

Terminale NSI - Algorithmique

Diviser pour régner. Tri fusion, présentation

2020/04/26

Tri fusion

On applique diviser pour régner pour trier un tableau.

Même principe :

Tri Fusion (`tableau`):

- Si `tableau` est de taille ≤ 1 on ne fait rien.
- Sinon, On sépare `tableau` en 2 parties `gauche` et `droite`,
- On appelle Tri fusion sur `gauche` et sur `droite`
- On fusionne `gauche` et `droite` dans `tableau`

`fusionner (`tableau`, `gauche`, `droite`):`

- * On parcourt les deux tableaux `gauche` et `droite` en même temps,
Pour chaque paire d'éléments, on place le plus petit dans `tableau`.
- * S'il reste des éléments dans `gauche` ou dans `droite` on les place à la fin de `tableau`

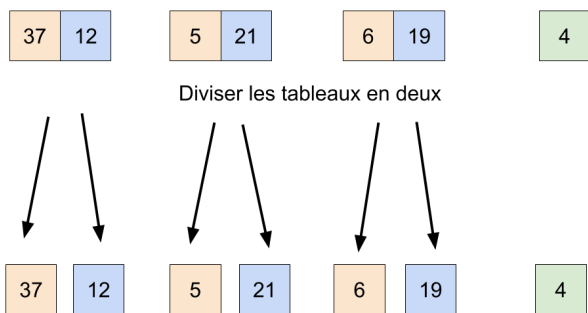
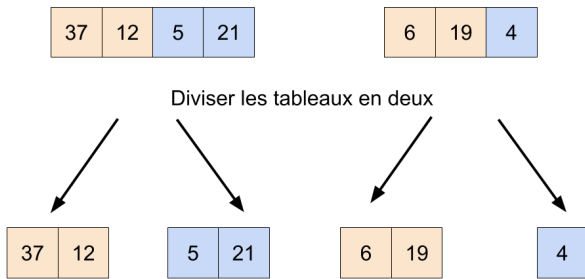
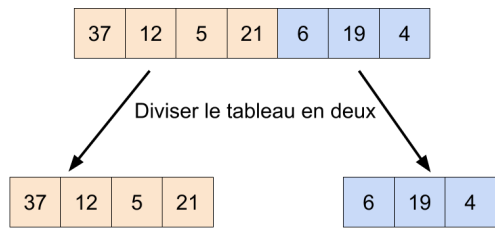
Exemple simple

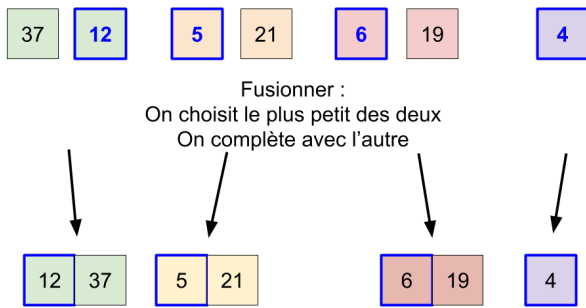
Vidéo Geek for geek

Exemple détaillé 0

37	12	5	21	6	19	4
----	----	---	----	---	----	---

Tri fusion de ce tableau





Fusionner :

On parcourt les tableaux par paire.
Chaque fois, le plus petit d'une paire d'éléments est choisi.

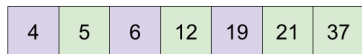
On complète avec ce qui reste



Fusionner :

On parcourt les tableaux par paire.
Chaque fois, le plus petit d'une paire d'éléments est choisi.

On complète avec ce qui reste



Présentation détaillée du tri fusion

<https://www.youtube.com/watch?v=TzeBrDU-JaY> (comme lui, même ordre, moins de détails)

Complexité

- La partie “diviser” est de complexité constante.
- La partie fusion utilise une boucle qui parcourt plusieurs tableaux en même temps.
 - On réalise à chaque étape la même chose :
 - * lire deux valeurs,
 - * comparer,
 - * ranger la plus petite. La complexité est linéaire.
- Combien d'étapes dans le tri fusion ? Autant d'étape qu'il en faut pour arriver à $\log_2(n)$ en effectuant des divisions par 2.

Exemple

Pour un tableau de taille $n = 64$ il faut :

$64/2 = 32, 32/2 = 16, 16/2 = 8, 8/2 = 4, 4/2 = 2, 2/2 = 1$: 6 étapes.

$2^6 = 64$.

Comme toujours quand on peut séparer le tableau en deux, la méthode diviser pour régner permet de ne réaliser que $\log_2 n$ étapes.

Utilisation du tri fusion

Contrairement au tri par sélection ou par insertion, le tri fusion est réellement utilisé en pratique.

Il a de nombreux avantages :

- complexité optimale (cela ne signifie pas qu'il est le plus rapide)
- stable (voir plus bas)
- facile à mettre en oeuvre

Cependant, il est possible d'améliorer la méthode :

`timsort`, le tri natif en Python et Javascript utilise une combinaison du tri fusion et du tri par insertion.