

# LISTES / TABLEAUX / TRIS : RÉSUMÉ

## I Listes (chaînées) vs Tableaux

### I.1 Spécificités

Listes :

- ⊙ Insertion possible d'un élément.
- ⊙ Accès au  $k^e$  élément en  $O(k)$ .

Tableaux :

- ⊙ Modification de la longueur impossible.
- ⊙ Accès au  $k^e$  élément en  $O(1)$ .

### I.2 Remarque

Python utilise des tableaux redimensionnables. C'est mieux, mais hors-programme.

### I.3 Un principe général

Pour l'implémentation des algorithmes (en général, en particulier les tris mais pas que= :

1. De par leurs spécificités, les tableaux sont naturellement adaptés à des traitements
2. De par leurs spécificités, les listes sont naturellement adaptées à des traitements

## II Algorithmes de tri

### II.1 Remarques

Remarque :

- Les listes sont des structures de données immuables (on dit aussi : persistantes). Pour trier une liste, on **retourne une nouvelle liste** ayant les mêmes éléments mais triée.
- Les tableaux sont des structures de données modifiables (on dit aussi : impératives). Pour trier un tableau, on ne retourne rien, mais on **modifie le tableau de départ** pour qu'il soit trié.

Remarque : un tri sur un tableau est dit "en place" lorsqu'il ne nécessite pas la création d'un tableau auxiliaire.

### II.2 Tri par sélection

Le tri par sélection consiste, pour trier une liste (ou un tableau) à :

- Déterminer l'élément minimal de la liste ou du tableau ;
- Le permuter pour le passer en tête.
- Recommencer sur les éléments restants.

Variante : on prend l'élément maximal.

Complexité dans le pire des cas :

### II.3 Tri par insertion

Le tri par insertion consiste, pour trier une liste (ou un tableau) à parcourir les éléments liste (/ le tableau) en insérant successivement chaque élément à sa place dans la liste déjà triée des éléments précédents (initialement vide).

Variante : on insère récursivement dans la liste déjà triée des éléments suivants.

Complexité dans le pire des cas :

## II.4 Tri par fusion

Le tri par fusion consiste, pour trier une liste (ou un tableau) à :

- Diviser la liste (/ le tableau) en deux moitiés ;
- Trier récursivement chacune des deux moitiés ;
- Fusionner les deux moitiés triées pour reconstituer la liste (/ le tableau) triée.

Complexité dans le pire des cas :

## II.5 Tri rapide (tri par pivot)

Le tri par pivot consiste, pour trier une liste (ou un tableau) à :

- Prendre le premier élément du tableau ou de la liste à trier ; on l'appelle le pivot ;
- Diviser le reste de la liste (/ du tableau) en deux moitiés : les éléments plus petits que le pivot, et les éléments plus grands ;
- Trier récursivement chacune des deux moitiés ;
- Raccorder les trois morceaux dans l'ordre.

Complexité dans le pire des cas :

Complexité en moyenne :

## II.6 Tri à bulles

Le tri à bulles consiste, pour trier une liste (ou un tableau) à :

- Parcourir la liste (/ le tableau) en permutant deux éléments consécutif à chaque fois qu'ils ne sont pas dans le bon ordre ;
- Recommencer sur les  $n-1$  premiers éléments de la liste (/ du tableau) ; etc.

Complexité dans le pire des cas :