Les types abstraits de données (TAD)

LES STRUCTURES LINÉAIRES

Piles & Files

Algorithmique & structures de Données

Liste Linéaire

Soit T un type de données.

Une liste linéaire de valeurs de type T est une suite de n ($n \geq 0$)

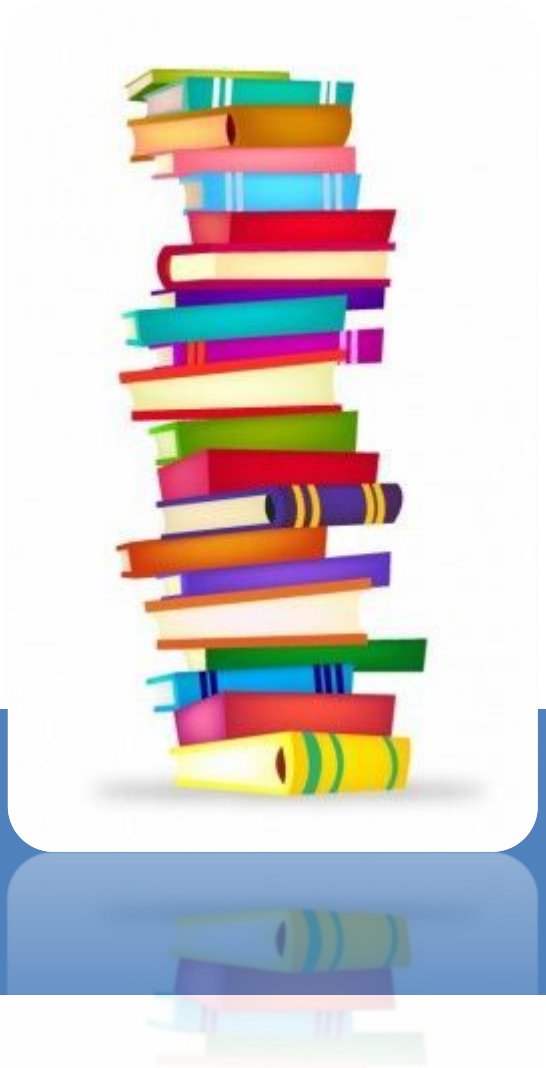
valeurs $v_1, ..., v_n$ rangées de façon à ce que :

- si $n > 0$, v_1 est la première valeur dans ce rangement et v_n est la dernière,
- si $1 < k < n$, v_k est précédée par la valeur v_{k-1} et suivie par la valeur v_{k+1}

2 exemples de Liste Linéaire : PILE et FILE

Les types abstraits de données (TAD)

LES PILES …

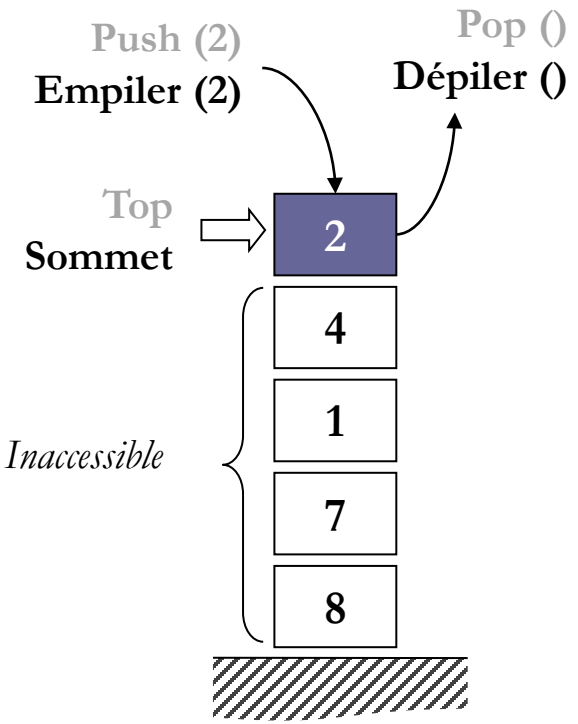


La Pile

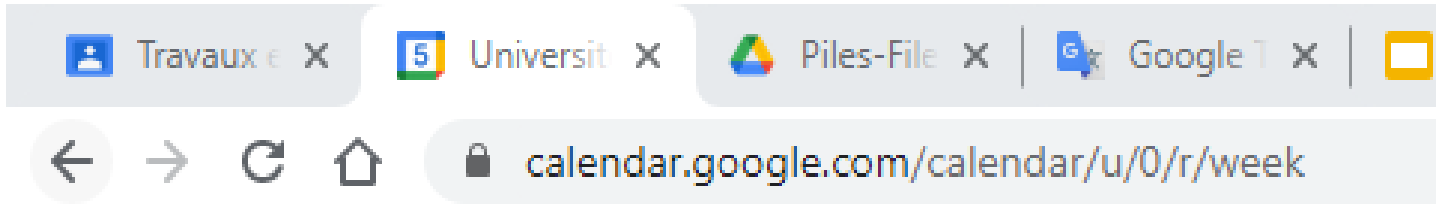
Une pile est une liste linéaire gérée par l'ordre **LIFO**

Dans une pile les insertions et les suppressions sont effectuées au sommet :

- la première valeur est le sommet de la pile,
- Empiler une valeur c'est l'insérer au début de la pile,
- Dépiler une valeur, si la pile est non vide, c'est sélectionner la première valeur de la supprimer de la pile.

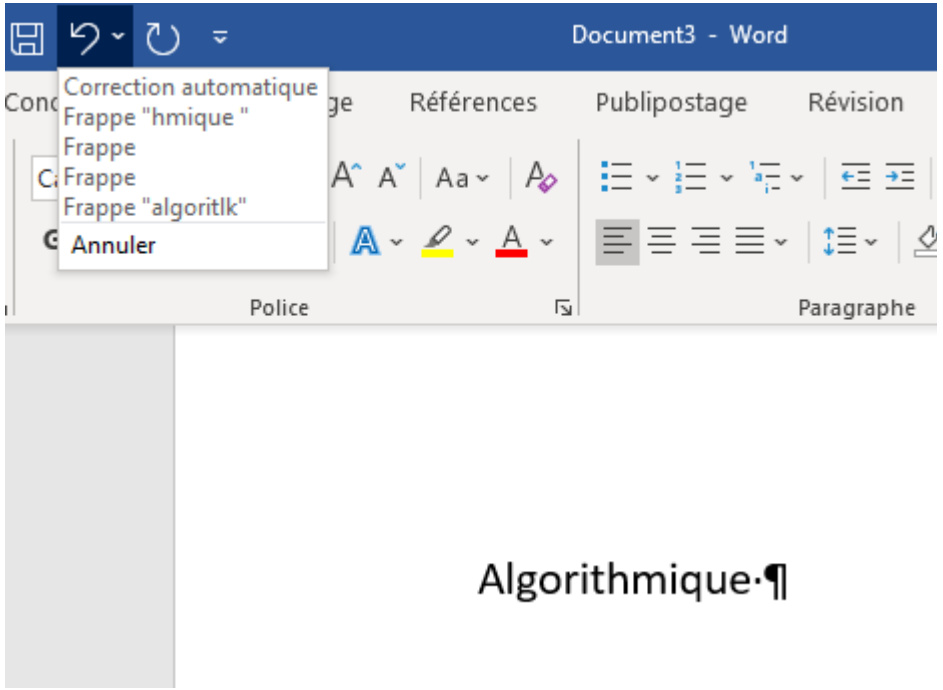


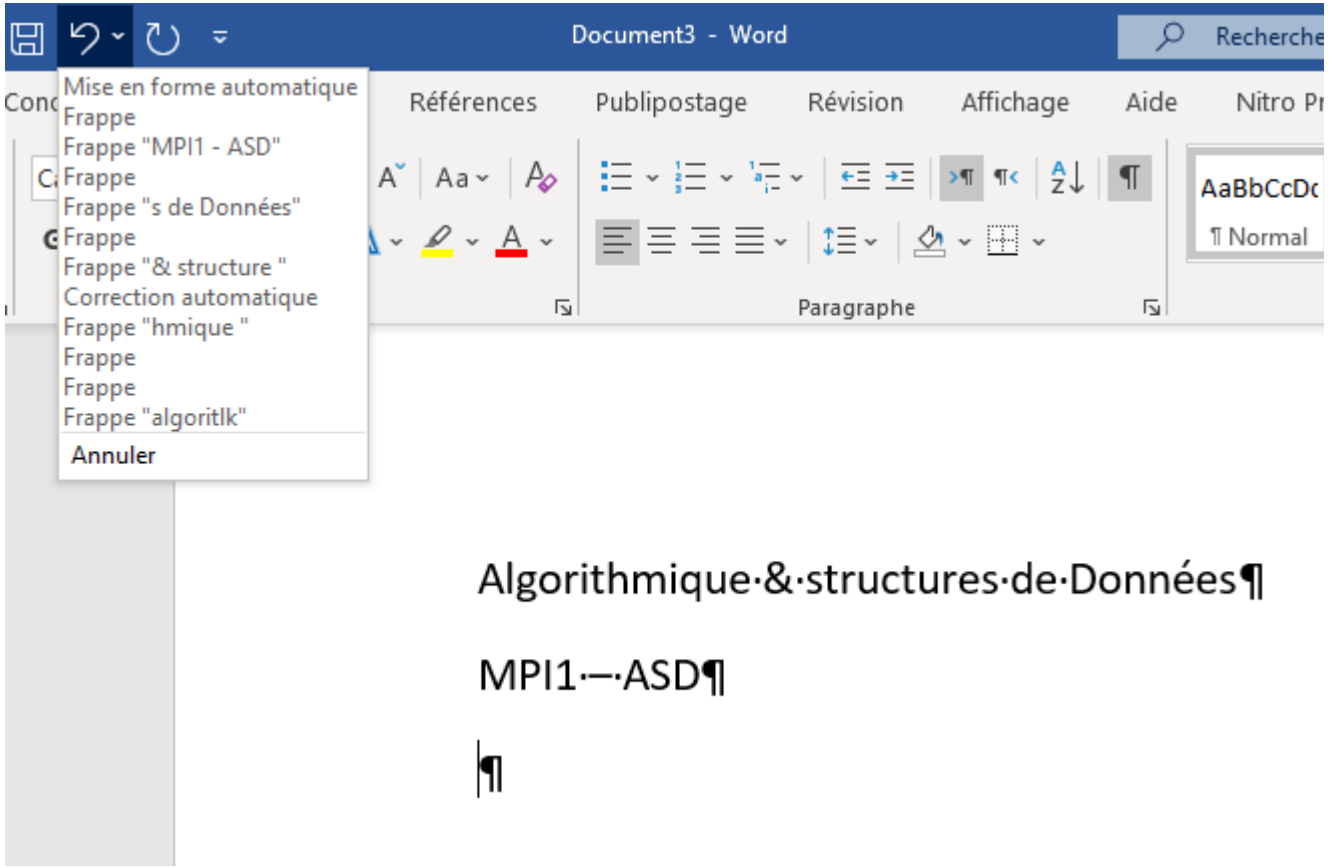
La Pile : application



- https://calendar.google.com/calendar/u/0/r/week
- https://classroom.google.com/u/0/w/MTgxODY2MzA3MTYz/t/all
- https://fr.wikipedia.org/wiki/Algorithme
- www.google.fr

La Pile : application





Opérations primitives sur les piles

On note :

Pile(T) : le type des piles de valeurs de type T

La Pile (Stack)

LIFO : last in, first out

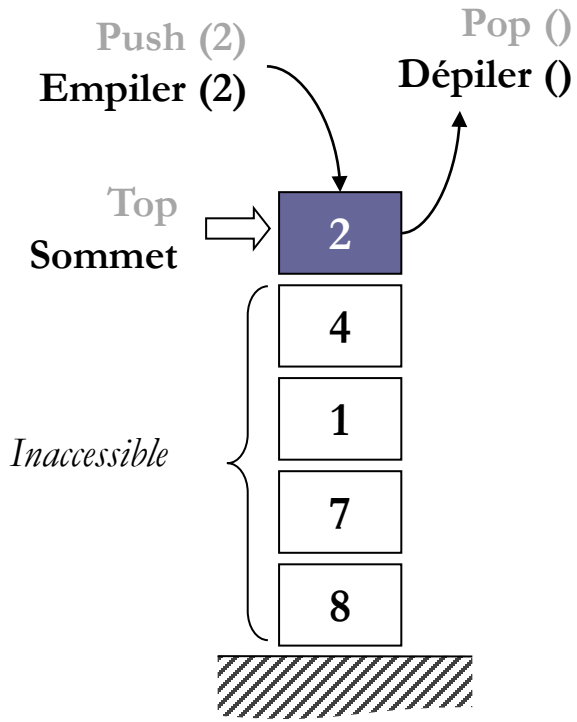
Opération	Paramètres	Le profil de l'opération		Spécification de l'opération	
		domaine	co-domaine	prés-conditions	post-condition
Init_P : *(Empty)*	**(var P : Pile)** *Procédure*	Pile →	Pile	~~	- Pile résultat : pile initialisée,
Vide : *(Empty)*	**(P : Pile) : booléen** *Fonction*	Pile →	booléen	~~	- Vrai si la pile est vide, - Faux si la pile n'est vide,
Pleine* : *(Full)*	**(P : Pile) : booléen** *Fonction*	Pile →	booléen	~~	- Vrai si la pile est pleine, - Faux si la pile n'est pleine,
Sommet : *(Top)*	**(P : Pile) : valeur** *Fonction*	Pile →	valeur	Pile non vide	- valeur du sommet de la Pile,
Empiler : *(Push)*	**(var P : Pile, x : valeur)** *Procédure*	Pile, valeur →	Pile	Pile non pleine	- Pile résultat : pile donnée + valeur comme sommet,
Dépiler : *(Pop)*	**(var P : Pile) : valeur** *Fonction*	Pile →	Pile, valeur	Pile non vide	- Pile résultat : pile donnée sans sommet, - Valeur du sommet,

Représentation d'une pile

Représentation Statique

La pile sera représentée par un vecteur tel que le sommet se trouvera à la **dernière position** du tableau

```
PILE = enreg
    stack : tableau[1..MAX] de valeur
    top : entier
 Fenreg
```



Représentation d'une pile

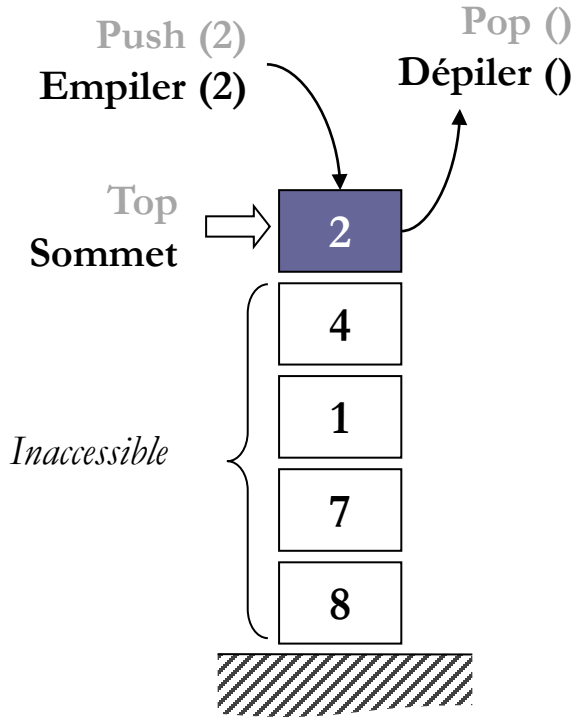
Opérations

Représentation Statique

```
Init-pile(var P:PILE)
Début
    P.top := 0
Fin

pile-vide(P : PILE) : booléen
Début
     pile-vide := (P.top = 0)
Fin

pile-pleine(P : PILE) : booléen
Début
     pile-pleine := (P.top = MAX)
Fin
```

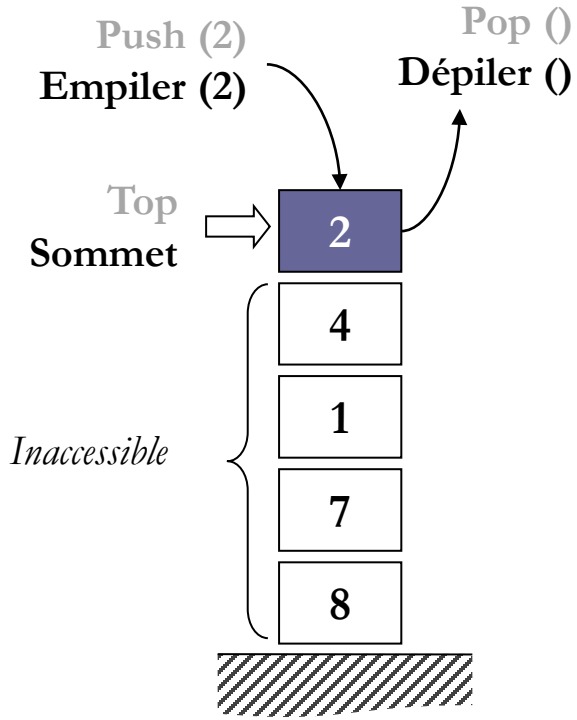


Représentation d'une pile

Opérations

Représentation Statique

```
empiler(var P : PILE; x : valeur)
Début
    P.top = P.top + 1
    P.stack[P.top] := x
Fin
```

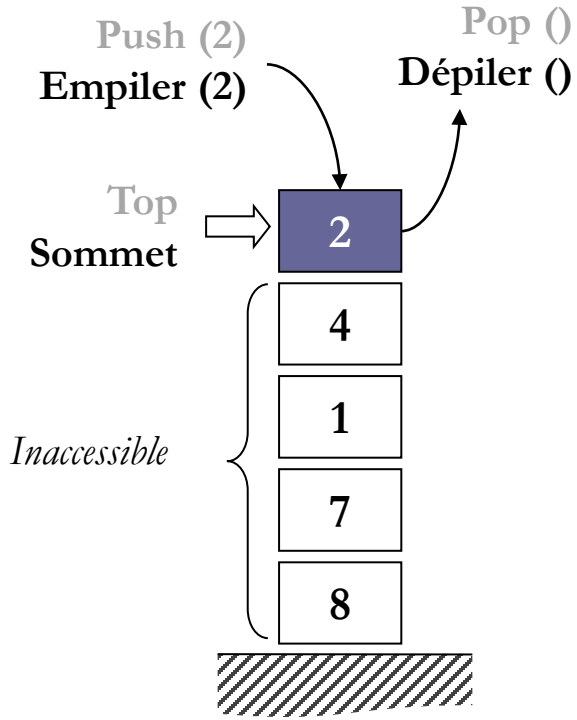


Représentation d'une pile

Opérations

Représentation Statique

```
dépiler(var P : PILE) : valeur
Début
    dépiler := P.stack[P.top]
    P.top = P.top - 1
Fin
```

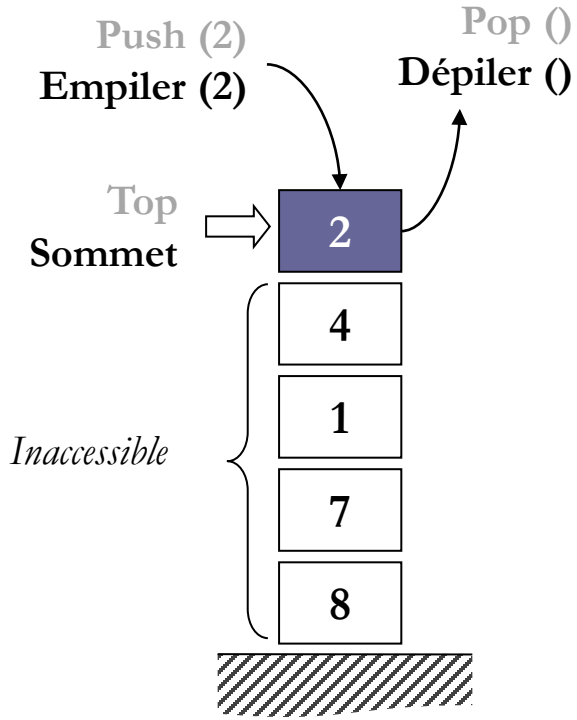


Représentation d'une pile

Opérations

Représentation Statique

```
sommet(P : PILE) : valeur
Début
     sommet := P.stack[P.top]
Fin
```



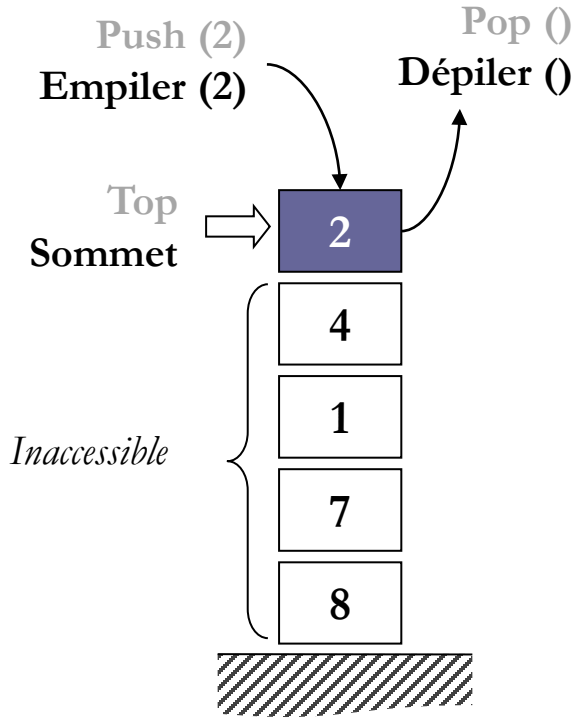
Représentation d'une pile

Représentation Dynamique

La pile sera représentée par une liste chaînée simple tel que le sommet de la pile sera le **premier élément** de la liste chaînée (tête de liste).

```
noeud = enreg
    Val   : valeur
    Next  : ^noeud
 Fenreg

LS1  :  ^noeud
```



Représentation d'une pile

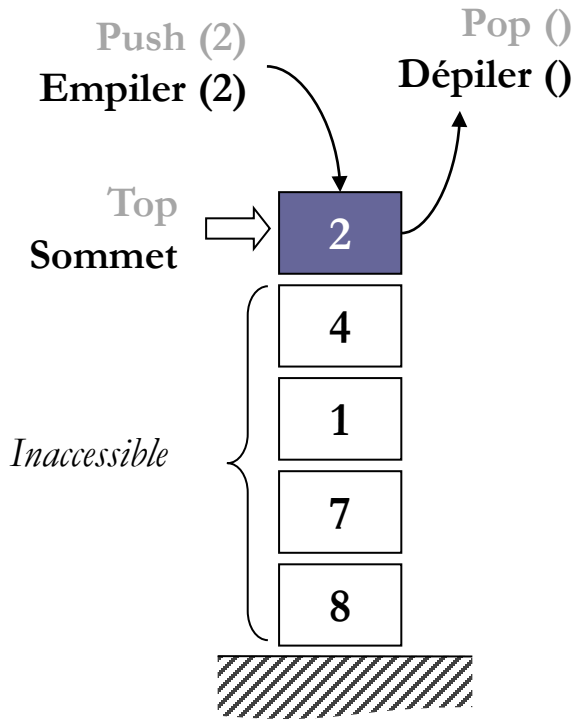
Opérations

Représentation Dynamique

```
Init-pile(var P:PILE)
Début
    P.top := NIL
Fin

pile-vide(P : PILE) : booléen
Début
    pile-vide := (P.top = NIL)
Fin

pile-pleine(P : PILE) : booléen
Début
    ----
Fin
```

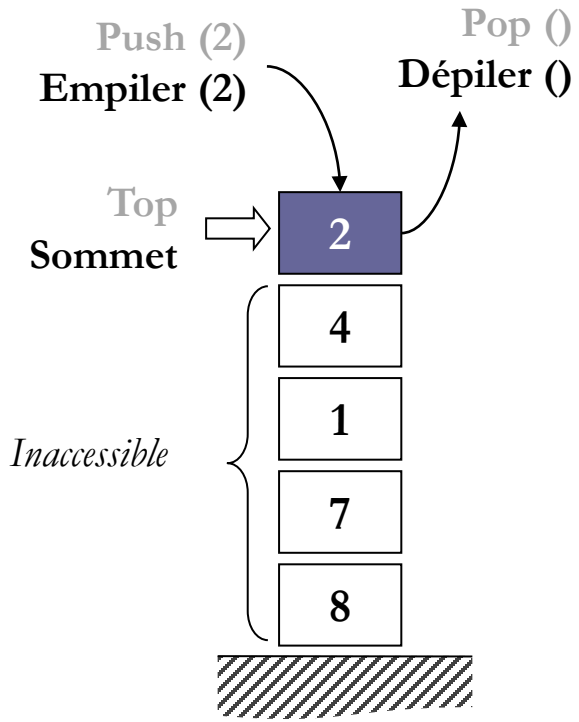


Représentation d'une pile

Opérations

Représentation Dynamique

```
empiler(var P : PILE; x : valeur)
Début
    insère_tête(P,x)
Fin
```

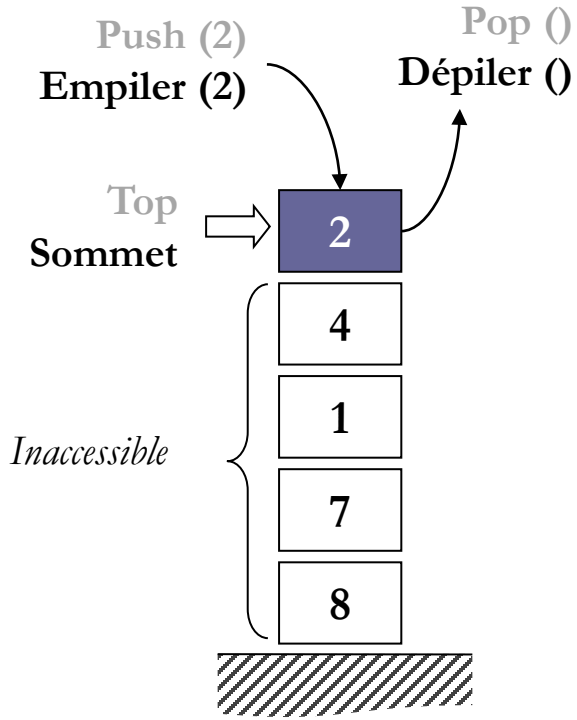


Représentation d'une pile

Opérations

Représentation Dynamique

```
dépiler(var P : PILE) : valeur
Début
    dépiler := supprime_tête(P)
Fin
```

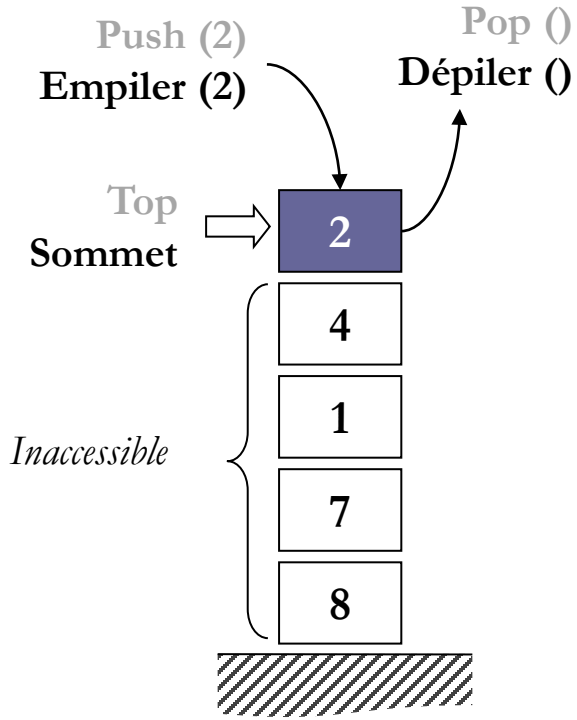


Représentation d'une pile

Opérations

Représentation Dynamique

```
sommet(P : PILE) : valeur
Début
     sommet := P^.val
Fin
```



Les types abstraits de données (TAD)

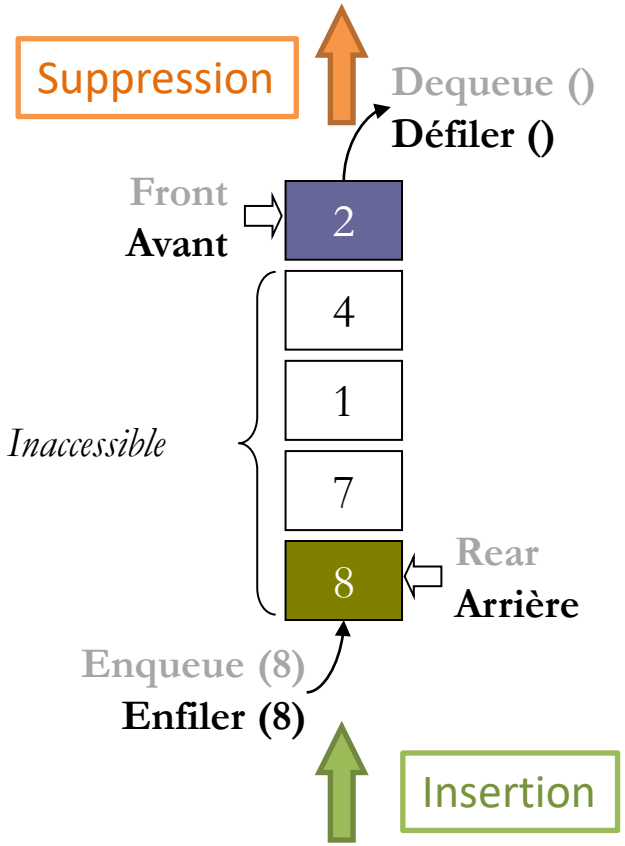
LES FILES …

La File

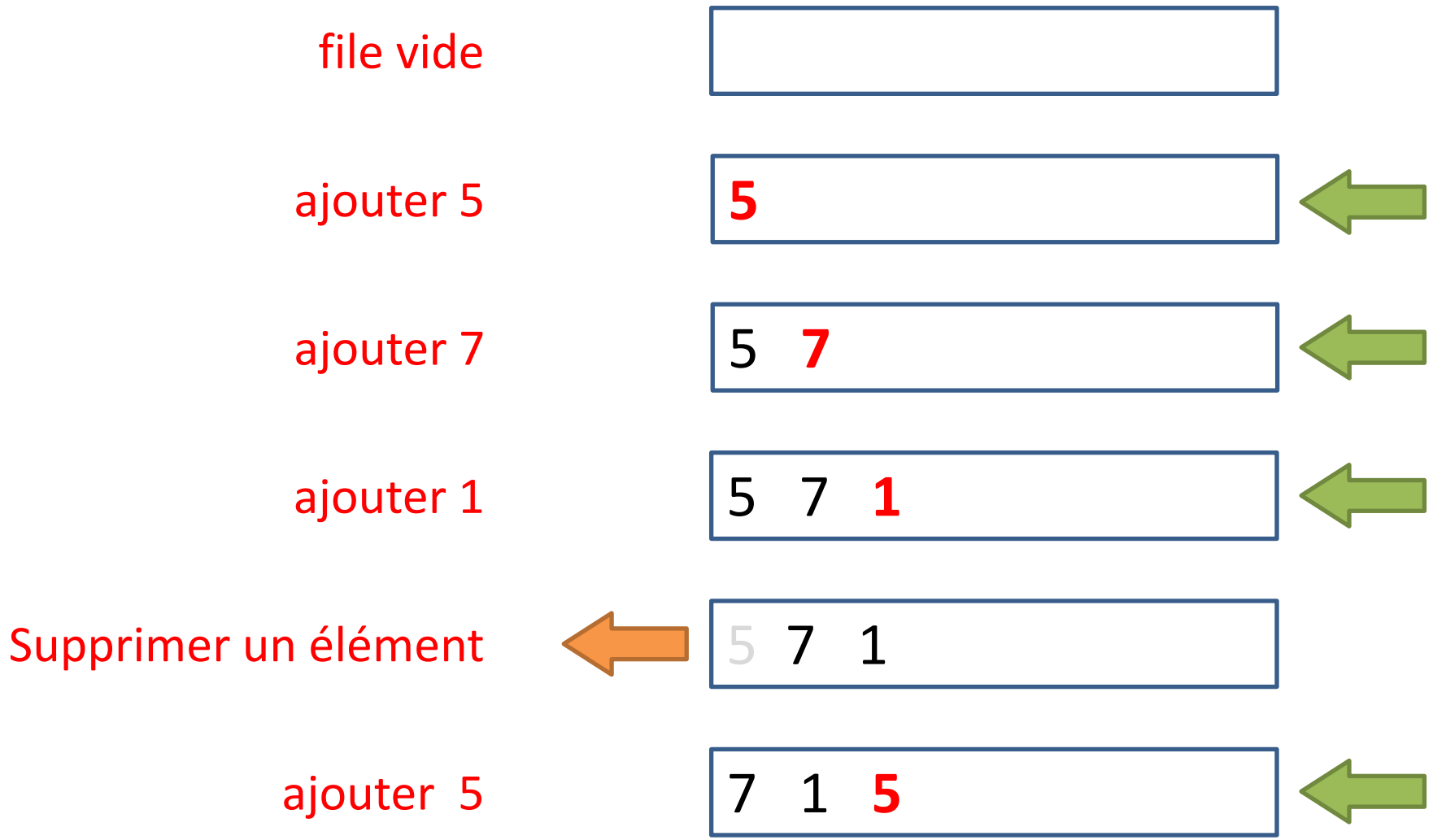
Une file est une liste linéaire gérée par l'ordre **FIFO**

Dans une file les insertions et les suppressions sont effectuées aux extrémités :

- **Enfiler** une valeur c'est l'insérer à la fin de la file,
- **Défiler** une valeur, si la file est non vide, c'est supprimer une valeur au début de la file.



La File


file vide
ajouter 5
5
ajouter 7
5 7
ajouter 1
5 7 1
Supprimer un élément
5 7 1
ajouter 5
7 1 5

Opérations primitives sur les files

On note :

File(T) : le type des files de valeurs de type T

La File (Queue)

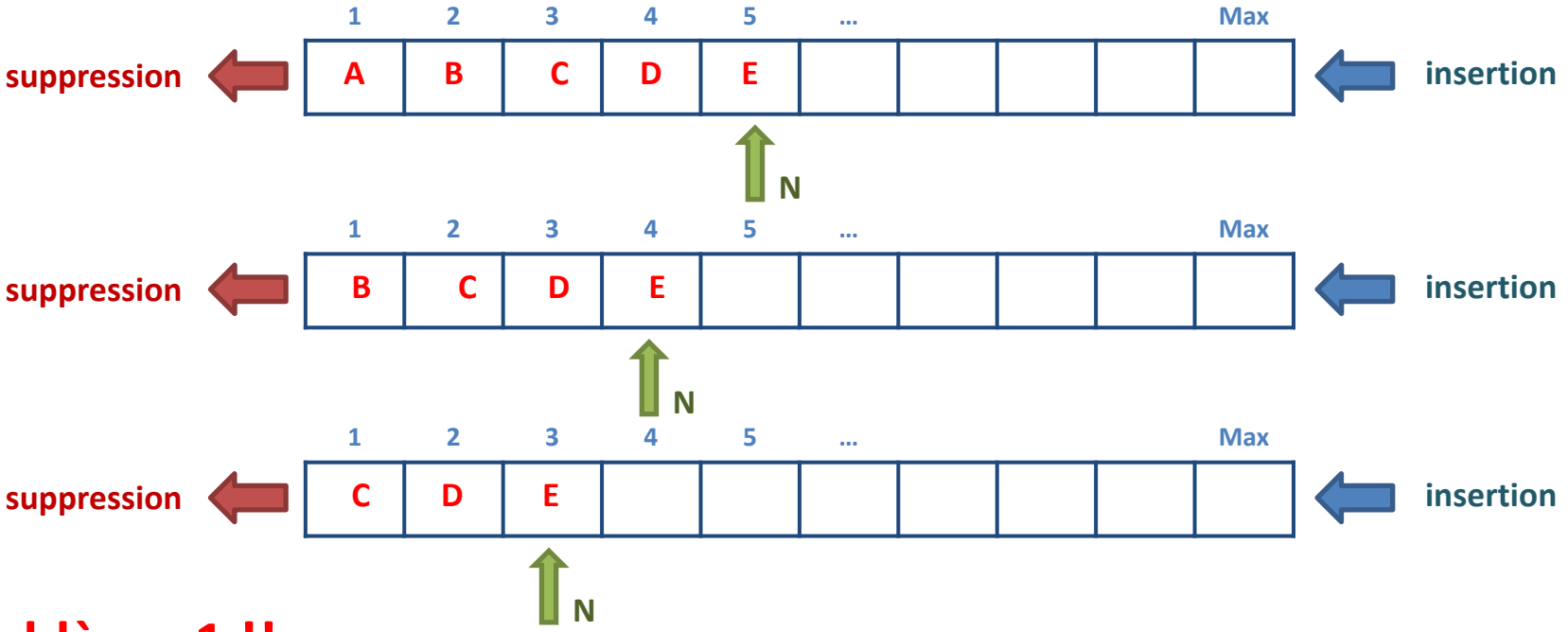
FIFO: first in, first out

Opération	Paramètres	Le profil de l'opération		Spécification de l'opération	
		domaine	co-domaine	prés-conditions	post-condition
Init_F : *(Empty)*	**(var F : File)** *Procédure*	File →	File	~~	- File résultat : file initialisée,
Vide : *(Empty)*	**(F : File) : booléen** *Fonction*	File →	booléen	~~	- Vrai si la file est vide, - Faux si la file n'est pas vide,
Pleine* **:** *(Full)*	**(F : File) : booléen** *Fonction*	File →	booléen	~~	- Vrai si la file est pleine, - Faux si la file n'est pas pleine,
Enfiler : *(Enqueue)*	**(var F : File, x : valeur)** *Procédure*	File, valeur →	File	File non pleine	- File résultat : File donnée + valeur comme arrière de la file
Défiler : *(Dequeue)*	**(var F : File) : valeur** *Fonction*	File →	File, valeur	File non vide	- File résultat : File donnée sans la valeur avant de la File - Valeur de l'avant de la File

Représentation d'une file

Représentation Statique

La file sera représentée par un tableau de N éléments :



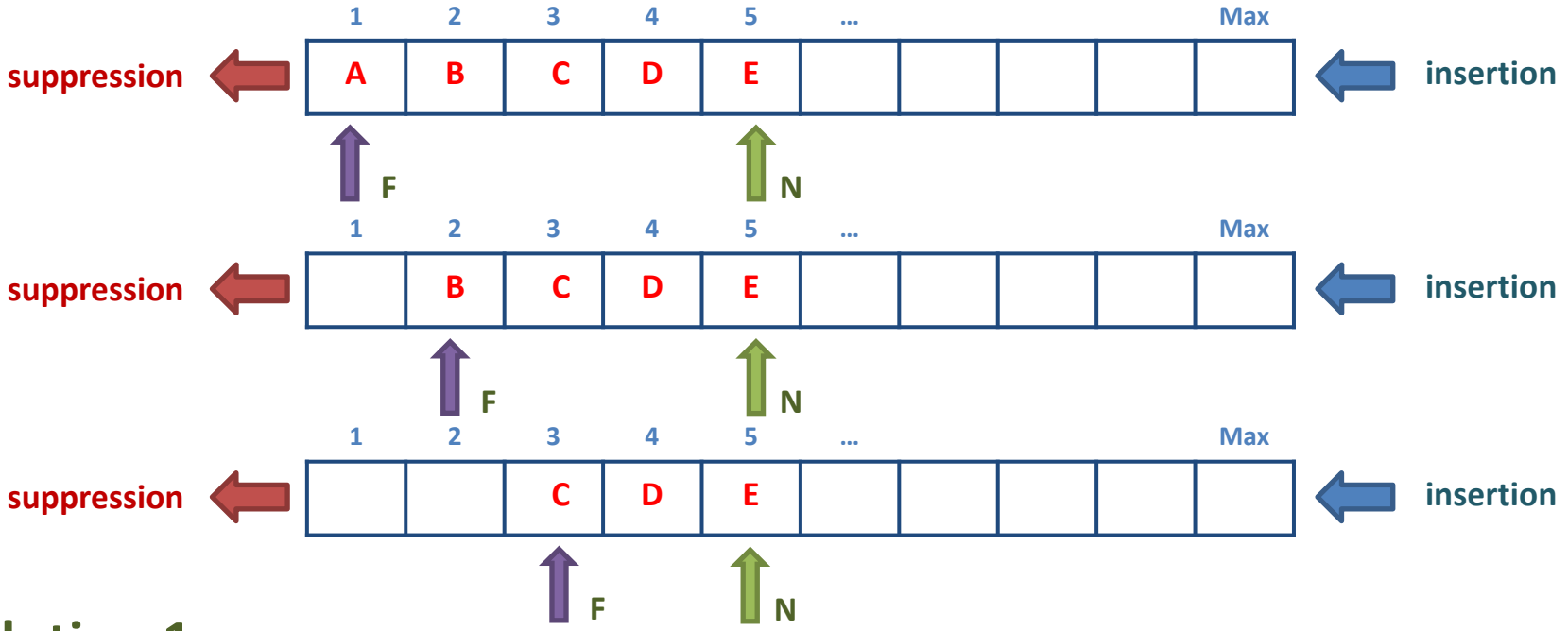
Problème 1 !!

Faire un décalage vers la gauche après chaque suppression

Représentation d'une file

Représentation Statique

La file sera représentée par un tableau de N éléments :



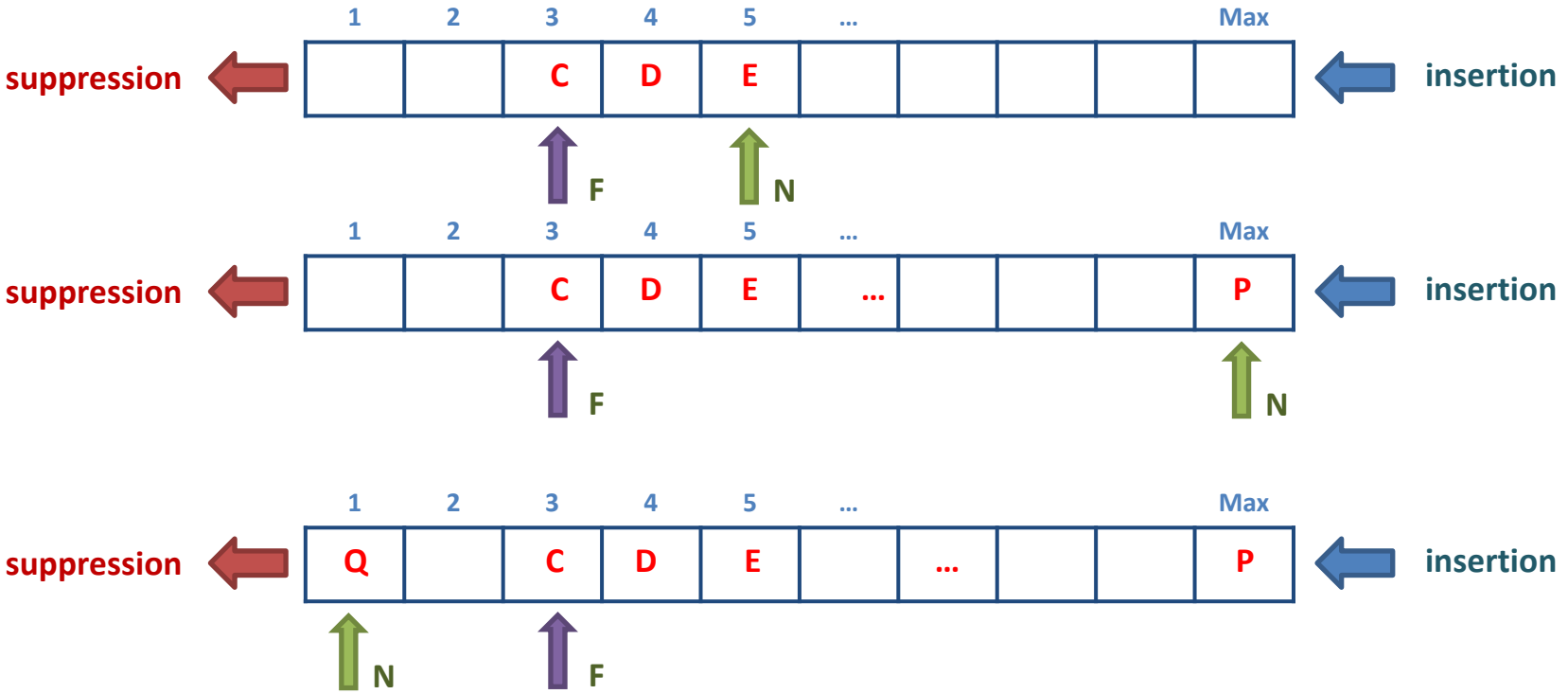
Solution 1

Le 1er élément prend une position dans le domaine 1-Max …

Représentation d'une file

Représentation Statique

La file sera représentée par un tableau de N éléments :



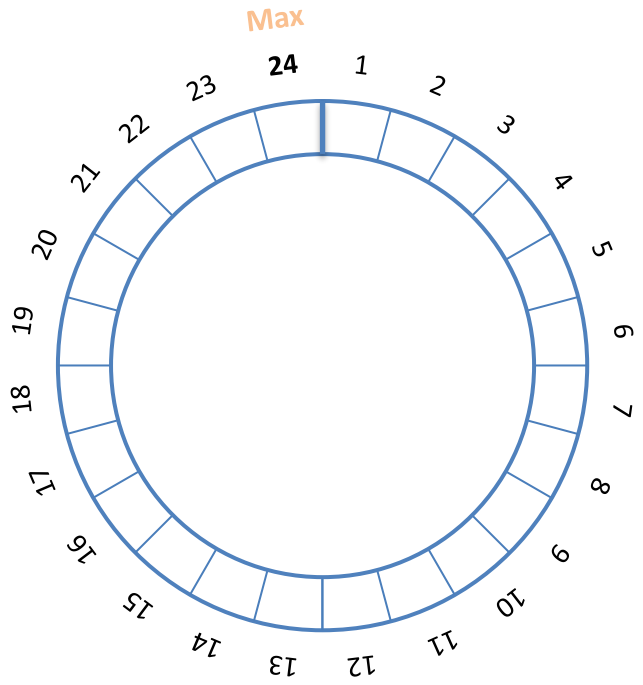
Solution 2 Tableau Circulaire

Représentation d'une file

Représentation Statique

La file sera représentée par un tableau circulaire :

```
FILE = enreg
    fifo : tableau[1..MAX] de valeur
    first : entier
    last : entier
 Fenreg
```

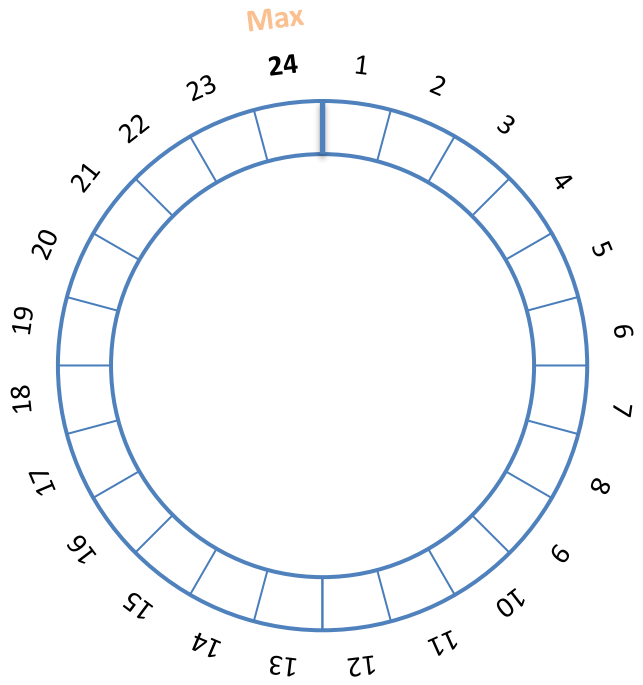


Représentation d'une file

Opérations

Représentation Statique

```
Init-file(var F : FILE)
Début
    F.first := 1
    F.last := 0
Fin
```

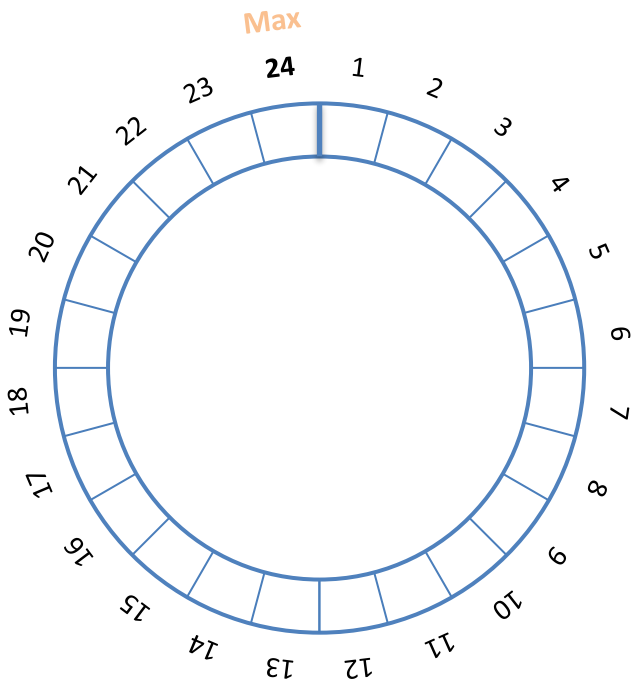


Représentation d'une file

Opérations Représentation Statique

```
file-vide(F : FILE): booléen
Début
     file-vide := (F.First = (F.last mod MAX) + 1)
Fin
```

```
Si F.last < Max Alors
     file-vide := F.First = F.last + 1
siNon
     file-vide := F.First = 1
Fin
```

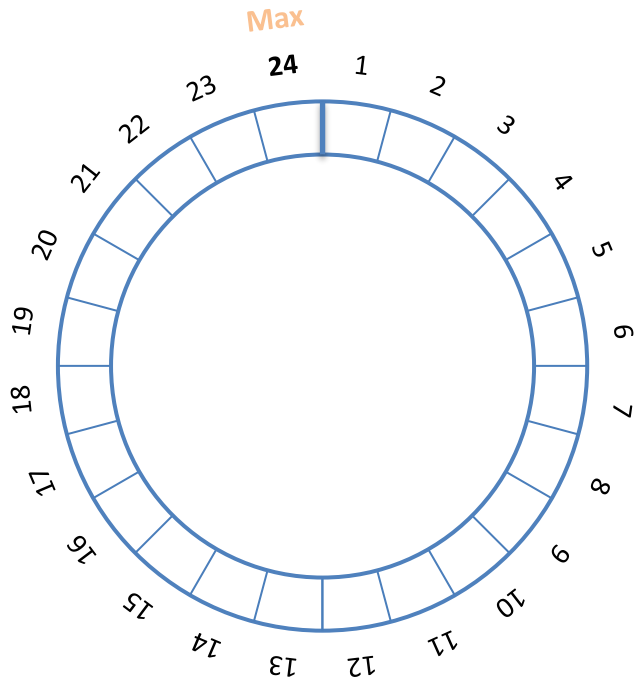


Représentation d'une file

Opérations Représentation Statique

```
file-pleine(F : FILE) : booléen
Début
     file-pleine := (F.First = (F.last + 1) mod MAX + 1)
Fin
```

```
Si F.last + 1 < Max Alors
    file-vide := F.First = (F.last + 1) + 1
siNon
    file-vide := F.First = 1
Fin
```



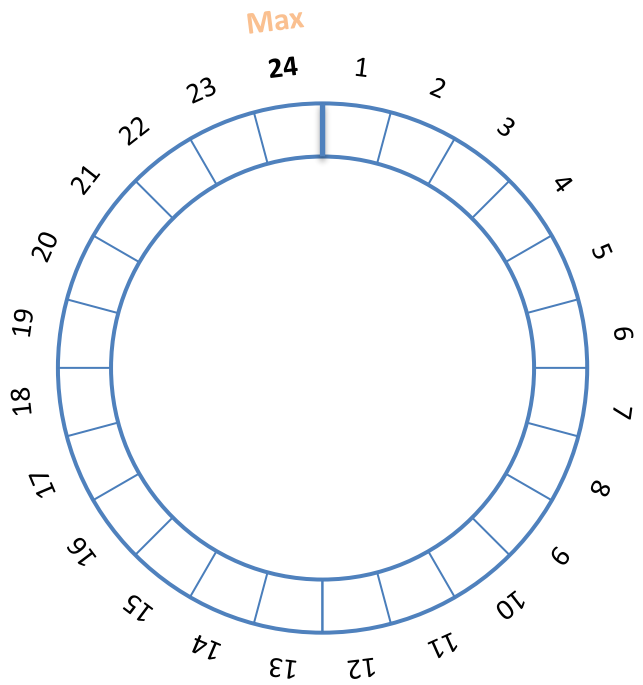
Représentation d'une file

Opérations

Représentation Statique

```
enfiler(var F : FILE; x : valeur)
Début
    F.last = (F.last mod Max) + 1
    F.fifo[F.last]:= x
Fin
```

```
Si F.last < Max Alors
     F.last := F.last + 1
siNon
    F.last := 1
Fin
```



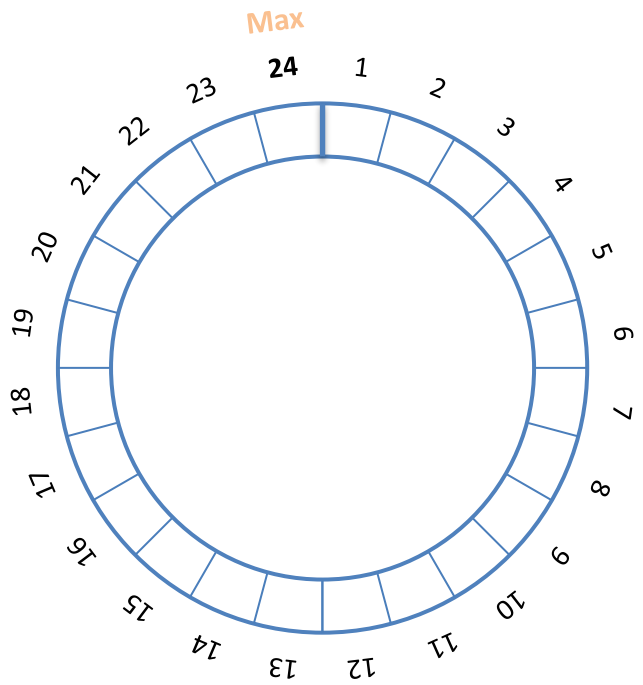
Représentation d'une file

Opérations

Représentation Statique

```
défiler(var F : FILE) : valeur
Début
    défiler := F.fifo[F.first]
    F.first = (F.first mod MAX) + 1
Fin
```

```
Si F.first < Max Alors
     F.first := F.first + 1
siNon
    F.first := 1
Fin
```



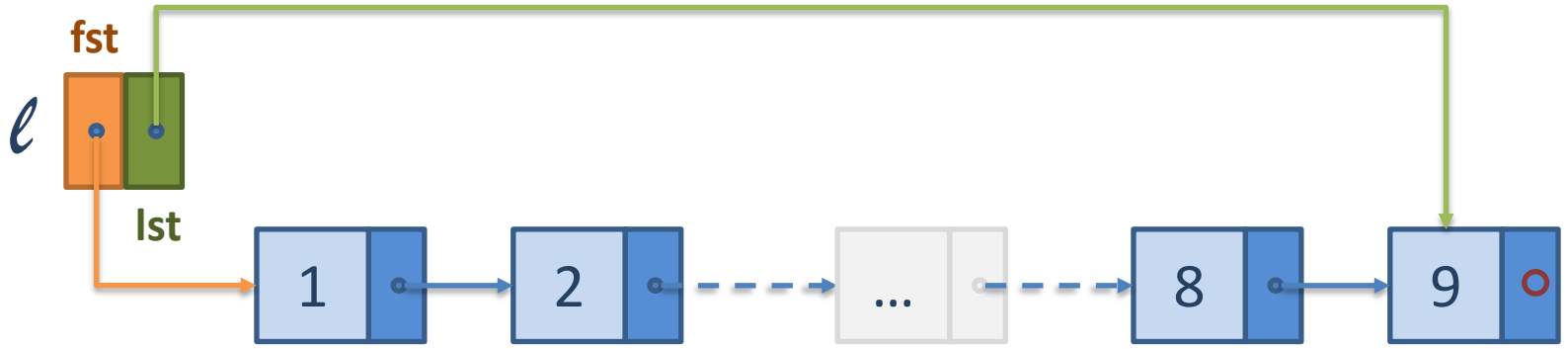
Représentation d'une file

Représentation Dynamique

La file sera représentée par une liste chaînée simple avec accès au premier et au dernier élément.

```
LS2 = enreg
    first : ^nœud
    last : ^nœud
  fenreg

nœud = enreg
    val   :  valeur
    next   : ^nœud
  fenreg
```



Représentation d'une file

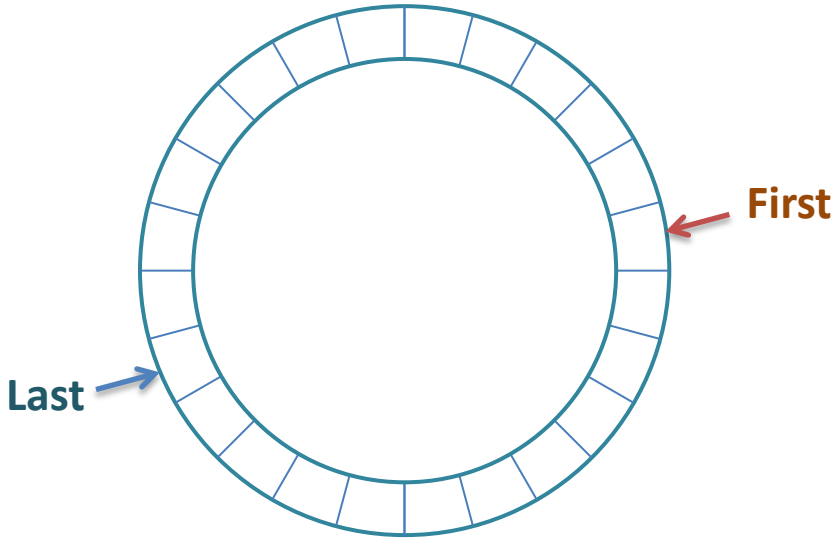
Opérations

Représentation Dynamique

```
Init-file(var F : FILE)
Début
    F.First := NIL
    F.Last := NIL
Fin

file-vide(F : FILE) : booléen
Début
     file-vide := ( F.First = NIL )
Fin

file-pleine(f : PILE) : booléen
Début
     ----
Fin
```



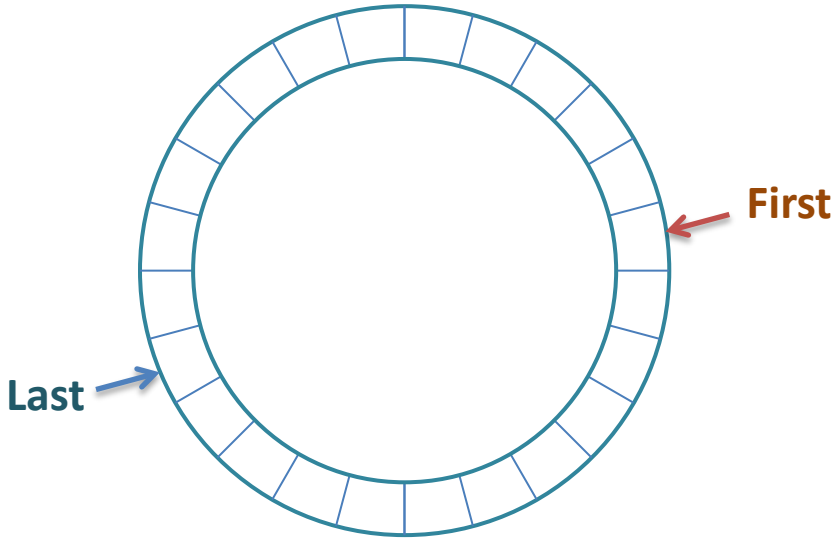
Représentation d'une file

Opérations

Représentation Dynamique

```
enfiler( var F : FILE ; x : valeur )
Début
     insère_queue(F, x)
Fin

défiler(var F : FILE) : valeur
Début
     défiler := supprime_tête(F)
Fin
```



Fin