

# Synthèse

## Les tableaux

Les tableaux (ou **tableaux statiques**) sont des structures de données fondamentales en informatique, et leur utilisation offre plusieurs avantages et inconvénients, notamment en termes de **complexité algorithmique**.

### Avantages des tableaux

#### 1. Accès rapide à un élément

- **Complexité :  $O(1)$**  (temps constant)
- Un des grands avantages des tableaux est l'accès direct à n'importe quel élément, grâce à l'indexation. Puisque les éléments sont contigus en mémoire, connaître l'indice permet de retrouver instantanément l'élément.

#### 2. Localité de la mémoire

- Les éléments sont stockés de manière contiguë en mémoire. Cela permet de profiter de la **localité spatiale**, ce qui améliore les performances du cache<sup>i</sup> processeur.

#### 3. Facile à utiliser

- Les tableaux sont simples à utiliser : il n'y a pas de gestion de liens entre les éléments.

## Inconvénients des tableaux

### 1. Insertion/Suppression coûteuses

- **Complexité :  $O(n)$**  (temps linéaire) pour insérer ou supprimer un élément à un index arbitraire.
  - Pour insérer ou supprimer un élément au milieu d'un tableau, il faut potentiellement décaler tous les éléments suivants, ce qui peut être coûteux en temps.
  - Par exemple, si l'on veut insérer un élément au début, tous les éléments doivent être déplacés d'une position.

### 2. Taille fixe

- Les tableaux sont généralement de taille **statique**, ce qui signifie que la taille doit être définie à l'avance et ne peut pas être modifiée après coup. Cela peut entraîner une perte de mémoire (si la taille allouée est trop grande) ou un dépassement (si la taille est trop petite).
- Solution : Les tableaux dynamiques comme les **listes** en Python corrigent cela, mais cela peut entraîner une augmentation de la complexité en termes de réallocation et de gestion de mémoire.

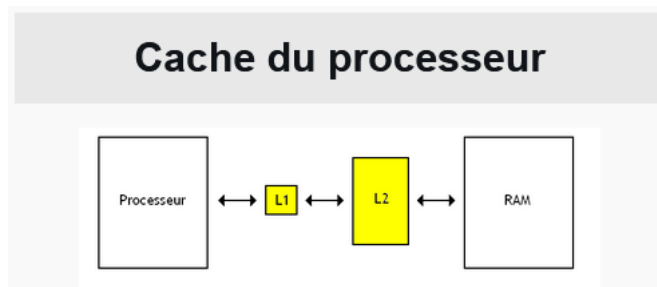
### 3. Mémoire contiguë nécessaire

- Comme les tableaux nécessitent un bloc continu de mémoire, il peut être difficile d'allouer de grands tableaux dans des environnements où la mémoire est fragmentée.

## Résumé des complexités courantes des opérations sur les tableaux

Opération	Complexité
Accès à un élément (via l'indice)	$O(1)$
Insertion à la fin (si de l'espace disponible)	$O(1)$
Insertion/Suppression au milieu ou au début	$O(n)$
Recherche d'un élément (sans tri)	$O(n)$

En résumé, les tableaux sont très efficaces pour l'accès direct et la gestion de petites séquences de données, mais peuvent devenir moins efficaces lorsqu'il s'agit de manipuler fréquemment leur contenu (insertion, suppression).



---

<sup>i</sup> Une mémoire cache est une mémoire qui enregistre temporairement des copies de données provenant d'une source, afin de diminuer le temps d'un accès ultérieur du processeur à ces données.