
MPSI - ITC - 18 MAI 2026 - TP 30

TRI RAPIDE

Exercice 1 (Implémentation simple (pas en place))

Implémentez le tri rapide vu en cours. Cette première implémentation n'a pas besoin d'être en place.

Exercice 2 (Constatation expérimentale des complexités)

Le but de cet exercice est de regarder quelle semble être en pratique la complexité du tri rapide dans différents scénarios :

- Pour les listes aléatoires
- Pour les listes déjà triées
- Dans le meilleur cas

On vous fournit une fonction `affiche_et_pente(tri, gen)` qui prend deux arguments en entrée :

- Une fonction `tri` qui prend en entrée une liste à trier et la trie
- Une fonction de génération de listes à trier, `gen`, qui prend en entrée la taille souhaitée pour la liste et génère une liste de cette taille-là

La fonction `affiche_et_pente` mesure, pour différentes tailles, le temps que met `tri` à s'exécuter sur des listes de cette taille-là (en générant les listes avec `gen`). Les mesures sont affichées en log-log, et une régression linéaire de la courbe log-log est faite.

1. Codez une fonction `liste_alea(n)` qui prend en entrée un entier `n` et génère une liste de taille `n`, choisie aléatoirement parmi toutes les listes de taille `n` ne contenant que des éléments compris entre 0 et 50.
Aide : Vous pouvez utiliser la fonction `randint` du module `random`. Vous pouvez utiliser `help` pour lire la documentation de cette fonction, une fois que vous l'aurez importée.
2. Codez une fonction `liste_triee(n)` qui prend en entrée un entier `n` et renvoie une liste de taille `n` triée par ordre croissant.
3. Codez une fonction `liste_meilleur_cas` qui prend en entrée un entier `n` et renvoie une liste de taille `n` particulièrement adaptée à votre implémentation du tri rapide (par exemple, le premier élément est la médiane de la liste, et cette propriété est récursivement vraie pour la sous-liste des éléments plus petits que le premier, et la sous-liste des éléments plus grands que le premier).
Votre fonction a le droit de ne marcher que pour certaines tailles de liste, par exemple les tailles de la forme $2^n - 1$.
4. Utilisez la fonction `affiche_et_pente` fournie pour observer la complexité du tri rapide dans ces différents scénarios. Cela est-il cohérent avec les complexités vues en cours ?

Exercice 3 (Implémentation en place)

Implémentez le tri rapide en place.

Aide 1 : Je vous conseille d'avoir une fonction auxiliaire locale qui prend deux arguments en plus : des indices `bord_gauche` et `bord_droit` entre lesquels il faut trier la liste. Écrivez dans un commentaire s'il faut comprendre les indices comme *inclus* ou *exclus*, et ayez un code qui est cohérent avec votre choix.

Aide 2 : Vous pouvez avoir deux indices

- `gauche` du prochain emplacement où mettre un nombre plus petit que le pivot
- `droite` du prochain emplacement où mettre un nombre plus grand que le pivot

et les faire évoluer au fur et à mesure :

- Si l'élément à l'indice `gauche` est plus petit que le pivot, on incrémente l'indice `gauche`
- Sinon, on échange les éléments aux indices `gauche` et `droite`, et on décrémente l'indice `droite`

Exercice 4 (Bonus : médiane des médianes)

Une façon de s'approcher d'une complexité en $\mathcal{O}(n \log(n))$ pour le tri rapide sur n'importe quelle liste, est de choisir comme pivot une *approximation de la médiane de la liste*. Il faut, pour avoir la bonne complexité, calculer cette approximation de la médiane en $\mathcal{O}(n)$.

Une façon de faire est de procéder récursivement comme suit :

- On découpe la liste en morceaux de taille 5
- On calcule la médiane de chaque morceau de taille 5 et on le met au milieu du morceau en question.
- On recommence avec la liste des médianes de chaque morceau

Bien sûr, c'est encore plus intéressant de faire ce calcul approché *en place*, pour pouvoir s'en servir dans une implémentation en place du tri rapide.

1. Justifiez que la complexité de ce calcul approché de médiane est bien en $\mathcal{O}(n)$.
2. Implémentez ce calcul approché de médiane ; et un tri rapide qui l'utilise.
3. Observez la complexité expérimentale de ce tri rapide, et appelez le professeur pour lui montrer les résultats.