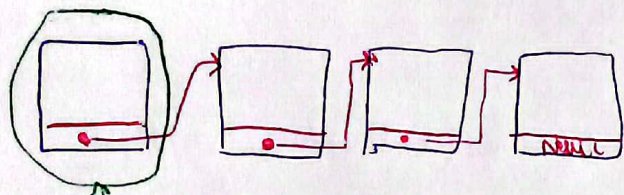


Ch3: Les listes chaînées

- un tableau est SD permettant de regrouper un nb fini d'éléments identiques. Alloués d'une manière contigue.
- une LS est une SD permettant de regrouper un ensemble d'éléments identiques.



cellule

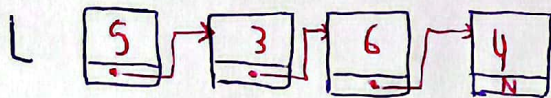
cellule: Enregistrement

Data: type (Entier, réel, Enreg)

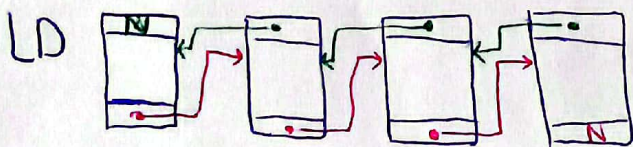
suiv: * cellule

fin

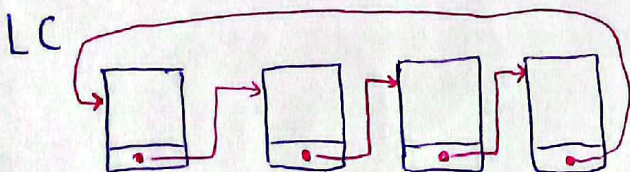
Les différents types de LS



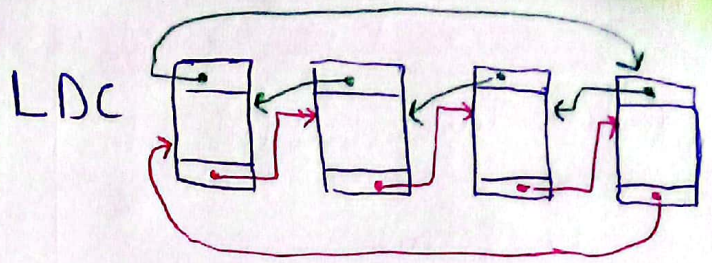
Liste simplement chaînée linéaire



Liste Doublement chaînée linéaire



Liste simplement chaînée circulaire



I/ Les listes simplement chaînées

linéaire:

Représentation (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: * Cellul

fin Enreg

Liste: Enregistrement

Tête: * cellule

Fin Enreg

Création d'1 liste vide:

Procédure Init (var L: liste)

Debut

L: * liste

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): booléen

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 elt à une liste:

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

var

$q: *cellule$

debut

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

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

(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

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

fin tantque

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

fin si

fin

$P_1: L: liste$

$P_3(L)$

$P_4(8L)$

$P_3: (L: liste)$

$P_2: L: *liste$

$P_4(L)$

$P_3(*L)$

$P_4(L: *liste)$

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

var

$p, q: *cellule$

$i: entier$

debut

si ($\text{Estvide}(xL) = \text{vrai}$ ou $k=1$) Alors

Ajout-tête (L, x)

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

$i \leftarrow 1$

tantque ($i \neq k$) faire

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

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

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

$i \leftarrow i+1$

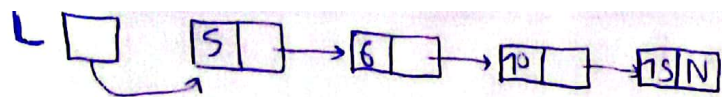
fin tq

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

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

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

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



X → la liste est vide ou $x \leq L \rightarrow \text{tête} \rightarrow \text{val}$
 chercher l'ID 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
    P ← L → tête
    tantque (P → suiv <> NULL) et
      P → suiv → val < x faire
      P ← P → suiv
    fin tantque
    q ← Allouer(1)
    q → val ← x
    q → suiv ← P → suiv
    P → suiv ← q
  fin si
fin
  
```

Recherche d'un elt d'une liste:

fonction Rech (L: liste, x: Entier):
 *cellule

```

var
  P: *cellule
debut
  P ← L.tête
  tantque (P <> NULL et P → val <> x) faire
    P ← P → suiv
  fin tantque
  Rech ← P
fin
  
```

Ex:

Ecrire une procédure permettant
 d'éclater une liste d'entiers en deux
 liste telq^{ue} la première contiendra
 les éléments négatifs dans l'ordre
 Inverse et sans redondance et la
 deuxième liste contiendra les
 éléments positifs dans le m^{ême} ordre
 de la liste initiale et sans
 redondance aussi.

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

```

var
  P: *cellule
debut
  Init(L1)
  Init(L2)
  P ← L.T.tête
  
```


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

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

var

$P: * \text{cellule}$

debut

si ($\text{Est vide}(*L)$ ou $q = \text{NULL}$) alors
Ecrire ("Pas de supp")

sinon

si ($L \rightarrow \text{tete} = q$) alors
supp - tete (L)

sinon

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

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

fin