



Licence Sciences
et Technologies

GU : MIAS41, UE : 411 MI ALGO
Épreuve d'Initiation à l'Algorithmique
Date : Septembre 2004 Durée : 1h30
Tous documents interdits.
Vous devez répondre directement sur le sujet qui comporte 4 pages
Vous insérez ensuite votre réponse dans une copie d'examen
comportant tous les renseignements administratifs
Épreuve de Mme Delest

Indiquez votre code d'anonymat : No:

La notation tiendra compte de la clarté de l'écriture des réponses

Barème

- Question 1 - Connaissances générales : 4 points
- Question 2 - Compréhension d'algorithmes de tri : 3 points
- Question 3 - Manipulation de tableaux : 6 Points
- Question 4 - Listes : 3 Points
- Question 5 - Arbres : 2,5 Points
- Question 6 - Table de hachage : 1,5 Points

Question 1. Cocher les affirmations qui sont correctes:

- Dans une boucle **tant que** le test s'effectue en fin de boucle
- Les instructions interne d'une boucle **Répéter** peuvent ne pas être exécutées
- Un tableau a un nombre d'élément constant
- Une liste n'est pas une table d'association à clé unique
- Le tri à Bulle s'exécute en temps minimum $O(n)$.
- La dichotomie ne sert pas dans les algorithmes de tri
- Le nombre de sommets d'un arbre binaire est toujours impair
- Une table de hachage n'est pas une table d'association à clé unique

Question 2. Soit la fonction suivante

```
fonction test(val n:entier):entier;  
  var i,c: entier;  
  début  
    si n=0 alors  
      retourner 1;  
    sinon  
      c=0;  
      pour i allant de 1 à n faire  
        si estImpair(n) alors  
          c=c+1;  
        sinon  
          c=c+test(n-1)  
        finsi  
      finpour  
      retourner c;  
    finsi  
  fin  
finfonction
```

1. Quel est le résultat de la fonction si $n=3$? Justifier. **La justification consistant à exécuter l'algorithme est correcte**

La fonction est réursive. Ainsi a chaque appel la fonction s'exécute dans son contexte c'est-à-dire avec des variables locales à la fonction.

Le test d'arrêt de la fonction porte sur la valeur de n qui doit être nulle.

Or, la boucle **pour** consiste à faire un appel récursif lorsque n est pair avec la valeur $n-1$ qui est donc impaire par suite le test ne sert que lors d'un appel $\text{test}(0)$ qui ne se produit jamais dans la boucle **pour**. Remarquons que $\text{test}(1)=1$.

La boucle **pour** compte le nombre d'entiers impair inférieurs ou égaux à n auquel est ajouté le résultat de la fonction pour tous les entiers impairs précédant un entier pair non nul, on a donc le calcul suivant pour $n=3$.

$\text{test}(3)=2+\text{test}(1)=3$.

2. Est-il possible de remplacer **val** par **ref** dans l'en-tête de la fonction. Justifier.

Il est impossible de mettre **ref** à la place de **val** car l'appel s'effectue avec $n-1$ qui donne donc un résultat qui doit être stocké dans un emplacement de la mémoire.

Question 3. *On prendra soin dans ce qui suit d'écrire soigneusement l'en-tête de la fonction et notamment de préciser les mots clés **ref** et **val**. Soit deux tableaux $T1$ et $T2$ de taille N contenant des caractères. On dit que $T1$ est le miroir de $T2$ si $\forall i \in [1..N], T1[i] = T2[N - i + 1]$.*

1. *Ecrire une fonction qui teste si deux tableaux sont miroir l'un de l'autre.*

<pre> fonction miroir(ref T1:tableau[1..N1] de car; ref T2:tableau[1..N2] de car):booléen; var i:entier; début si N1 <> N2 alors retourner(faux); sinon pour i allant de 1 à N1 faire </pre>	<pre> si T1[i] <> T2[N1-i+1] alors retourner(faux); finsi; finpour; retourner(vrai); finsi; fin finfonction </pre>
--	--

2. *Donnez la(les) complexité(s) de votre algorithme . Réponse: $C^<(n) = O(1)$ et $\overline{C}(n) = C^>(n) = O(n)$*

3. *Ecrire une fonction qui calcule le maximum du nombre d'apparition d'une lettre donnée L dans les deux tableaux.*

<pre> fonction maxCar(ref T1:tableau[1..N1] de car; ref T2:tableau[1..N2] de car; val L:car):entier; var i,c1,c2:entier; fonction compte(ref U:tableau[1..N] de car; ref c:entier); début si U[i]=L alors c=c+1; finsi; fin fonction maxCarUn(ref U:tableau[1..NU] d'éléments; ref c:entier):vide; début tant que i <= NU faire compte(U,c) </pre>	<pre> i=i+1; fintantque; fin; début i=1; c1=0;c2=0; tant que i <= N1 et i <= N2 faire compte(T1,c1); compte(T2,c2); i=i+1; fintantque; maxCarUn(T1,c1); maxCarUn(T2,c2); retourner(max(c1,c2)); fin; finfonction </pre>
---	---

1. *Donnez le fonctionnement de la fonction insérer dans les listes simplement chaînées?*
On ne peut insérer que comme suivant d'un élément, l'en-tête de la fonction est:

```
fonction insérerAprès(val x:type_prédéfini;ref L:type_liste;P:~type_predefini):vide;
```

2. *Ecrire une fonction qui prend en paramètre un tableau d'entiers et dont le résultat est une liste doublement chaînée contenant les éléments du tableau dans l'ordre décroissant de l'indice.*

```
fonction tableauListe(ref T:tableau[1..N] d'entiers):listeDC;
  var L: listeDC d'entiers;
  var P:~entier;
  var i:entier;
  début
    creerListeDC(L);
    insererEnTete(T[N],L);
    P=premier(L);
    pour i allant de N-1 à 1 faire
      insérerAprès(T[i],L,P);
      P=suivant(L,P);
    finpour;
  fin;
finfonction
```

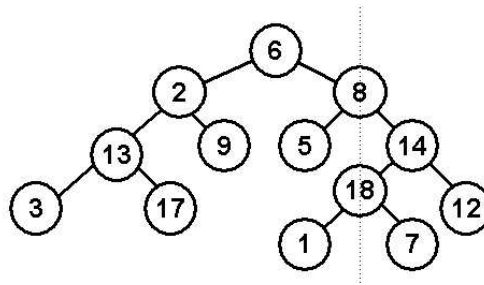


FIG. 1 – *Figure question 5.*

Question 5.

1. *Pour l'arbre binaire dessiné Figure 1, répondre aux questions suivantes.*
 - *Donnez la liste des étiquettes du sous arbre gauche du sommet d'étiquette 2.* Réponse: 2,13,3,17,9 (dans l'ordre préfixe, précision non demandée).
 - *Donnez la hauteur du sous-arbre issu du sommet d'étiquette 8.* Réponse: 3.
2. *Dessinez un arbre binaire ayant 13 sommets* Réponse Figure 2.
3. *Dessinez tous les arbres planaires ayant 4 sommets* Réponse Figure 3.

Question 6. *On considère la fonction de hachage qui à tout entier naturel n associe*

$$h_k(n) = n[k * (k - 1)]$$

1. *Dans le cas de l'adressage chaîné, quel sera la taille de la table de hachage pour cette fonction.*
Réponse: $k(k-1)$.

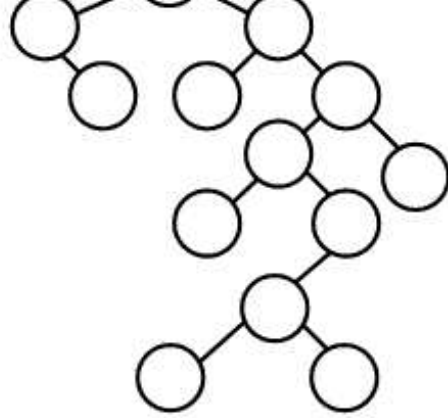


FIG. 2 – Réponse Exercice 5 question 3.

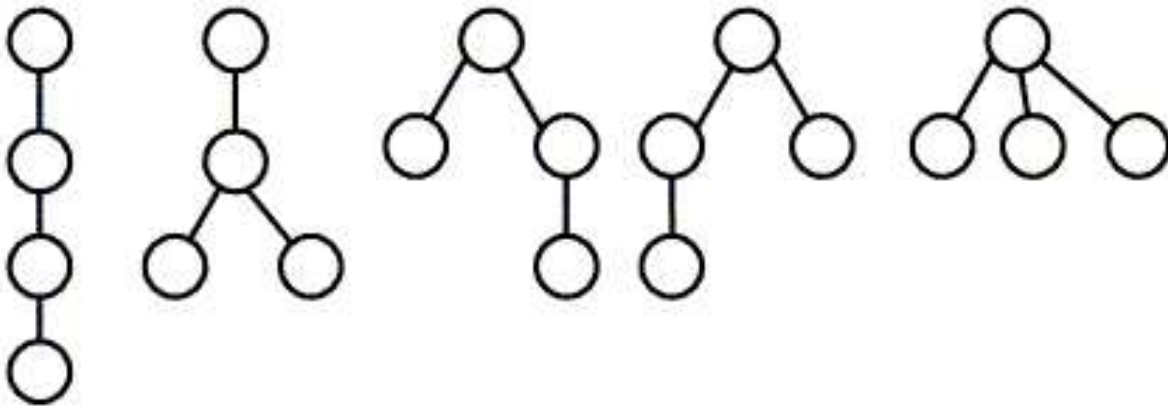


FIG. 3 – Réponse Exercice 5 question 3.

2. Soit $k = 4$. Donnez l'état de la table après l'insertion de la suite de valeur 2,12,26,5,36.

Indice	0	1	2	3	4	5	6	7	8	9	10	11		
Contenu	L0	NIL	L2	NIL	NIL	L5	NIL	NIL	NIL	NIL	NIL	NIL		

Les éléments de la table sont des listes simplement ou doublement chaînées.

L0 =(12,36)

L2=(2,26)

L5=(5)

FIN.