



Licence
Sciences et Technologies

ANNEE : 2005/2006

SESSION DE MAI 2006

ETAPE : INF2 MAT2 MAI4

UE : INF102

Epreuve : Initiation à l'algorithmique

Date :

Heure :

Durée : 1 H 30

Documents : Tous documents interdits

Vous devez répondre directement sur le sujet qui comporte 4 pages.

Insérez ensuite votre réponse dans une copie d'examen comportant tous les renseignements administratifs

Epreuve de Mme Delest.

Indiquez votre code **d'anonymat** :

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

Barème

- Question 1 – Connaissances générales : 4 points
- Question 2 – Compréhension d'algorithme de tris : 3 points
- Question 3 – Compréhension d'algorithme récursif : 3 points
- Question 4 – Ecriture de fonctions : 3 points
- Question 5 – Manipulation de tableau : 3 Points
- Question 6 – Manipulation de listes : 4 points

Question 1. Cochez les affirmations qui sont correctes :

- La boucle *pour* comporte un test
- Les instructions d'une boucle *répéter* sont toujours exécutées
- Une variable passée par *référence* à une fonction est recopiée dans l'environnement de la fonction.
- Les méthodes diviser pour régner permettent toujours d'obtenir une meilleure complexité en temps
- La complexité minimale d'un algorithme de tri est en $O(n)$
- Dans le tri à rapide, les k premiers éléments du tableau sont à leur place après k appels récursifs
- Les listes ont toujours une complexité mémoire plus élevée que les tableaux.
- Dans une liste simplement chaînée de n éléments, l'accès au dernier élément se fait en $O(n)$

Question 2. Soit la fonction suivante :

```
fonction mystere(ref T:tableau[1..NMAX] d'entier, val N :entier) : entier ;
  var k,tmp,compt :entier ;
  var tmp :tableau[1..N]d'entiers ;
  début
    init(tmp,0) ;
    k=2;
    itmp=0;
    compt=0 ;
    tant que k<=N faire
      si T[k] ==T[k-1]+1 alors
        itmp=itmp+1 ;
        tmp[itmp]=k ;
        compt=compt+1;
      finsi ;
      k=k+1 ;
    fintantque;
    copie(tmp,T,1,itmp ,1);
    retourner(compt) ;
  fin ;
finfonction
```

où *init* et *copie* sont les fonctions décrites dans le cours.

3. Dans la fonction récursive, le tableau est passé par référence. S'il était passé par valeur, quel serait l'effet sur la complexité ?

Question 4. Soit un tableau B d'entiers de taille N .

1. Ecrire la fonction *pic* qui prend pour paramètre un index K et le tableau B et dont le résultat est vrai si $B[K] = B[K-1] + 1$.
2. Ecrire une fonction *comptePic* qui utilise la fonction *pic* et dont le résultat est le nombre de fois où un entier est égal à l'entier qui le précède plus 1 dans le tableau. Par exemple si le tableau contient (0,1,0,0,1,0,1,1,0,1), le résultat est 4.

Question 5. Soit un tableau T de dimension M de tableaux de taille N d'entiers triés en ordre croissant . Par exemple, $M=3$, $N=4$, un exemple de tableau T est :

$T[1]=$	2	5	6	12
$T[2]=$	1	3	7	10
$T[3]=$	5	6	9	14

En utilisant la fonction *fusion* décrite dans le cours, écrire une fonction *multiTri* qui reçoit en entrée un tel tableau et dont le résultat est un tableau d'entiers à une seule dimension contenant tous les entiers du tableau T triés en ordre croissant. On ne demande pas de réécrire la fonction *fusion*. On rappelle l'en-tête de cette fonction décrite en cours :

*fonction fusion(ref T1:tableau[D1..N1] d'entier; ref T2:tableau[D2..N2] d'entier;
val DT1,FT1,DT2,FT2:entier):tableau[1..FT1+FT2-DT1-DT2+2] d'entier;*

Question 6.

1. Quels sont les avantages et les inconvénients de l'utilisation des listes par rapport à l'utilisation de tableaux pour implémenter une séquence de nombres ?

2. Soit L une liste simplement chaînée d'entiers.

- Que contient L et P après la séquence d'instructions suivante

```
var L :listeSC d'entier ;  
var P :^entier ;  
creerListe(L) ;  
insererEnTete(4,L) ;  
P=premier(L) ;  
insererAprès (3,L,P) ;  
insererAprès(4,L,P) ;  
P=trouverDernier(L) ;  
ajouterAprès(5,L,P) ;  
P=premier(L) ;  
supprimerAprès(L,P) ;  
P=suivant(L,premier(L)) ;
```

P=.....
Contenu(P)=.....
L=.....

- Ecrire une fonction *comptePicListe* dont le résultat est le nombre de fois où un entier est égal à l'entier qui le précède plus un dans la liste L. On s'inspirera de la question 4.

FIN