

Partie II : Algorithmique et Programmation

SOMMAIRE

Chapitre : Piles & File

I.	Structures de données	2
1.	Pile – Stack (Principe LIFO).....	2
2.	Pile (Primitives)	2
3.	Une liste comme une Pile	3
II.	File – Queue (Principe FIFO)	3
1.	File (Primitives)	3
2.	Une liste comme une File.....	3

Pile & File

I. Structures de données

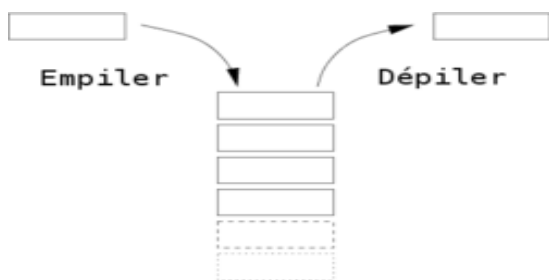
Une structure de données est une organisation logique des données permettant de simplifier ou d'accélérer leur traitement.

Exemples de structure de données : (liste, chaîne de caractères, tuples, dictionnaires, pile, file, ...)

1. Pile – Stack (Principe LIFO)

En informatique, une pile (en anglais stack) est une structure de données fondée sur le principe « dernier arrivé, premier sorti » (ou LIFO pour Last In, First Out), ce qui veut dire que les derniers éléments ajoutés à la pile seront les premiers à être récupérés.

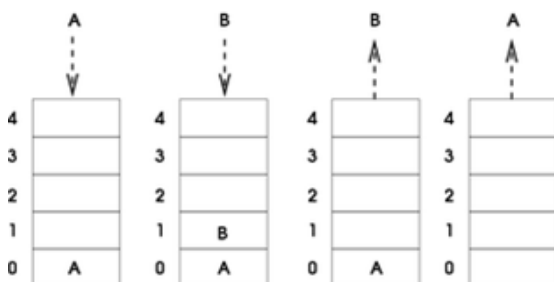
Le fonctionnement est donc celui d'une pile d'assiettes : on ajoute des assiettes sur la pile, et on les récupère dans l'ordre inverse, en commençant par la dernière ajoutée.



2. Pile (Primitives)

Voici les primitives communément utilisées pour manipuler des piles :

- « **empiler** » : ajoute un élément sur la pile. Terme anglais correspondant : «Push».
- « **dépiler** » : enlève un élément de la pile et le renvoie. En anglais : «Pop».
- « **vide** » : renvoie vrai si la pile est vide, faux sinon. En anglais : «Empty».
- « **remplissage** » : renvoie le nombre d'éléments dans la pile. En anglais : «count, size or length».



3. Une liste comme une Pile

En python, Les méthodes des listes rendent très facile leur utilisation comme des piles, où le dernier élément ajouté est le premier récupéré (“dernier entré, premier sorti”, ou LIFO pour “last-in, first-out”).

Méthodes de la liste pour la gérer en tant que pile :	
pile=[] #déclaration d'une pile	
empiler	pile.append(elem)
dépiler	pile.pop()
vide	pile==[] ou bien len(pile)==0
remplissage	len(pile)

II. File – Queue (Principe FIFO)

Une file (en anglais queue) est une structure de données fondée sur le principe « premier arrivé, premier sorti » (ou FIFO pour First In, First Out), ce qui veut dire que les premiers éléments ajoutés à la file seront les premiers à être récupérés.

Le fonctionnement ressemble à une file d'attente : les premières personnes à arriver sont les premières personnes à sortir de la file.

1. File (Primitives)

Voici les primitives communément utilisées pour manipuler des files :

- « **Enfiler** » : ajoute un élément dans la file. Terme anglais correspondant : « Enqueue ».
- « **Défiler** » : renvoie le prochain élément de la file, et le retire de la file. Terme anglais correspondant : « Dequeue ».
- « **La file est-elle vide ?** » : renvoie « vrai » si la file est vide, « faux » sinon.
- « **Remplissage** » : renvoie le nombre d'éléments dans la file.

2. Une liste comme une File

En python, Les méthodes des listes rendent très facile leur utilisation comme des files, où le premier élément ajouté est le premier récupéré (“premier entré, premier sorti”, ou FIFO pour “first-in, first-out”).

On peut effectuer ça sur les listes avec deux manières :

Insertion en queue, récupération en tête.	
file = [] # déclaration d'une file vide au départ.	
enfiler	file.append(elem)
défiler	file.pop(0)
vide	file==[]
remplissage	len(file)

Insertion en tête, récupération en queue.	
file = [] # declaration d'une file vide au départ.	
enfiler	file.insert(elem,0)
défiler	file.pop()
vide	file==[]
remplissage	len(file)