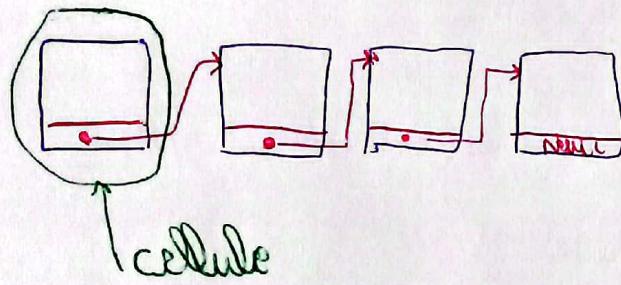


ch3 : Les listes chainées

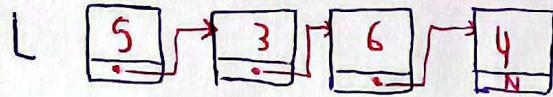
- un tableau est SD permettant de regrouper un nb fini d'élément identiques. Alloués d'une manière contigüe.

- une LS est une SD permettant de regrouper un ensemble d'éléments identiques

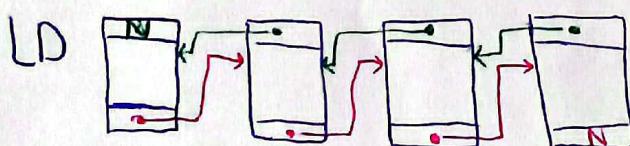


cellule : Enregistrement
Data : type(Entier, reel, Enreg)
suiv : * cellule
fin

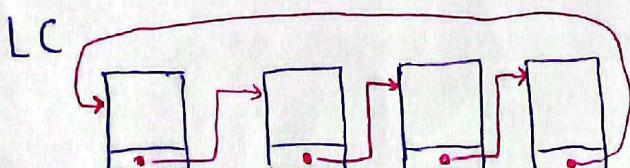
Les différents types de LS



Liste simplement chaînée linéaire

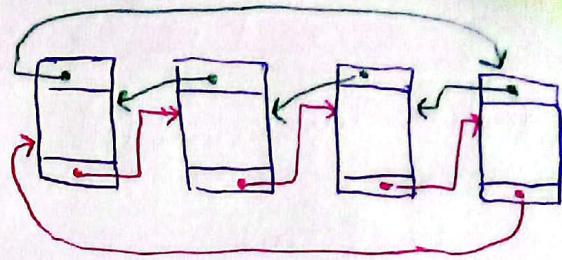


Liste Doublement chaînée linéaire



Liste simplement chaînée circulaire

LDC



I/ Les listes simplement chaînées linéaires

Representation (SD):

Cellule : Enregistrement
Val : type
suiv : * cellule
fin Enreg

Liste : Enregistrement
Tête : * cellule
:

Fin Enreg

Principales opérations:

On considère une liste chaînée d'entiers

cellule : Enregistrement
val : Entier
suiv : * cellule
fin Enreg

Liste : Enregistrement
Tête : * cellule

Fin Enreg

Création d'une liste vide:

Procédure Init (var L : Liste)

Début

L.Tête ← NULL

fin

Determiner la taille d'une liste :

fonction Taille(L; liste) : Entier

var

P : *cellule

Nb : Entier

Debut

Nb \leftarrow 0

P \leftarrow L.Tête

Tantque (P \neq NULL)

Nb \leftarrow Nb + 1

P \leftarrow P \rightarrow suiv

Fin Tantque

Taille \leftarrow Nb

Fin

Creation d'une liste vide :

Cellule : Enregistrement

Val : Entier

Suiv : *cellule

Fin Enreg

Liste : Enregistrement

*Tête : *cellule

Fin Enreg

Procédure Init(L : *Liste)

Debut

L \rightarrow Tête \leftarrow NULL

Fin

Fonction Estvide(L; liste) : boolean

Debut

Estvide \leftarrow (L.Tête = NULL)

Fin

fonction Taille(L; liste) : Entier

var

P : *cellule

nb : entier

Debut

nb \leftarrow 0

P \leftarrow L.Tête

Tantque (P \neq NULL) Faire

nb \leftarrow nb + 1

P \leftarrow P \rightarrow suiv

Fin Tantque

Taille \leftarrow nb

Fin

Procédure Afficher(L; liste)

var

L1 : liste

Debut

Si (Estvide(L) = faux) Alors

Ecrire(L.Tête \rightarrow val)

L1.Tête \leftarrow L.Tête \rightarrow suiv

Afficher(L1)

Fin Si

Fin

Algorithme principal

var

L : liste

Debut

Init(&L)

Si (Estvide(L) = vrai) Alors

L'ajout (Insertion) d'un élément à une liste:

Procédure Ajout-Tête ($L: *liste$, $x: Entier$)

Var

$q: *cellule$

debut

$q \leftarrow \text{Allouer}(1)$

$q \leftrightarrow \text{val} \leftarrow x$

(1) $q \rightarrow \text{suiv} \leftarrow L \rightarrow \text{Tête}$

(2) $L \rightarrow \text{Tête} \leftarrow q$

Fin

Procédure Ajout-Queue ($L: *liste$, $x: Entier$)

Var

$P, q: *cellule$

Debut

si ($\text{EstVide}(*L) = \text{vrai}$) Alors

Ajout-Tête (L, x)

Sinon

$q \leftarrow \text{Allouer}(1)$

$q \rightarrow \text{val} \leftarrow x$

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

$P \leftarrow L \rightarrow \text{Tête}$

Tantque ($P \rightarrow \text{suiv} \neq \text{NULL}$) faire

$P \leftarrow P \rightarrow \text{suiv}$

Fintantque

$P \rightarrow \text{suiv} \leftarrow q$

fin si

Fin

$P_1: L: liste$

$P_3(L)$

$P_4(\&L)$

$P_3: (L: liste)$

$P_2: L: liste$

$P_4(L)$

$P_3(\&L)$

$P_4(L: *liste)$

Procédure Ajout-Pos ($L: *liste$, $x: entier$, $k: entier$)

Var

$P, q: *cellule$

$i: entier$

debut

si ($\text{EstVide}(*L) = \text{vrai}$ ou $k = 1$) Alors

Ajout-Tête (L, x)

$P \leftarrow L \rightarrow \text{Tête}$

$i \leftarrow 1$

Tantque ($i \leq k - 1$) faire

$P \leftarrow P \rightarrow \text{suiv}$

Tantque ($P \rightarrow \text{suiv} \neq \text{NULL}$ et $i \leq k - 2$) faire

$P \leftarrow P \rightarrow \text{suiv}$

$i \leftarrow i + 1$

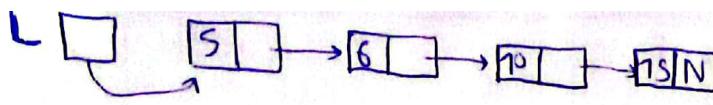
fin tq

$q \leftarrow \text{Allouer}(1)$

$q \leftarrow \text{val} \leftarrow x$

$q \rightarrow \text{suiv} \leftarrow P \rightarrow \text{suiv}$ (1)

$P \rightarrow \text{suiv} \leftarrow q$ (2)



- X la liste est vide ou $x \leq L \rightarrow \text{tête} \rightarrow \text{val}$
 inserer l'@ de l'élément après lequel
 on va ajouter x

Procédure Ajout-triée (L: *liste, x: entier)

var

P, q : *cellule

debut

si (EstVide (*L)) = vrai ou
 $x \leq L \rightarrow \text{tête} \rightarrow \text{val}$ alors

Ajout-tête (L, x)

sinon sinon

P $\leftarrow L \rightarrow \text{tête}$

tant que ($p \rightarrow \text{suiv} \neq \text{NULL}$) et

P $\rightarrow \text{suiv} \rightarrow \text{val} < x$ faire

P $\leftarrow p \rightarrow \text{suiv}$
 fintantque

q $\leftarrow \text{Allouer}(1)$

q $\rightarrow \text{val} \leftarrow x$

q $\rightarrow \text{suiv} \leftarrow p \rightarrow \text{suiv}$

P $\rightarrow \text{suiv} \leftarrow q$

finsi

fin

Redéredre d'un elt d'une liste:

fonction Redr (L: Liste, x: Entier):

*cellule

var

P : *cellule

debut

P $\leftarrow L \cdot \text{tête}$

tant que ($p \neq \text{NULL}$ et $p \rightarrow \text{val} > x$) faire

P $\leftarrow p \rightarrow \text{suiv}$

fintantque

Redr $\leftarrow p$

fin

Ex:

Écrire une procédure permettant d'éclater une liste d'entiers en deux liste tel que la première contiendra les éléments négatifs dans l'ordre inverse et sans redondance et la deuxième liste contiendra les éléments positifs dans le même ordre de la liste initiale et sans redondance aussi.

Procédure éclater (L: liste, L1: *liste,
 L2: *liste)

Var:

p : *cellule

debut

Init(L1)

Init(L2)

P $\leftarrow L \cdot T \cdot \text{tête}$

```

Tantque ( p <> NULL ) faire
  si ( p → val < 0 ) alors
    si ( Rech( *L1, p → val ) = faux )
      alors
        Ajout-Tête ( L1, p → val )
      finsi
    sinon
      si ( Rech( *L2, p → val ) = faux )
        alors
          Ajout-queue ( L2, p → val )
        finsi
      finsi
    P ← p → suiv
  fintantque
fin

```

La suppression d'un élément à partir d'une liste:

- déterminer l'adresse de l'élément à supp (dans q)
- effacement
- Drainage
 - { déterminer p; l'adresse de l'elt précédent de q
 - [MAJ des champs]
 - $P \rightarrow suiv \leftarrow p \rightarrow suiv \rightarrow suiv$
 - q
- libérer l'espace qui a été alloué à l'element libérer (q)

```

Procedure supp-Tête ( L : * liste )
var
  q : * cellule
debut
  si ( estvide (*L) = vrai ) Alors
    Ecrire ("supp imp, liste vide")
  sinon
    q ← L → tête
    L → tête ← L → tête → suiv
    q → suiv ← NULL
    libérer (q)
  finsi
fin

Procedure supp-Queue ( L : * liste )
var
  P, q : * cellule
debut
  si ( Estvide (*L) ou Taille (*L)-1 ) Alors
    supp-Tête ( L )
  sinon
    P ← L → tête
    Tantque ( P → suiv → suiv <> NULL ) faire
      P ← P → suiv
    fintantque
    q ← P → suiv
    P → suiv ← NULL
    q → suiv ← NULL
    libérer (q)
  finsi
fin

```

~~Supp~~ Supprime de l'élément à une
adresse donnée.

Procédure supp-Adr ($L: * \text{liste}$,
~~ps~~ $q: * \text{cellule}$)

Var

$p: * \text{cellule}$

debut

si ($\text{Est vide}(*L)$ ou $q = \text{NULL}$) alors

Ecrire ("Pas de supp")

sinon

si ($L \rightarrow \text{tête} = q$) alors

supp-tête (L)

sinon

$P \leftarrow L \rightarrow \text{tête}$

tantque ($p \neq \text{NULL}$ et $p \rightarrow \text{suiv} \neq q$)

$P \leftarrow p \rightarrow \text{suiv}$

fin tantque

si ($P \neq \text{NULL}$) Alors

$P \rightarrow \text{suiv} \leftarrow q \rightarrow \text{suiv}$

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

liberer (q)

fin