Algorithme

29/04/2023

⑥ → Algorithme récursive.
① → Recherche dichotomique.
② → structure dynamique.

Recherche dichotomique par la récursivité:

```
Fonction Recherche (X:entier, T:tab, D, F:entier):booleen
Defvar M:entier.
Debut
| Si (D>F) alors retourner (Faux)
|          sinon
|              M ← (D+F) div 2
|              si (T[M]=X) alors retourner (Vrai)
|              sinon
|                  si (T[M]<X) alors retourner (recherche(X,T,M+1,F)
|                  sinon retourner (recherche(X,T,D,M-1)
|                  FinSi
|              FinSi
| FinSi
Fin
```

Les allocations dynamiques

P → Nil → Pointeur sur entier / réel / Tableau / C-C / enregistrement
P → @

allouer(P) Liberer(P)

allouer ← verbe utiliser pour définir un pointeur.

déclaration: P : ^entier ⇒ P le pointeur d'un entier

Ona: P et P^ ;

P: pointeur , P^ est la variable pointé

→ Si P pointeur sur un tableau ⇒ P pointe sur 1er élement du Tableau

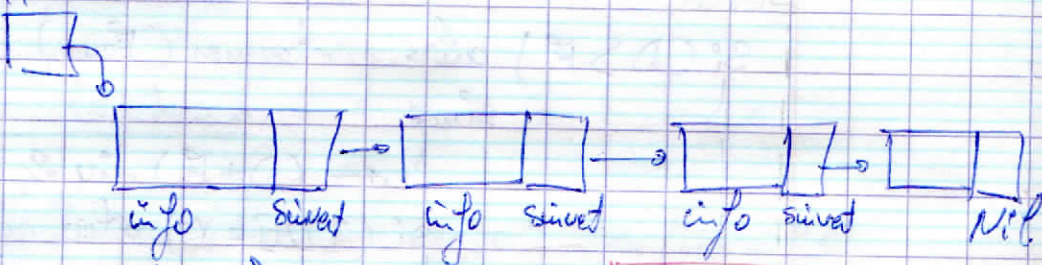
P = T ⇒ P = &T[1]

⇒ P+i = &T[i]

⇒ ^(P+i) = T[i] (en langage C ^(P+i) = T[i].)

Les listes chaînées :

Simple :



Déclaration L.S.Chaînée : algorithme Création par insertion en tête

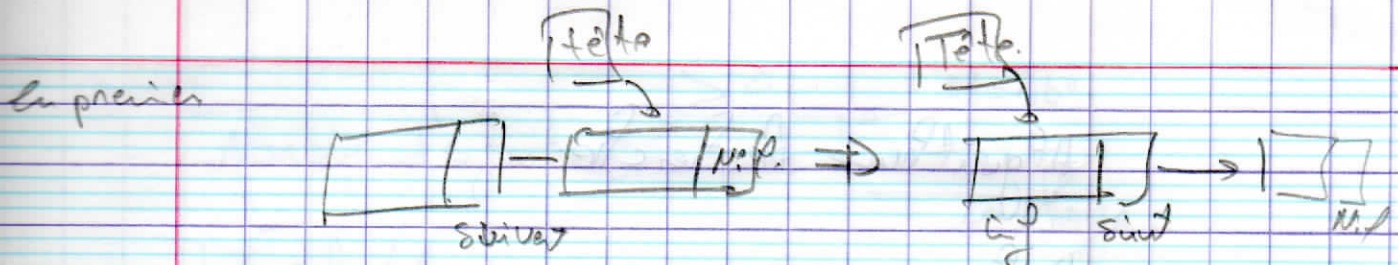
Def type

Element = Structure
| info : Type
| suivant : ^Element
Fin struct

Def var
Tête : ^Element ; P : ^Element

Création Liste simplement chaînée :

debut
allouer (tête)
Lire (tête^.info)
Tête^.suivant ← Nil




```
Pour i de 1 à 19 faire
|   allouer (P)
|   Lire (P^.info)
|   P^.suivant ← tête
|   Tête ← P
Fin Pour
```

Afficher les elements de la liste par cours liste

```
Defvar
   P, Tête : ^Element

Debut:
   si (Tête = Nil) alors Ecrire ("liste vide")
      sinon
            P ← tête
         Tantque ( P ≠ Nil ) faire
            |   Ecrire (P^.info)
            |      P ← P^.suivant
         fin Tantque.
```

afficher les elements d'une liste chainée simple

exercice

- recherche d'une valeur entier dans une liste

Algorithme Recherche.

```
Defvar
    Tête : ^Element
    Val : T
    Test : booléen
Debut
    si (tête = Nil) alors écrire ("Liste vide")
    | sinon
    |     Ecrire ("donne valeur recherchée")
    |     Lire (Val)
    |     P ← tête . Test ← Faux
    |     Tantque (P ≠ Nil ET Test = faux) faire
    |         si (P^.info = val) alors
    |             Test ← vrai
    |         sinon
    |             P ← P^.suivant
    |         Fin Si
    |     Fin Tantque.
    Fin Si
    Si (Test = Vrai) alors écrire ("existe")
                     sino Ecrire ("n'existe pas")
    Fin Si
```