

EMP

Structures de données

Examen1 // 2010 2011

Barème probable : (6) + (4+4+4+2)

Durée : 1h30

Tableaux

1. Donner l'algorithme qui recherche une valeur V dans un tableau ordonné $T[1..N]$ selon le principe suivant :

$a=1$; $b := N$

a) Générer un nombre aléatoire entre a et b , soit i ($a \leq i \leq b$)

b) Si $T[i] = V$ fin avec succès.

c) Si $V > T[i]$ poser $a := i + 1$ sinon poser $b := i - 1$

d) Reprendre à partir de a)

Le processus s'arrête quand la valeur V est trouvée ou bien a devient supérieur à b .

On utilisera la fonction $\text{Rand}(n)$ qui fournit un nombre aléatoire entre 0 et $n-1$.

Listes

Soit L une liste ordonnée.

1. Ecrire le module $\text{Rechercher}(Val, L1, L2, \text{Trouv})$ qui recherche une valeur Val donnée dans une sous liste $[L1, L2]$ ordonnée et qui retourne le résultat dans la variable booléenne Trouv .

2. Construire une liste L' dont les éléments sont des adresses vers les maillons de la liste L . Le maillon de rang i de la liste L' contient l'adresse de l'élément de rang i de la liste L . (Pas donné)

3. Utiliser les deux listes (L et L') pour rechercher rapidement une valeur V donnée.

4. Discuter le nombre moyen de comparaisons.

SOIT

T UN VECTEUR (10) ;
L UNE LISTE ;
Lprime UNE LISTE DE LISTE ;
Recherche2 UNE ACTION ;
Recherche UNE ACTION ;
Rech UNE ACTION ;
Construire UNE ACTION ;
Pas UN ENTIER ;

DEBUT

Pas := 5 ;
INIT_VECT (T , [12 , 23 , 45 , 56 , 60 , 66 , 70 , 75 , 78 , 98]) ;
APPEL Recherche (98) ;
CREER_LISTE (L , [12 , 23 , 45 , 56 , 60 , 66 , 70 , 75 , 78 , 98 , 123 , 167 , 200 , 236 , 267 , 289])
;
APPEL Construire (L , Lprime) ;
APPEL Recherche2 (Lprime , L , 23) ;

FIN

ACTION Recherche (Val)

SOIT

Bi , Bs , N DES ENTIERS ;
Val UN ENTIER ;
Trouv UN BOOLEEN ;

DEBUT

Bi := 1 ;
Bs := 10 ;
Trouv := FAUX ;
TQ (Bi <= Bs) ET NON Trouv
N := ALEANOMBRE (Bs - Bi + 1) ;
Ecrire ('n=' , N) ;
SI Val = ELEMENT (T [Bi + N])
Trouv := VRAI
SINON
SI Val < ELEMENT (T [Bi + N])
Bs := Bi + N - 1
SINON
Bi := Bi + N + 1
FSI
FSI
FTQ ;
Ecrire (Trouv)

FIN

ACTION Construire (L , Lprime) ;

SOIT

L , P DES LISTES ;
Lprime UNE LISTE DE LISTE ;
Q , Prec DES LISTES DE LISTE ;

I UN ENTIER ;

DEBUT

Lprime := NIL ;

P := L ;

I := 1 ;

TQ P <> NIL

 I := I + 1 ;

 SI I = Pas

 Pas := Pas + Pas ;

 ALLOUER (Q) ;

 AFF_VAL (Q , P) ;

 AFF_ADR (Q , NIL) ;

 SI Lprime = NIL

 Lprime := Q

 SINON

 AFF_ADR (Prec , Q) ;

 FSI ;

 Prec := Q

FSI ;

P := SUIVANT (P)

FTQ

FIN

ACTION Rech (Val , L1 , L2 , Trouv)

SOIT

L1 , L2 , P DES LISTES ;

Val UN ENTIER ;

Trouv , Arret DES BOOLEENS ;

DEBUT

P := L1 ;

Trouv := FAUX ;

Arret := FAUX ;

TQ (P <> L2) ET NON Trouv ET NON Arret

 SI VALEUR (P) = Val

 Trouv := VRAI

 SINON

 SI VALEUR (P) > Val

 Arret := VRAI

 SINON

 P := SUIVANT (P)

 FSI

FSI

FTQ

FIN

ACTION Recherche2 (Lprime , L , Val) ;

SOIT

Lprime , Q , Prec DES LISTES DE LISTE ;

L UNE LISTE ;

Trouv , Arret DES BOOLEENS ;
Val UN ENTIER ;
Der UNE LISTE ;

DEBUT

```
Trouv := FAUX ;
Arret := FAUX ;
Q := Lprime ;
TQ ( Q <> NIL ) ET NON Trouv ET NON Arret
  SI VALEUR ( VALEUR ( Q ) ) = Val
    Trouv := VRAI
  SINON
    SI Val < VALEUR ( VALEUR ( Q ) )
      SI Q = Lprime
        APPEL Rech ( Val , L , VALEUR ( Q ) , Trouv )
      SINON
        APPEL Rech ( Val , VALEUR ( Prec ) , VALEUR ( Q ) , Trouv )
    FSI ;
    Arret := VRAI
  SINON
    Prec := Q ;
    Q := SUIVANT ( Q )
  FSI
FSI
FTQ ;
SI Q = NIL
  Der := NIL ;
  APPEL Rech ( Val , VALEUR ( Prec ) , Der , Trouv )
FSI ;
Ecrire ( Trouv )
```

FIN