Les piles et les files

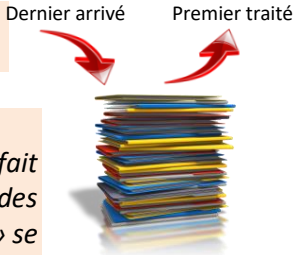
1. Introduction

Les **piles** et les **files** sont des **listes** particulières : on accède aux éléments par les **extrémités**, c'est-à-dire au début ou à la fin.
Elles sont utilisées par exemple pour des programmes qui doivent traiter des données qui arrivent au fur et à mesure.
On distingue :
- les piles (« stacks ») : le premier empilé est le dernier à être dépilé, « LIFO » (Last In first Out),
- les files : le premier entré est le premier à sortir, « FIFO » (First In First Out).

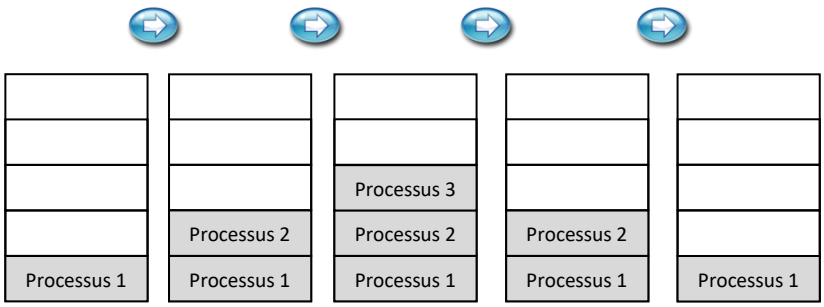
L'implémentation des piles et des files peut se faire avec des **listes chaînées** ou des **tableaux**.

2. Les piles

Exemple 1 : Empilement de dossiers
Dans une pile de dossiers, le dernier arrivé est le premier traité.

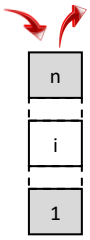


Exemple 2 : Gestion de processus par un système d'exploitation
Dans un ordinateur, lorsqu'un « processus 1 » fait appel à un « processus 2 » qui fait lui-même appel à un « processus 3 », l'ensemble est stocké dans une table des processus, propre au noyau du système d'exploitation. Lorsque le « processus 3 » se termine, le système sait qu'il doit revenir au « processus 2 », puis au « processus 1 ».



NOTA : On retrouve cet empilement dans les appels récursifs de fonction. Lors d'un débordement de pile, le message « stack overflow » apparaît.

Exemple 3 :
Dans un navigateur web, une pile sert à mémoriser les pages Web visitées. L'adresse de chaque nouvelle page visitée est empilée et l'utilisateur désempile l'adresse de la page précédente en cliquant le bouton « Afficher la page précédente »



Les principales fonctions associées aux piles sont :
- Ajouter au sommet : **empiler** (« **push** »),
- Supprimer du sommet : **dépiler** (« **pop** »),
- **Est_vide**, **Est_pleine**.

2.1. Structures de données

Definition 1 :

- Un ***objet*** représente un concept, une idée ou toute entité du monde physique ou virtuel.
- Un ***conteneur*** est un objet permettant de stocker d'autres objets (non forcément distincts).
- Une ***structure de données*** est un conteneur dynamique (i.e. de longueur modifiable). Elle est destinée à organiser des données informatiques dans le but de maximiser l'efficacité de certaines opérations (moindre coût temporel, moindre espace en mémoire).

Un conteneur S peut être étudié selon trois points de vue :

- l'accès, c'est-à-dire la manière d'accéder aux éléments du conteneur;
- le stockage, c'est-à-dire la manière de stocker les éléments du conteneur ;
- le parcours, c'est-à-dire la manière de parcourir les éléments du conteneur.

2.2 Structure LIFO

Définition 2 : Soit E un ensemble non vide. Une pile p d'éléments de E est une structure de données qui met en œuvre le principe :

dernier entré, premier sorti ou LIFO (Last - In - First - Out)

avec les conventions suivantes : une pile P est :

- soit Ø (pile vide, de profondeur 0) ;
- soit une pile de profondeur n ≥ 1, de la forme p = (q; s) où s est un élément de E, appelé ***sommet*** de p. et q une pile de profondeur n -1, appelé ***queue*** de p.

Applications en "ingénierie inverse" : "retour à la page précédente", " Undo"...

$$p = [\underbrace{p[0],\ p[1],\ \ldots,\ p[n-2]}_{\text{queue } q \text{ : non dynamique}}, \underbrace{p[n-1]}_{\text{sommet } s}]$$

Définition pratique en Python

Pour simplifier, nous nous appuierons sur la structure de liste.

Une pile est une structure LIFO sans parcours possible : on ne peut accéder qu'à l'élément situé au sommet.

P[2] est donc interdit par exemple. On doit utiliser p[-1] ou la méthode p.pop()

Contrainte dans tous les scripts du TP :

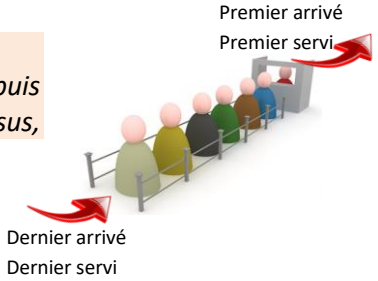
la pile p passée en paramètre doit être restituée en l'état.

Il faut donc reconstruire la pile avant de ressortir de la fonction

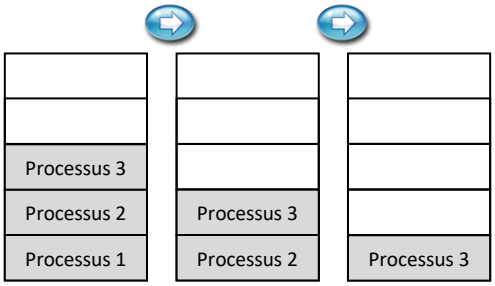
3. Les files

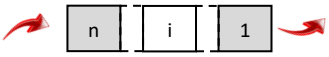
Exemple 5 : File d'attente
Dans une file d'attente, le premier arrivé est le premier servi.

Exemple 6 : Gestion de processus par un système d'exploitation
Dans un ordinateur, lorsque des appels aux processus « 1 », « 2 », puis « 3 » se succèdent, l'ensemble est stocké dans une table des processus, propre au noyau du système d'exploitation. Lorsque le « processus 1 » se termine, le système sait qu'il doit passer au « processus 2 », puis au « processus 3 ».



Processus 1 en cours
Processus 2 et 3 en attente



NOTA : On retrouve cette notion de file dans les appels successifs de fonction, la gestion des travaux d'impression, les « buffers », etc.

Les principales fonctions associées aux files sont :
- Ajouter en queue : **enfile**r (« **enqueue** »),
- Supprimer tête : **défile**r (« **dequeue** »),
- **Est_vide**, **Est_pleine**.