

Examen

Barème: 1.5 + 2.5 + 3 + 3 + 3 + 4 + 3

On souhaite dans cet exercice traiter des listes linéaires chaînées de nombres entiers.

Exemple :

$Tete \mapsto [4] \mapsto [1] \mapsto [2] \mapsto [1] \mapsto [3] \mapsto Nil$

1. Donner les déclarations permettant d'utiliser de telles listes.
2. Soit la fonction f suivante :

```
Fonction  $f(P : \text{Poniteur}(\text{TMaillon})) : \text{entier};$   
var  $x : \text{entier};$   
Début  
  Si ( $P = Nil$ ) Alors  
     $f \leftarrow 0;$   
  Sinon  
     $x \leftarrow f(\text{Suivant}(P)) + 1;$   
     $f \leftarrow \text{Valeur}(P) \times x;$   
  Fin Si};  
Fin};
```

- (a) Donner la valeur retournée par l'appel de la fonction f sur la liste de l'exemple ci-dessus.
 - (b) Calculer la complexité de la fonction f en fonction de la longueur n de la liste passée en paramètre.
3. Écrire la fonction récursive :

Fonction Triée($L : \text{pointeur}(\text{TMaillon})$) : booléen ;

permettant de retourner vrai si la liste L est triée dans l'ordre croissant et faux sinon.

4. Écrire la procédure :

Procédure Inverser($L : \text{pointeur}(\text{TMaillon})$) ;

permettant d'inverser la liste L sans créer de nouveaux éléments.

5. Écrire la procédure

Procédure Trier($L : \text{pointeur}(\text{TMaillon})$) ;

permettant de trier la liste d'entiers L en utilisant la méthode par Bulle.

6. Donner l'arbre binaire de recherche obtenu en insérant dans l'ordre les clés suivantes : **18, 25, 22, 9, 19, 4, 31, 14.**

Corrigé type

1. Déclarations permettant d'utiliser de telles listes.

```
Type TMailon = Structure  
  Valeur : entier ;  
  Suivant : Pointeur(TMailon) ;  
Fin ;  
Var Tete : Pointeur(TMailon ;
```

2. (a) Valeur retournée par l'appel de la fonction f sur la liste de l'exemple = **48**
(b) Complexité de la fonction f en fonction de la longueur n de la liste passée en paramètre.
Le temps $T(n)$ d'exécution de la fonction :

$$T(n) = \begin{cases} a & \text{si } n = 0 (P = Nil) \\ T(n-1) + b & \text{sinon } (si P \neq Nil : n \neq 0) \end{cases}$$

- La constante a englobe le temps du test ($n \neq Nil$) et le temps de l'affectation ($f \leftarrow 0$)
- La constante b englobe :
 - * le temps du test ($P = Nil$),
 - * le temps de l'opération $+$ entre $f(Suivant(P))$ et 1 l'affectation à x .
 - * et le temps de l'opération \times entre $Valeur(P)$ et x et l'affectation au résultat final.

$$\begin{aligned} T(n) &= T(n-1) + b \\ &= [b + T(n-2)] + b \\ &= T(n-2) + 2b \\ &= [b + T(n-3)] + 2b \\ &= \dots \\ &= T(n-i) + ib \\ &= \dots \\ &= T(0) + nb == a + nb \\ T(n) &= nb + a \end{aligned}$$

Donc la complexité est en $O(n)$ car a et b sont des constantes positives.

3. Fonction récursive **Triée**(L : **pointeur**(TMailon)) : **booléen** ;

```
Fonction Triée(  $L$  : Pointeur(TMailon)) : booléen;  
Début  
  Si ( $L=Nil$  ou  $Suivant(L)=Nil$  ) Alors  
    | Triée  $\leftarrow$  Vrai ;  
  Sinon  
    Si ( $Valeur(L)>Valeur(Suivant(L))$  ) Alors  
      | Triée  $\leftarrow$  Faux ;  
    Sinon  
      | Triée  $\leftarrow$  Triée( $Suivant(L)$ ) ;  
    Fin Si ;  
  Fin Si ;  
Fin ;
```

4. Procédure **Inverser**(L : **pointeur**(TMailion));

```

Procédure Inverser(  $L$  : Pointeur(TMailion));
var NouvelleL,  $P$  : Pointeur(TMailion);
Début
    NouvelleL  $\leftarrow$  Nil;
    Tant que (( $L \neq$  Nil) ) faire
         $P \leftarrow L$ ;
         $L \leftarrow$  Suivant( $L$ );
        Aff_adr( $P$ , NouvelleL);
        NouvelleL  $\leftarrow P$ ;
    Fin TQ;
     $L \leftarrow$  NouvelleL;
Fin;

```

5. Procédure **Trier**(L : **pointeur**(TMailion));

```

Procédure Trier(  $L$  : Pointeur(TMailion));
var  $P, fin$  : Pointeur(TMailion);
     $x$  : entier;
Début
     $fin \leftarrow$  Nil;
    Tant que (( $L \neq fin$ )) faire
         $P \leftarrow L$ ;
        Tant que (( $P \neq fin$ ) et ( $Suivant(P) \neq fin$ )) faire
            Si ( $Valeur(P) > Valeur(Suivant(P))$ ) Alors
                 $x \leftarrow Valeur(P)$ ;
                Aff_val( $P, Valeur(Suivant(P))$ );
                Aff_val( $Suivant(P), x$ );
            Fin Si;
             $P \leftarrow Suivant(P)$ ;
        Fin TQ;
         $fin \leftarrow j$ ;
    Fin TQ;
Fin;

```

6. Donner l'arbre binaire de recherche obtenu en insérant dans l'ordre les clés suivantes : **18, 25, 22, 9, 19, 4, 31, 14.**

