

TD6 Piles et files

Les piles (*stack*) et les files (*queue*) sont des structures de données. Elles peuvent être implémentées par des listes, mais obéissent à certaines règles pour lire, retirer ou ajouter des données.

Une **pile** est une structure de type *LIFO* (*Last In First Out*) : les derniers éléments ajoutés à la pile sont les premiers à être retirés. Les listes Python se prêtent bien à l'implémentation de pile car les fonctions `append` et `pop` agissent sur le dernier élément.

Une **file** est une structure de type *FIFO* (*First In First Out*) : les premiers éléments ajoutés à la file sont aussi les premiers à être récupérés. Les files peuvent s'implémenter avec le module `queue` de Python. Dans ce TP néanmoins, les files sont simulées avec des listes (même si la complexité des opérations n'est pas optimale).

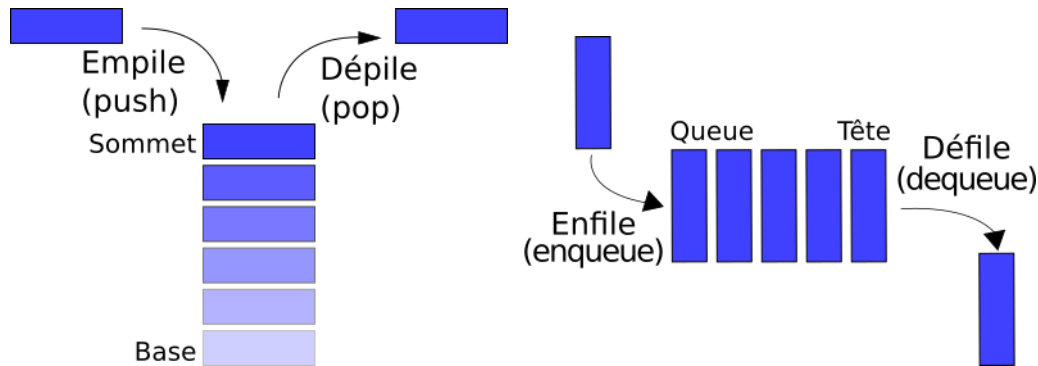


Figure 1: à gauche une pile, à droite une file.

Les files peuvent être utilisées afin de mémoriser des données en attente de traitement. Les piles peuvent être utilisés pour implémenter des algorithmes récursifs dans un style itératif (ex plus bas : affichage préfixe d'un arbre).

Fonctions de base

On utilise les fonctions suivantes pour les piles et les files :

```
# PILES
def empiler(e, p):
    p.append(e)

def depiler(p):
    p.pop()
# Rq : p.pop() permet d'accéder à la valeur du sommet de la pile en même temps que
    depiler.

def sommet(p):
    return p[len(p)-1]

# FILES
```

```
def enfiler(e, f):
    f.append(e)

def defiler(f):
    for i in range (len(f)-1):
        f[i] = f[i + 1]
        f.pop()
#Rq : f.pop(0) permet d'accéder à la valeur de la tête de file en même temps que
defiler.

def tete(f):
    return f[0]
```

Petits exercices

1. Inversion d'une pile

Écrivez une fonction `inverser(p)` qui inverse l'ordre des éléments d'une pile en utilisant une file.

2. Inversion d'une file

Écrire une fonction qui inverse une file d'entiers. On utilisera une pile pour faire l'inversion.

3. Parenthésage correct

Une liste chaînée de caractères contient des parenthèses. Une liste est correcte si toute parenthèse ouvrante se trouve avant la parenthèse fermante correspondante et si la sous liste entre ces deux parenthèses est aussi correcte. Par exemple `(() (()) ())` est correcte mais `) ()` ne l'est pas. Écrire une fonction qui détermine si une liste est correcte.

4. Palindrome

Un palindrome est une chaîne de caractères qui se lit de la même manière de gauche à droite et de droite à gauche. En utilisant une pile et une file, écrire une fonction qui détermine si une liste chaînée de caractères est un palindrome.

5. Affichage d'un arbre selon parcours préfixé

Écrire une fonction **non réursive** qui étant donné un arbre binaire (cf TPs précédents), imprime l'étiquette de chaque noeud de l'arbre, selon un parcours préfixé. (*Utiliser une pile*)

6. Affichage d'un arbre en largeur

Écrire une fonction qui étant donné un arbre binaire (cf TPs précédents), imprime l'étiquette de chaque noeud de l'arbre, selon un parcours en largeur (d'abord la racine puis ses fils, puis les noeuds de profondeur 2, etc.). (*Utiliser une file*)