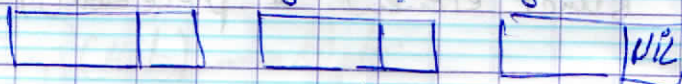


Listes:

Simplement chaînées  
Doublement chaînées  
Circulaire

Piles:

Toto



→ pour réversibilité; LIFO: Last in First out

Le type

Element-Pile = structure

info : T

suivant : ^ Element-Pile

Fin structure

Pile = ^ Element-Pile

InitPile (P: Pile) initialise Pile.

fonc PileVide (P: Pile) → Booléen.

Procédure Empiler (P: Pile, Val: T.)

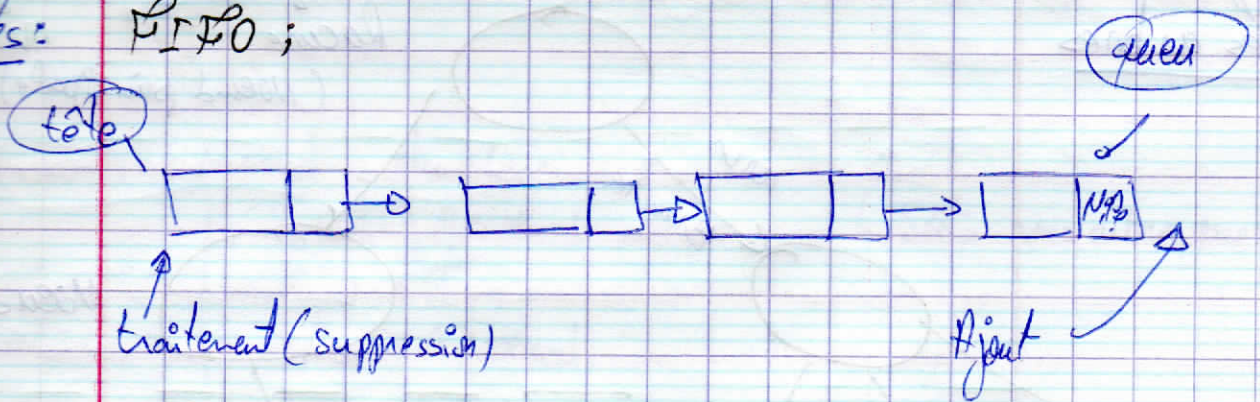
" Depiler (P: Pile)

fonc Sommet (P: Pile) : T

! avec  
idéer sur  
le cours



Files: FIFO;



Def type

Elmt File = structure  
 |  
 info : T  
 suivant : ^ Elmt File.  
 Fin struct

File = structure  
 |  
 tête : ^ Elmt File  
 queue : ^ Elmt File.  
 Fin struct

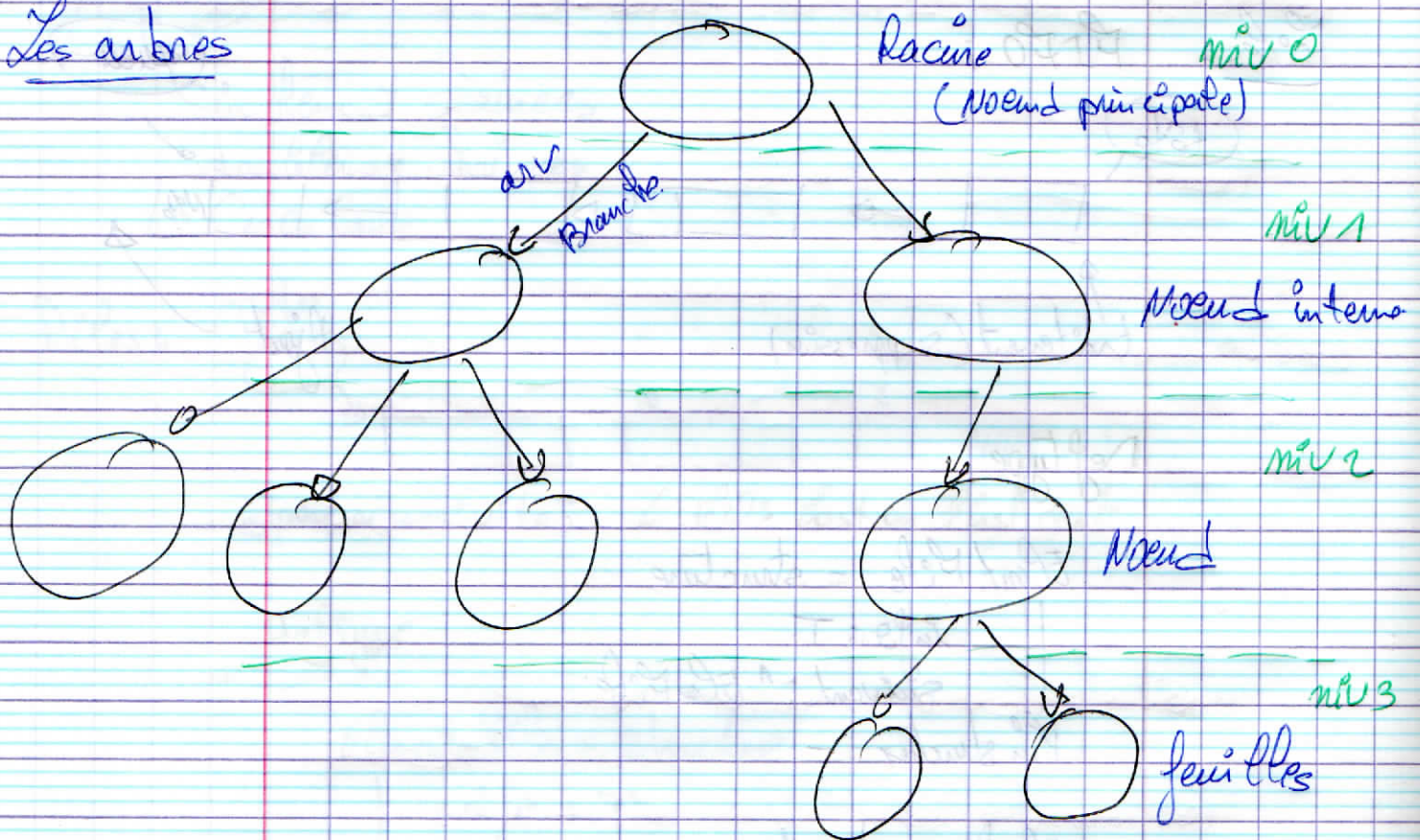
def var  
 F: File

Init File (F: File)  
 File Vide (F: File)  
 Enfiler (F: File, Val: T)  
 De Filer (F: File)  
 Sommet (F: File)

⚠ voir cours.

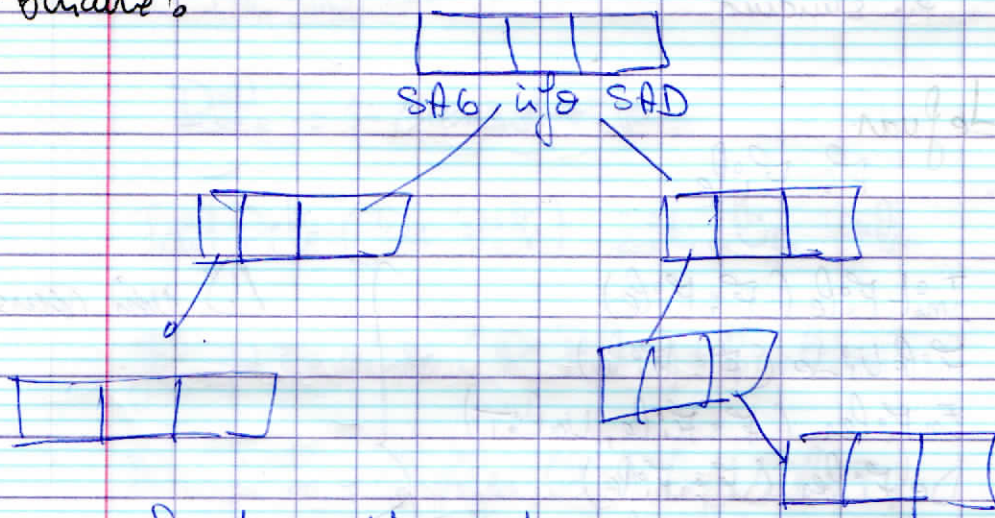


## Les arbres



arbre binaire : en plus 2 noeuds par chaque père

arbre binaire :



- hauteur d'un arbre
- hauteur d'un noeud, profondeur
- degré à nombre des fils



De type

Noeud = structure

| SAG : ^Noeud  
| info : T  
| SAD : ^Noeud  
Fin structure

(SAG: sous arbre gauche)

Def var

Racine : ^Noeuds

Création d'un arbre vide:

① Fonction CreerArbre() : ^Noeud

debut

| retourner(Nil)

Fin

② Fonction est-Vide (Racine : ^Noeud) : Booléen.

debut

| retourner (Racine = Nil)

Fin

③ Fonction Initialisation (val : T) : ^Noeud

debut

Racine : ^Noeud

debut

| Affecter (Racine)  
| Racine.info ← val  
| Racine.SAG ← Nil  
| Racine.SAD ← Nil  
| Retourner (Racine)



④ fonction Est-feuille (Racine :  $\wedge$  Noeud) : Booléen

Debut  
| si est-vide (Racine) alors retourner (Faux)  
| sinon  
| retourner (est-vide (Racine<sup>!</sup>.SAG) ET  
| est-vide (Racine<sup>!</sup>.SAD))  
Fin Si  
Fin

⑤ Fonction Hauteur (Racine :  $\wedge$  Noeud) : entier

Debut  
| si est-vide (Racine) alors retourner (0)  
| sinon  
| retourner (1 + Max (Hauteur (Racine<sup>!</sup>.SAG),  
| Hauteur (Racine<sup>!</sup>.SAD))  
Fin Si  
Fin

⑥ fonction Max (int a, int b)

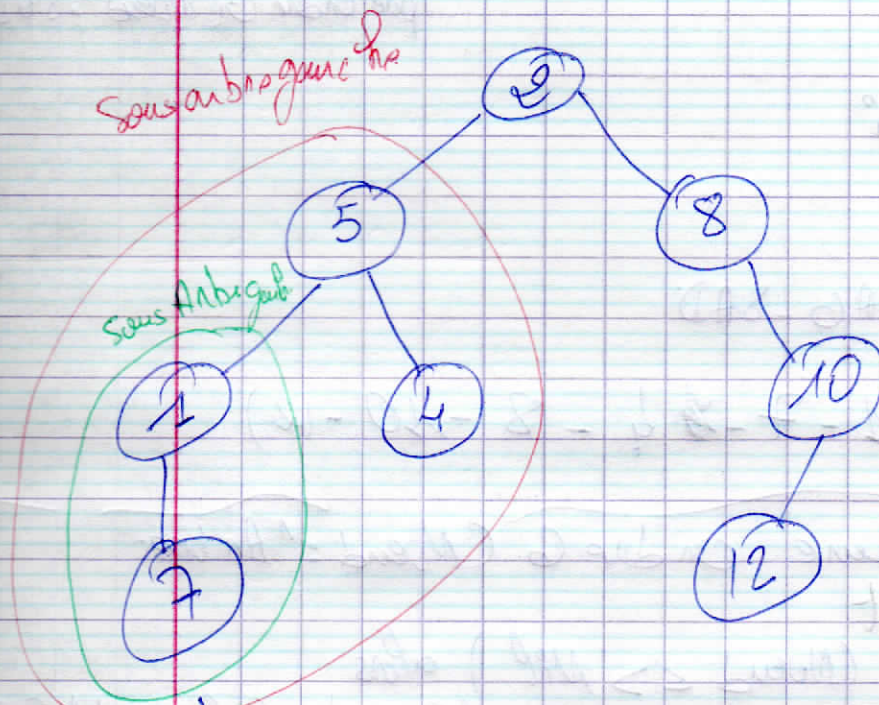
Debut  
| si (a > b) alors retourner (a)  
| sinon retourner (b)  
Fin



Parcours post finé : SAG - SAD - Racine  
SAD - SAG - Racine

parcours pré finé : Racine - SAG - SAD  
Racine - SAD - SAG

Parcours in finé : (au Ch. de)  
SAG - Racine - SAD  
SAD - Racine - SAG



(parcours post finé)

Procédure Postordre G (Noeud : 'Arbre')

Début

si (Noeud  $\leftrightarrow$  Nil) alors

Postordre G (Noeud : SAG)

Postordre G (Noeud : SAD)

Traiter (Noeud : info)

Fin si

Fin

(parcours post finé : 7 - 1 - 4 - 5 - 12 - 10 - 8 - 2)



parcours préfixe / 12-10-8-4-7-1-3-2)  
postfixe

SAD-SAG-Parcours

Procédure Préordre G (Noeud : ^ Arbre)

debut

si (Noeud  $\neq$  Nil) alors  
Traverse (Noeud ^ info)  
préordre G (Noeud ^ SAG)  
préordre G (Noeud ^ SAD)

Fin si

Fin

→ racine - SAG - SAD

(2-5-1-7-4-8-10-12)

Procédure Ordre G (Noeud : ^ Arbre)

debut

si (Noeud  $\neq$  Nil) alors  
Enordre G (Noeud ^ SAG)  
Traverse (Noeud ^ info)  
Enordre G (Noeud ^ SAD)

Fin si

Fin

→ SAG - racine - SAD

(1-7-5-4-2-8-12-10)