Tri fusion

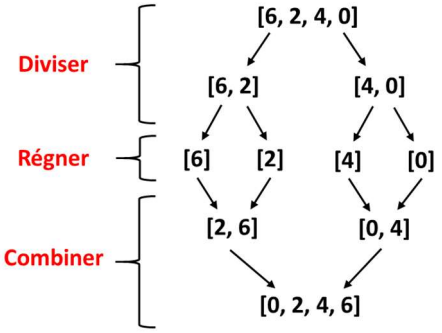
1. Introduction

Des algorithmes de tri ont déjà été étudié : le tri par insertion et le tri par sélection.
Le tri fusion (**merge sort** en anglais) est une nouvelle méthode de tri. Comme les autres algorithmes, il prend en entrée un tableau non trié et donne en sortie le même tableau, mais trié.
Le fonctionnement de l'algorithmique tri-fusion est basé sur la méthode diviser pour régner.

2. Principe du tri-fusion

Il comprend les étapes suivantes :

- On divise le tableau en deux sous tableaux et ainsi de suite par récursivité jusqu'à avoir un seul élément.
- Comme le tableau ne comporte plus qu'un seul élément, il est par conséquent trié (cas trivial).
- Ensuite on fusionne les deux tableaux triés en un seul tableau trié et ainsi de suite afin d'obtenir le tableau de départ trié.



Pour résumer, le principe du tri-fusion comprend trois étapes :

- **Diviser** : couper tableau en deux
- **Régner** : trier les éléments du tableau
- **Combiner** : fusionner deux tableaux.

✍ À l'aide du principe du tri-fusion, élaborer le schéma correspondant au tri de la liste suivante :
`[9, 8, 7, 6, 5, 4, 3, 2, 1]`.

3. Complexité

Le tri par insertion et le tri par sélection ont tous les deux une complexité **$O(n^2)$**.

3.1. Quelle complexité pour le tri fusion ?

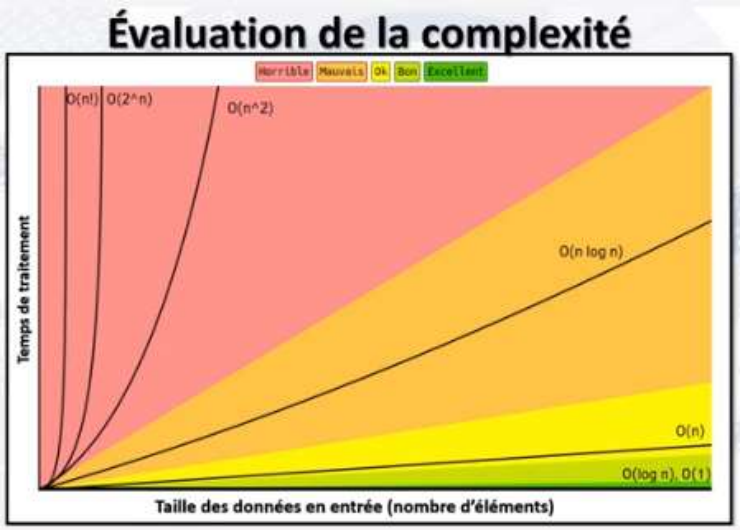
La partie **diviser** consiste à couper les tableaux en deux plusieurs fois de suite donc une complexité en **log(n)**.
Exemple pour un tableau de taille n = 64 :

- 64 / 2 = 32
- 32 / 2 = 16
- 16 / 2 = 8
- 8 / 2 = 4
- 4 / 2 = 2
- 2 / 2 = 1

Il faut 6 étapes ($2^6 = 64$).

La partie combiner (fusion) consiste à faire des comparaisons entre les premiers éléments de chaque tableau à fusionner, donc pour un tableau de n éléments, on aura n comparaisons. Par conséquent la complexité est linéaire donc une complexité en n.

En combinant les complexités des deux parties, on en déduit que la complexité du tri-fusion est en **O(n.log(n))**.



La comparaison des courbes des complexités **$O(n^2)$** (tri par insertion et tri par sélection) et **O(n.log(n))** (tri-fusion) montre que l'algorithme de tri-fusion est plus efficace que les algorithmes de tri par insertion et tri par sélection.

4. Algorithme du tri-fusion

L'algorithme va se composer de deux algorithmes :

- L'algorithme fusion qui va réaliser la partie combiner en fusionnant deux tableaux triés en un seul triés.
- L'algorithme tri-fusion qui réalise les parties diviser et régner en divisant le tableau en deux sous-tableaux récursivement jusqu'à obtenir un seul élément dans chaque tableau.

4.1. Algorithme fusion

Tant que les deux tableaux sont non vides, on boucle :

- Si une des deux listes est vide alors il suffit de rajouter au tableau des résultats la liste non vide.
- Sinon on compare la première valeur de chacun des tableaux, on prend la plus petite de ces valeurs et on la met dans le tableau des résultats et on la supprime du tableau initial.

Puis on retourne la liste des résultats.

✍ Implémenter de manière impérative la fonction **`fusion`**.

4.2. Algorithme tri-fusion

- Si la liste contient un seul élément alors on la retourne.
- Sinon :
  - On coupe le tableau en deux tableaux
  - On trie le tableau de gauche avec un appel récursif à ce même algorithme tri-fusion.
  - On trie le tableau de droite avec un appel récursif à ce même algorithme tri-fusion.
  - Puis on fusionne les deux tableaux (appel à l'algorithme fusion).

✍ Implémenter la fonction **`tri_fusion`**.
✍ Compléter le docstring avec un doctest significatif.

Pour les plus rapides :
✍ Implémenter la fonction **`fusion`** de manière récursive.

Source : https://lewebpedagogique.com/infoloth/
http://nsi4noobs.fr/Principe-du-tri-fusion