La structure de données Pile (Stack)

1. Définition

Une **Pile** est une structure de données linéaire qui suit le principe **LIFO** (*Last In, First Out*), c'est-à-dire **Dernier entré, premier sorti**. Cela signifie que le dernier élément ajouté est le premier à être retiré, comme une pile de livres où on retire toujours celui du dessus.

2. Principales opérations

- **Empiler (push)** : Ajoute un élément en haut de la pile.
- **Dépiler (pop)** : Retire et renvoie l'élément en haut de la pile.
- **Est vide (is_empty)** : Vérifie si la pile est vide.
- **Consulter le haut (peek/top)** : Regarde l'élément au sommet de la pile sans le retirer (optionnel).

3. Complexité algorithmique

- **Empiler (push)** : O(1)— L'ajout d'un élément en haut de la pile est une opération constante.
- **Dépiler (pop)** : O(1)— Le retrait de l'élément en haut de la pile est également une opération constante.
- **Consulter le haut (peek)** : O(1)— Accéder à l'élément au sommet de la pile est une opération instantanée.
- **Est vide** : O(1)— Vérifier si la pile est vide prend un temps constant.

Remarque : Ces complexités sont valables pour des piles implémentées efficacement (comme avec une liste chaînée).

4. Avantages

- **Simplicité d'implémentation** : Les piles sont faciles à implémenter et permettent de gérer l'ajout et le retrait d'éléments de manière très efficace.
- **Gestion des opérations imbriquées** : Elles sont idéales pour des situations où des éléments doivent être traités dans l'ordre inverse de leur insertion (ex. : annulations d'actions, évaluation d'expressions mathématiques).

5. Inconvénients

- **Accès restreint** : Il n'est possible d'accéder qu'au dernier élément ajouté. Pour accéder aux éléments en bas ou au milieu, il faut dépiler les éléments au-dessus.
- **Pas adaptée pour toutes les structures de données** : Une pile n’est pas optimale pour des situations où l’ordre FIFO est requis ou où un accès aléatoire aux éléments est nécessaire.

6. Cas d'utilisation

- **Gestion des appels imbriqués** : Lorsqu'une fonction est appelée par une autre, la pile conserve l'historique des appels, notamment au niveau du système (pile d'appels).
- **Algorithmes de parcours en profondeur (DFS)** : La pile est utile dans les parcours de graphes en profondeur, où les nœuds sont traités dans l'ordre inverse de leur arrivée.
- **Historique d'actions (annuler/refaire)** : Dans de nombreuses applications, la pile est utilisée pour gérer les actions que l'on peut annuler, comme dans les logiciels de traitement de texte.