جـامعة هواري بومـدين للعلـوم و التكنـولوجيــا

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique

D é p a r t e m e n t   d'I n f o r m a t i q u e

Concours d'accès au Doctorat 3 ième Cycle Informatique 2017 – 2018

Le 29/10/2017

Matière 1 : Algorithmique avancée et complexité,
Coefficient **1**, durée **1 h 30**
(Spécialités : IA, MFA, SIGL)

Exercice 1 : (12 points)

Soit $A(n, m), B(n', m')$ deux tableaux à deux dimensions tel que $n' < n$ et $m' < m$. Il s'agit de rechercher l'élément B dans A.

1- En supposant que les éléments de A et B ne sont pas triés, écrire un algorithme qui retrouve B dans A. Evaluez sa complexité.

2- En supposant que chacune des lignes de A et B est triée par ordre croissant (voir figure 2), écrire un algorithme non naïf de complexité minimale pour trouver B dans A. Evaluez cette complexité.

2	2	2	3	5	7	8	17	24	24	54	67	76
3	4	4	5	6	6	6	8	11	12	33	81	85
12	14	23	26	26	26	31	34	44	45	52	87	90
6	6	17	24	24	54	56	61	67	81	87	90	108
2	2	2	3	5	7	8	17	24	24	54	67	76
3	4	4	5	6	6	6	8	11	12	33	81	85
12	14	23	26	26	26	31	34	44	45	52	87	90
6	6	17	24	24	54	56	61	67	81	87	90	108
12	14	23	26	26	26	31	34	44	45	52	87	90
6	6	17	24	24	54	56	61	67	81	87	90	108

Tableau A

24	54	56
5	7	8
6	6	6

Tableau B

Figure 2. Exemple de tableaux A et B triés

3- En supposant que le tableau A n'est pas trié, pour comparer B à une portion du tableau A (de position i, j) nous définissons la mesure d par l'équation (1).

$$d = \sum_{p=0}^{n'-1} \sum_{q=0}^{m'-1} |B(p,q) - A(i+p, j+q)| \quad \ldots .(1)$$

On dira qu'une portion de A **correspond** à B si $d = 0$.

On suppose que si B **correspond** à une portion de A alors plus on s'éloigne de celle-ci, la mesure d augmente (voir figure 3).

Ecrire un algorithme qui recherche B dans A. Evaluez la complexité de cet algorithme.

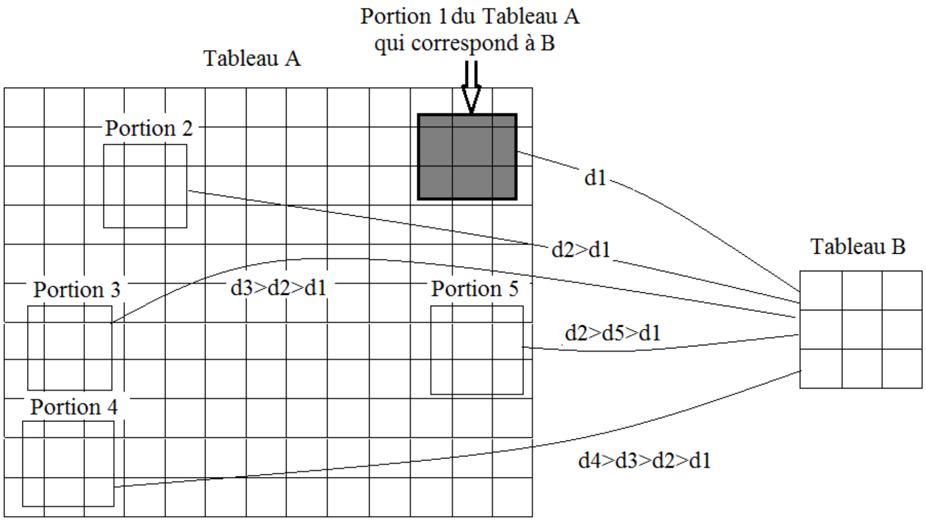

Figure 3. Exemple de calcul des mesures d entre le tableau B et différentes portions du tableau A.

Notons que la portion qui correspond à B (Portion 1 dans cette figure 3) est de position inconnue.

Exercice 2 : (8 points)

Recherche d'un point fixe d'une fonction

Soit $T[1..n]$ $n \geq 1$ un tableau trié en ordre croissant d'entiers relatifs tous distincts. On cherche à calculer le point fixe de T (chercher s'il existe ou non $i \in [1..n]$ tel que $T[i] = i$) .

1. Donner la réponse lorsque :
   $T1 = \{-5, 0, 1, 3, 4, 6, 9, 10, 12, 13\}$
   $T2 = \{-5, -1, 2, 4, 5, 9, 12, 13\}$
2. Construire une solution à ce problème. Donnez sa complexité.
3. On suppose que $T[1..n]$ contient des entiers naturels positifs tous distincts. Donnez une nouvelle solution du problème du point fixe et donnez sa complexité.

جـامعة هواري بومـدين للعلـوم و التكنـولوجيـا
Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Electronique et d'Informatique
Département d'Informatique

Concours d'accès au Doctorat 3 ième Cycle Informatique 2017 – 2018

Le 29/10/2017

Matière 1 : Algorithmique avancée et complexité,
Coefficient **1**, durée **1 h 30**
(Spécialités : IA, MFA, SIGL)

Corrigé Exercice 1 :

Corrigé question 1 :

```
Début
Boolean Different=False;
i=0;
While((i ≤ n − n')&&(! Different))
     { j=0;
      While((j ≤ m − m')&&(! Different))
          { p=0;
            While ((p ≤ n − n') )&&(! Different))
                {q=0 ;
                 While((q < m')&&((A(i + p, j + q) == B(p, q))) {q++;}
                 if(q<m') Different =True ;
                p++;
                }
          }
   }
Fin
```

Complexité si n' est de même ordre de grandeur que n, de même pour m et m', alors l'algorithmes est en $O(n^4)$, sinon, il est $O(n^2)$.

Corrigé question 2 :

Comme les lignes de A sont triées, nous utilisons la recherché dichotomique pour trouver A(i,j) identique à B(0,0) et qui sera le point de départ pour vérifier si les autres éléments (des lignes et colonnes) de B coïncident avec ceux de A, sinon on avance sur la même ligne s'il existe plusieurs éléments de A identique à B, dans l'autre cas on cherchera B(0,0) sur la ligne suivante de A.

La recherche de B(0,0) se fera que sur les n-n' premières lignes en raison de la taille n' de B.

<u>Début</u>

Boolean Identique=true, Trouvé=False;

RechercheDichotomique(A, i, B(0,0), indice)

Cet appel permet de communiquer à la fonction *RechercheDichotomique* le tableau A, la ligne numéro i et l'élément à rechercher B(0,0), elle retourne l'indice de l'élément dans la ligne numéro i de A s'il existe, sinon elle retourne -1.

```
i=0;
While((i <= n − n')&&(!Trouvé))
 {
      RechercheDichotomique(A, i, B(0,0), indice) ;
      if(indice !=-1)
      {
       //se positionner au premier élément identique à B(0,0) dans le cas
      // de présence de multiples valeurs identiques
       while((indice>0) &&(A(i,indice)==B(0,0)) indice--;
        p=0;
        while((indice≤ m − m')&&(A(i,indice)==B(0,0)) )&&(!Trouvé)
          {   //verifier la première ligne de B avec A à partir de j=indice
             q=0; j=indice;
             While((A(i + p, j + q) == B(p, q)&&(q < m'))
             { q++;}
             if(q == m') then // vérifier les lignes suivantes de B
               {
                  p++ ;
                  j=indice ; p=0; Identique=true;
                 While((i + p <= n − n')&&(Identique))
                     {  q=0;
                      While((j + q ≤ m − m')&&(A(i + p, j + q) == B(p, q))&&(Identique))
                         {q + +}
                      If(q < m' − 1) then Identique =false; else p + +;
                     }
                  If(p>n'-1) Trouvé=true;
                }
           indice++;
        }
    }
    i++;
 }
```

Fin

<u>Complexité:</u>

Si n' est de même grandeur que n alors la complexité est $(n^3 \log n)$. La recherche dichotomique $(log\ n)$ est exécutée $(n - n')$ fois. Le n^2 est le résultat des comparaisons de B avec le voisinage du premier élément trouvé dans A égal à $B(0,0)$.

Si n' est une constante ou réduite relativement à n, alors la complexité est $O(nlogn)$

Corrigé question 3 :

Début

Soient

A(k,l) l'élément central de A avec k=n/2, l=m/2,

B(k',l') l'élément central de B avec k'=n'/2 et l'=m'/2.

1- Si(n>=n')

alors Calculer la mesure d de B avec la portion centrale de A en positionnant B(k',l') sur A(k,l).
Sinon Echec (retourner)

2- Si (d !=0) alors
Calculer la mesure d de B avec les quatre portions de A positionnées aux centres des quatre quarts du tableau A.
3- Si une des mesures d est nulle,
Alors
La portion recherchée est trouvée.
Sinon
Choisir la partie de la matrice considérée ayant d minimale et refaire l'action (1) pour
cette portion.

Fin

Complexité :

Elle de $O(n^2 2log\ n)$ où n^2 correspond au calcul de d si n' est de même grandeur.
Sinon, le tableau B à une dimension réduite et constante, la complexité est $O(log\ n)$.

Corrigé Exercice 2

1. Le point fixe de T1 est 6 et T2 a deux points fixes 4 et 5
2. Si T[1]>1 ou T[n]<n alors le point fixe ne peut pas exister.

```
Fonction RecherchePointFixe(T : tableau[n] ; n : entier) : booléen
Début
  Si (T[1]<=1  et T[n]>=n) alors retourner PointFixe(T,1,n) ;
  Sinon retourner faux ;
Fin.
Fonction PointFixe(T :tableau[n] ; i, s :entier) :booléen ;
Début
  Si (inf=sup) alors retourner vrai   //T[i] = i
  Sinon m=(i+s)/2 ;
        Si T[m]>=m alors retourner (PointFixe(T, i, m)
        Sinon retourner(PointFixe(T, m+1,s) ;
        Finsi ;
  Finsi ;
Fin ;
```

Complexité :
C(n)=$\theta(\log_2 n)$ si t[1]<=1 et T[n]>=n
C(n)= θ (1) sinon

3. Puisque T[1]>=1, T est toujours sur ou au-dessus de la fonction identité. Si l'ensemble des points fixes de T n'est pas vide, 1 est l'un d'eux. Il suffit donc de comparer T[1] à 1.

```
Fonction RecherchePointFixe(T : tableau[n] ; n : entier) : booleen
Début
  Si (T[1]=1) alors retourner vrai ;
  Sinon retourner faux ;
Fin.
```

Complexité constante : $\theta(1)$