Université de Béjaïa
Faculté des Sciences Exactes
Département Informatique

Module : Programmation Avancée
Année d'étude Master 1
25/02/ 2017

EMD
Programmation Avancée

Exercice 1. On donne une matrice carrée M de taille NXN et la fonction suivante:

```
Fonction Cal (Var M: Mat; N: entier): réel
Var i,j :  entier;
Debut
        P<---1
        Pour i allant de 1 à N Faire
            Pour j allant de 1 à i  Faire
                P <---  P * M[i, j];
            Fin Pour
        Fin Pour
        Retourner (P)
Fin
```

1. Que fait la fonction *Cal*?
2. Montrer que sa complexité T(N) en nombre de multiplications est $= N(N+1)/2$.
3. Quel est le type de cette complexité?

Exercice 2. Etant donnés x et n en entrée et soit $Som1 = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^i}{i!} + \ldots + \frac{x^n}{n!}$

1. Etablir la relation entre le $i^{ème}$ terme et le $(i-1)^{ème}$ terme.
2. Ecrire une fonction efficace qui retourne la valeur de *Som1*.
3. Montrer que sa complexité en nombre de multiplications est de l'ordre de $O(n)$.

Exercice 3. On définit la $F(n)$ comme suit :

$$F(n) = \begin{cases} 1 \text{ si } n = 0,\ 1,\ 2 \\ F(n-1) + F(n-2) + F(n-3) \quad \text{sinon} \end{cases}$$

1. Ecrivez un algorithme récursif calculant $F(n)$.
2. Montrez que la complexité en nombre d'additions de cet algorithme est en $\Omega(3^{n/3})$.
3. Ecrivez un algorithme récursif qui calcule, pour $n > 1$, le triplet $(F(n), F(n-1), F(n-2))$.
4. Utilisez l'algorithme précédent pour écrire un nouvel algorithme calculant $F(n)$.
5. Quelle est la complexité (en nombre d'additions) de cet algorithme.

Barème : 4(1+2+1)+6(1+3+2)+10(2+2+2+2+2)

xercice 1 .

a

. Cal calcule le produit des elts de l

atrice triangulaire inférieure M.

$$\begin{array}{c} \qquad\qquad\qquad N \\ \left[\begin{array}{cccc} x & & & \\ x & x & & \\ x & & x \ddots & \\ x & & & x \end{array}\right] \end{array}$$

$M_{11} * M_{21} * M_{22} * M_{31} * M_{32} * M_{33} * \ldots$

$+ M_{i1} * M_{i2} * \ldots * M_{ii} * \ldots * M_{N1} * M_{N}$

b

T(N) = ?

```
Pour i de 1 à N faire
  | Pour j de 1 à i faire
  |   | P <- P * M[i,j]
FP FP
```

$i = 1$ j de 1 à 1 $\longrightarrow$ 1

$i = 2$ $j = 1, 2$ $\longrightarrow$ 2

$i = 3$ $j = 1, 3$ $\longrightarrow$ 3

$\vdots$

$i = N$ $j = 1, N$ $\longrightarrow$ N

$T(N) = 1 + 2 + \ldots + N = N(N+2)/2$

Type de complexité :

$T(N) = (N^2 + 2N)/2 = O(N^2) \Rightarrow$ complexité

1

relation entre le i^{e} et le $(i-1)^{e}$ terme

$$\frac{x^i}{i!} = \left(\frac{x^{i-1}}{(i-1)}\right) * \frac{x}{i} \Rightarrow \quad U_i = U_i * \frac{x}{i}$$

Fonction Som (x : reel, n : entier) : reel

<u>Debut</u>

```
| S ← x ; T ← x ;
| Pour (i de 2 à N) faire
|     T ← T * x/i ;
|     S ← S + T ;
| F.Pour
| Retourner (S) ;
Fin
```

Complexite T(N) en n^{bre} de '*'

```
Pour i de 2 à N
|   T ← T * x/i .        N - 1   '*'
F.
```

$$T(N) = (N-1) = O(N)$$

Début

$\overbrace{\text{Si } n=0 \text{ ou } n=1 \text{ ou } n=2}^{n \leq 2}$ alors

retourner (1);

Sinon

retourner $(F(n-1)+F(n-2)+F(n-3))$;

Fsi

Fin

$f = \Omega(g) \Leftrightarrow \exists n_0, \exists c \geq 0 \ \forall n \geq n_0 \ g(n)$

$T(n) \geq c \cdot 3^{n/3}$

2

On démontre par recurrence que $T(n) \geq c \cdot 3^{n/3}$

à determiner

1 - Supposons que la relation est vraie pour

$n-1$ [i.e $T(n-i) \geq c \cdot 3^{(n-i)/3} \ \forall i \geq 1$]

On va montrer que la relation est vraie

$T(n) = T(n-1) + T(n-2) + T(n-3) + 2 \geq c \cdot 3^{(n-1)/3} + c \cdot 3^{(n-2)/3} + c$

$\Rightarrow T(n) \geq 3 \cdot c \cdot 3^{(n-3)/3} = c \cdot 3^{n/3} \Rightarrow \boxed{T(n) \geq c \cdot 3}$

2 - Il reste à montrer que la relation est vraie pour les valeurs de départ.

$n=0$, $n=1$ et $n=2 \Rightarrow T(n) = 0$ on part donc

~~ex~~, $n=3$ et $n=4$

pour $n=3$ $F(3) = F(2) + F(1) + F(0) \Rightarrow T(3) = 2$

$F(4) = F(3) + F(2) + F(1) \Rightarrow T(4) = 2 + 2 =$

Debut

Si $n \leq 2$ **Alors**

retourner (1)

Sinon

$(x, y, z) \leftarrow F\_T(n)$;

retourner (x);

Fsi

Fin

Complexité $T(n)$ de F = Complexité F_T

$$\begin{cases} T(n) = T(n-1) + 2 \\ T(3) = 2 \end{cases}$$

$\Rightarrow T(n) = T(n-2) + 2*2 = T(n-3) +$

$= \ldots = T(n-i) + i*2 = \ldots T(3) + (n-3$

$= 2 + (n-3)*2 = (n-2)*$

$\Rightarrow T(n) = 2n - 4.$