

Structures de données abstraites

Sandrine Vial
`sandrine.vial@uvsq.fr`

Octobre 2020

Structures de Données Abstraites

- ▶ Mise en œuvre d'un ensemble dynamique
- ▶ Définition de données (**structuration** et **propriétés**)
- ▶ Définition des opérations pour **manipuler** les données.

Quelques structures classiques

1. Pile *" Dernier arrivé Premier Servi"*
2. **File** *"Premier arrivé Premier Servi"*
3. Tables de hachage *Généralisation des tableaux*
4. Dictionnaire *Et si l'index était une chaîne de caractères ?*
5. Tas *Je sais où est le maximum (ou le minimum)*
6. Files de priorité *On suit les priorités*
7. Arbres *Et si on mémorisait les liens entre les éléments ?*
8.

Une File

Définition

Analogie avec une file d'attente :

FIFO (First In First Out ou **Premier Arrivé Premier Servi**)

- ▶ On rajoute un élément à la fin de la file
- ▶ On supprime l'élément qui est en tête de file.

Mise en œuvre

A l'aide d'un tableau (*nombre maximum d'éléments dans la file fixé*)

Type de données

```
Enregistrement File {  
    T[NMAX] : entier;  
    Début : entier;  Indice de l'élément la plus ancien de la file  
    Fin : entier;    Indice du prochain élément à insérer dans la file  
}
```

Mise en œuvre : un tableau

Algorithme 1 La file est-elle vide ?

FileVide(f : File) : booléen

▷ *Entrée : F (une file)*

▷ *Sortie : vrai si la file est vide, faux sinon.*

 Debut

 si ($f.\text{Début} = f.\text{Fin}$)

 retourner vrai ;

 sinon

 retourner faux ;

 fin si

Fin

Complexité : $O(1)$

Mise en œuvre : un tableau

Algorithme 2 La file est-elle pleine?

FilePleine(f : File) : booléen

▷ *Entrée* : f (une file)

▷ *Sortie* : vrai si la file est pleine, faux sinon.

Debut

si ($f.\text{Début} = (f.\text{Fin} + 1) \bmod N\text{MAX}$)

retourner vrai ;

sinon

retourner faux ;

fin si

Fin

Complexité : $O(1)$

Mise en œuvre : un tableau

Algorithme 3 Insertion d'un élément

Insertion(f : File, elt : entier)

▷ *Entrée* : f (une file) et elt (un entier)

▷ *Sortie* : la file f dans laquelle elt a été inséré

 Debut

 si ($FilePleine(f) = \text{faux}$)

$f.T[f.Fin] \leftarrow elt$;

$f.Fin \leftarrow (f.Fin + 1) \bmod NMAX$;

 sinon

 Afficher un message d'erreur

 fin si

Fin

Complexité : $O(1)$

Mise en œuvre : un tableau

Algorithme 4 Suppression d'un élément

Suppression(f : File) : entier

▷ *Entrée* : f (une file)

▷ *Sortie* : renvoie l'élément le plus ancien de la file f et supprime l'élément de la file

▷ *Variable locale* :

elt : entier ;

Début

si (FileVide(f) = faux)

elt \leftarrow $f.T[f.Début]$;

$f.Début \leftarrow (f.Début + 1) \bmod NMAX$;

retourner elt ;

sinon

Afficher un message d'erreur

fin si

Fin

Complexité : $O(1)$