STRUCTURES DE DONNÉES ET ALGORITHMES · CHAPITRE 6

Retirer le premier élément ajouté

Appliquez le contrat FIFO, utilisez deque en Python et une file préallouée MATLAB, puis comprenez le principe du tampon circulaire.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Appliquer le contrat FIFO sans décaler tous les éléments.
- FIFO conserve l'ordre d'arrivée en ajoutant à l'arrière et retirant à l'avant.
- Une deque ou des indices de tête évitent de décaler toute la collection à chaque retrait.

Prérequis. Avoir étudié les chapitres DSA précédents ; savoir accéder à une collection et suivre des indices.

Partir d'un besoin

Un service reçoit les tâches 10 puis 20. Pour respecter l'ordre d'arrivée, il doit traiter 10 d'abord. Retirer toujours le dernier élément inverserait cette règle.

Définir la structure et les conventions

Une **file** suit **FIFO**, First In, First Out : premier entré, premier sorti. Enfiler ajoute à l'arrière ; défiler retire à l'avant. La file exprime une règle de service, pas une obligation de décaler le stockage.

Python fournit `deque` dans le module collections. `from collections import deque` rend cet outil disponible ; append ajoute à droite et popleft retire à gauche. MATLAB utilise ici un tableau de capacité trois et deux indices : head désigne la prochaine case à lire ; tail désigne la dernière case écrite. La file est vide si head dépasse tail.

Cette représentation MATLAB avance sans réutiliser les anciennes cases. Un **tampon circulaire** les réutilise en revenant au début du tableau : en indices à partir de 0, la position suivante est `(index + 1) MOD capacity`. Un compteur d'occupation permet de distinguer vide et plein lorsque les indices se rejoignent.

Suivre les états

Trace FIFO

Opération	Éléments en attente	Sortie
Enfiler 10	[10]	Aucune
Enfiler 20	[10, 20]	Aucune
Défiler	[20]	10

Comparer les réalisations

La sortie est 10 et la file contient encore 20. La conversion list(queue) crée une liste pour afficher l'état ; elle ne fait pas partie de l'opération de retrait. Dans une file générale, vérifiez aussi la capacité avant d'écrire.

Pseudo-code

```
// example.pseudo
// use_queue
// ENQUEUE adds at the back; DEQUEUE removes the front.
// ENQUEUE 10; ENQUEUE 20; DEQUEUE returns 10.
// The remaining queue contains 20.
```

Python

```python
# use_queue.py
from collections import deque
queue = deque()
queue.append(10)
queue.append(20)
removed = queue.popleft()
remaining = list(queue)
print(removed, remaining)
```

MATLAB

```matlab
% use_queue.m
storage = zeros(1, 3);
head = 1;
tail = 0;
tail = tail + 1;
storage(tail) = 10;
tail = tail + 1;
storage(tail) = 20;
if head > tail
    error('Empty queue');
end
removed = storage(head);
head = head + 1;
remaining = storage(head:tail);
disp(removed);
disp(remaining);
```

Justifier et compter

Les opérations aux extrémités d'une deque Python sont approximativement O(1). Le tableau préalloué effectue ici des accès Θ(1). En revanche, pop(0) sur une liste Python décale les éléments et coûte Θ(n). Transformer une file en liste ou extraire son contenu pour l'afficher coûte Θ(n).

Prévoir puis expérimenter

Enfilez 30 après le retrait de 10, puis prédisez les deux retraits suivants. Ils donnent 20 puis 30. Dans le tableau linéaire, une quatrième écriture ne peut pas réutiliser la première case sans une convention circulaire.

Exercice corrigé

▼ Une file circulaire de capacité 3 a un compteur égal à 0. Est-elle pleine si les indices avant et arrière sont égaux ?

Non. Avec cette convention, le compteur d'occupation indique le vide (0) ou le plein (3). Les indices seuls peuvent être identiques dans les deux situations.

À retenir

- FIFO conserve l'ordre d'arrivée en ajoutant à l'arrière et retirant à l'avant.
- Une deque ou des indices de tête évitent de décaler toute la collection à chaque retrait.
- Un tampon circulaire doit distinguer le vide du plein, par exemple avec un compteur d'occupation.

Version web : https://www.clearcode.fr/courses/programming/dsa/linear/queues