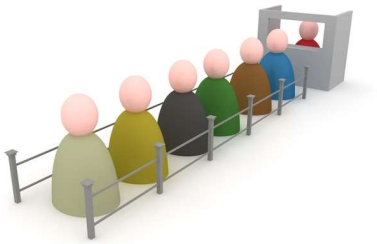
Les piles et les files

Dernière mise à jour le : 27/10/2023



Les piles et les files sont deux structures de données linéaires qui permettent, au même titre que les listes, de gérer des séquences d'éléments. Ainsi, dans une pile et dans une file chaque élément est également repéré par sa position, il y a un premier, un dernier, chaque élément a un successeur (sauf le dernier) et un prédecesseur (sauf le premier).

Les opérations disponibles pour ces deux structures sont assez proches car dans les deux cas, on veut pouvoir :

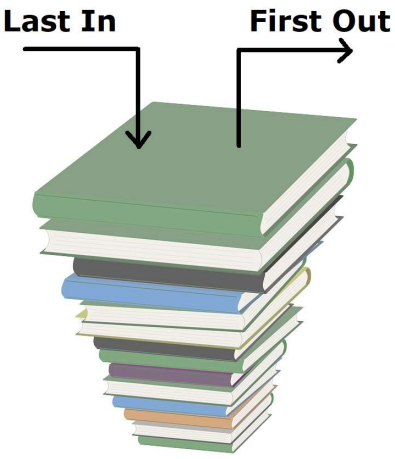
- créer une pile/file vide
- connaître sa taille
- lui ajouter un élément
- lui retirer un élément
- (accéder à un élément particulier)

Cependant, la politique d'ajout/retrait des éléments dans la séquence n'est pas la même. Le nom des opérations diffèrent également pour mieux distinguer les deux structures.

Les piles

Il faut se représenter une pile comme... une pile de livres ! Seul le livre disposé sur le dessus est accessible : l'ajout et le retrait d'un livre ne peut donc se faire que sur le *sommet* de la pile.

On dit que les piles sont en mode **LIFO** (*Last In, First Out* qui signifie « dernier entré, premier sorti »).



Interface d'une pile

Le jeu d'opérations disponibles pour une pile est :

- `construire_pile()` : crée une pile vide
- `taille(P)` : accès au nombre d'éléments dans la pile `P`
- `empiler(P, e)` : ajoute l'élément `e` au sommet de la pile `P`.
- `depiler(P)` : retire l'élément au sommet de la pile `P`. **Précondition** : `P` n'est pas vide.
- `sommet(P)` : pour accéder (en lecture) au sommet de la pile `P` (sans le retirer de la pile). **Précondition** : `P` n'est pas vide.

En anglais, une pile se dit *stack*, l'opération `empiler` est souvent notée `push`, l'opération `depiler` est souvent notée `pop` et l'opération `sommet` est souvent notée `top` ou `peek`.

Remarque : Certaines signatures algorithmiques peuvent légèrement varier. Par exemple, on peut parfois voir l'opération `est_vide` (qui teste si une pile est vide) à la place de `taille` (une pile est vide si et seulement si sa taille vaut 0) ou encore l'opération `depiler` qui renvoie également le sommet (donc l'opération `sommet` n'est plus nécessaire). C'est un choix libre qui ne change pas la nature de la structure de données abstraite mais la façon d'écrire des algorithmes.

Représentation d'une pile et exemple

Une pile contenant les éléments 'a', 'b' et 'c' ('a' étant le sommet et donc 'c' le fond de la pile) sera représentée :

$$>\text{'a', 'b', 'c'}]$$

Exemple : On considère la pile `P` : >'a', 'b', 'c']. Voici comment la manipuler :

Opération	Contenu de la pile
`empiler(P, 'e')`	>'e', 'a', 'b', 'c']
`depiler(P)`	>'a', 'b', 'c']
`depiler(P)`	>'b', 'c']
`sommet(P)`	renvoie 'b'
`depiler(P)`	>'c']
`empiler(P, 'm')`	>'m', 'c']
`taille(P)`	renvoie 2

Applications des piles

Les piles sont très utilisées en informatique. Voici quelques usages caractéristiques :

- Les algorithmes récursifs utilisent une pile d'appel pour mémoriser les contextes d'exécution de chaque appel. (déjà abordé)
- 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 dépile l'adresse courante pour accéder à la page précédente en cliquant le bouton « Afficher la page précédente ».
- La fonction « Annuler la frappe » (en anglais Undo, le célèbre CTRL+Z) d'un traitement de texte mémorise les modifications apportées au texte dans une pile.
- On peut aussi utiliser une pile pour parcourir *en profondeur* un graphe et mémoriser les sommets visités. (voir Thème 5 : Algorithmique)
- La vérification du bon parenthésage d'une expression peur également se faire à l'aide d'une pile.
- etc.

Certains de ces exemples seront abordés dans les activités.

Implémentations

Une pile est généralement implémentée par :

- un tableau (redimensionnable ou non)
- ou par une liste chaînée.

Selon le cas, il faudra veiller à ce que l'implémentation soit la plus efficace possible.

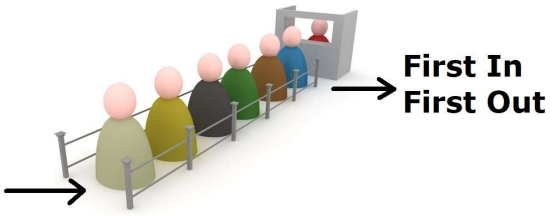
- Si on utilise un tableau, les opérations `empiler` et `depiler` seront plus efficaces si elles se font à la fin du tableau plutôt qu'au début car cela ne nécessite pas de décaler les autres éléments.
- En revanche, si on utilise une liste chaînée, elles seront plus efficaces si elles ont lieu au début (car pour accéder au dernier élément il faut parcourir tous les éléments de proche en proche à partir du premier qui est le seul accessible).

> Nous implémenterons une pile dans les activités.

Les files

Il faut se représenter une file comme... une file d'attente ! On ne peut entrer dans la file qu'en dernière position et on ne peut la quitter que si on est le premier. L'ajout d'un élément dans une file ne peut se faire qu'à la fin (en dernière position) et le retrait d'un élément ne peut se faire qu'au début (en première position).

On dit que les files sont en mode **FIFO** (*First In, First Out* qui signifie « premier entré, premier sorti »).



Interface d'une file

Le jeu d'opérations disponibles pour une file est :

- `construire_file()` : crée une file vide
- `taille(F)` : accès au nombre d'éléments dans la file `F`
- `enfiler(F, e)` : ajoute l'élément `e` en dernier dans la file `F`.
- `defiler(F)` : retire le premier élément de la file `F`. **Précondition** : `F` n'est pas vide.
- `premier(F)` : pour accéder (en lecture) au premier élément de la file `F` (sans le retirer de la file). **Précondition** : `F` n'est pas vide.

> En anglais, une file se dit *stack*, l'opération `enfiler` est souvent notée `enqueue`, l'opération `defiler` est souvent notée `dequeue` et l'opération `premier` est souvent notée `front` ou `peek`.

Remarque : Comme pour les piles, on pourrait remplacer l'opération `taille` par l'opération `est_vide` et choisir que `defiler` renvoie également le premier élément pour s'économiser l'opération `premier`.

Représentation d'une file et exemple

Une file contenant les éléments 'a', 'b' et 'c' ('a' étant le premier et 'c' le dernier) sera représentée :

$$<\text{'a'}, \text{'b'}, \text{'c'}<$$

Exemple : Voici comment manipuler une file F :

Opération	Contenu de la file F
`F = construire_file()`	<<
`taille(F)`	renvoie 0
`enfiler(F, 'a')`	<'a'<
`enfiler(F, 'b')`	<'a', 'b'<
`enfiler(F, 'c')`	<'a', 'b', 'c'<

`premier(F)`	renvoie 'a'
`defiler(F)`	<'b', 'c'<
`enfiler(F, premier(F))`	<'b', 'c', 'b'<

Applications des files

Les piles sont très utilisées en informatique. Leur usage caractéristique concerne les files d'attentes :

- Un système d'exploitation gère l'ordonnancement des processus par des files (voir Thème 3 : Architectures matérielles, systèmes d'exploitation et réseaux)
- Une imprimante gère les tâches d'impression avec des files : chaque nouvelle tâche est insérée dans une file d'attente, et celles-ci sont traitées dans l'ordre d'arrivée.
- On peut aussi utiliser une file pour parcourir *en largeur* un graphe et mémoriser les sommets visités. (voir Thème 5 : Algorithmique)
- etc.

> Certains de ces exemples seront abordés dans les activités.

Implémentations

Il existe différentes façons d'implémenter une file, on peut par exemple utiliser :

- un tableau (redimensionnable ou non)
- une liste chaînée
- deux piles

Avec ces implémentations, il faudra en général faire un compromis sur l'efficacité des opérations car celles-ci nécessitent de travailler sur les deux extrémités de la file (pour enfiler/défiler).

- Si on utilise un tableau, les opérations en début sont coûteuses et celles à la fin ne le sont pas. On peut alors décider d'enfiler en fin de tableau (peu coûteux) mais il faudra défiler en début de tableau (coûteux). Si on fait le choix inverse, c'est l'opération `defiler` qui sera peu coûteuse et l'opération `enfiler` qui sera coûteuse.
- Si on utilise une liste chaînée, c'est l'inverse (efficace en tête et coûteux en queue) mais le problème reste le même : une des deux opérations sera moins efficace.

> Il existe en réalité une implémentation plus efficace mais nous n'en parlerons pas ici.

Références :

- Documents ressources de l'équipe éducative du DIU EIL, Université de Nantes, Christophe JERMANN et Christophe DECLERCQ.
- Article Wikipédia sur les piles pour les exemples d'applications .

Germain BECKER, Lycée Mounier, ANGERS

Voir en ligne : info-mounier.fr/terminale_nsi/structures_donnees/piles-files