

# *Algorithmique et Structures de Données II*

## *CH4: Les piles et les files*

ENSEIGNANT: FETHI MGUIS

Sections: LFSI1/LARI1

A.U: 2016/2017

## 1 Introduction

Les notions de pile et de file sont deux stratégies de manipulation des structures de données regroupant un ensemble de données tel que les tableaux et les listes chaînées.

## 2 La notion de pile

### 2.1 Caractéristiques

Lorsqu'une structure de données est manipulée selon la stratégie Pile, alors les deux opérations d'ajout et de récupération se font uniquement en tête de la pile, appelé aussi le sommet de la pile. Autrement dit, le seul élément accessible est le premier élément.

### 2.2 Principales opérations

Dans cette section on présente un aperçu sur les principales opération sur une pile. Pour développer ces module, on a choisit de manipuler une liste simplement chaînée selon la stratégies pile.

#### 2.2.1 Structures nécessaires

Cellule: <b>Enregistrement</b>   Val : <b>Entier</b>   Suiv : <b>*Cellule</b> <b>Fin Enreg</b> Pile: <b>Enregistrement</b>   Sommet : <b>*Cellule</b> <b>Fin Enreg</b>
--

### 2.3 Initialisation d'une pile vide

<b>Procédure</b> Init( P : <b>*Pile</b> ) <b>Début</b>   P→Sommet←NULL <b>Fin</b>
--

Algorithme 1: Initialisation d'une pile vide

### 2.4 Vérification si une pile est vide

<b>Fonction</b> Vide( P : <b>Pile</b> ) : <b>Booleen</b> <b>Début</b>   Retourner(P.Sommet=NULL) <b>Fin</b>
--

Algorithme 2: Vérification si une pile est vide

## 2.5 Ajout d'un élément : Empiler

L'opération d'ajout d'un élément à une pile est appelée **Empiler**.

```
Procédure Empiler( P : *Pile; X : Entier)
Var
  q : *Cellule
Début
  q ← Allouer(1)
  q → Val ← X
  q → Suiv ← P → Sommet
  P → Sommet ← q
Fin
```

Algorithme 3: Ajout d'un élément : Empiler

## 2.6 Récupération d'un élément de la pile : Dépiler

L'opération de récupération d'un élément de la pile consiste à récupérer la valeur du sommet en le supprimant de la pile. Cette opération est appelée **Dépiler**. La fonction dépiler ne peut être appelée que lorsque la pile est non vide

```
Fonction Dépiler( P : *Pile) : Entier
Var
  q : *Cellule X : Entier
Début
  X ← P → Sommet → Val
  q ← P → Sommet
  P → Sommet ← P → Sommet → Suiv
  Libérer(q)
  Retourner(X)
Fin
```

Algorithme 4: Récupération d'un élément de la pile : Dépiler

# 3 La notion de file

## 3.1 Caractéristiques

Lorsqu'une structure de données est manipulée selon la stratégie File, alors l'opération d'ajout n'est possible qu'en queue de la file et la récupération d'un élément n'est possible qu'en tête de la file. Autrement dit, les seuls éléments accessibles sont le premier élément pour la récupération et la queue pour l'ajout. D'où la nécessité de sauvegarder les adresses du premier et dernier élément d'une file.

## 3.2 Principales opérations

Dans cette section on présente un aperçu sur les principales opération sur une file. Pour développer ces module, on a choisit de manipuler une liste simplement chaînée selon la stratégies file.

### 3.2.1 Structures nécessaires

```
Cellule: Enregistrement
  Val : Entier
  Suiv : *Cellule
Fin Enreg
Pile: Enregistrement
  Tête : *Cellule
  Queue : *Cellule
Fin Enreg
```

### 3.3 Initialisation d'une File vide

```
Procédure Init( F : *File))  
Début  
  F→Tête←NULL  
  F→Queue←NULL  
Fin
```

Algorithme 5: Initialisation d'une file vide

### 3.4 Vérification si une file est vide

```
Fonction Vide( F : File)) : Booleen  
Début  
  Retourner(F.Tête=NULL)  
Fin
```

Algorithme 6: Vérification si une file est vide

### 3.5 Ajout d'un élément : Enfiler

L'opération d'ajout d'un élément à une file est appelée **Enfiler**.

```
Procédure Enfiler( F : *File; X : Entier)  
Var  
  q : *Cellule  
Début  
  q←Allouer(1)  
  q→Val←X  
  q→Suiv←NULL  
  Si (Vide(*F)=Vrai) Alors  
    F→Tête←q  
  Sinon  
    P→Queue→Suiv←q  
  Fin Si  
  P→Queue←q  
Fin
```

Algorithme 7: Ajout d'un élément : Enfiler

### 3.6 Récupération d'un élément de la file : Défiler

L'opération de récupération d'un élément de la file consiste à récupérer la valeur la tête en la supprimant de la file. Cette opération est appelée **Défiler**. La fonction dépiler ne peut être appelée que lorsque la file est non vide

```
Fonction Défiler( F : *File) : Entier  
Var  
  q : *Cellule X : Entier  
Début  
  X←F→Tête→Val  
  q←F→Tête  
  P→Tête←F→Tête→Suiv  
  Si (F→Tête=NULL) Alors  
    F→Queue←NULL  
  Fin Si  
  Libérer(q)  
  Retourner(X)  
Fin
```

Algorithme 8: Récupération d'un élément de la file : Défiler