément)
- $T1 \leftarrow$ la première moitié de T
- $T2 \leftarrow$ la seconde moitié de T
- On renvoie la fusion(trifusion(T1),trifusion(T2))

Voici l'algorithme de fusion

fonction fusion(T1,T2):

- Si T1 est vide:
 - On renvoie T2
- Si T2 est vide:
 - On renvoie T1
- Si $T1[0] < T2[0]$:
 - On renvoie $[T1[0]] + \text{fusion}(T1[1:],T2)$
- Sinon:
 - On renvoie $[T2[0]] + \text{fusion}(T1,T2[1:])$



À FAIRE 1:

Expliquer l'algorithme de fusion



À FAIRE 2:

Implémenter ces deux algorithmes et tester votre programme.