

# Ch4 - Les piles et les files

Initier:

## 1) La Notion de Pile

on considère une pile d'entiers :

cellule : Enregistrement

val : Entier

suit : \*cellule

Fin Enreg

Pile : Enregistrement

sommet : \*cellule

Fin Enreg

Procédure Init (P: \*Pile)

debut

P → sommet ← NULL

fin

fonction Est vide (P: Pile) : booléen

debut

Estvide ← (P.sommet = NULL)

fin

fonction valsommet (P: Pile) : entier

debut

valsommet ← P.sommet → val

Procédure Empiler (P: \*Pile, x: entier)

Var

q: \*cellule

debut

q ← Allouer (1)

q → val ← x

q → suiv ← p → sommet

p → sommet ← q

fin

fonction Dépiler (P: \*Pile) : Entier

Var

x: Entier

q: \*cellule

debut

x ← p → sommet → val

q ← p → sommet

P → sommet ← P → sommet → suiv

libérer (q)

Dépiler ← x

Fin

Ex:

Ecrire une procédure permettant de trier une pile d'entiers dans l'ordre croissant (plus petit est au sommet), en utilisant une pile intermédiaire et une variable simple

Procédure Tri (P: \*Pile)

Var

Q: Pile

x: entier

Debut

Init (&Q)

Tantque (Estvide (&P) = faux) faire

x ← Dépiler (P)

Tantque (Estvide (Q) = faux)

Et valsommet (Q) > x faire

Empiler (P, Dépiler (&Q))

Fintantque

Empiler (&Q, x)

Fintantque

Tantque (Estvide (Q) = faute) faire

Empiler (P, Dépiler (&Q))

Fintantque

Fin

Expression Arithmétique

- Infixé A+B
- Postfixé AB+
- Préfixé +AB

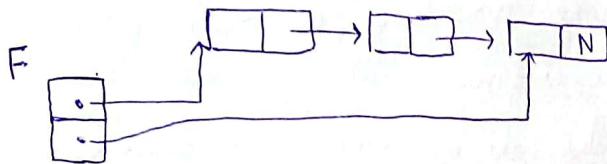
### l'algorithme de conversion Inf - Postf

Inf:  $A + B - (C * (2 / 4 * 5)) \% 2 * E$



Output: AB+C24/5\*%2%E\*-

### La notion de File :



Cellule : Enregistrement

Val : Entier

suiv : \*cellule

Fin enreg

File : Enregistrement

Tête : \*cellule

Queue : \*cellule

Fin enreg

Procédure Init(F : \*File)

Début

$F \rightarrow \text{Tête} \leftarrow \text{NULL}$

$F \rightarrow \text{queue} \leftarrow \text{NULL}$

Fin

fonction Estvide (F : File) : boolean

début

$\text{Estvide} \leftarrow (F \rightarrow \text{Tête} = \text{NULL})$

fin

Procédure Enfiler(F : \*File, x : entier)

Var

q : \*cellule

début

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

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

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

si ( $\text{Estvide}(F) = \text{vrai}$ ) Alors

$F \rightarrow \text{Tete} \leftarrow q$

sinon

$F \rightarrow \text{queue} \rightarrow \text{suiv} \leftarrow q$

fin si

$F \rightarrow \text{queue} \leftarrow q$

fin

fonction defiler(F : \*File) : Entier

Var

x : Entier

q : \*cellule

début

$x \leftarrow F \rightarrow \text{Tete} \rightarrow \text{val}$

$q \leftarrow F \rightarrow \text{Tete}$

$F \rightarrow \text{Tete} \leftarrow F \rightarrow \text{Tete} \rightarrow \text{suiv}$

si ( $F \rightarrow \text{Tete} = \text{NULL}$ ) Alors

$F \rightarrow \text{queue} \leftarrow \text{NULL}$

fin si

liberer(q)

defiler  $\leftarrow x$

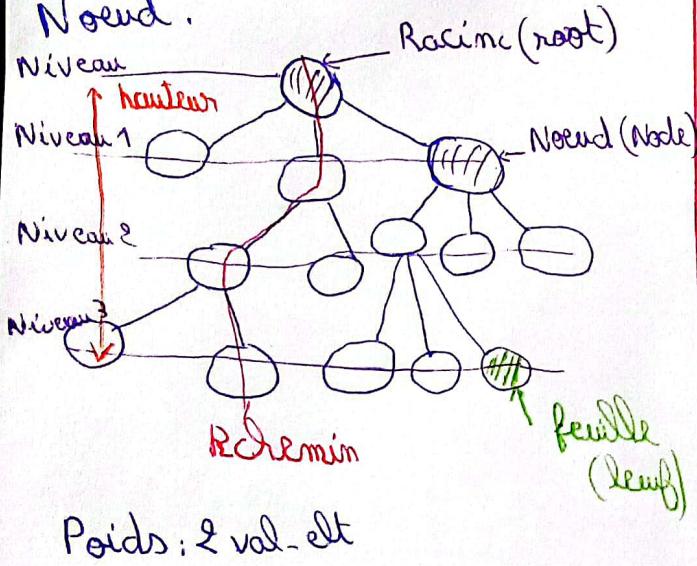
fin

## Ch 5 - Les arbres

Def: un arbre est un DS permettant de regrouper plusieurs identiques éléments

des éléments peuvent être alloués d'une manière non contiguë.

Chaque élé possède un seul prédecesseur (père), sauf le premier (Racine) et zero ou plusieurs successeurs (fils). un élément dans un arbre est appelé Nœud.



Nœud: Enregistrement

val: type

AF1: \*Nœud

AF2: \*Nœud

;

AFN: \*Nœud

fin enreg

nb max de fils = 2 : Arbre binaire

=> un arbre binaire est un arbre dont chaque Nœud peut avoir un maximum 2 fils

1) Arbre binaire:

on considère un binaire d'entier:  
 Nœud, Enregistrement

Val: Entier

FG: \*Nœud

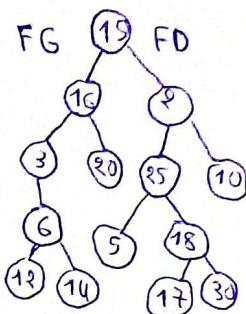
FD: \*Nœud

fin Enreg

Arbre: Enregistrement

Racine: \*Nœud

fin Enreg



INIT

Initialiser un arbre vide:

Procédure Init (A: \*Arbre)

debut

A  $\leftarrow$  Racine  $\leftarrow$  NULL

fin

verifier si l'arbre est vide ou non

afficher

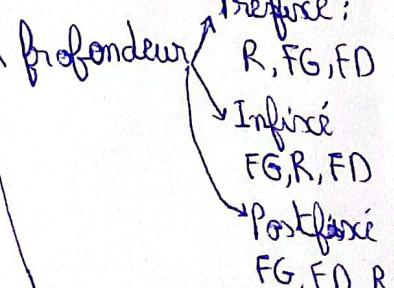
fonction Estvide (A: Arbre): booléen

debut

Estvide  $\leftarrow$  (A.Racine = NULL)

fin

Parcours d'un arbre



Largeur: Niveau/Niveau

## Procédure Préfixe (A: Arbre)

Var

AG, AD : Arbre

Début

Si (Est-vide(A) = faux) Alors  
 AG.Racine  $\leftarrow$  A.Racine  $\rightarrow$  FG  
 AD.Racine  $\leftarrow$  A.Racine  $\rightarrow$  FD  
 Ecrire (A.Racine  $\rightarrow$  val)  
 Préfixé (AG)  
 Préfixé (AD)

fin si

fin

Préfixé : 15 - 16 - 3 - 6 - 12 - 14 - 20 - 2 - 25 . 5  
 - 18 - 17 - 30 - 70

## Procédure infixe (A: Arbre)

Var

PPostfixe

AG, AD : Arbre

Début

Si (Est-vide(A) = faux) Alors  
 AG.Racine  $\leftarrow$  A.Racine  $\rightarrow$  FG  
 AD.Racine  $\leftarrow$  A.Racine  $\rightarrow$  FD

Postfixé

|   |                                     |
|---|-------------------------------------|
| 1 | Préfixé (AG)                        |
| 3 | Ecrire (A.Racine $\rightarrow$ val) |
| 2 | Préfixé (AD)                        |

fin si

fin

Infixe : 3 - 12 - 6 - 14 - 16 - 20 - 15 - 5 - 25 -  
 17 - 18 - 30 - 2 - 10

Postfixe : 12 - 14 - 6 - 3 - 20 - 16 - 5 - 17 - 30  
 - 18 - 25 - 10 - 2 - 15

## Procédure Longueur (A: Arbre)

Var

F: File de Noeud

X: Noeud

Début

Si (Est-vide (A) = faux) Alors  
 Enfiler(&F, \*(A.Racine))  
 tant que (Est-vide(F) = faux) faire  
 x  $\leftarrow$  Défiler(&F)  
 Ecrire (x.val)  
 Si (x.FG < NULL) Alors  
 Enfiler(&F, \*(x.FG))  
 fin si  
 Si (x.FD < NULL) Alors  
 Enfiler(&F, \*(x.FD))  
 fin si  
 fin tant que

fin

fin

15 16 23 20 85 10 6 5 18 12 14 30 8

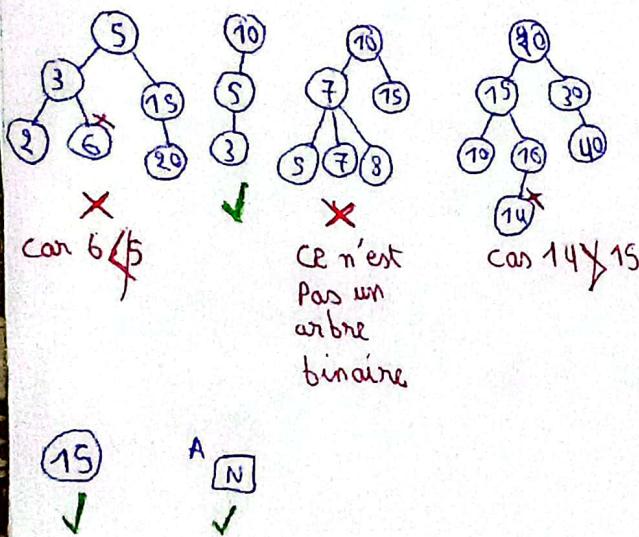
sorité : 15 16 23 20 95 106 5 18 12 14 17 30

## 2) Arbre Binaire de Recherche ABR:

def: un ABR est un arbre binaire dont chaque NOEUD doit vérifier la propriété suivante:

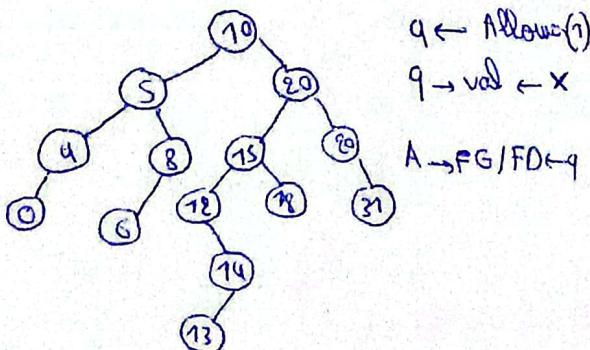
Tout él<sup>t</sup>  $\in$  SAG  $\leftarrow$  val de Noeud  $\leq$  tout él<sup>t</sup>  $\in$  SAD

Ex. parmi les 3 arbres suivant identifier les ABR :



Les principales opérations:

Ajout d'un él<sup>t</sup>:



Supp. d'un él<sup>t</sup> à partir d'un ABR

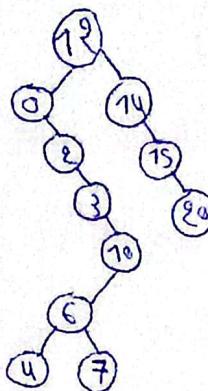
on distingue 3 cas :

- Noeud à supprimer est une feuille
- Noeud à supprimer possède 1 seul fils (FC/FD)
- Noeud à supprimer possède 2 fils

Ex:

1) construire un ABR avec l'ensemble suivant:

12 - 14 - 0 - 2 - 3 - 10 - 15 - 6 - 20 - 7 - 4



2) Ajouter l'elt  $x = 13$

3) supp les elts: 2, 20, 12

on considère un AB dont le nb de Noeud = n

Le nb de feuille = f  
l'hauteur = h

1) Quelle est l'hauteur maximale de l'arbre?  
 $\Rightarrow h_{\max} = n$

2) Quel est le nb max feuille en fonction de h?  $\Rightarrow f_{\max} = ?$