

Tri rapide

Fiche de synthèse

❗ **Important.** Le tri rapide est hors programme : il illustre le paradigme diviser pour régner et se compare au tri fusion, qui est au programme.

① PRINCIPE

☰ Tri rapide.

1. Choisir un élément du tableau, le **pivot**.
2. **Partitionner** : placer à gauche les éléments plus petits que le pivot, à droite les plus grands.
3. Trier récursivement la partie gauche et la partie droite.

Avec le tableau $[7, 2, 1, 6, 8, 5, 3, 4]$ et le pivot 4, on obtient les éléments plus petits $[2, 1, 3]$, le pivot $[4]$ et les éléments plus grands $[7, 6, 8, 5]$; on trie ensuite récursivement $[2, 1, 3]$ et $[7, 6, 8, 5]$.

```
def tri_rapide(tab):
    if len(tab) <= 1:
        return tab
    pivot = tab[0]
    petits = [x for x in tab[1:] if x < pivot]
    egaux = [x for x in tab if x == pivot]
    grands = [x for x in tab[1:] if x > pivot]
    return (tri_rapide(petits) + egaux
            + tri_rapide(grands))
```

② DIVISER POUR RÉGNER

Le tri rapide divise le problème en deux sous-problèmes, les résout récursivement, puis combine les résultats par une simple concaténation. Le travail principal se fait lors de la **division**, par la partition. Dans le tri fusion, au contraire, la division est immédiate (couper au milieu) et le travail principal se fait lors de la **combinaison**, par la fusion.

③ COMPLEXITÉ

- **Meilleur cas** : le pivot partage toujours le tableau en deux parties de même taille. Il y a $\log_2 n$ niveaux d'appels, et les partitions d'un même niveau parcourent n éléments en tout : la complexité est en $O(n \log n)$, comme le tri fusion.
- **Pire cas** : le pivot est toujours le plus petit ou le plus grand élément, par exemple en prenant le premier élément d'un tableau déjà trié. Chaque partition ne retire que le pivot : il y a n niveaux, et la complexité est en $O(n^2)$.
- **Cas moyen** : $O(n \log n)$.

Choisir le pivot au hasard, ou la médiane de trois éléments, rend le pire cas très improbable.

④ COMPARAISON AVEC LE TRI FUSION

Critère	Tri fusion	Tri rapide
Meilleur cas	$O(n \log n)$	$O(n \log n)$
Cas moyen	$O(n \log n)$	$O(n \log n)$
Pire cas	$O(n \log n)$	$O(n^2)$
Mémoire	nouvelles listes	en place possible
Stable	oui	non en version en place

Définition 1 (Tri stable) Un tri est **stable** quand deux éléments égaux restent dans l'ordre où ils étaient avant le tri.

Le tri rapide écrit avec des listes en compréhension crée de nouvelles listes ; sa version en place, qui échange les éléments dans le tableau lui-même, n'utilise presque pas de mémoire supplémentaire, mais n'est pas stable. En pratique, le tri rapide est souvent plus rapide que le tri fusion. On préfère le tri fusion quand il faut garantir $O(n \log n)$ ou un tri stable.

📌 À retenir.

- Le tri rapide partitionne autour d'un pivot, puis trie récursivement les deux parties.
- Il est en $O(n \log n)$ en moyenne et en $O(n^2)$ dans le pire cas.
- Son travail se fait à la division, celui du tri fusion à la combinaison.