

INFO2009-Introduction à l'informatique

Aide à la réussite - Les piles, les files, et autres types abstraits

Bertrand Alexis

Université de Liège

2024

Les Structures de données

Les structures de données possèdent une interface et une implémentation. L'interface reste inchangée et correspond à une sorte de documentation permettant d'utiliser la structure de données. L'implémentation correspond à comment la structure et ses fonctions ont été écrites dans le code.

Les listes

Une liste est une structure de donnée capable de retenir une collection de valeurs. Une liste permet à l'utilisateur d'ajouter un nouvel élément et de retirer un élément librement (peu importe sa place).

Les listes sont la base de structures de données plus complexes. Celles-ci peuvent être implémentées de deux manières :

- ▶ Tableau
- ▶ Liste chaînée
 - ▶ Liste simplement chaînée
 - ▶ Liste doublement chaînée

Les piles

Une pile est une structure de données capable de retenir une collection de valeurs. Une pile permet à l'utilisateur d'ajouter un nouvel élément et de retirer un élément en suivant le principe *LIFO* (Last In First Out).

L'interface contient les fonctions suivantes :

- ▶ `push(p, v)` : empile (ajoute au-dessus) la valeur `v` sur la pile `p`.
- ▶ `pop(p)` : dépile (enlève au-dessus) une valeur de `p` et retourne cette valeur.
- ▶ `top(p)` : retourne la valeur au-dessus de la pile.
- ▶ `size(p)` : retourne la taille de `p`.
- ▶ `is_empty(p)` : retourne une valeur booléenne indiquant si la pile est vide ou pas.

Les files

Une file est une structure de données capable de retenir une collection de valeurs. Une file permet à l'utilisateur d'ajouter un nouvel élément et de retirer un élément en suivant le principe *FIFO* (First In First Out).

L'interface contient les fonctions suivantes :

- ▶ `send(f, v)` : ajoute la valeur `v` à la file `f`.
- ▶ `receive(f, v)` : extrait la valeur la plus ancienne de la file et la retourne.
- ▶ `size(f)` : retourne la taille de `f`.
- ▶ `is_empty(f)` : retourne une valeur booléenne indiquant si la pile est vide ou pas.

D'autres types abstraits

- ▶ Arbre binaire
- ▶ Graphe
- ▶ File de priorité
- ▶ Dictionnaire
- ▶ ...

Exercices

Les types abstraits peuvent être implémentés en bien des manières.

1. Implémenter une pile et une file à l'aide de vecteurs. Créer le type structuré ainsi que les fonctions correspondant à ce type abstrait et son implémentation par vecteur.
2. Implémenter une pile et une file à l'aide de listes liées. Créer le type structuré ainsi que les fonctions correspondant à ce type abstrait et son implémentation par liste liée.