



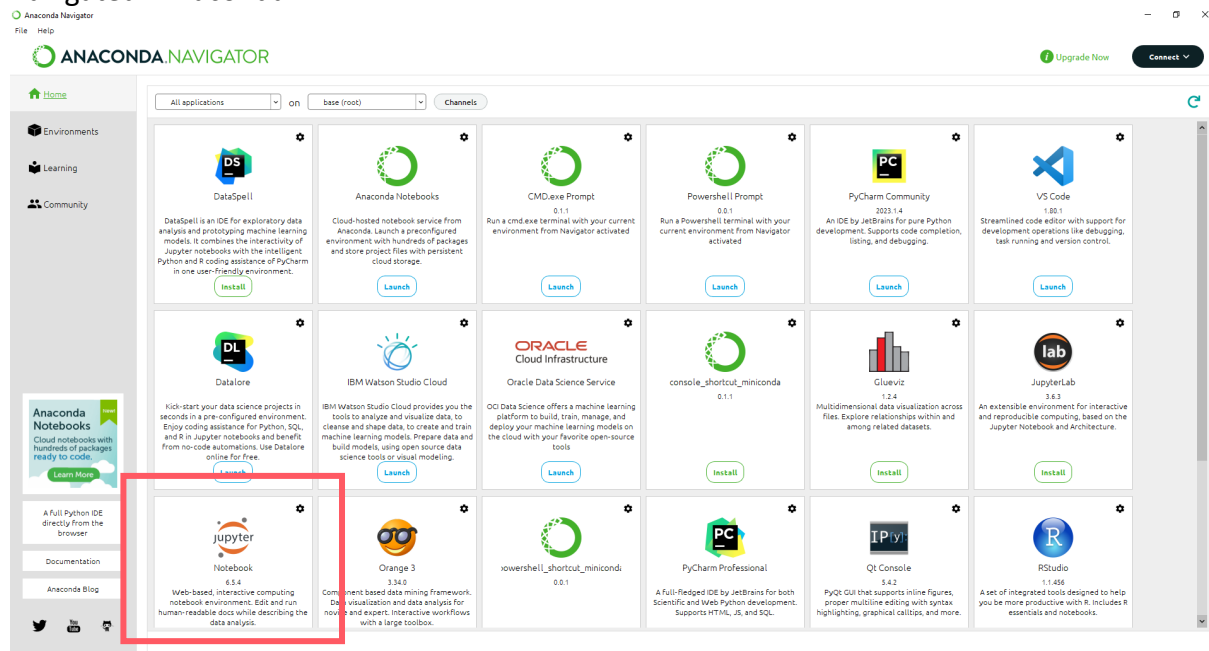
Licence 2

Algorithmique, structures de données et programmation

TD7 – File en Python

Objectif : Dans ce TD, nous utiliserons le conteneur `deque` du module `collections` pour représenter des files en Python et la `list` pour représenter des piles en Python.

Téléchargez le fichier [TD7.ipynb](#) de la plate-forme de cours Moodle. Enregistrez-le dans votre dossier de travail et ouvrez-le en utilisant l'IDE [Jupyter Notebook](#) accessible dans le navigateur Anaconda.



Implémentation du module "File"

Nous utiliserons le conteneur `deque` du module `collections` pour représenter des files en Python. En se basant sur le cours 7 d'« Algorithmique, structure de données et programmation », vous avez une implémentation des fonctions primitives du module "File" dans le fichier [TD7.ipynb](#) :

- `queue_create()` – la fonction qui crée une nouvelle file.
- `queue_isempty()` – la fonction qui teste si une file est vide.
- `queue_in()` – la fonction qui ajoute un nouvel élément dans une file.
- `queue_out()` – la fonction qui supprime un élément d'une file non vide.

Attention :

Pour réaliser tous les exercices suivants vous devez utiliser **exclusivement** les fonctions primitives disponibles dans les modules [TD6_pile.py](#) et [TD7.ipynb](#), sauf le cas contraire indiqué au début de l'exercice.

Exercice 1 : Création d'une file

Créez une file en utilisant le module "File". Remplissez-la avec quelques éléments de votre choix.

Exercice 2 : Copie d'une file

Écrivez une fonction `queue_copy(q)` recevant une file `q` comme argument et renvoyant une copie `q2` de cette file.

Remarque : la file `q` doit être conservée.

Exercice 3 : Inversion d'une pile

Écrivez une fonction qui inverse une pile `P1` d'entiers en utilisant une file auxiliaire. Pour inverser une pile on aurait besoin d'utiliser une file dans laquelle on enfiler ce qui a été dépiler, puis on empile à partir de la file pour retrouver les éléments de la pile originale dans l'ordre inverse.

Exercice 4 : Traitement d'éléments d'une pile

On a une pile `P1` contenant des entiers positifs. Écrivez un programme pour déplacer les entiers de `P1` dans une pile `P2` de façon à avoir dans `P2` tous les nombres pairs en dessus des nombres impairs en gardant l'ordre d'apparition des nombres pairs et en inversant l'ordre d'apparition des nombres impairs.

Remarque : pour la résolution de cet exercice, vous pouvez utiliser une pile auxiliaire `P3`.

Exercice 5 : La moyenne mobile

Remarque : Pour cet exercice vous pouvez utiliser directement des fonctions/méthodes de la bibliothèque `deque`.

Écrivez une fonction de calcul de la moyenne mobile en utilisant la bibliothèque `deque`.

Remarque : La moyenne mobile est une moyenne qui au lieu d'être calculée sur l'ensemble des N valeurs d'une série, est calculée tour à tour sur chaque sous-ensemble de n valeurs consécutives ($n \leq N$). [Wikipédia]

Mesures	2	3	5	8	8	7	8	5
Moyenne glissante	néant	$(2 + 3 + 5)/3$ 3,3333	$(3 + 5 + 8)/3$ 5,3333	$(5 + 8 + 8)/3$ 7	$(8 + 8 + 7)/3$ 7,6666	$(8 + 7 + 8)/3$ 7,6666	$(7 + 8 + 5)/3$ 6,6666	$(8 + 5 + 2)/3$ 5

Exercice 6 : Implémentation alternative d'une file

Implémentez une file en utilisant deux piles : écrire les opérations enfiler, défiler de ce cas de figure.

Pour aller plus loin

Exercice 7 :

Implémentez une pile en utilisant deux files : écrire les opérations empiler, dépiler de ce cas de figure.