

complexité

Examen

Soit $T1$ et $T2$ deux tableaux d'entiers de n et m éléments.

Une sous-séquence commune de $T1$ et $T2$ est une suite d'entiers consécutifs qui se retrouve à la fois dans $T1$ et $T2$. Exemples :

1	5	3	2	4	4	8	7	9	9	0	5	3	2	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	5	3	2	4	40	4	4	8	7	5
---	---	---	---	---	----	---	---	---	---	---

Ont pour sous-séquences communes $\{5, 3, 2, 4\}$ $\{4, 4, 8, 7\}$ et $\{0, 5, 3, 2, 4\}$

Nous nous intéressons à la longueur de la plus grande sous-séquence commune.

Question 1. (4 points)

Evaluer la complexité de la méthode `longueur`, dans le pire cas.

```
int longueur( int[] t1, int it1, int [] t2, int it2){  
    // retourne la longueur du plus grand préfixe commun  
    // à T1 commençant en it1, et T2 commençant en it2  
    if(it1 == t1.length || it2 == t2.length) return 0;  
    else  
        if( t1[it1]==t2[it2] ) return 1+ longueur(t1, it1+1, t2, it2+1);  
        else return 0;  
}
```

Question 2. (10 points)

On se propose d'évaluer la complexité de la méthode suivante, qui permet de calculer la longueur de la plus grande sous-séquence commune de deux tableaux. Dans ce but, on évalue le nombre d'appels de la méthode `algo1`, en fonction des nombres n et m d'éléments des tableaux $T1$ et $T2$.

```
int algo1( int[] t1, int it1, int [] t2, int it2){  
    if(it1 == t1.length || it2 == t2.length) return 0;  
    else  
        return max( longueur(t1, it1, t2, it2 ),  
                    max( algo1 (t1, it1, t2, it2 + 1),  
                        algo1 (t1, it1 + 1, t2, it2)));  
}
```

Démontrer que $N_n^m = 2 \times C_{n+m}^n - 1$ où C_i^j est le coefficient du binôme. En déduire que pour $n=m$ l'algorithme est exponentiel.

Question 2. (6 points)

Evaluer la complexité de la méthode `algo2`, en fonction des nombres n et m d'éléments des tableaux $T1$ et $T2$.

```
int algo2( int[] t1, int [] t2){  
    int res = 0;  
    for( int it1 = 0; it1<t1.length; ++it1)  
        for( int it2 = 0; it2<t2.length; ++it2)  
            res = max(res, longueur(t1, it1, t2, it2));  
    return res;  
}
```